Complexity

Frontiers in Data-Driven Methods for Understanding, Prediction, and Control of Complex Systems 2021

Lead Guest Editor: Andrea Murari
Guest Editors: Jesus Vega, Gonzalo Farias, Teddy Craciunescu, and Michela Gelfusa

Frontiers in Data-Driven Methods for Understanding, Prediction, and Control of Complex Systems 2021

Complexity

Frontiers in Data-Driven Methods for Understanding, Prediction, and Control of Complex Systems 2021

Lead Guest Editor: Andrea Murari
Guest Editors: Jesus Vega, Gonzalo Farias, Teddy Craciunescu, and Michela Gelfusa

Contents

Hindawi
Complexity
Volume 2022, Article ID 4830411, 13 pages
https://doi.org/10.1155/2022/4830411

WILEY | Hindawi

Research Article

Predicting and Preventing Crime: A Crime Prediction Model Using San Francisco Crime Data by Classification Techniques

Muzammil Khan,[1] **Azmat Ali**,[2] **and Yasser Alharbi**[3]

[1]*Department of Computer & Software Technology, University of Swat, Swat, Pakistan*
[2]*School of Computer Science, Wuhan University, Wuhan, China*
[3]*College of Computer Science, University of Hail, Hail, Saudi Arabia*

Correspondence should be addressed to Muzammil Khan; muzammilkhan86@gmail.com

Received 3 August 2021; Accepted 25 January 2022; Published 25 February 2022

Academic Editor: Gonzalo Farias

The crime is difficult to predict; it is random and possibly can occur anywhere at any time, which is a challenging issue for any society. The study proposes a crime prediction model by analyzing and comparing three known prediction classification algorithms: Naive Bayes, Random Forest, and Gradient Boosting Decision Tree. The model analyzes the top ten crimes to make predictions about different categories, which account for 97% of the incidents. These two significant crime classes, that is, violent and nonviolent, are created by merging multiple smaller classes of crimes. Exploratory data analysis (EDA) is performed to identify the patterns and understand the trends of crimes using a crime dataset. The accuracies of Naive Bayes, Random Forest, and Gradient Boosting Decision Tree techniques are 65.82%, 63.43%, and 98.5%, respectively, and the proposed model is further evaluated for precision and recall matrices. The results show that the Gradient Boosting Decision Tree prediction model is better than the other two techniques for predicting crime, based on historical data from a city. The analysis and prediction model can help the security agencies utilize the resources efficiently, anticipate the crime at a specific time, and serve society well.

1. Introduction

Data mining is the knowledge discovery process used to collect and analyze a large dataset and summarize it with helpful information. It is critical in different fields of science to serve analytical purposes and plays an essential role in human life and fields such as education, business, medicine, health, and science. Data mining is an attractive process of discovering a valid, understandable, helpful pattern and valuable information in large amounts of data [1]. The main goal of data mining is to find out fascinating and concealed knowledge in the data and summarize it in a significant form [2–4]. Similarly, the results should be in the form that conveys the inside information effectively [5–7]. Therefore, classification techniques are among the most important and commonly used techniques in data mining, and supervised class prediction techniques allow nominal class labels for predictions [8].

San Francisco is one of the largest cities in the United States of America. Therefore, it is vital to understand the pattern of crimes to ensure the safety of the citizens. San Francisco Crime Classification is an open-source dataset available for an online competition administrated by Kaggle Inc. The main task in the dataset is to predict the crime category based on a given set of geographical and time-based variables. The limited and constrained police resources prove insufficient to handle the city's law and order issues. Therefore, it is vital to study and understand the distribution of different types of crimes in the city based on the occurrence time and the location for security agencies to channelize resources efficiently. Naive Bayes, Random Forest, and Gradient Boosting Decision Tree are used for prediction and classification of crimes into two types of violent and nonviolent crimes.

In this paper, the main goal is to propose a prediction model that predicts crime based on past criminal records.

The proposed model contains three techniques and performs evaluation through accuracy, precision, and recall evaluation matrices. The data is descriptively analyzed and statistical crime distribution over space and time is visualized to help attain potential patterns. The features are extracted from the original dataset, and the classification is performed using Naive Bayes, Random Forest, and Gradient Boosting Decision Tree techniques. The experimental results show that the Gradient Boosting Decision Tree prediction model is better than the other two techniques for predicting crime, based on historical data from a city. The analysis and prediction model can help the security agencies utilize the resources efficiently, anticipate the crime at a specific time, and serve society well. Conclusions of the study and future directions for further research are presented in the last section of the paper.

2. Related Work

Data mining has been frequently used in crime prediction models for the last couple of years, considering different features. Yehya used variables such as longitude (X), latitude (Y), address, day of week, date (YYYY-mm-dd: hh : MM : ss), district, resolution, and category to analyze and predict San Francisco crime data. The study used different techniques and principal component analysis to classify the accuracy and avoid overfitting. He also used four different classifiers: K-NN, XGB Decision Tree, Bayesian, and Random Forest, applied them to the task, and obtained the log-loss of 2.39031 by the Random Forest classifier [9]. Wenbin Zhu et al. conducted an experiment for the classification of crime based on the San Francisco dataset. According to their explanation, it was mentioned that crime classification helps police to keep the city safe. They predicted crime categories based on time and location. They used Naive Bayes, logistic regression, and the Random Forest as baseline classifiers with best prediction results [10]. Umair Saeed et al. experimented with data mining techniques to identify and predict crimes and compared the experiment results of Naive Bayes and Decision Tree classifiers. They observed that the Naive Bayes classifier performed better and accurately predicted crime prediction [11]. Somayeh Shojaee et al. conducted an experimental study for crime prediction using supervised classification learners. They used two different feature selection methods executed on real crime datasets. They compared these two methods based on AUC (i.e., Area Under the Curve) values. They found that Naive Bayes, K-Nearest Neighbor (KNN), and Neural Networks are better classifiers against Decision Tree (J48) and Support Vector Machine (SVM). The Chi-square feature selection technique is used in their experiment for the performance measurement of the classifiers. The investigation is conducted in a RapidMiner environment to enhance the quality of crime mining [12]. Junbo et al. predicted crime categories from 2003 to 2015 surrounding San Francisco city based on a dataset derived from SFPD Crime Incident Reporting System. They investigated Naive Bayes, K-NN, and Gradient Tree Boosting classification models and analyzed their advantages and disadvantages on that prediction task. According to their results, Naive Bayes did not perform as a perfect model for that task because some features did not represent the count or frequency. On the other hand, K-Nearest Neighbor improved the prediction result to a large extent. Gradient Tree Boosting performed as the best model in their experiment, but it was slightly slow. Gradient Tree Boosting model generated a score of 2.39383 and was ranked 93 among 878 teams [9]. R. Iqbal et al. (2013) conducted an experimental study for the classification algorithms. They experimented with the prediction of crime categories for the different states of USA. They compared Naive Bayes with the Decision Tree classifier for crime prediction. Naive Bayes achieved 70.81% accuracy and the Decision Tree classifier achieved 83.95% accuracy, which shows that the Decision Tree classifier performs better for the crime classification problems [13].

3. San Francisco Dataset

The study uses a dataset from Kaggle to build up the model [2]. The dataset (training set/data) has different attributes, each having a different connection. The training dataset contains the incidents taken from Kaggle on San Francisco crimes. The data ranges from January 2003 to May 2015. The dataset contains almost 12 years of criminal reports from San Francisco. The dataset has classified categories of all crimes, which contain different crime types. The training set consists of 878049 observations and the testing set consists of 884263 observations. The dataset is used to check the accuracy of the classification techniques with new unclassified data. The training set consists of nine variables as shown in Table 1.

The study arbitrarily mixes the original training dataset and divides it into a training dataset and testing dataset with 80% and 20% sizes, respectively.

3.1. Exploratory Data Analysis. A simple script is run and explores several unique categories of crimes in the dataset, and 39 different crime categories are identified. The figure also shows the distribution of crime and change in the type of crime since 2003. For example, from the below plot, larceny/theft is the most common type of crime. Further, there appears to be a skewness in the type of crimes. For example, there have been 174,900 incidents of larceny/theft, whereas there have been only 6 of TREA since 2003.

From Figure 1, it is found that the top 10 crimes are larceny/theft, other offenses, noncriminal, assault, drug/narcotic, vehicle theft, vandalism, warrants, burglary, and suspicious OCC, accounting for 83.5% of the whole records statistically [10]. It is reasonable to suggest allocating more police resources to deal with these crimes as they are more likely to occur.

Figure 2 indicates that the lower overall density of sex offenses compared to the other categories of crime is expected, as there are fewer crimes of this category in the data. The overall structure here indicates the aggregate with the most prominent hot spot in the north area centered in the

Table 1: Selected features for analysis.

Attributes	Descriptions
Dates	Date is the timestamp of the moment when the crimes occurred
Category	Category shows the crime category
Description	Description shows the short description of the crime
DayOfWeek	DayOfWeek shows the day on which the crime occurred
PdDistricts	PdDistricts shows the district of the city where the crime was committed
Resolution	Resolution shows a short description of the crime resolution
Address	Address shows the address of the crime where it was located.
X	X shows the latitude of the crime position
Y	Y shows the longitude of the crime position

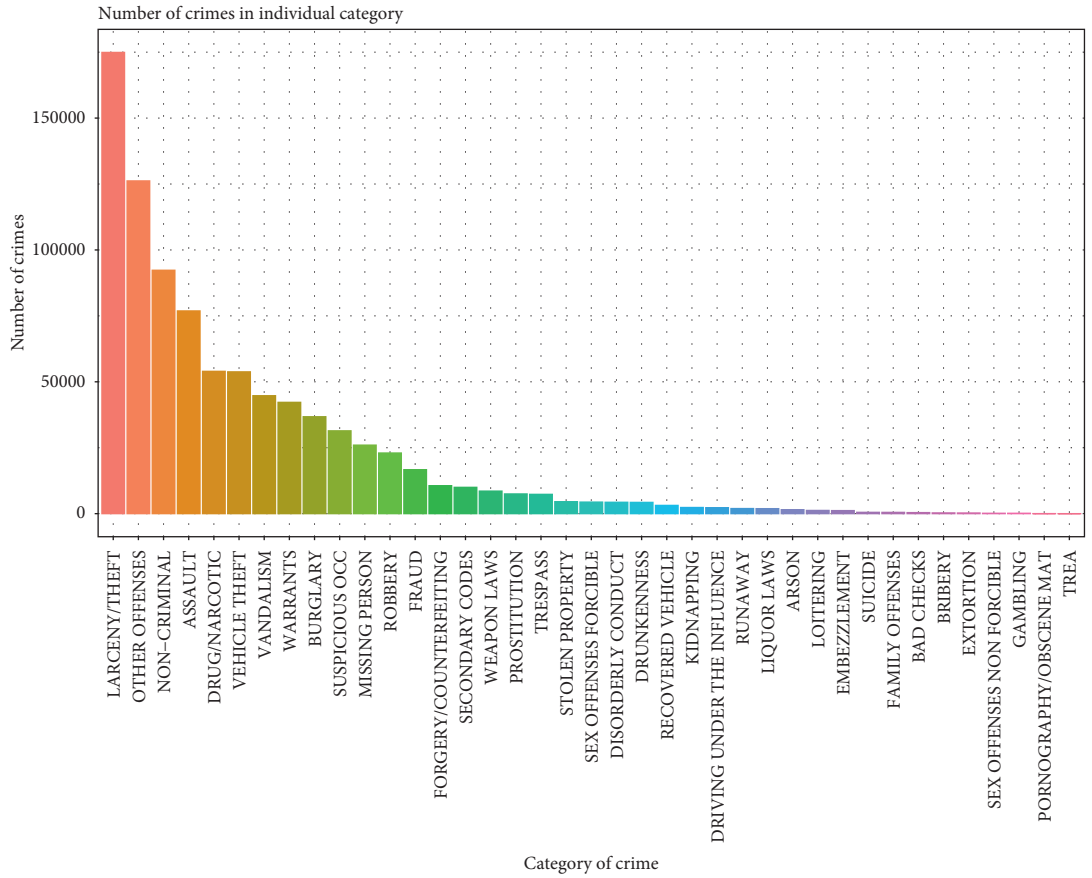


Figure 1: Number of crimes in individual category.

Tenderloin neighborhood area. Larceny/theft, other offenses, noncriminal, and assaults seem to be more concentrated on the map. However, four crimes seem to cover a larger area: vehicle theft, vandalism, burglary, and suspicious OCC. At the same time, other crimes come into a smaller area with larger density crime. It is interesting to explore further other columns of the dataset to help us extract useful features. What are the distributions for day of week, hour, month, and even year for the record of the crimes? We visualize how their occurrences alter with year, month, day of week, and hour for the ten most occurring crimes.

Figure 3 shows interesting figures and results based on years. This map reveals the increase or decrease in the top ten crimes in different years in San Francisco from 2003 to 2015.

Figure 4 shows monthly reports of the top ten crimes in San Francisco, revealing the expansion and reduction of crime month-wise. However, the interesting point is that all crimes (top 10) are increased after three months and also decreased after three months, which reveals that the top ten crimes in the San Francisco area based on seasonal pattern increased in the 3rd month (March) with same pattern in Spring, decreased in the 6th month (June) with the same pattern in Summer, and increased again in September, Autumn.

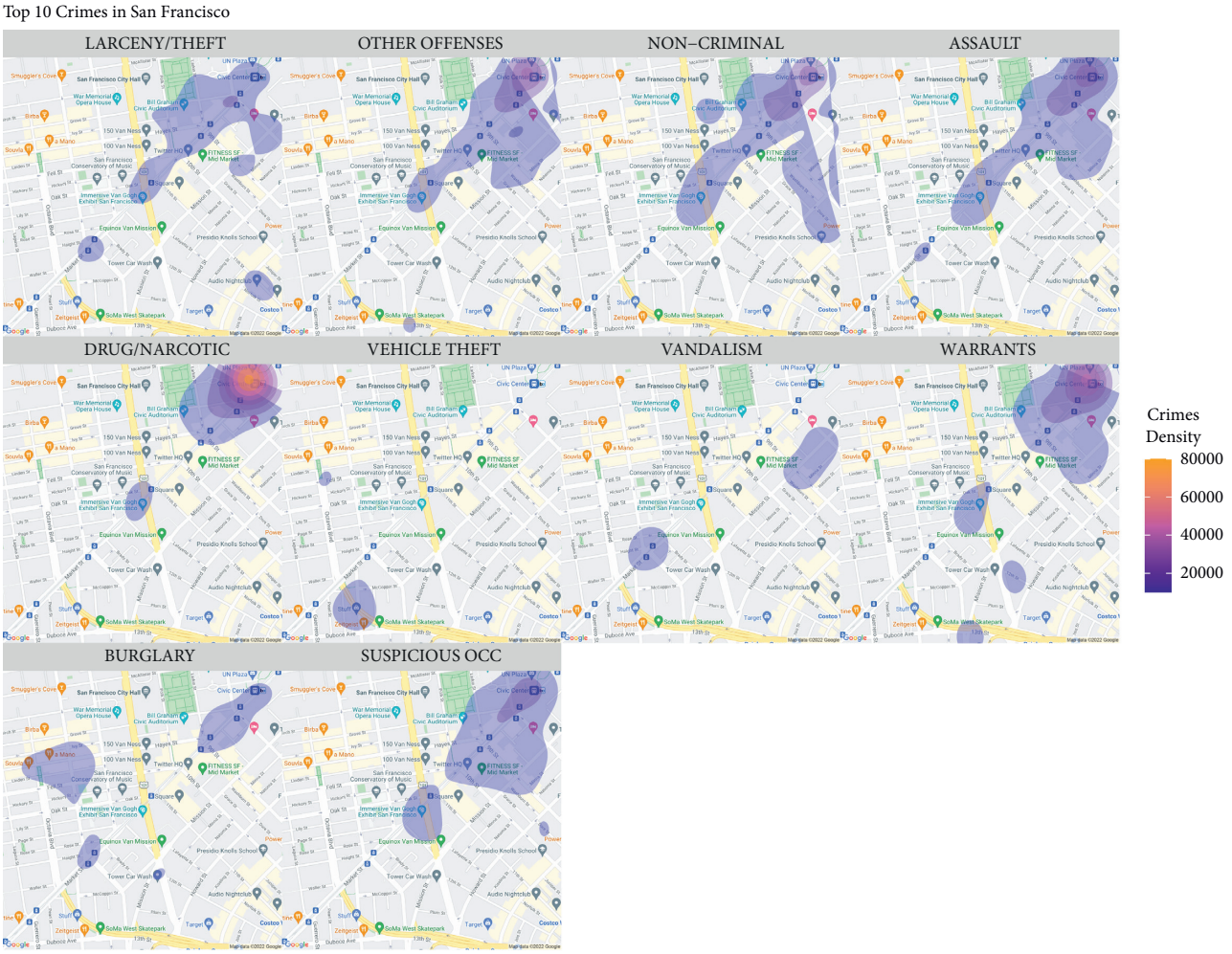


Figure 2: Top 10 crimes' density in San Francisco.

Figure 5 shows the top ten crimes' ratio (increase or decrease) for days of the week. The crime is more concentrated in northern areas on Friday, Saturday, and Wednesday. Larceny crime, vehicle crime, and vandalism crime increased on Friday and Saturday with the same pattern, while the rate of suspicious OCC crime occurred and increased on Friday and Wednesday. Burglary crime increased on Friday, and assault crime increased on Saturday and Sunday. Drug/narcotics and warrants crime occurred and increased on Wednesday. All these crimes indicate the ratio and occurrence of crime in San Francisco based on days (days of weeks).

Figure 6 shows the aggregate of the crime and the crime rate in each hour. In this graph, the results suggest that all the top ten crimes decreased between 3 : 00 AM and 6 : 00 AM but reached their second peak at midnight and the first peak around 5 : 00 PM to 6 : 00 PM. So, when police resources are limited, our suggestion is to allocate more police from noon to midnight.

There are seasonal patterns in data, where although the total crime counts were different, the normalized values followed similar trends. When normalized by mean and standard deviations, seasonal patterns in a month appear. Similar patterns emerge for hours also. Different lines represent crimes for different categories (top 10 only) in Figures 7 and 8, respectively.

3.2. Variable Selection. The variable "Category" is the dependent variable for prediction. The variables "Resolution" and "Description" are irrelevant for the analysis because of their nature and were dropped from the dataset during preprocessing steps. The remaining variables are considered the independent variables, used for predicting the dependent variable.

3.3. Variable Transformation. Few variables are transformed to enrich the features of the dataset:

(1) The "Date" variable is divided into four separate variables: year of the incident (2003–2015), month and place of the incident (1–12), day of the incident (1–31), and the hour of the day when the incident happened (0–23).

(2) The variables DayOfWeek and PdDistrict are indexed and replaced by numbers (i.e., DayOfWeek: 1,2, . . ., 7, and PdDistrict: 1, 2, . . ., 10)

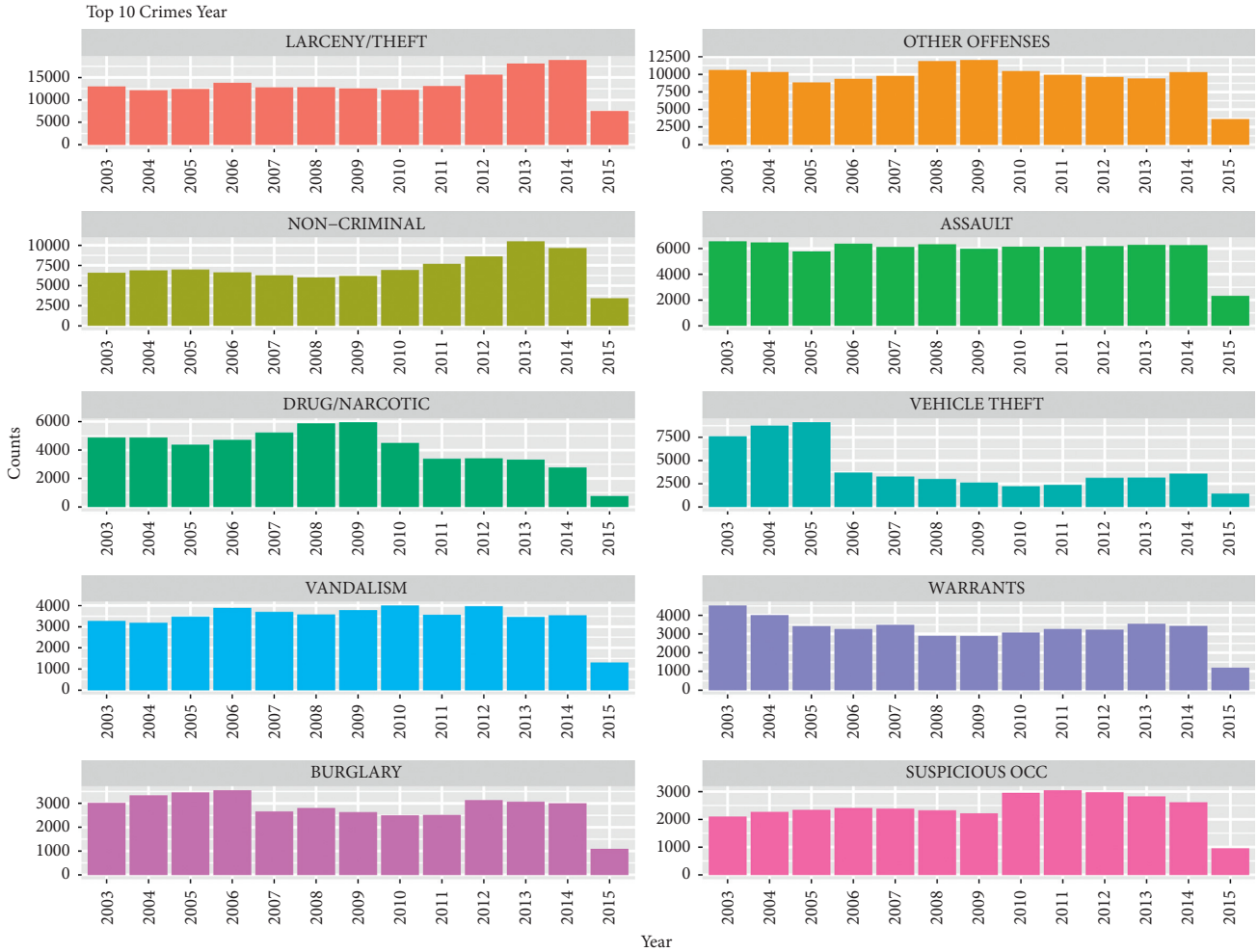


Figure 3: Year-wise top 10 crimes.

4. Prediction Model

The prediction model is based on Naive Bayes, Random Forest, and Gradient Boosting Decision Tree prediction techniques, briefly discussed below.

4.1. Naive Bayes. Naive Bayes is based on the Bayesian theorem, and it is a conditional probabilities method that calculates the probability by counting frequent values [14].

Naive Bayes is summarized as follows:

(1) A simple classification process classifier

(2) Best suited for historical data and prediction

(3) Classification technique analysis of the relationship between each attribute and the class instance

(4) A supervised learning method that can solve categorical and probabilistic problems

(5) A popular classification technique in text categorization [14].

This Naive Bayes classifier was introduced in 1995 [14]. It is known with different names in the community of data mining and machine learning, such as simple bases and independence Bayes [15]. The Naive Bayes classifier is commonly used in many applications like sentiment classifications and in different ensemble prediction models [16–18].

Using the Naive Bayes classifier, two types of quantities need to be calculated from the dataset, that is, Class Probabilities and Conditional Probabilities.

The method of the Bayesian classifier is given in the following equation:

$$P\left(\frac{C}{X}\right) = P\left(\frac{X}{C}\right)\frac{P(C)}{P(X)}. \quad (1)$$

Here, $P(C\text{-}X)$ is a maximum posterior hypothesis, $P(C)$ is prior, $P(X)$ is evidence, and $P(X\text{-}C)$ is the likelihood of the hypothesis [8].

4.2. Random Forests. Leo Breiman and Ahele Culter developed the Random Forest algorithm. In 1995, Tin Km Ho (Bell Labs) used for the first time the term Random Decision Tree. Ensemble learning method, Random Forests, or Random Decision Forest is a very famous classification and regression method. It is building numbers of the classifier on the training dataset which makes good predictions. This technique is also

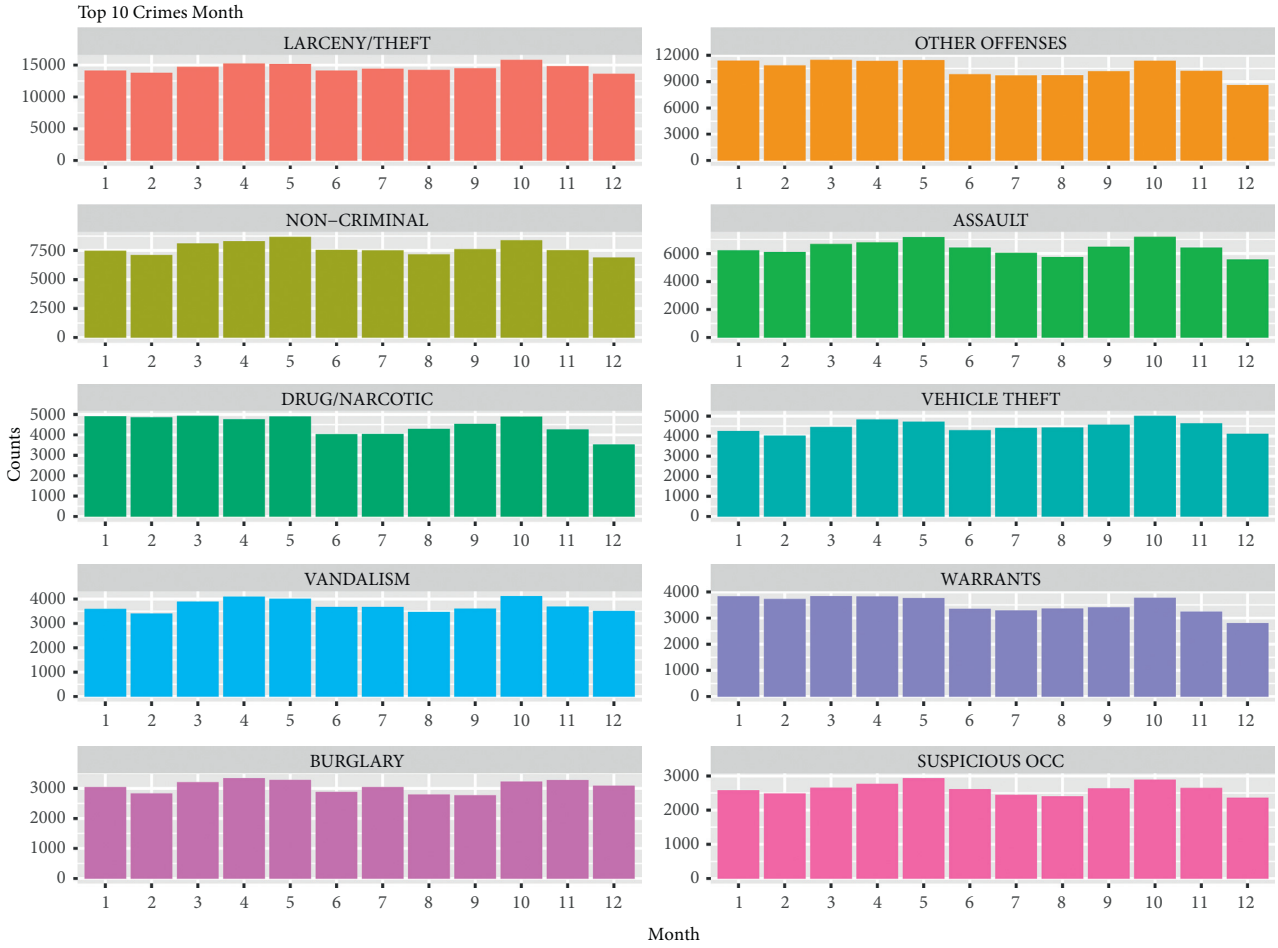


Figure 4: Month-wise top 10 crimes.

used for the predictions of handwriting character, digital pattern recognition, semantic analysis, language feature extraction, and hybrid models [19–22]. In this technique, every tree depends on randomly selected values sampled and independently corresponding distribution for every single tree around it. The numbers of trees increase in the forest general error for the forests converges as become to the limit for the forest's trees. The generalization error of the classifier depends on the correlation and individual strength between the trees of the forest. Each node in the Random Forest is split and randomly selected; the features yield an error rate that is better as compared with AdaBoost.

Definition. Random Decision Forests or Random Forest is a technique consisting of a tree-structured classifier $h(x, k)$, $k = 1,\ldots,$ where k represents independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

Correlation and Strength. In Random Decision Trees or RF, the generalization error can be obtained in terms of two parameters: how the single classifier measures the value accurately and the dependence between them [23].

Random Decision Forests correct for Decision Trees' habit of overfitting to their training set, and a Random Forest produces a large number of decision trees. For data including categorical variables with a different number of levels, Random Forests are biased in favor of those attributes with more levels. Categorical variables also increase the computational complexity to create trees [24].

4.3. Gradient Boosting Tree. Gradient Boosting Tree is a machine learning technique for classification and regression problems. This technique makes a prediction model that uses typically Decision Trees in the form of an ensemble of the weak prediction model. In this technique, the models are built in the same way as in other boosting models. It constructs the model in a stage-wise way as other boosting methods do, and it generalizes it by allowing optimization of an arbitrary differentiable loss function. The idea of gradient boosting originated in the observation by Leo Breiman where boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed by Jerome H. Friedman simultaneously with the more general functional gradient boosting perspective of Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Freeman.

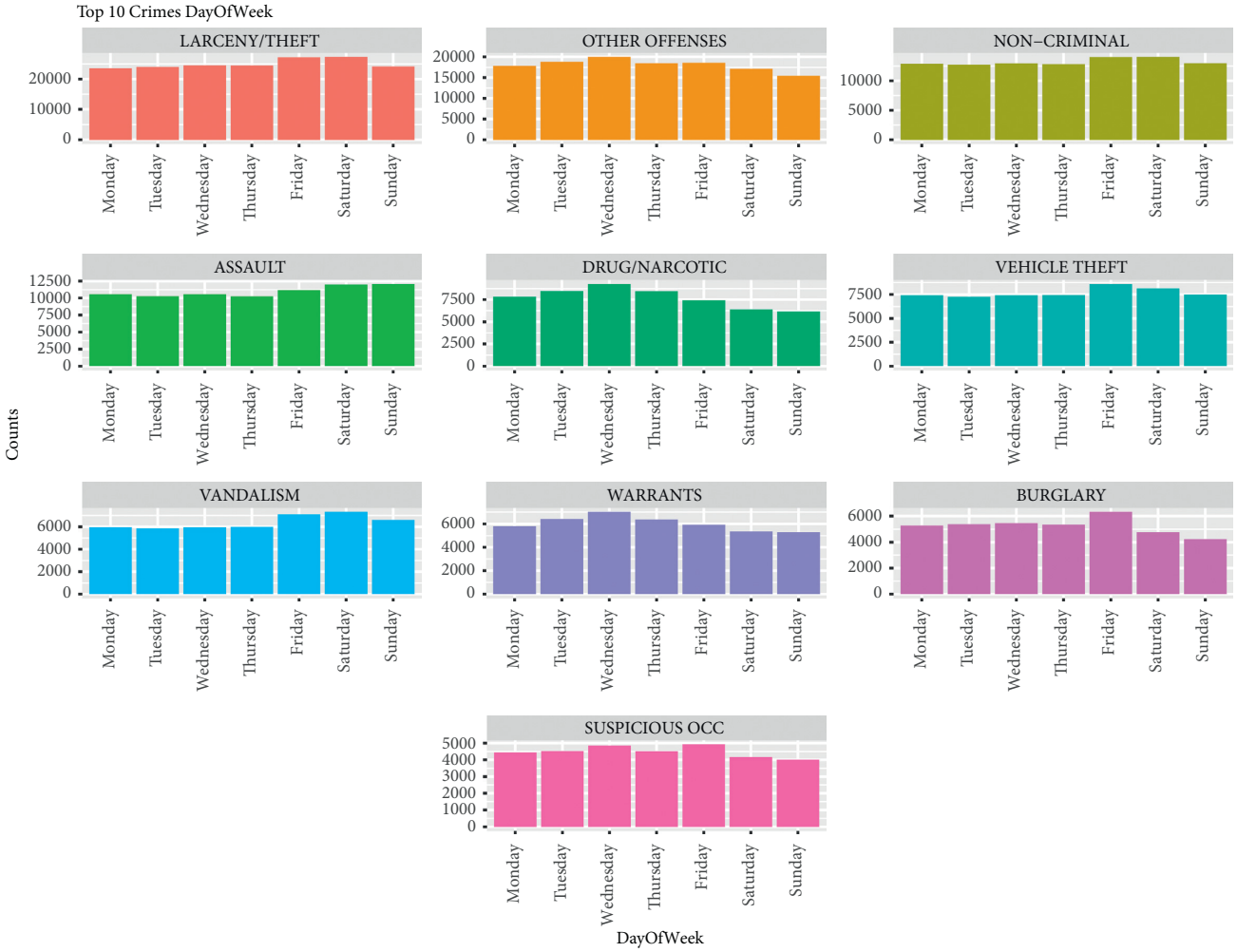


Figure 5: Top 10 crimes per day of week.

4.4. Performance Evaluation Metrics. The proposed prediction models are evaluated on the accuracy, precision, and recall, and ROC and Lift are the performance metrics for estimating the classification models [25]. Therefore, it is imperative to compare the accuracy using an alternative method, precision and recall; because of a two-class problem, the performance of a classifier is presented using the "confusion matrix" in Table 2.

The following are standardized equations for computing accuracy, sensitivity/recall, specificity, and precision.

$$\text{Accuracy} = \text{TP} + \text{TN}/\text{TP} + \text{FP} + \text{TN} + \text{FN}.$$

$$\text{Sensitivity} = \text{recall} = \text{TP}/t = \text{TP}/(\text{TP} + \text{FN}).$$

$$\text{Specificity} = \text{TN}/n = \text{TN}/(\text{TN} + \text{FP}).$$

$$\text{Precision} = \text{TP}/p = \text{TP}/(\text{TP} + \text{FP}).$$

TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative in the confusion matrix presented in Table 2. Precision in this context refers to the actual percentage of crime predicted by the classification model, which translates into the returns on the cost of categories. On the other hand, recall measures the percentage of crime identified and needed to be targeted. Thus, at last, specificity measures how good a test is at avoiding false alarms.

5. Experiment Results and Performance Evaluation

All three models were trained and presented in the previous section with different setting parameters and feature selections. The data exploration section observes that both the time-related features and geographic features are important. For analysis, all the three models are trained and tested, that is, the training dataset with 878,049 records from Kaggle, and they are divided into two parts in the ratio of 80 : 20 for all the models. Thus, 80% of the dataset were used to train the model, whereas 20% were used to test the model. The subsections discuss the performance and results.

6. Naive Bayes

In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. In Table 3, each column holds the reference (or actual) data and within each row is the prediction. The diagonal represents instances where our observation correctly predicted the class of the item. The table

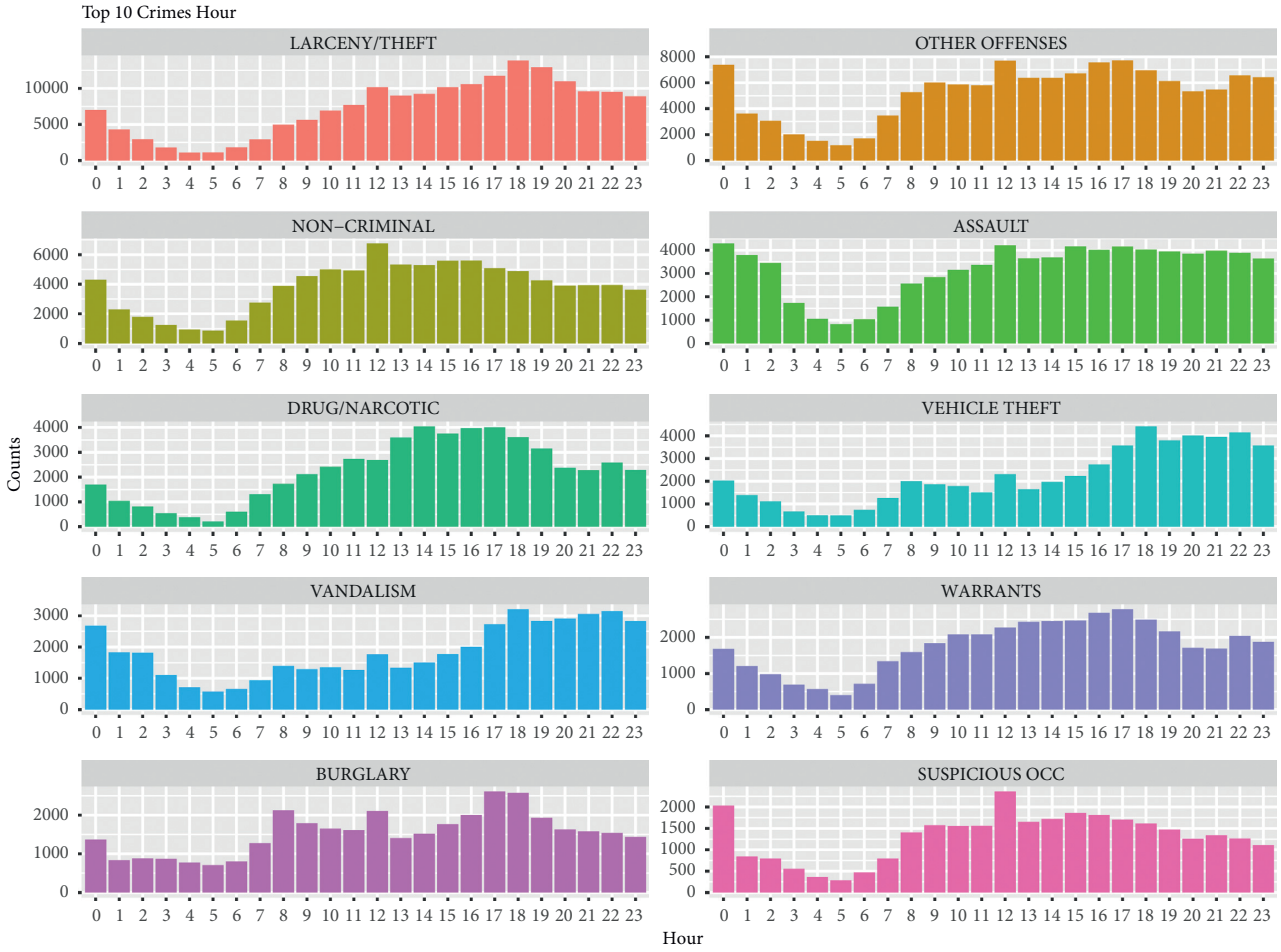


Figure 6: Top 10 crimes per hour.

classifies nonviolent crime and violent crime classes using the Naive Bayes algorithm for the training set. For each class, the result of a confusion matrix is discussed below:

There are 345,082 items classified into the nonviolent crime class.

(1) In the nonviolent crime class, the correctly classified items are 226,209.

(2) In the violent crime class, the wrongly classified items are 118,873.

There are 357,357 items classified into the violent crime class.

(1) In the violent crime class, the correctly classified items are 236,107.

(2) In the nonviolent crime class, the wrongly classified items are 121,250.

In Table 4, each column holds the reference (or actual) data and within each row is the prediction. The diagonal represents instances where our observation correctly predicted the class of the item. The table classifies nonviolent crime and violent crime classes using the Naive Bayes algorithm for the testing set. For each class, the result of a confusion matrix is discussed below.

There are 86,399 items classified into the nonviolent crime class.

(1) In the nonviolent crime class, the correctly classified items are 55,282.

(2) In the violent crime class, the wrongly classified items are 31,117.

89,211 items are classified into the violent crime class.

(1) n the violent crime class, the correctly classified items are 57,693.

(2) In the nonviolent crime class, the wrongly classified items are 31,518.

6.1. Random Forest. Random Forest technique is an ensemble learning method for classification, regression, and other tasks, operated by constructing a multitude of Decision Trees at training time and outputting the class, that is, the mode of the classes (classification) or means prediction (regression) of the individual trees. Random Decision Forests correct for Decision Trees' habit of overfitting to their training set. In this experiment, Random Forest was selected as a technique to estimate the predictors (Table 5).

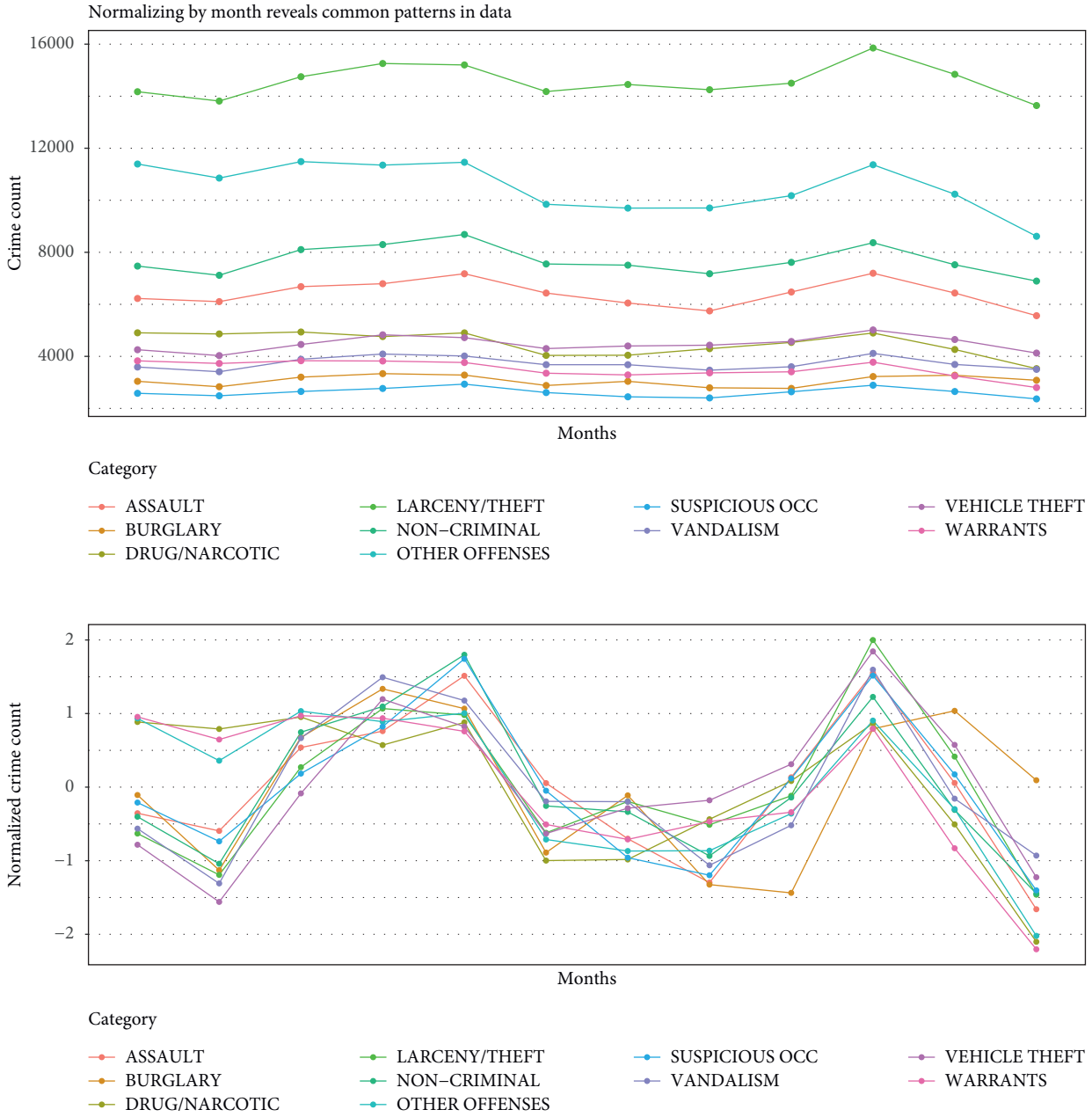


Figure 7: Normalizing by month reveals common pattern in data.

In Table 6, each column holds the reference (or actual) data and within each row is the prediction. The diagonal represents instances where our observation correctly predicted the class of the item. The table classifies nonviolent crime and violent crime classes using the Random Forest algorithm for the training set. For each class, the result of a confusion matrix is discussed below.

There are 349,230 items classified into the nonviolent crime class.

(1) In the nonviolent crime class, the correctly classified items are 280,840.

(2) In the violent crime class, the wrongly classified items are 68,390.

353,209 items are classified into the violent crime class.

(1) In the violent crime class, the correctly classified items are 287,017.

(2) In the nonviolent crime class, the wrongly classified items are 66,192.

In Table 7, each column holds the reference (or actual) data and within each row is the prediction. The diagonal represents instances where our observation correctly predicted the class of the item. The table classifies nonviolent crime and violent crime classes using the Random Forest algorithm for the testing set. For each class, the result of a confusion matrix is discussed below.

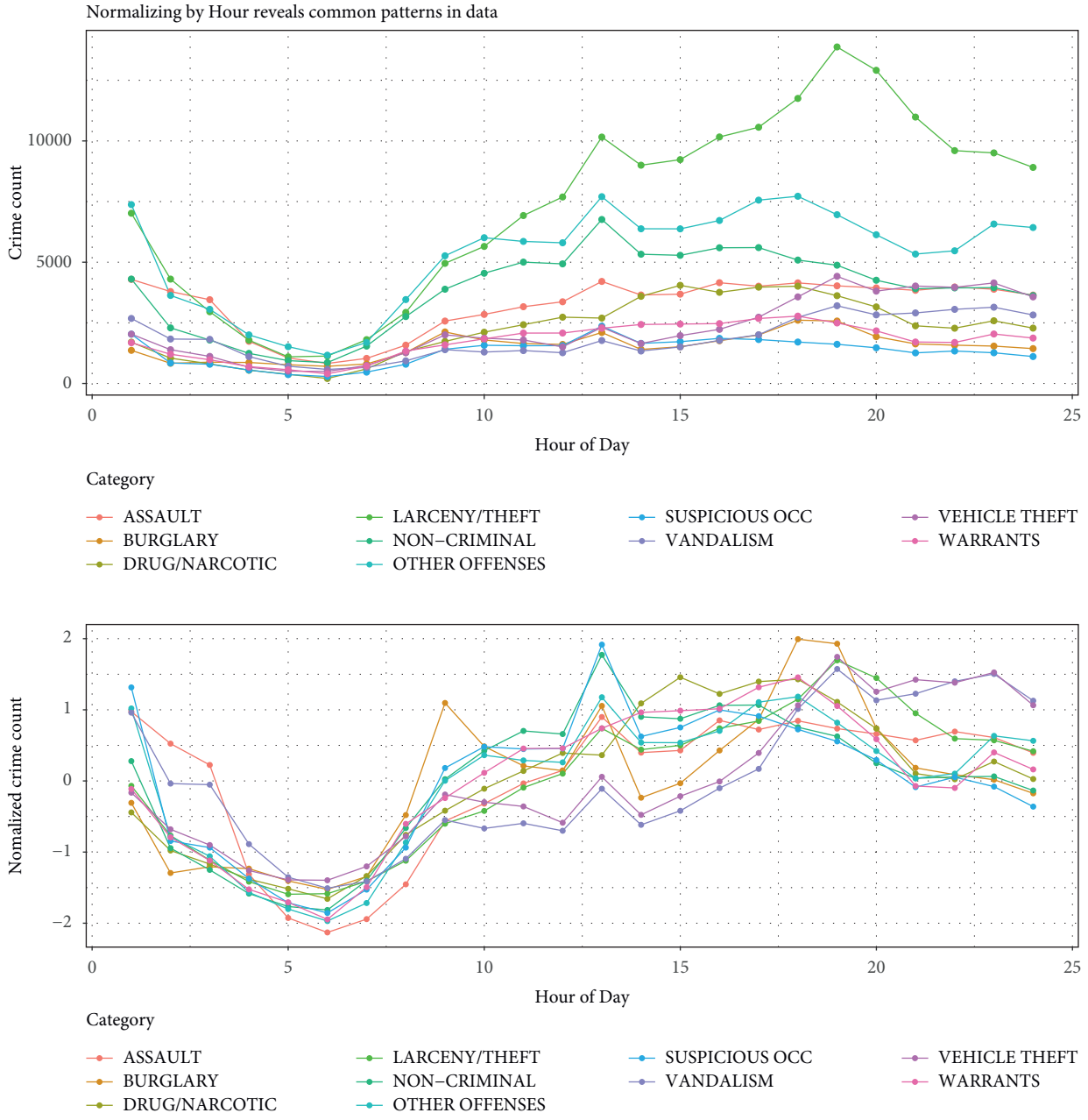


Figure 8: Normalizing by hour reveals common pattern in data.

Table 2: Confusion matrix.

		Yes	No
Actual class	Yes	True positive (TP)	False negative (FN)
	No	False positive (FP)	True negative (TN)

Table 3: Confusion matrix results of Naive Bayes on training data.

Predictions/references	Nonviolent crime	Violent crime
Nonviolent crime	226,209	118,873
Violent crime	121,250	236,107

Table 4: Confusion matrix results of Naive Bayes on testing data.

Predictions/references	Nonviolent crime	Violent crime
Nonviolent crime	55,282	31,117
Violent crime	31,518	57,693

Table 5: Accuracy, incorrectly classified instances, recall, and precision for Naive Bayes on training and testing data.

Method	Accuracy (correctly classified instances)	Incorrectly classified instances	Recall	Precision
Naive Bayes (training data)	65.82%	34.18%	65.55%	65.10%
Naive Bayes (testing data)	64.33%	35.67%	64.67%	63.8 8%

Table 6: Confusion matrix results of Random Forest on training data.

Predictions/references	Nonviolent crime	Violent crime
Nonviolent crime	280,840	68,390
Violent crime	66,192	287,017

Table 7: Confusion matrix results of Random Forest on testing data.

Predictions/references	Nonviolent crime	Violent crime
Nonviolent crime	254,779	31,766
Violent crime	32,448	56,617

There are 86,545 items classified into the nonviolent crime class.

(1) In the nonviolent crime class, the correctly classified items are 54,779.

(2) In the violent crime class, the wrongly classified items are 31,776.

89,065 items are classified into the violent crime class.

(1) In the violent crime class, the correctly classified items are 56,617.

(2) In the nonviolent crime class, the wrongly classified items are 32,448.

6.2. Gradient Boosting Trees. Gradient Boosting Decision Trees is a robust machine learning technique used in predictive modeling due to its high prediction accuracy compared to other modeling techniques. Gradient Boosting Decision Trees produces a prediction model in the form of an ensemble of weak prediction models, that is, Decision Trees. It builds the model in a stage-wise fashion as other boosting methods do, and it generalizes it by optimizing an arbitrary differentiable loss function (Table 8).

In Table 9, each column holds the reference (or actual) data and within each row is the prediction. The diagonal represents instances where our observation correctly predicted the class of the item. The table classifies nonviolent crime and violent crime classes using the Gradient Boosting Decision Trees algorithm for the training set. For each class, the result of a confusion matrix is discussed below.

There are 351,145 items classified into the nonviolent crime class.

Table 8: Accuracy, incorrectly classified instances, recall, and precision for Random Forest on training and testing data.

Method	Accuracy (correctly classified instances)	Incorrectly classified instances	Recall	Precision
Random Forest (training data)	80.84%	19.16%	80.41%	80.93%
Random Forest (testing data)	63.43%	36.57%	63.29%	62.80%

Table 9: Confusion matrix results of Gradient Boosting Decision Trees on training data.

Predictions/references	Nonviolent crime	Violent crime
Nonviolent crime	347,260	3,885
Violent crime	0	351,294

(1) In the nonviolent crime class, the correctly classified items are 347,260.

(2) In the violent crime class, the wrongly classified items are 3,885.

351,294 items are classified into the violent crime class.

(1) In the violent crime class, the correctly classified items are 351,294.

(2) In the nonviolent crime class, the wrongly classified items are 0.

In Table 10, each column holds the reference (or actual) data and within each row is the prediction. The diagonal represents instances where our observation correctly predicted the class of the item. The table classifies nonviolent crime and violent crime classes using the Gradient Boosting Decision Trees algorithm for the testing set. For each class, the result of a confusion matrix is discussed below.

There are 86,569 items classified into the nonviolent crime class.

(1) In the nonviolent crime class, the correctly classified items are 86,569.

(2) In the violent crime class, the wrongly classified items are 0.

Table 10: Confusion matrix results of Gradient Boosting Decision Trees on testing data.

Predictions/references	Nonviolent crime	Violent crime
Nonviolent crime	86,569	0
Violent crime	430	88,611

Table 11: Accuracy, incorrectly classified instances, recall, and precision for Gradient Boosting Decision Trees on training and testing data.

Method	Accuracy (correctly classified instances)	Incorrectly classified instances	Recall	Precision
Gradient Boosting Decision Trees (training data)	99.44%	0.66%	98.89%	100%
Gradient Boosting Decision Trees (testing data)	99.75%	0.25%	100%	99.50%

89,041 items are classified into the violent crime class.

(1) In the violent crime class, the correctly classified items are 88,611.

(2) In the nonviolent crime class, the wrongly classified items are 430.

Tables 5, 8, and 11 present the accuracies of Naive Bayes, Random Forest, and Gradient Boosting Decision Tree techniques, respectively, and it is shown that the Gradient Boosting Decision Trees technique has better results.

7. Conclusions and Future Directions

The study presents exploratory data analysis using a prediction model based on classification techniques and compares the results of San Francisco crime data. The Naive Bayes, Random Forest, and Gradient Boosting Decision Tree are used for predicting the crime category attribute labeled "violent" and "nonviolent." The techniques are implemented in R languages, and the experimental results for all three algorithms manifest that Gradient Boosting Decision Tree performed better than Naive Bayes and Random Forest for the crime classification. The Gradient Boosting Decision Tree achieved 98.5%, 96.96%, and 100% for accuracy, precision, and recall, respectively. Law enforcement agencies can take great advantage of using machine learning algorithms like Gradient Boosting Decision Tree to fight crime effectively, channelize the resources efficiently, anticipate the crime up to some extent, and serve society. The proposed prediction models can be implemented to any dataset or crime data for predictions and resource management.

In the future, the same models using more advanced classification algorithms can be applied to the crime dataset and evaluate their prediction performance to discover trends and improve the subject knowledge. To design a comprehensive framework for the prediction that helps law enforcement agencies manage the resources in a specific area quickly, it is believed that higher accuracy can be achieved when employing more feature engineering in the address field. A more temporal analysis can be performed to determine the number and intensity of criminal activities using time series analysis, a mix of temporal and spatial analysis, which can help allocate resources more efficiently and effectively.

Data Availability

The dataset used in this research is available in the UCI repository online.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. M. Weiss and N. Indurkhya, *Predictive Data Mining: A Practical Guide*, Elsevier Science, Amsterdam, Netherlands, 1998.

[2] B. Kochar and R. Singh Chhillar, "An effective data warehousing system for rfid using novel data cleaning, data transformation and loading techniques," *The International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 208–216, 2012.

[3] P. Santhi and V. Murali Bhaskaran, "Performance of clustering algorithms in healthcare database," *International Journal for Advances in Computer Science*, vol. 2, no. 1, pp. 26–31, 2010.

[4] A H. Wahbeh, Q. A. Al-Radaideh, M. N. Al-Kabi, and E M. Al-Shawakfa, "A comparison study between data mining tools over some classification methods," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, pp. 18–26, 2011.

[5] M. Khan, S. S. Khan, and M. D. Awan, "Comparative exploration of features for data mining results by legend navigation interactive technique," *International Journal of Database Theory and Application*, vol. 9, no. 9, pp. 49–58, 2016.

[6] M. Khan, S. S. Khan, K. Ullah, and G. Ullah, "Evaluating interactive visualization techniques on small touch screen devices," *International Journal of Grid and Distributed Computing (IJGDC)*, vol. 12, no. 02, pp. 31–48, 2019.

[7] M. Khan, A. Shah, and I. Ahmad, "Framework for interactive data mining results visualization on mobile devices,"

International Journal of Database Theory and Application, vol. 7, no. 4, pp. 23–36, 2014.

[8] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, Amsterdam, Netherlands, 2011.

[9] Y. Abouelnaga, "San francisco crime classification," arXiv preprint arXiv:1607.03626, 2016.

[10] T. A. Shen, W. Wang, and S. Chyou, *San francisco Crime Classification*, University of California, California, USA, 2015.

[11] U. Saeed, M. Sarim, A. Usmani, A. Mukhtar, S. Abdul Basit, and S. Kashif Riffat, "Application of machine learning algorithms in crime classification and classification rule mining," *Research Journal of Recent Sciences*, vol. 4, no. 3, pp. 106–114, 2015.

[12] S. Shojaee, A. Mustapha, Fatimah Sidi, and M. A. Jabar, "A study on classification learning algorithms to predict crime status," *International Journal of Digital Content Technology and its Applications*, vol. 7, no. 9, p. 361, 2013.

[13] R. Iqbal, M. A. Azmi Murad, A. Mustapha, P. H. Shariat Panahy, and N. Khanahmadliravi, "An experimental study of classification algorithms for crime prediction," *Indian Journal of Science and Technology*, vol. 6, no. 3, pp. 4219–4225, 2013.

[14] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia*, vol. 18, no. 60, pp. 1–8, 2006.

[15] K. M. Leung, *Naive Bayesian Classifier*, Polytechnic University Department of Computer Science/Finance and Risk Engineering, Hong Kong, China, 2007.

[16] A. Onan, "An ensemble scheme based on language function analysis and feature engineering for text genre classification," *Journal of Information Science*, vol. 44, no. 1, pp. 28–47, 2018.

[17] A. Onan, "Ensemble of classifiers and term weighting schemes for sentiment analysis in Turkish," *Scientific Research Communications*, vol. 1, no. 1, 2021.

[18] A. Onan, S. Korukoğlu, and H. Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification," *Expert Systems with Applications*, vol. 62, no. 1–16, 2016.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] A. Onan, "Hybrid supervised clustering based ensemble scheme for text classification," *Kybernetes*, vol. 46, no. 2, pp. 330–348, 2017.

[21] A. Onan, S. Korukoğlu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification," *Information Processing & Management*, vol. 53, no. 4, pp. 814–833, 2017.

[22] M. A. Toçoğlu and A. Onan, *International Conference on Intelligent and Fuzzy Systems*, pp. 1693–1700, Springer, Berlin, Germany, 2020.

[23] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, no. 7, pp. 1545–1588, 1997.

[24] Xi Xia, W. Zhao, X. Rui et al., "A comprehensive evaluation of air pollution prediction improvement by a machine learning method," in *Proceedings of the 2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI)*, November 2015.

[25] R. Caruana and A. Niculescu-Mizil, "Data mining in metric space: an empirical analysis of supervised learning performance criteria," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 69–78, ACM, Seattle WA USA, Auguest 2004.

Hindawi
Complexity
Volume 2022, Article ID 1845571, 11 pages
https://doi.org/10.1155/2022/1845571

WILEY | Hindawi

Research Article

Prediction of Students' Performance Based on the Hybrid IDA-SVR Model

Huan Xu

Department of Public Teaching, Hefei Preschool Education College, Hefei 230013, China

Correspondence should be addressed to Huan Xu; xhhjy001@126.com

Received 28 June 2021; Revised 25 October 2021; Accepted 26 October 2021; Published 18 February 2022

Academic Editor: Murari Andrea

Students' performance is an important factor for the evaluation of teaching quality in colleges. The aim of this study is to propose a novel intelligent approach to predict students' performance using support vector regression (SVR) optimized by an improved duel algorithm (IDA). To the best of our knowledge, few research studies have been developed to predict students' performance based on student behavior, and the novelty of this study is to develop a new hybrid intelligent approach in this field. According to the obtained results, the IDA-SVR model clearly outperformed the other models by achieving less mean square error (MSE). In other words, IDA-SVR with an MSE of 0.0089 has higher performance than DT with an MSE of 0.0326, SVR with an MSE of 0.0251, ANN with an MSE of 0.0241, and PSO-SVR with an MSE of 0.0117. To investigate the efficacy of IDA, other parameter optimization methods, that is, the direct determination method, grid search method, GA, FA, and PSO, are used for a comparative study. The results show that the IDA algorithm can effectively avoid the local optima and the blindness search and can definitely improve the speed of convergence to the optimal solution.

1. Introduction

In recent years, computer technology has been widely used in the field of education. The prediction of students' academic performance has always been an important part of education. At present, students' performance is still the main standard to measure students' level of knowledge acquisition, and an important factor to judge the teaching quality of schools and teachers. With the increase of enrollment scale, the growth of teachers and the number of students is out of proportion, which affects the teaching quality and students' performance. Therefore, it is very important to accurately predict students' performance in education management. The prediction of students' performance can guide teachers to adjust students' learning behavior in time and improve students' performance.

The common performance prediction methods can be divided into two categories. The first is to establish statistical models, such as multivariate linear regression model and sparse factor analysis model. Sravani and Bala [1] predicted the students' performance based on a linear regression model. The second is based on data-driven performance prediction methods, such as logistic regression (LR) [2], Naive Bayes (NB) [3], decision tree (DT) [4], artificial neural network (ANN) [5], support vector regression (SVR), and so on. These methods do not require the participation of professionals but only extract the model from the relevant data. Table 1 shows the comparison of common machine learning methods. Table 2 shows the time and space complexity of common machine learning methods. As shown in Table 2, in small samples, SVR performs well in both time and space complexity. Borkar and Rajeswari [6] used education data mining and artificial neural network to predict the students' academic performance. Ghorbani and Ghousi [7] compared the performance of various machine learning such as random forest, k-nearest neighbor, support vector machine, and decision tree in predicting students' performance.

The students' performance is influenced by a variety of studying behaviors and varies greatly from individual to individual. Therefore, the traditional statistical model may be ineffective at some time. The data-driven approach

Table 1: Comparison of common machine learning methods.

Model	Advantages	Disadvantages
Logistic regression	1. Simple calculation and fast speed 2. Avoid overfitting through regularization	The performance is poor when faced with the multivariate or nonlinear decision boundary
Naive Bayes	Perform well on small-scale data	Very sensitive to the expression of input data
Decision tree	1. Able to apply to samples with missing attribute values 2. Strong robustness to outliers	Easy to overfit
Artificial neural networks	Perform well on nonlinear data	1. Long training time 2. The computational complexity is proportional to the network complexity
Support vector regression	1. Strong generalization ability 2. Can apply to high-dimensional nonlinear data 3. Low computational complexity	Sensitive to the selection of parameters and kernel function

Table 2: The time and space complexity of common machine learning methods.

Model	Time complexity	Space complexity
Logistic regression	$O(n * m)$	$O(m)$
Naive Bayes	$O(n * m * c)$	$O(m * c)$
Decision tree	$O(n * \log(n) * d)$	$O(p)$
Artificial neural networks	$O(t * \sum n_1 n_2 + n_2 n_3 + \cdots)$	$O(t * \sum (n_1 n_2 + n_2) + (n_2 n_3 + n_3) + \cdots)$
Support vector regression	$O(n^2)$	$O(n_{SV})$

attempts to predict the students' performance directly from the student behavior data. The establishment of the data-driven students' performance prediction model only needs to collect enough performance data. Second, among the common data-driven models, SVR is more suitable for analyzing the student behavior data from Tables 1 and 2. Therefore, SVR is selected to predict the students' performance in this paper.

Note: n is the number of the training set, m is the dimensions of the sample, c is the number of categories of Naive Bayes, p is the number of nodes in the tree, n_i is the number of neurons in i^{th} layer, d is the maximum depth of the decision tree, t is the training times, p is the number of interneurons, and n_{SV} is the number of the support vectors in SVR.

Recent advances achieved in common machine learning methods are as follows. Zhou et al. [8] proposed a novel graph-based ELM (G-ELM) for imbalanced epileptic EEG signal recognition. Zhang et al. [9] combined deep learning-based image recognition methods and serological specific indicators for diagnosis of atrophic gastritis (AG). Yan et al. [10] proposed an improved early distinctive shapelet classification method for early classification on time series. Bai et al. [11] classified time series based on multifeature dictionary representation and ensemble learning. Ramanan et al. [12] developed a learning algorithm based on functional-gradient boosting methods for logistic regression, and the empirical evaluation on standard data sets demonstrated the superiority of the proposed approach over other methods for learning LR. Zhang et al. [13] proposed attribute and instance weighted Naive Bayes (AIWNB), and the experimental results validated that it indeed improved the prediction accuracy of NB. Schidler and Szeider [14] proposed the SAT-based decision tree method by combining heuristic and exact methods in a novel way, which successfully decreased the depth of the initial decision tree in almost all cases. Khoo et al. [15] applied artificial neural networks to solve parametric PDE problems, and the simplicity and accuracy of the approach are demonstrated through notable examples. Cheng and Lu [16] developed an adaptive Bayesian support vector regression model for structural reliability analysis, and the proposed method outperformed other methods for medium-dimensional problems. Each method has its own advantages in a specific area. Through the analysis of Tables 1 and 2, support vector regression has better learning performance for the problem in this paper. It overcomes the requirement of traditional methods for large samples and can solve the problem of small sample and nonlinearity. In this paper, SVR is used to predict the students' performance.

The data used in the prediction includes two attributes. One is the students' previous performance, and the other is the students' basic behavior attributes, including the students' age, gender, attendance rate, self-study frequency, library access records, and so on. Bunkar et al. [17] used students' class test scores, seminar scores, homework scores, class attendance, and lab work to predict students' scores at the end of the semester. The second attribute often contains many redundant features, which may have a bad effect on the computational complexity and prediction accuracy of the model. Therefore, it is necessary to remove the redundant information before detecting product quality, and feature selection is a crucial method to deal with such a problem. Feature selection is an important preprocessing step for many high-dimensional quality classification problems [18]. With the increase of the number of features,

the search space of the feature subset grows exponentially. Most traditional feature selection algorithms are of low efficiency, so many scholars turn to using intelligent algorithms with stronger search ability, such as genetic algorithm [19], particle swarm optimization [20], and so on.

However, to obtain satisfactory prediction accuracy, it is not only related to the input characteristics of SVR but also closely related to the selection of SVR model parameters. The empirical method and the grid search method are the common SVR parameter selection methods. The empirical method is too subjective and the grid search method is time-consuming. In addition, the above two methods can only modify the parameters individually, and cannot achieve collaborative optimization among the parameters. At present, more and more scholars have applied the intelligent optimization algorithm to parameter selection of the SVR model. Luo et al. [21] proposed a novel artificial intelligence approach to predict the vertical load capacity of driven piles in cohesionless soils using SVR optimized by genetic algorithm (GA). Huang et al. [22] applied SVR to predict the strength of steel fibre reinforced concrete and used the firefly algorithm (FA) to tune the hyperparameters of SVR. Liu et al. [23] analyzed surface acoustic wave (SAW) yarn tension sensor's measured data by SVR and used the PSO algorithm to optimize the hyperparameters of SVR. Intelligent algorithms have been proved to be effective in solving parameter optimization problems. It does not depend on the specific domain of the problem and has strong robustness to the various types of problem. DA algorithm is an effective global optimization algorithm. After the duel between individuals, individuals continue to evolve and get closer to the optimal solution to the problem. Therefore, the DA algorithm is selected to optimize the SVR model parameters and the features collaboratively in this paper.

The rest of the paper is organized as follows. In Section 2, the IDA-SVR model is established. In Section 3, a real example about the academic performance of students is given to illustrate the proposed model. Finally, we summarize this paper and put forward future research directions in Section 4.

2. Methodology

This section will introduce the necessary background knowledge and the proposed model. First, the SVR model and DA algorithm are elaborated. Next, the proposed IDA-SVR model is described in detail.

2.1. Support Vector Regression. SVR is the application of a support vector machine (SVM) in regression learning. Suppose $(x_1, y_1), \ldots, (x_n, y_n)$, $x_i \in R^m, y_i \in R$, are the sample data. Such a linear function, namely SVR function, is as follows:

$$f(x) = \omega^T \varphi(x) + b, \tag{1}$$

where $\omega = (\omega_1, \omega_2, \ldots, \omega_m)^T$ is a vector normal to the maximum-margin hyperplane and b is the deviation. $\varphi(\cdot)$ is a nonlinear mapping.

The problem can be treated as the following optimization problem:

$$\min_{\omega,b,\varepsilon} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\left(\xi_i^+ + \xi_i^-\right)$$

$$\text{s.t.} \begin{cases} w^T\varphi(x) + b - y_i \le \varepsilon + \xi_i^+, \\ y_i - \omega^T\varphi(x) - b \le \varepsilon + \xi_i^-, \\ \xi_i^+, \xi_i^- \ge 0, \quad i = 1, 2, \ldots, l, \end{cases} \tag{2}$$

where C is the regularization factor and ξ_i^- and ξ_i^+ are slack variables representing lower and upper constraints on the outputs of the model. ε is a positive constant. Errors are calculated only if the deviation between $f(x)$ and y_i is greater than ε.

The above problem is a quadratic problem with linear constraints, so the Kalush–Kun–Tuck (KKT) optimal conditions are necessary and sufficient. The solution, which can be obtained from the dual problem, is a linear combination of a subset of sample points denominated support vectors (s.v.) as follows:

$$\omega = \sum_{\text{S.V.}} \beta_i \varphi(x_i) \Rightarrow f_{\omega,b}(x) = \sum_{\text{S.V.}} \beta_i \langle \varphi(x_i), \varphi(x) \rangle + b. \tag{3}$$

Let $\kappa(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$, which is called the kernel function. It can map points from low- to high-dimensional space. Then, equation (3) can be rewritten as follows:

$$\omega = \sum_{\text{S.V.}} \beta_i \varphi(x_i) \Rightarrow f_{\omega,b}(x) = \sum_{\text{S.V.}} \beta_i k\langle x_i, x \rangle + b. \tag{4}$$

Kernel selection is one of the key technologies to improve the ability of SVR. This paper uses the radial basis function as shown in the following equation:

$$\kappa\left(x_i, x_j\right) = e^{-\sigma\left\|x_i - x_j\right\|^2}. \tag{5}$$

The prediction accuracy of the SVR model depends on the good settings of the hyperparameters C and ε and the kernel parameter σ. Therefore, the selection of the parameters is an important issue. Next, we will introduce the IDA algorithm to optimize SVR parameters.

2.2. DA Algorithm. Duelist algorithm (DA) is a new algorithm based on a genetic algorithm proposed by Biyanto [24] from the perspective of human combat and learning ability. The process of the DA algorithm is shown in Figure 1.

2.2.1. Encoding. In this paper, the encoding of the DA algorithm is composed of parameters (C, ε, σ) and feature subsets, as shown in Figure 2.

$b_C^1 \sim b_C^{n_C}$, $b_\varepsilon^1 \sim b_\varepsilon^{n_\varepsilon}$, $b_\sigma^1 \sim b_\sigma^{n_\sigma}$, and $b_f^1 \sim b_f^{n_f}$ are the binary strings of parameters C, ε, σ, and features, respectively. $n_C, n_\varepsilon, n_\sigma$, and n_f are the numbers of binary digits of C, ε, σ, and features, respectively.

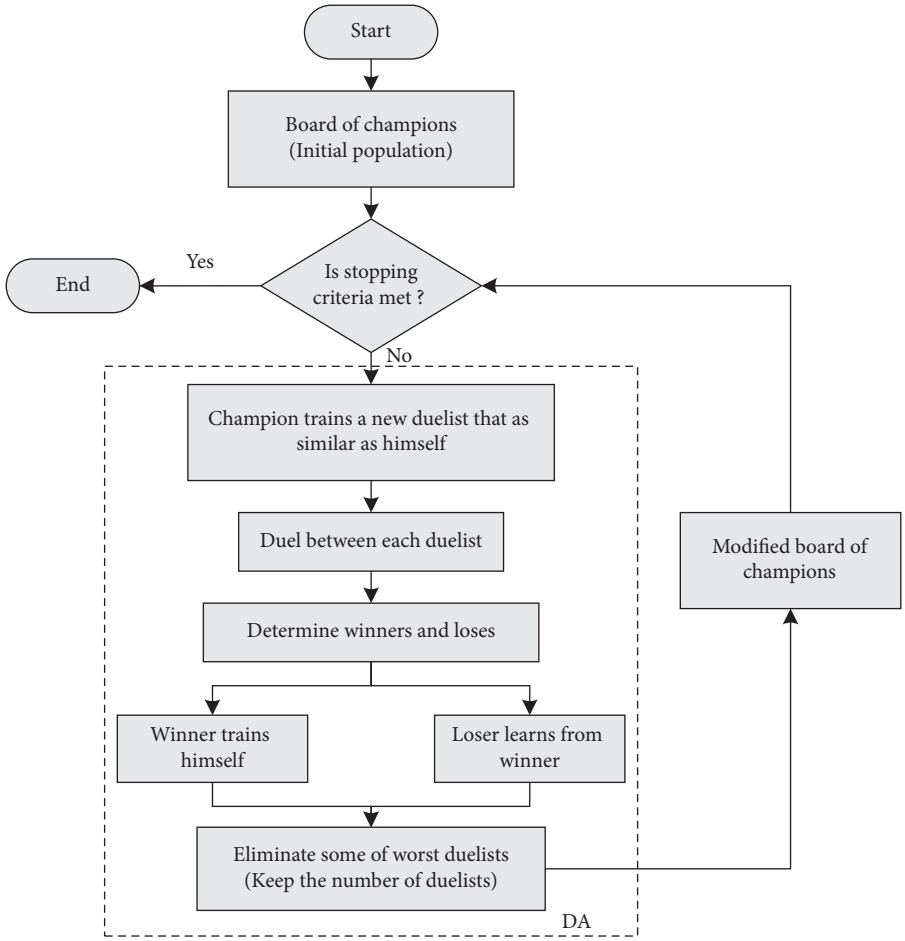


Figure 1: Flowchart of DA algorithm.

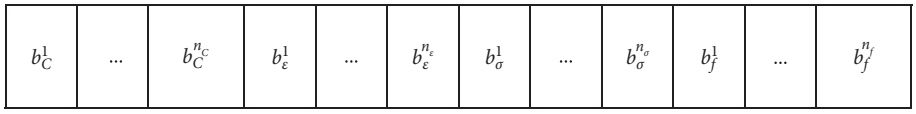


Figure 2: Encoding of DA-VNS algorithm.

2.2.2. Fighting Capability Function. In this paper, we take the mean squared error (MSE) as the fighting capability. Let y_i be the observed values and $\widehat{y}_i$ be the predicted values, then

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2. \tag{6}$$

Here, n is the number of samples. A smaller MSE value indicates a better fighting capability.

2.2.3. Duel Scheduling between Duelists. DA algorithm optimizes the solution by one-to-one dueling between duelists. The pseudocode of the duel process is shown in Algorithm 1.

2.2.4. Duelist's Improvement. In this step, both the loser and the winner need to improve their fighting capabilities. The pseudocode of duelist's improvement is displayed in Algorithm 2.

2.3. IDA-SVR Model. After in-depth analysis, it is found that the DA algorithm has four shortcomings.

(1) The value of the initial solution is generated randomly. The random process cannot guarantee the uniform distribution of the initial population and the quality of the individual. Some of the solutions are far away from the optimal solution.

(2) By analyzing the whole process of the DA algorithm, we can conclude that the luck coefficient has a great impact on the performance of the algorithm. The larger the luck coefficient, the more random the new individual. It follows that the fitness fluctuation of the solution becomes larger, and the speed of convergence to the optimal solution becomes slower. On

the contrary, the smaller the luck coefficient is, the weaker the randomness of the new individual will be, which leads to the slower speed of obtaining the optimal solution. Therefore, the setting of the luck coefficient is crucial to the effectiveness of the algorithm.

(3) Each duelist is categorized into winner and loser after the duel. To improve duelists' fighting capability, each loser is trained by learning from the winner, and winners evolve on their own. Therefore, it can be seen that the loser's improvement is based on the information exchange between two individuals, which will lead to the slow convergence of the algorithm.

(4) Like other swarm intelligent optimization algorithms, the DA algorithm is also prone to local optimization and low search accuracy in the search process.

In view of the shortcomings of the DA algorithm, this paper has made improvements in the following aspects:

(1) The chaotic sequence is used to initialize the population. It not only improves the diversity of the population by using chaos but also does not change the randomness of the optimization algorithm during initialization. There are various mathematical models for generating chaotic sequences. In this paper, a logistic equation is used to construct chaotic sequences as follows:

$$x(t+1) = \mu x(t)(1 - x(t)), \quad t = 0, 1, 2, \ldots, \tag{7}$$

where μ is the control parameter. When $0 < x(0) < 1$ and $\mu = 4$, the logistic equation is in a complete chaotic state. In this case, $x(t)$ is chaotic and in the interval $(0, 1)$. Given the initial value $x(0) \in (0, 1)$, the time series $x(1), x(2), \ldots,$ can be generated.

(2) According to the statistics principle, there are more chances to search for more optimal solutions around the optimal solution. That is, we can set the luck coefficient a little bigger at the beginning. Then the solution will be more random and easier to find the optimal solution. When close to the optimal solution, a small luck coefficient allows the algorithm to search for more optimal solutions around it. Therefore, based on the above analysis, we define the adaptive luck coefficient c:

$$c = \frac{i_{\max}}{\lambda(i_{\max} + i + 1)}. \tag{8}$$

Here, $i_{\max}$ is the total number of iterations, i is the current iteration number, and λ is the adjustment coefficient of step length, which is determined according to the feasible regions of different optimization problems.

(3) For the loser's improvement, each loser is trained by learning from one of the winners after a duel. The roulette method is used to determine the winner that the loser will learn from.

(4) The chaotic sequence search method is used to generate the neighborhood solutions. The randomness and ergodicity of chaotic variables can make the algorithm jump out of the local optimization. In this way, the global searching ability of the algorithm is improved. First, the chaotic sequence is generated by equation (7) based on the optimal position currently. Then the optimal position of the chaotic sequence is used to replace the position of a duel. Through the above steps, neighborhood solutions of the local optimal solution can be generated in the iteration, which can help the current solution escape from the local optimal solution.

The four strategies are to improve the algorithm in different steps without overlapping. Strategy (a) is an improvement on the initial value. Strategy (b) makes the lucky coefficient adjust adaptively and enables the algorithm to converge to the optimal solution faster. Strategy (c) is to increase the diversity of solutions in the duelist improvement step. Strategy (d) can make the newly generated solution jump out of the local optimum. The above four strategies guarantee the prediction accuracy and the speed of convergence to the optimal solution of the algorithm together.

Then, we will use the improved DA algorithm to optimize the parameters of SVR. Figure 3 shows the flowchart of the hybrid IDA-SVR model developed in this research work.

3. Experimental Study

In this paper, the mathematical performance data of 240 students from five classes in grade two in a vocational college in Hefei are selected as the research object. Among them, 180 samples are used as the training data and the remaining 60 samples are used as the testing data. Each sample contains 18 features, as shown in Table 3.

3.1. Data Preparation. To eliminate the influence of the different dimensions on the numerical values, further normalization of data is needed. The normalization formula is as follows:

$$a'_{ij} = \frac{a_{ij} - a_{i\min}}{a_{i\max} - a_{i\min}}, \tag{9}$$

where a_{ij} is the initial sample data to be normalized and $a_{i\min}$ and $a_{i\max}$ are the minimum and maximum values in the column sample values, respectively.

3.2. Experimental Study. All experiments are run on Intel Core i5-1035 8 GB, the Microsoft Windows 10 operating system, and the development environment of Python 3.6.6, PyCharm 2021.1. The parameter settings are shown in Table 4.

```
(1) Duelist A and B, Luck_coeff
(2) FC = Fighting capability; LC = Luck_coeff
(3) A(Luck) = A(FC) ∗ (LC + (random(0, 1) ∗ LC));
(4) B(Luck) = B(FC) ∗ (LC + (random(0, 1) ∗ LC));
(5) if ((A(FC) + A(Luck)) > = (B(FC) + B(Luck)))
(6) A(Winner) = 1;
(7) B(Winner) = 0;
(8) else
(9) A(Winner) = 0;
(10) B(Winner) = 1;
(11) end if
```

Algorithm 1: Determination of the winner and the loser.

```
(1) Duelist A and B, Duelist_length, Prob_innovate; Prob_learn
(2) if A(Winner) = 1
(3) for i = 1:(Duelist_length)
(4) r = random(0,1)
(5) if r < Prob_innovate
(6) if A[i] = 1
(7) A[i] = 0
(8) else A[i] = 1
(9) end if
(10) end if
(11) end for
(12) else
(13) fori = 1: (Duelist_length)
(14) r = random(0, 1)
(15) if r < Prob_learn
(16) B[i] = A[i]
(17) end if
(18) end for
(19) end if
```

Algorithm 2: Duelist's Improvement.

We apply the IDA-SVR model to predict the students' performance, and the results are shown in Figure 4 and Table 5.

By analyzing the students' performance prediction model based on SVR, it is found that the number of support vectors used in the prediction model is 153. It can be seen that only the data of 153 students is needed to realize the prediction of performance in the sample set composed of 240 students. Due to the limited length of the article, two examples are listed below for analysis.

Take the example of a support vector with index number 36, which is a student with a performance of 90. In the model, the top five important learning behavior features are the number of assignments submitted, the average number of hands raised, the time of study this course outside the class, the frequency of distraction, and the number of absenteeism. The corresponding feature weight vector of this sample is [1.0335, 0.8327, 0.8133, −0.7415, −0.5448]. It can be seen that the number of assignments submitted has a great influence on the performance. Next, we will look at a student with low performance.

Now we analyze the support vector with index number 228, which is a student with lower performance of 72. In the model, the top five important learning behavior features are work as a student cadre, the number of absenteeism, the frequency of distraction, the number of assignments submitted, and the average number of hands raised. The corresponding feature weight vector of this sample is [1.2814, −0.9738, −0.7122, 0.5415, 0.4388]. Compared with the student with index number 36, this student has a lower weight in the number of assignments submitted and the average number of hands raised. Therefore, the above two features have a significant impact on students' achievement. In addition, absenteeism is one of the main reasons for a student's poor performance.

Based on the above model analysis, the following teaching suggestions are proposed. Teachers should explore targeted measures to improve students' performance based on the results of experimental analysis and the actual condition. To improve students' performance, teachers should reasonably use teaching methods in the teaching process to stimulate students' learning engagement and motivation.

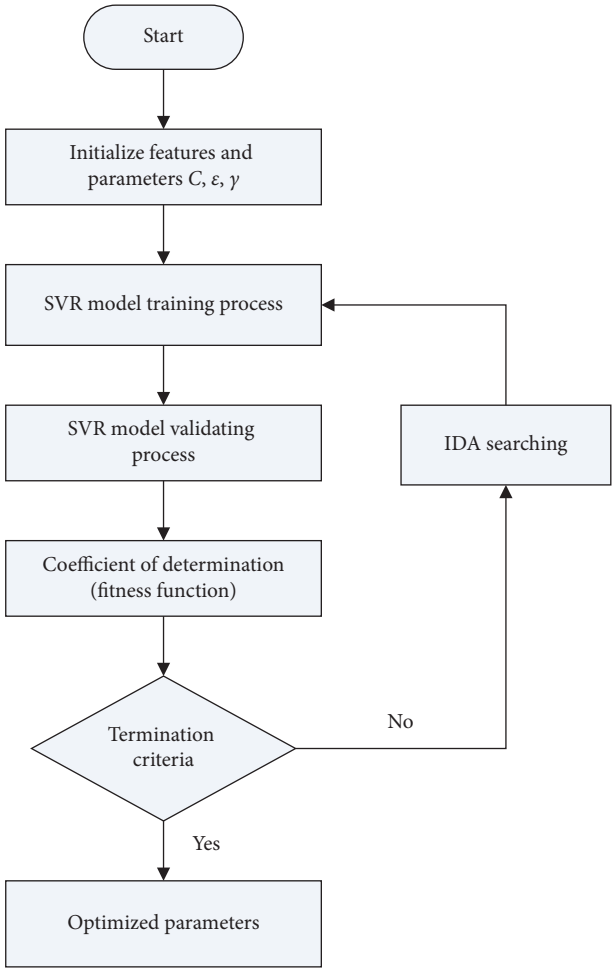


Figure 3: Flowchart of the IDA-SVR model.

Figure 5 shows the accumulative Pearson's correlation coefficient between the actual and predicted values. As shown in Figure 5, with the number of predicted students' performance increases, we can see a strong correlation that exists between the actual and predicted values. The Pearson's correlation coefficient of the total testing data is 0.9214. Figure 6 describes the degree of fluctuation of errors between the actual and predicted values. As shown in Figure 6, the errors are between −0.2 and 0.2. In addition, there are almost no outliers. From the comprehensive analysis of Figures 4–6, we can conclude that the proposed method has a good performance in predicting students' performance. (All the data are normalized.)

To demonstrate the superiority of the IDA-SVR model, four other models, basic SVR, PSO-SVR, decision tree (DT), and artificial neural networks (ANN), are selected to compare with the IDA-SVR model. The constructed ANN model includes three hidden layers with the ReLU activation function. In the decision tree algorithm, the maximum depth of the decision tree is set to 5. Table 6 recapitulates the results of the proposed method and the comparative methods on the students' performance prediction problem. We find that the proposed model outperforms the selected comparative methods in prediction accuracy.

Note: pop is the population size of the intelligent algorithm and iter is the number of the iteration times of the intelligent algorithm.

As shown in Table 6, the proposed IDA-SVR model has the highest time complexity and running time, which can be said that it trades time for accuracy. How to reduce the time complexity and running time is a future research direction for us. First, the runtime may be further reduced by exploring more computational-efficient SVR algorithms and a faster parameter tuning mechanisms. Moreover, parallelization techniques and methods are worth exploring and utilizing to improve learning performance and reduce the computational cost in the model.

3.3. Comparative Experiment. We design a set of comparative experiments to evaluate the performance of the IDA algorithm in optimizing the parameters of SVR for students' performance prediction problem. Parameter optimization methods, that is, the direct determination method [25], grid search method [26], genetic algorithm (GA) [21], firefly algorithm (FA) [22], and particle swarm optimization (PSO) [23], with SVR are used to compare with the IDA-SVR. The results of the direct determination method, grid search method, GA-SVR, FA-SVR, and PSO-SVR are reported in Table 7.

Table 7 displays the results of the prediction performance of six algorithms. The direct determination method has the worst classification accuracy among these algorithms. The possible reason is that it requires high data quality, and it is not suitable for students' performance data. The grid search method is to divide the parameters to be optimized into grids in a certain spatial range and then search for the optimal parameters by traversing all points in the grid. It has a good effect in a small interval, but a poor effect in a large interval or multiparameter case. GA, FA, PSO, and IDA are all heuristic algorithms. To investigate the efficacy of IDA algorithms in optimizing the SVR parameters, we compare the running results after 500 iterations of GA-SVR, FA-SVR, PSO-SVR, and IDA-SVR, respectively. The results are shown in Figures 7 and 8.

As shown in Figure 7, in general, the solutions obtained by IDA-SVR is superior to GA-SVR, FA-SVR, and PSO-SVR. The solutions obtained by PSO-SVR are mostly clustered around the optimal solution. In addition, the solutions obtained by GA-SVR and FA-SVR are more concentrated than those obtained by DA and DA-VNS algorithms. In other words, the search scope of the GA and FA algorithms is smaller than the other two algorithms. The IDA algorithm has the largest search range, which can effectively avoid the local optima and the blindness search.

Figure 8 shows the evolution of the minimum MSE for GA-SVR, FA-SVR, PSO-SVR, and IDA-SVR over 500 iterations. From Figure 8, we can see that the PSO and IDA algorithms can make the solutions converge to the optimal solution continuously. Compared with PSO, the speed of convergence to the optimal solution of IDA is faster. In

Table 3: The features of each student sample.

Category	No.	Feature	Value
Basic information	1	Sex	{1,2}
	2	Native place	{1,2,3,4,5}
	3	Semester	{1,2}
	4	Education level of parents	{1,2,3,4,5}
	5	Work as a student cadre	{0,1}
Interest	6	Interest in the course	{1,2,3,4}
	7	The degree of keeping up with the class	{1,2,3,4}
	8	Learning initiative	{1,2,3,4}
Behavior in class	9	Number of absenteeism	[0,25]
	10	Frequency of distraction	[5,47]
	11	Average number of hands raised	[0,12]
	12	Number of questions answered	[0,9]
	13	Number of assignments submitted	[0,20]
	14	Number of interactions between teachers and students	[0,17]
	15	Number of group discussions attended	[0,10]
Behavior outside class	16	Time of study this course outside the class	[5,65]
	17	Study extracurricular material time	[0,12]
	18	Online viewing time	[0,60]

Table 4: List of preset parameters in DA-VNS.

Parameters	Value
Population size	100
Iteration times	500
Search range of penalty parameter C	$[10^{-3}, 10^{3}]$
Allowable error ε	$[10^{-3}, 10^{-1}]$
Search range of kernel width σ	$[10^{-6}, 10^{1}]$
Luck coefficient (LC)	0.1

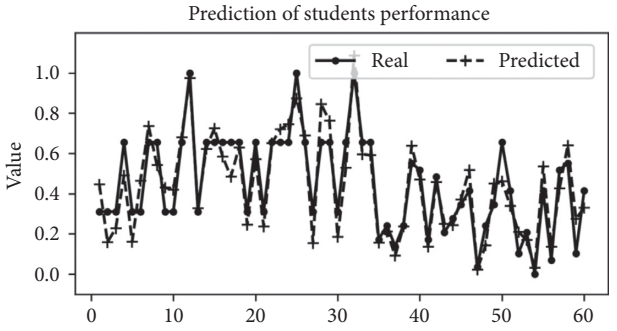


Figure 4: The prediction results of students' performance based on the IDA-SVR model.

Table 5: The results of IDA-SVR in students' performance prediction.

Model	MSE	Selected features
IDA-SVR	0.0092	6

addition, among the four algorithms, the IDA algorithm takes the least number of iterations to find the optimal solution. It successfully finds the global optimal solution of

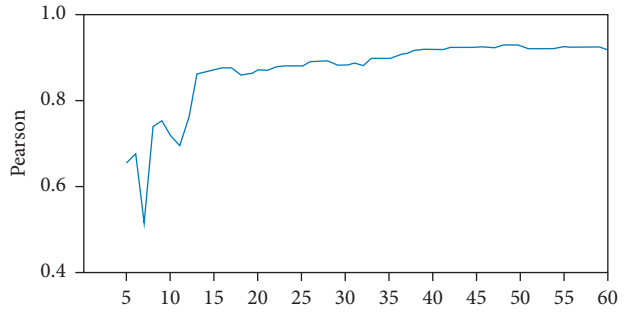


Figure 5: The accumulative Pearson's correlation coefficient between the actual value and the predicted value.

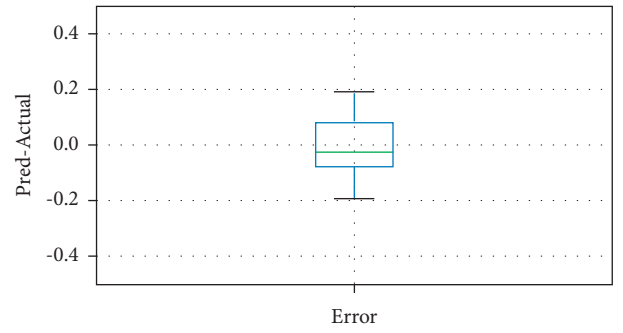


Figure 6: Boxplot representing the difference between the actual value and the predicted value.

input feature set and hyperparameters within 200 iterations when GA uses 380 iterations, FA 338 iterations, and PSO 372 iterations. After the above analysis, the IDA algorithm can definitely improve the speed of convergence to the optimal solution.

Table 6: Comparison of performance between IDA-SVR and other models.

Model	MSE	Time complexity	Running time (second)
SVR	0.0251	$O(n^2)$	0.33
PSO-SVR	0.0117	$O(\text{pop} * \text{iter} * n^2)$	90.58
DT	0.0326	$O(n * \log(n) * d)$	1.2
ANN	0.0241	$O(t * \sum n_1 n_2 + n_2 n_3 + \cdots)$	3.59
IDA-SVR	0.0092	$O(\text{pop} * \text{iter} * n^2)$	105.17

Table 7: Comparison of performance between IDA-SVR and other methods.

Method	MSE
Direct determination method [20]	0.227
Grid search method [21]	0.203
GA-SVR [21]	0.0168
FA-SVR [22]	0.0149
PSO-SVR [23]	0.0117
IDA-SVR	0.0092

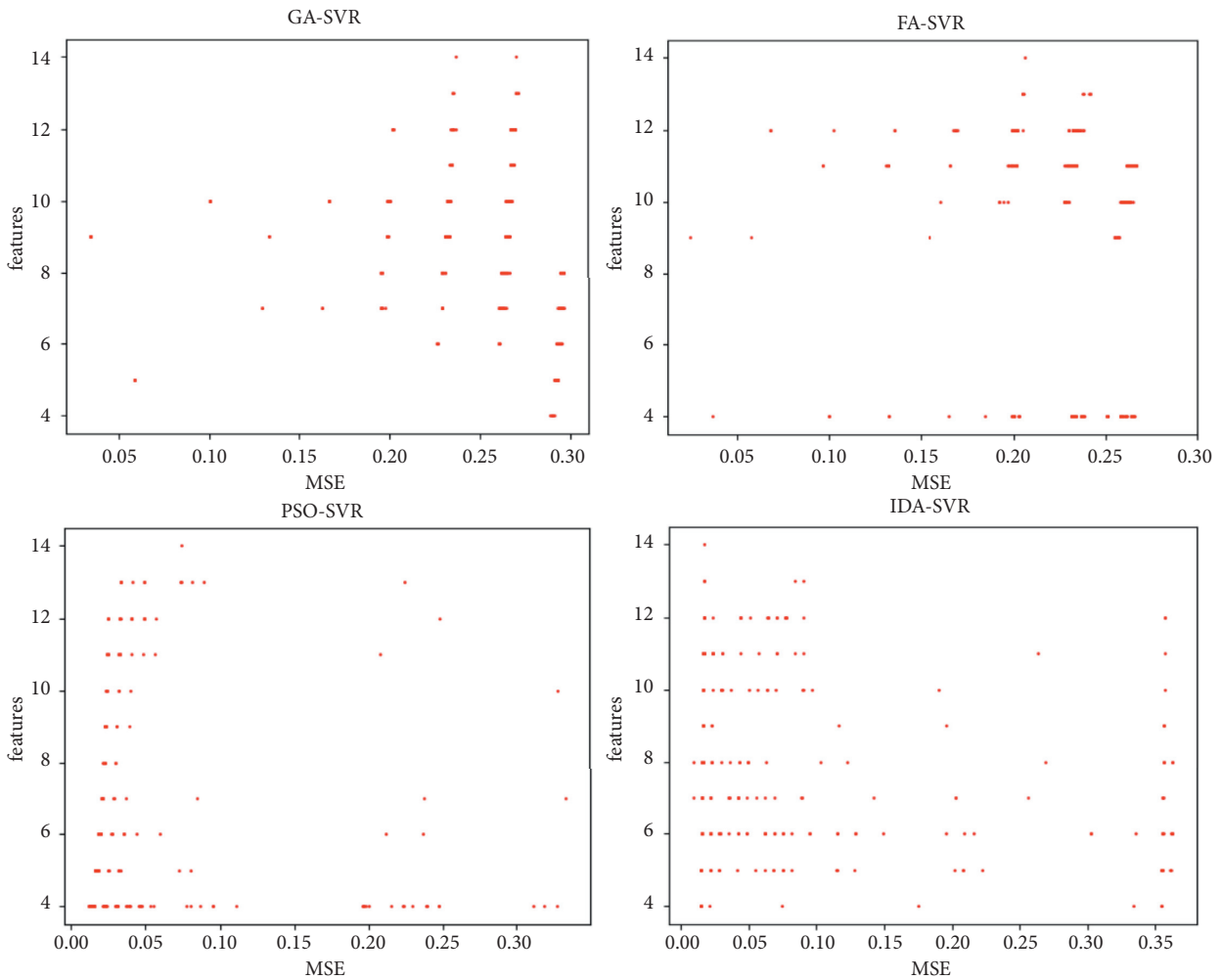


Figure 7: Running results after 500 iterations of GA-SVR, FA-SVR, PSO-SVR, and IDA-SVR.

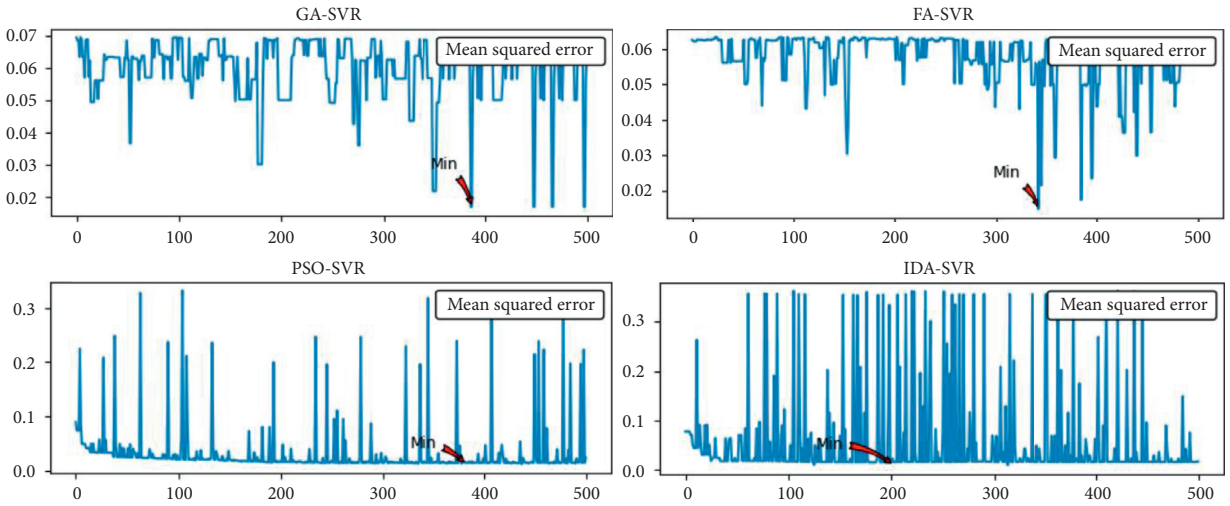


Figure 8: The evolution of the minimum MSE for GA-SVR, FA-SVR, PSO-SVR, and IDA-SVR over 500 iterations.

4. Conclusions

In this paper, a hybrid IDA-SVR model is proposed to predict the students' performance. The main contributions of this paper can be summarized as follows: (1) a novel intelligent approach is proposed to predict students' performance based on student behavior using support vector regression. Some experiments are conducted using the mathematics score data of students. The experimental results show that the proposed model has excellent performance in solving the prediction problem of students' performance. (2) The improved duel algorithm is designed to optimize the kernel parameters of SVR and select the features. Compared with other parameter optimization methods, the IDA algorithm can effectively avoid the local optima and the blindness search and can definitely improve the speed of convergence to the optimal solution.

The method proposed in this paper aims at solving students' performance prediction problem. However, it also can be applied to other problems in other fields. Because the proposed hybrid method is essentially a prediction algorithm for the small sample data with labels. It is applicable to any field that meets the above point, such as prediction of some economic indicators, environmental indicators, abnormal detection of ECG signals, diagnosis of circuit failures, and so on.

Although the proposed model performs well among many models, it still has some limitations. First, the improved DA algorithm has some instability. For example, the initial values of the parameters to be optimized are given randomly, and different initial values will have different effects on the results. In addition, even though the improved DA algorithm provides the possibility of global search, it cannot ensure that it converges to the global best. Second, SVR can get much better results than other algorithms on a small sample training set. But when the sample dimension is large, the time complexity of SVR will increase, which will greatly reduce the efficiency of the predictor. Third, the improved DA algorithm optimizes the parameters of SVR by training individuals on the training set and evaluating the scores on the testing set. The more iterations of optimization, the higher the accuracy. In other words, the proposed model trades time for accuracy to a large extent.

To solve the above limitations, our study can be extended in the following future research directions. With the development of computer technology, the number of layers of neural networks that can handle is increasing, and the performance of deep learning methods has surpassed machine learning in many fields. To improve the performance of SVR, it is necessary to improve the objective function, constraint conditions, and kernel function of the SVR model based on the problem itself.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] B. Sravani and M. M. Bala, "Prediction of student performance using linear regression," in *Proceedings of the 2020 International Conference for Emerging Technology (INCET)*, pp. 1–5, IEEE, Belgaum, India, June 2020.

[2] A. C. J. W. Janssens, Y. Deng, G. J. J. M. Borsboom, M. J. C. Eijkemans, J. D. F. Habbema, and E. W. Steyerberg, "A new logistic regression approach for the evaluation of diagnostic test results," *Medical Decision Making*, vol. 25, no. 2, pp. 168–177, 2005.

[3] I. Rish, "An empirical study of the naive bayes classifier," *Journal of Universal Computer Science*, vol. 1, no. 2, p. 127, 2001.

[4] J. R. Quinlan, "Induction on decision tree," *Machine Learning*, vol. 1, 1986.

[5] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application

in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, no. 5, pp. 717–727, 2000.

[6] S. Borkar and K. Rajeswari, "Attributes Selection for predicting students' academic performance using education data mining and artificial neural network," *International Journal of Computer Applications*, vol. 86, no. 10, pp. 25–29, 2014.

[7] R. Ghorbani and R. Ghousi, "Comparing different resampling methods in predicting students' performance using machine learning techniques," *IEEE Access*, vol. 8, pp. 67899–67911, 2020.

[8] J. Zhou, X. Zhang, and Z. Jiang, "Recognition of imbalanced epileptic EEG signals by a graph-based extreme learning machine," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5871684, 12 pages, 2021.

[9] J. Zhang, J. Yu, S. Fu, and X. Tian, "Adoption value of deep learning and serological indicators in the screening of atrophic gastritis based on artificial intelligence," *The Journal of Supercomputing*, vol. 77, pp. 8674–8693, 2021.

[10] W. Yan, G. Li, Z. Wu, S. Wang, and P. S. Yu, "Extracting diverse-shapelets for early classification on time series," *World Wide Web*, vol. 23, no. 6, pp. 3055–3081, 2020.

[11] B. Bai, G. Li, S. Wang, Z. Wu, and W. Yan, "Time series classification based on multi-feature dictionary representation and ensemble learning," *Expert Systems with Applications*, vol. 169, Article ID 114162, 2021.

[12] N. Ramanan, G. Kunapuli, and T. Khot, "Structure learning for relational logistic regression: an ensemble approach," *Data Mining and Knowledge Discovery*, vol. 35, pp. 2089–2211, 2021.

[13] H. Zhang, L. Jiang, and L. Yu, "Attribute and instance weighted naive Bayes," *Pattern Recognition*, vol. 111, Article ID 107674, 2021.

[14] A. Schidler and S. Szeider, "SAT-based decision tree learning for large data sets," in *Proceedings of the AAAI 21, the Thirty-Fifth AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2021.

[15] Y. Khoo, J. Lu, and L. Ying, "Solving parametric PDE problems with artificial neural networks," *European Journal of Applied Mathematics*, vol. 32, no. 3, pp. 421–435, 2021.

[16] K. Cheng and Z. Lu, "Adaptive bayesian support vector regression model for structural reliability analysis," *Reliability Engineering & System Safety*, vol. 206, Article ID 107286, 2021.

[17] K. Bunkar, U. K. Singh, B. Pandya, and R. Bunkar, "Data mining: prediction for performance improvement of graduate students using classification," in *Proceedings of the 2012 Ninth International Conference on Wireless and Optical Communications Networks (WOCN)*, pp. 1–5, IEEE, Indore, India, September 2012.

[18] B. Frénay, G. Doquire, and M. Verleysen, "Is mutual information adequate for feature selection in regression," *Neural Networks*, vol. 48, no. 6, pp. 1–7, 2013.

[19] S. Li, H. Wu, D. Wan, and J. Zhu, "An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 40–48, 2011.

[20] N. Allias, M. N. Ismail, and K. de Silva, "A hybrid gini PSO-SVM feature selection: an empirical study of population sizes on different classifier," in *Proceedings of the International Conference on Artificial Intelligence*, Kota Kinabalu, Malaysia, December 2013.

[21] Z. Luo, M. Hasanipanah, H. Bakhshandeh Amnieh, K. Brindhadevi, and M. M. Tahir, "GA-SVR: a novel hybrid data-driven model to simulate vertical load capacity of driven piles," *Engineering with Computers*, vol. 37, no. 2, pp. 823–831, 2021.

[22] Y. Huang, J. Zhang, F. Tze Ann, and G. Ma, "Intelligent mixture design of steel fibre reinforced concrete using a support vector regression and firefly algorithm based multi-objective optimization model," *Construction and Building Materials*, vol. 260, Article ID 120457, 2020.

[23] S. Liu, P. Xue, J. Lu, and W. Lu, "Fitting analysis and research of measured data of SAW yarn tension sensor based on PSO-SVR model," *Ultrasonics*, vol. 116, Article ID 106511, 2021.

[24] T. Biyanto, H. Fibrianto, E. Listijorini, and T. Budiati, "Duelist algorithm: an algorithm inspired by how duelist improve their capabilities in a duel," in *Proceedings of the Seventh International Conference on Swarm Intelegence*, Bali, Indonesia, 2015.

[25] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004.

[26] Y. Sun, S. Ding, Z. Zhang, and W. Jia, "An improved grid search algorithm to optimize SVR for prediction," *Soft Computing*, vol. 25, no. 7, pp. 5633–5644, 2021.

Hindawi
Complexity
Volume 2022, Article ID 9557722, 16 pages
https://doi.org/10.1155/2022/9557722

Research Article

Influential Nodes in the OBOR Fossil Energy Trade Network Based on D-S Theory: Detection and Evolution Analysis

Cuixia Gao,[1,2,3] **Simin Tao**,[2] **Kehu Li**,[2] **and Yuyang He**[4]

[1]*School of Environmental Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*
[2]*School of Mathematical Sciences, Jiangsu University, Zhenjiang, Jiangsu, China*
[3]*Center for Energy Development and Environmental Protection, Jiangsu University, Zhenjiang, Jiangsu, China*
[4]*School of Mechanical Engineering, Jiangsu University, Zhenjiang, Jiangsu, China*

Correspondence should be addressed to Cuixia Gao; cxgao@ujs.edu.cn

Received 10 September 2021; Revised 18 November 2021; Accepted 30 November 2021; Published 10 January 2022

Academic Editor: Gonzalo Farias

The structure formed by fossil energy trade among countries can be divided into multiple subcommodity networks. However, the difference of coupling mode and transmission mechanism between layers of the multirelationship network will affect the measurement of node importance. In this paper, a framework of multisource information fusion by considering data uncertainty and the classical network centrality measures is build. Then, the evidential centrality (EVC) indicator is proposed, by integrating Dempster–Shafer evidence theory and network theory, to empirically identify influential nodes of fossil energy trade along the Belt and Road Initiative. The initial result of the heterogeneity characteristics of the constructed network drives us to explore the core node issue further. The main detected evidential nodes include Russia, Kazakhstan, Czechia, Slovakia, Egypt, Romania, China, Saudi Arabia, and Singapore, which also have higher impact on network efficiency. In addition, cluster analysis discovered that resource endowment is an essential factor influencing country's position, followed by geographical distance, economic level, and economic growth potential. Therefore, the above aspects should be considered when ensuring national trade security. At last, the rationality and comprehensiveness of EVC are verified by comparing with some benchmark centralities.

1. Introduction

Fossil fuel is an essential resource for economic and social development. It has a vital bearing on the national economy and human survival. Although more and more voices are emphasizing low-carbon fuels, in particular, solar PV and wind, the dominant position of fossil energy cannot be eliminated over a short period, due to the fact that 100% renewable energy must go through a process [1, 2]. Therefore, the issue of fossil energy-related carbon emission reduction is still rigorous towards the current carbon neutrality target. In 2019, for instance, global energy-related CO_2 emissions reached 33.4 Gt, with 14,798 Mt in coal, 11,344 Mt in oil, and 7,250 Mt in natural gas, respectively [3].

As the embodiment of international division of labor, trade is a significant factor in explaining the change in carbon emissions in many countries [4]. Especially for developing countries, international trade is an important way to achieve their energy supply-demand balance and supply security. As the world's largest carbon emitter, China's continued dependence on imported crude oil and natural gas as well as its growing fossil energy demand is also expanding its influence on world energy markets. Simultaneously, China's pivotal role in international fossil energy trade market has only increased since the Belt and Road Initiative (BRI) in 2013.

Fossil energy trade is at the core of the BRI due to unevenly distributed resources. After seven years of construction, the initiative has become new solutions to improve global governance in the new era [5]. "China's Trade and Investment Cooperation under the Belt and Road Initiative" published by the Shanghai Academy of Social Sciences showed that exported-oriented fuel and mineral resources are the most common trading type along the "One

Belt One Road" (OBOR) nations which accounts for nearly 40 percent [6]. The remaining proved reserves for oil and gas under OBOR account for 58.7% and 77.7%, respectively. The production of oil and gas exceeds 50% of the world's [7]. In addition, seven of the world's top 10 oil and gas exporters are concentrated in the OBOR. As a result, it is beneficial for China to promote the construction of the initiative by clarifying the fossil energy trade patterns. Furthermore, it is necessary to better understand the pattern changes of fossil energy trade in OBOR countries to achieve national energy transition and carbon neutrality targets.

The trade relationship of fossil energy among the OBOR countries is firstly a system, involving a large number of countries and intricate links between countries. Secondly, each country plays a different role in trade, such as oil exporter, gas importer, coal net exporter, and oil and gas net importer. Thirdly, due to the influence of uncertainties including multienergy complementarity, energy structure adjustment, clean energy transitions, limitation by increasing aggregate demand, and renewable energy substitutability, countries in various roles or relationships are expected to interact with each other and evolve over time. Importantly, the participators are likely been influenced heterogeneously owing to their unique energy structure and trade structure. Thus, a comprehensively and systemic study by considering uncertainties is needed. Ultimately, the potential impact of uncertainties on fossil energy trade among OBOR countries is embedded into the model as "cognitive uncertainty" in this paper, because it is difficult and sometimes proves impossible to accurately judge what the impacts of these uncertainties are.

The primary contributions of this study can be summarized as follows:

(i) First, the constructed network is the first such measure to consider uncertainty in modeling fossil energy trade relationships.

(ii) Second, the proposed evidential centrality has benefits in terms of measurement, which allow it to provide a more comprehensively calculation from the perspective of local, global, and uncertainty.

(iii) Third, the top-ranked countries for fossil energy trade under OBOR are identified, and the possible underlying driving forces are exposed empirically.

The rest of this paper is organized as follows: Section 2 reviews the literature on fossil energy trade network and uncertainty analysis methods. Section 3 introduces a trace of preliminaries and describes the details of the proposed evidence-based method for identifying the influential nodes. Section 4 presents the data sources. Section 5 illustrates the empirical results. Finally, Section 6 draws the conclusions.

2. Literature Review

Most of the existing studies related to fossil energy trade are focusing on the world level and most developing countries. Moreover, developing countries are the main consumers, who have obvious contradiction between resource shortage and economic development, while the OBOR is an important group in global trade. With the increasing importance of regional trade, literature studies on OBOR countries and their trade relations is not rare. Exemplarily, Zhang revealed the status and prospects of the oil and gas trade among the OBOR countries and discovered that the general trading relationship between China and the other participating countries was closely related to their gas and oil trade [8].

On the other hand, complex network analysis has been proposed for exploring the patterns of fossil energy trade. Table 1 lists the recent ones that used complex network theory in the study of international fossil energy trade. In all, the existing studies typically addressed single-layer networks in fossil energy trade, such as separate oil network, natural gas network, or coal network. Certainly, a few researchers consider fossil fuels as a whole, but the data used represent a linear sum of the observed values between countries. For example, Gao et al. [17] constructed the international coal, oil, and natural gas network models by gathering nodes into a single plane, without considering the influence of uncertainty factors. That is, most literature studies just provide an intuitive description of the volume between two countries' trade. Few scholars have paid attention to incorporating uncertainties as parameters when constructing the network model formed by all three (coal, oil, and natural gas) layers. In addition, research on the OBOR energy trade using complex network analysis is scarce.

Researchers have demonstrated that both global and local information of the fossil energy trade relations can fully unfold in influential nodes. Hence, influential node detection has become the focus of existing literature. It can be seen from Table 1 that the commonly adopted centrality indicators include degree centrality (DC), betweenness centrality (BC), and closeness centrality (CC). However, these evaluation methods are based on node's structural characteristics; that is, their importance is evaluated based on the structural parameters of nodes themselves. For example, degree centrality only considers node's influence capability from local information. It does not make an in-depth quantification of their surrounding environment, such as target node's position and neighborhood attributes within multisteps [21–23]. BC and CC are both based on the shortest distance between node pairs, reflecting the control force of network flow. But, they have high time complexity of $O(N^3)$ which is not suitable for large-scale networks [24, 25]. As mentioned above, it is still an open issue providing a new and comprehensive method, to identify influential nodes with high accuracy and low time complexity [26] .

However, in the existing network models of energy trade, the influence of uncertainty factors, such as trade strategy, policy implementation, and data uncertainty, is still not adequately taken into account [27]. The impact of uncertainty can be roughly divided into two categories: one is external, such as financial crisis and geopolitics, and the other is the linkage effect of trade structure within different kinds of fossil energy caused by substitution and price. How do we evaluate the influential nodes integrally as well as the complexed interactions among them? How will the

Table 1: List of fossil energy trade network studies in recent years.

Authors (year)	Modeling	Type of network	Category	Scope and time	Evaluation indexes
Peng et al. (2021) [9]	$G = (V, E, W)$	Transportation	LNG	World, 2013–2017	DC, L, CC, C,
Bu et al. (2020) [10]	$G = (V, E, W)$	Consumption	Gas	China, 2005–2017	DC, BC, CC, LMDI
Wang and Li (2019) [11]	$G = (V, E, W)$	Transportation	Coal	China, 1997–2016	DC, BC,L, CC,
Wang et al. (2019) [12]	$G = (V, E, W)$	Trade	Coal	World, 1996–2015	DC, BC, C
Xi et al. (2019) [13]	$G = (V, E, W)$	Trade	Oil	OBOR, 2009–2016	DC, BC, CC
An et al. (2018) [14]	$G = (V, E, W)$	Trade	Oil	World, 2014–2017	PMI, C
Guan and An (2017) [15]	$G = (V, E)$	Trade	Oil, coal, gas, PV	World, 2014	BC, ND, RankS, LP
Zhong et al. (2017) [16]	$G = (V, E)$	Trade	Coal, oil, gas	World, 2000–2013	C, NMI
Gao et al. (2015) [17]	$G = (V, E, W)$	Trade	Coal, oil, gas	World, 2002–2013	DC, C, NMI
Ji et al. (2014) [18]	$G = (V, E)$	Trade	Oil	World, 2010	EI, DC, CC, C, NMI
Zhong et al. (2014) [19]	$G = (V, E, W)$	Trade	Oil	World, 2002–2011	C, NMI
An et al. (2014) [20]	$G = (V, E, W)$	Trade	Oil	World, 1993–2012	DC, CC, C, stability

Note. DC: degree centrality, L: shortest path length, CC: closeness centrality, C: community structure, BC: betweenness centrality, NMI: normalized mutual information, LMDI: logarithmic mean Divisia index, PMI: pointwise mutual information, ND: network density, RankS: ranking score, LP: link prediction, and EI: export intensity.

influential nodes affect the efficiency of network operation? Of course, it is not straightforward to incorporate these factors into the model. In this paper, we will introduce the multisource information fusion technology that has an extensive practicality, effectiveness, and applicability, to solve the uncertainty problem mentioned above. As an effective uncertain information processing method, it has been well developed, such as GIQ [28], CaFtR [29], evidence theory [30], D numbers theory [31], Z numbers [32], and entropy-based approaches [33].

As one of the most effective tools of uncertainty reasoning, Dempster–Shafer (D-S) evidence theory uses multievent hypothesis to express a nondetermination state directly [34]. Besides, the D-S evidence theory has the advantages of expressing "uncertain" and "unknown" directly and has great application value in classification and information fusion, which provides strong support for the evaluation of complex uncertain systems. Furthermore, it has been widely applied in lots of fields, including but not limited to reliability analysis, classification, quantum information decision, and fault diagnosis [35, 36]. Moreover, it has the ability to combine observations from different sources, so as to reduce system uncertainty effectively.

In this paper, the trade data for coal, oil, and natural gas can be seen as multisource information. Due to the factors such as multienergy complementarity and energy structure adjustment, the existing energy trade networks are insufficient to consider these kinds of information, so that some information would be lost during the modeling and processing of data. Therefore, the fusion method of evidence theory can be used in multirelationship energy trade network studies.

3. Methodology

3.1. Preliminaries

3.1.1. D-S Evidence Theory. The D-S evidence theory-based method is among the alternative algorithmic approach to multisensor data fusion that tries to achieve refined estimates of "uncertainty" [37–39]. It employs a reliability function rather than probability to measure uncertainty, and it is widely used in the field of information and decision-making. It provides a favorable Dempster combination rule for information fusion, which has the superiority such as the commutative law and the associative law and can realize fusion between evidence without the support of prior probability. The basic framework of multisource information fusion based on D-S theory is shown in Figure 1. The basic concepts and definitions used in this paper are shown as follows. Much more detail is provided in references [40–42].

Definition 1 (frame of discernment). Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ be a finite nonempty set, and let 2^Θ be the power set of Θ; thus, $2^\Theta = \{\varnothing, \theta_1, \ldots, \theta_N, \{\theta_1, \theta_2\}, \{\theta_1, \theta_3\}, \ldots, \Theta\}$.

Definition 2 (basic probability assignment (BPA)). For a frame of discernment Θ, BPA is a mapping m: $2^\Theta \longrightarrow [0, 1]$, satisfying

$$m(\varnothing) = 0 \tag{1}$$

and

$$\sum_{\varnothing \subseteq 2^\Theta} m(\theta) = 1, \tag{2}$$

where $\varnothing$ is an empty set and θ is any element of 2^Θ. $m(\theta)$ reflects the exact degree of trust in the proposition θ, namely, the probability assigned. Condition (1) reflects no confidence in the null set; condition (2) reflects the sum of the basic probability assignment of all propositions equal to 1.

Definition 3 (Dempster combination rule). Suppose m_1 and m_2 are independent BPAs from different evidence resources, respectively. The fusion result of m_1 and m_2, denoted by $m = m_1 \oplus m_2$, under Dempster's rule of combination is defined as follows:

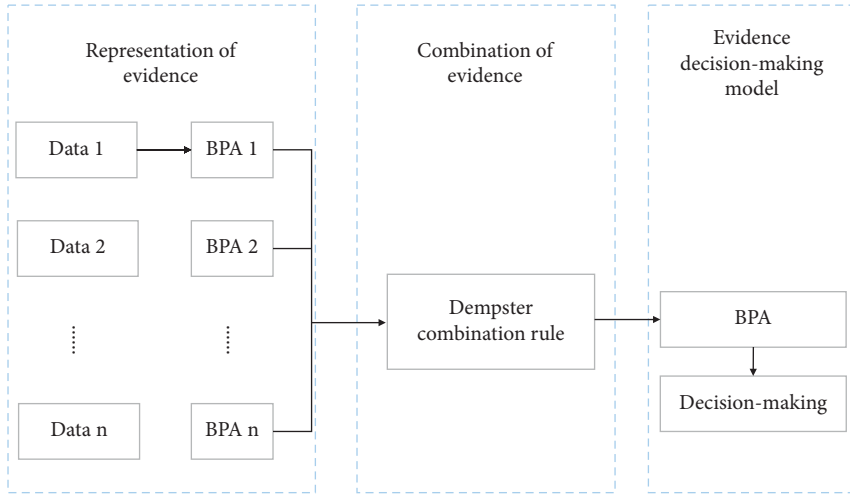


Figure 1: Framework of multisource information fusion based on D-S theory.

$$m(\theta) = [m_1 \oplus m_2](\theta) = \begin{cases} 0, & \theta = \varnothing \\ \dfrac{\sum_{B \cap C = 0} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \varnothing} m_1(B) m_2(C)}, & \theta \neq \varnothing \end{cases}, \tag{3}$$

where θ, B, and C are the elements of 2^{Θ}. Thereinto, $\oplus$ is an orthogonal operator.

3.1.2. Benchmark Centrality Measures for Influential Nodes

Definition 4 (degree centrality (DC)). The DC of node i, denoted as d_i, is defined as

$$d_i = \sum_{j=1}^{N} a_{ij}, \tag{4}$$

where N is the total number of nodes in the network and $a_{ij} = 1$ if node i is connected to node j and $a_{ij} = 0$ otherwise [43, 44].

Definition 5 (betweenness centrality (BC)). The BC of node i, denoted as b_i, is defined as

$$b_i = \sum_{j,k \neq i} \frac{g_{jk}(i)}{g_{jk}}, \tag{5}$$

where g_{jk} denotes the total number of shortest paths between nodes j and k and $g_{jk}(i)$ is the number of those paths that go through node i [45, 46].

Definition 6 (closeness centrality (CC)). The CC of node i, denoted as c_i, is defined as the reciprocal of the sum of geodesic distances to all other nodes and calculated by the following formula:

$$c_i = \left[\sum_{j=1, j \neq i} d_{ij} \right]^{-1}, \tag{6}$$

where d_{ij} is the geodesic distance between nodes i and j [47, 48].

Definition 7 (eigenvector centrality (EC)). The EC of node i, denoted as e_i, is defined as

$$e_i = \lambda_{\max}^{-1} \sum_{j=1}^{N} a_{ij} e_j, \tag{7}$$

where $\lambda_{\max}^{-1}$ is the maximum eigenvalue of the adjacent matrix and its corresponding eigenvector is $e = [e_1, e_2, \ldots, e_N]^T$. $a_{ij} = 1$ if node i is connected with node j, and $a_{ij} = 0$, otherwise [49, 50].

Definition 8 (PageRank centrality (PC)). Let $E(u)$ be some vector over the Web pages that corresponds to a source of rank. Then, the PageRank of a set of Web pages is an assignment R' to the Web pages which satisfies

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u), \tag{8}$$

such that c is maximized and $R_1' = 1$ [51, 52].

3.2. A Relevance Matrix-Based BPA Method. The relevance matrix can be obtained by converting the multiproduct trade matrixes with certain processing approaches. The correlation among nodes in the network can be fully reflected in the construction of BPA. The primary question is how to build an effective BPA, which is also the critical step. The detailed subsequent steps are as follows (see Figure 2).

Step 1. The representation of evidence. Firstly, we build a weighted matrix for each fossil energy trade $W^k \equiv (w_{ij}^k)_{n \times n} (i = 1, 2, \ldots, n; j = 1, 2, \ldots, n)$ with elements of w_{ij}^k, where $k = 1, 2, 3$ represents coal, oil, and natural gas, respectively. w_{ij}^k represents the trade volume from node i to node j (unit in USD). Then, we can build the relevance matrix for each energy $S^k \equiv (s_{ij}^k)_{n \times n}$, where $s_{ij}^k = w_{ij}^k / w_{\max}^k$ is

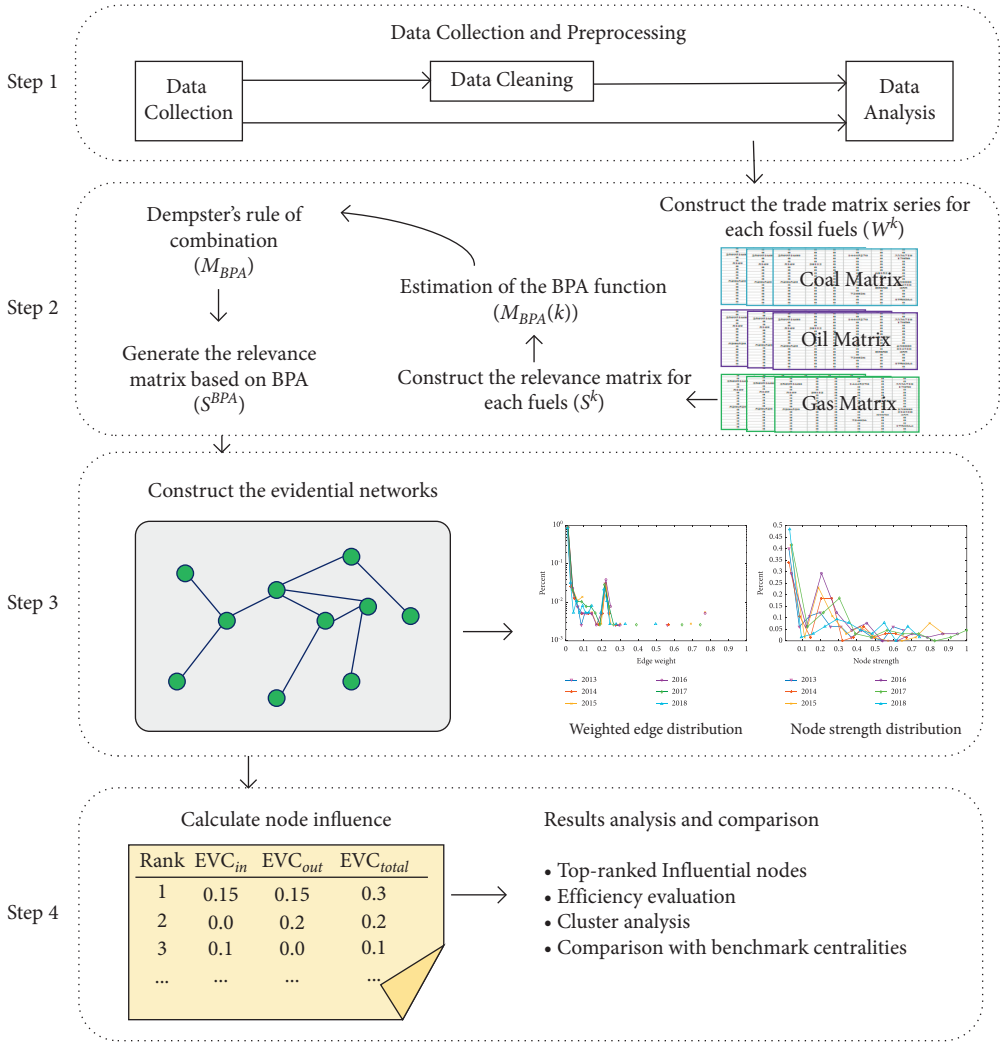


Figure 2: Flowchart of the proposed D-S based network method.

proportional to node strength and $w^k_{\max}$ is the maximum of elements in W^k. However, existing studies have verified that the distribution of fossil energy trade displayed power-law characteristics [9]. In order to avoid too many small values to reflect variability, we adopt $s^k_{ij} = w^k_{ij}/w^k_{\text{local-max}}$, where $w^k_{\text{local-max}}$ is the maximum in the local connected networks of node i and j.

Step 2. Estimation of the BPA function. Each element in the relevance matrix S^k can be transformed to the elements in the BPA matrix of $M_{\text{BPA}} \equiv (m_{ij}(Y), m_{ij}(N), m_{ij}(Y,N))_{n\times n}$. The elements in the BPA matrix M_{BPA} are defined as follows:

$$m_{ij}(Y) = \frac{\left|s_{ij} - \min(S)\right|}{\text{summ}},$$
$$m_{ij}(N) = \frac{\left|s_{ij} - \max(S)\right|}{\text{summ}}, \quad (9)$$
$$m_{ij}(Y,N) = \frac{\left|s_{ij} - (\max(S) + \min(S))/2\right|}{\text{summ}},$$

where $\text{summ} = |s_{ij} - \max(S)| + |s_{ij} - \min(S)| + |s_{ij} - (\max(S) + \min(S))/2|$ and $m_{ij}(Y) + m_{ij}(N) + m_{ij}(Y,N) = 1$. In this paper, each single-layer matrix of coal, oil, and

natural gas corresponds to a BPA matrix, defined as $M_{\mathrm{BPA}}(c) \equiv (c_{ij})_{n\times n}$, $M_{\mathrm{BPA}}(o) \equiv (o_{ij})_{n\times n}$, and $M_{\mathrm{BPA}}(g) \equiv (g_{ij})_{n\times n}$, respectively.

Step 3. Dempster's rule of combination. Let $M_{\mathrm{BPA}} \equiv (l_{ij})_{n\times n}$ be the BPA matrix of total fossil energy, where

$$\begin{aligned} l_{ij} &= \left[l_{ij}(Y), l_{ij}(N), l_{ij}(Y,N)\right] \\ &= c_{ij} \oplus o_{ij} \oplus g_{ij}. \end{aligned} \tag{10}$$

If $t_{ij} = c_{ij} \oplus o_{ij}$, then $l_{ij} = t_{ij} \oplus g_{ij}$.

$$t_{ij} = \left[t_{ij}(Y), t_{ij}(N), t_{ij}(Y,N)\right], \tag{11}$$

where

$$\begin{aligned} t_{ij}(Y) &= \frac{c_{ij}(Y) * o_{ij}(Y) + c_{ij}(Y) * o_{ij}(Y,N) + c_{ij}(Y,N) * o_{ij}(Y)}{1 - \left(c_{ij}(Y) * o_{ij}(N) + c_{ij}(N) * o_{ij}(Y)\right)}, \\ t_{ij}(N) &= \frac{c_{ij}(N) * o_{ij}(N) + c_{ij}(N) * o_{ij}(Y,N) + c_{ij}(Y,N) * o_{ij}(N)}{1 - \left(c_{ij}(Y) * o_{ij}(N) + c_{ij}(N) * o_{ij}(Y)\right)}, \\ t_{ij}(Y,N) &= \frac{1}{1 - \left(c_{ij}(Y) * o_{ij}(N) + c_{ij}(N) * o_{ij}(Y)\right)} \left(c_{ij}(Y,N) * o_{ij}(Y,N)\right). \end{aligned} \tag{12}$$

Step 4. Probabilistic conversion and decision-making. We transform the BPA matrix for total fossil energy into the relevance matrix again and make final decision judgement. Let the relevance matrix for total fossil energy be $S^{\mathrm{BPA}} \equiv (s_{ij}^{\mathrm{BPA}})_{n\times n}$, and

$$\begin{aligned} s_{ij}^{\mathrm{BPA}} &= l_{ij}(Y) + \frac{l_{ij}(Y,N)}{|Y,N|} \equiv \mathrm{Bet}P_{l_{ij}}(Y), \\ |Y,N| &= 2, \\ \mathrm{Bet}P_m(Y) &= m(Y) + \frac{m(Y,N)}{|(Y,N)|}. \end{aligned} \tag{13}$$

3.3. Network Construction and Node Influence. In this paper, the network can be constructed as a directed weighted network. The model of the directed weighted network is given as a triple $G = (V, E, W)$, where $V = (v_1, v_2, \ldots, v_n)$ represents the set of nodes and $E = (e_1, e_2, \ldots, v_m)$ represents the set of edges. Here, each link of a graph has an associated numerical value, called a weight. $W = (w_{ij})_{n\times n} = (s_{ij}^{\mathrm{BPA}})_{n\times n}$ is the weighted matrix of connected edges, where $w_{ii} = 0$ and $w_{ij} \geq 0$ and $w_{ij}(i \neq j)$ represents the weight of edge from node v_i to node v_j. Then, a time series of the influence network, which is called evidential network (EN), among OBOR countries is build.

Normally, when two nodes have high relevance, then nodes' influences will increase automatically. Given a network $G = (V, E, W)$, for any node $i \in V$, the incoming and outgoing influence capability of node i, denoted as $\mathrm{EVC}_{\mathrm{in}}(i)$ and $\mathrm{EVC}_{\mathrm{out}}(i)$, are given by

$$\begin{aligned} \mathrm{EVC}_{\mathrm{in}}(i) &= \sum_{j\in\Phi(i)} w_{ji}, \\ \mathrm{EVC}_{\mathrm{out}}(i) &= \sum_{j\in\Phi(i)} w_{ij}, \end{aligned} \tag{14}$$

where $\Phi(i)$ is the set of nearest neighbors of node i.

The total evidence centrality ($\mathrm{EVC}_{\mathrm{total}}$) measures the sum of the total incoming influence of all inlinks and the total outgoing influence by all outlinks, which can be calculated as follows:

$$\mathrm{EVC}_{\mathrm{total}}(i) = \mathrm{EVC}_{\mathrm{in}}(i) + \mathrm{EVC}_{\mathrm{out}}(i), \tag{15}$$

where $\mathrm{EVC}_{\mathrm{total}}(i)$ is the total influence of node i. The greater the total evidence centrality value of a node, the more influential the node.

4. Data Description

The data for the three kinds of fossil energy trade were downloaded from the United Nations Commodity Trade Statistics Database (UN COMTRADE). HS codes are 270100, 270900, 271111, and 271121, respectively. In the original data sources, natural gas consists of two categories: gasified natural gas and liquefied natural gas. In addition, the data presented in the UN COMTRADE are unit in both US dollars (trade value) and kilograms (trade volume). We finally select the data for trade value due to serious data lack of trade volume.

More importantly, the statistical data are from both importing and exporting countries. However, there exist data errors between them, as shown in Figure 3. Due to space constraints, only part of the country names is marked in

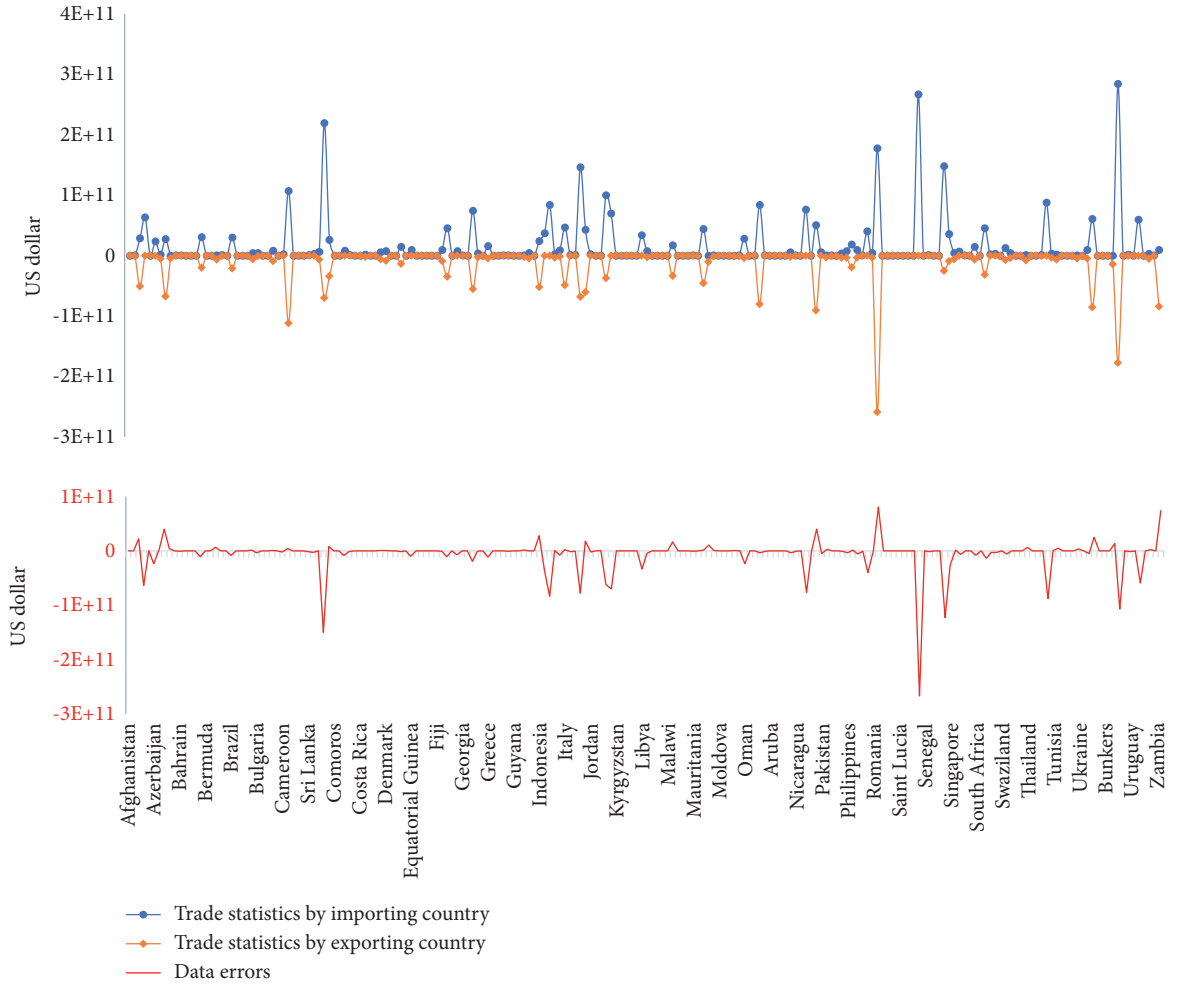


Figure 3: Data uncertainty example of oil export in 2013.

Figure 3. Consistent with most existing studies, we use the trade data released by exporting counties ultimately. In fact, the errors existing here can be seen as one kind of data uncertainty.

Nest, we conducted statistical analysis on the evolution and proportion of fossil energy trade within the countries under OBOR, as shown in Figure 4. Although the trade volume of OBOR countries accounted for less than 20 percent, its proportion has been increasing since the BRI was put forward, from 8 percent in 2013 to 14 percent in 2018. This indicates that the status of fossil energy trade for countries in the OBOR route has increased in the international market. This also proves the necessity of studying such issues.

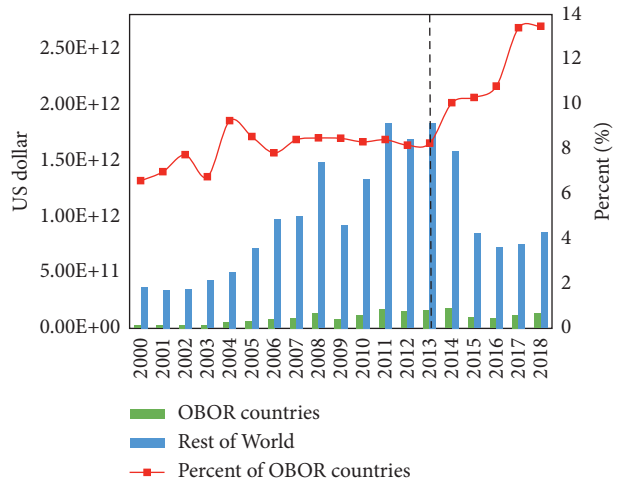
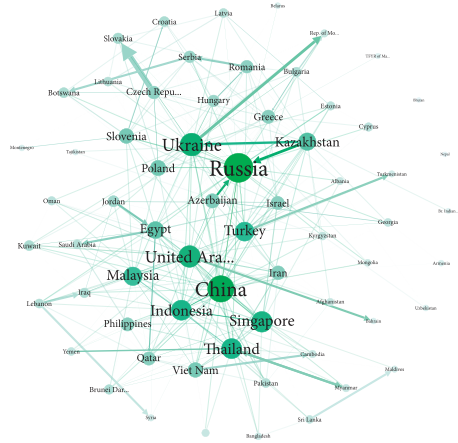


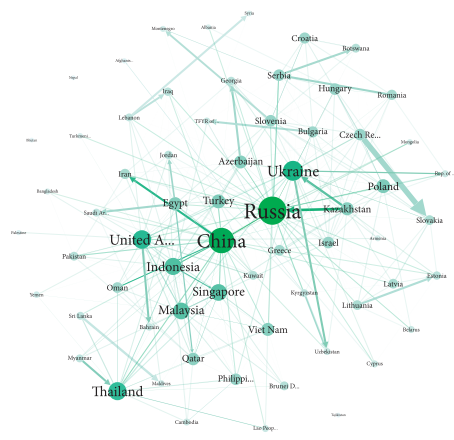
Figure 4: Changes in volume and share of fossil energy trade under OBOR since 2000.

5. Result Analysis

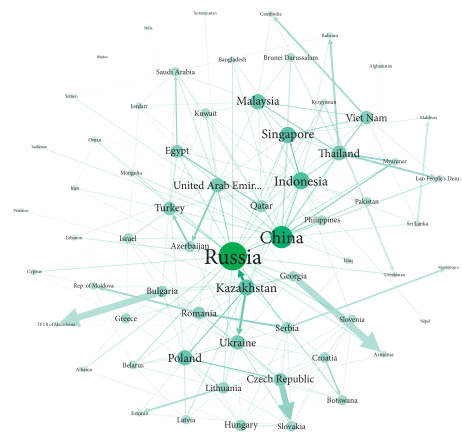
5.1. Topological Structure Analysis of Evidential Networks. According to Sections 3.2 and 3.3, we can obtain the fossil energy trade evidential networks during 2013–2018. Then, the network structure is drawn by using Gephi, a general network analysis visualization software tool, as depicted in Figure 5. In Figure 5, node size is proportional to node degree–the number of links incident to the node. The thickness of link indicates the connection weight between the two counties, and the thicker the link, the greater the connection, and vice versa.



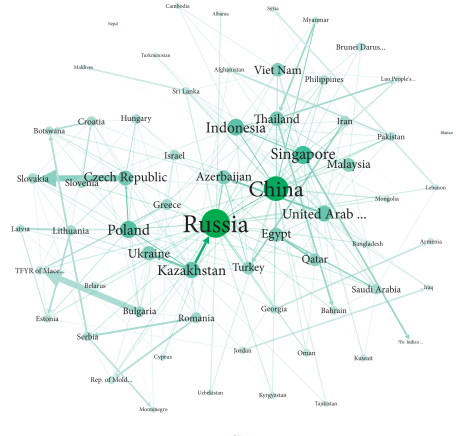
2013



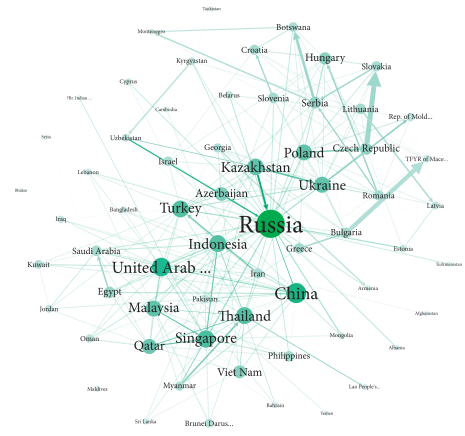
2014



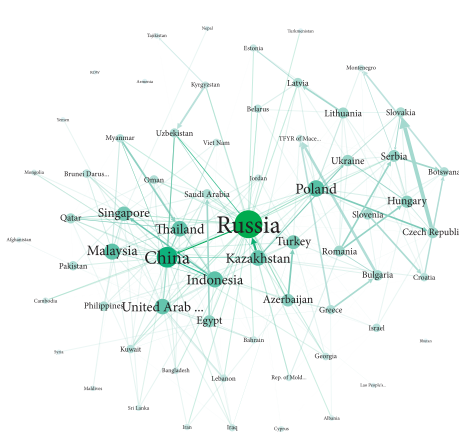
2015



2016



2017



2018

Figure 5: Structure of the evidential networks during 2013–2018.

From Figure 5, firstly, Russia dominated the system in terms of node degree. In other words, Russia has the largest number of partners owing to its abundant energy resources. Russia's top three trading partners are concentrated in Kazakhstan, China, and Azerbaijan. It can be seen that resource advantage dominates a country's position. Secondly, as the world's largest energy consumer, China's top five trading partners include the United Arab Emirates,

Kazakhstan, Azerbaijan, Iran, Singapore, and Indonesia. Of course, Russia is also an important trading partner for China. With the construction or undergoing construction of oil and gas pipelines in the Far East, the energy relation between China and Russia has become increasingly close. Both countries are committed to build an energy security system conducive to each other and expected to form an energy community with a shared future. There are also some notable countries, namely, Ukraine, United Arab Emirates, Kazakhstan, Azerbaijan, Singapore, Indonesia, and Turkey, whose positions were not very much changed over time.

Furthermore, Figure 5 shows that the connecting links between most countries are thin, while those between a few country pairs are thick. That is, the weights gather on a very few connections, which can be depicted from the weighted edge distribution (Figure 6(a)). Figure 6(a) shows that the distribution of weighted links in all years presents similar power-law heterogeneity, which is similar to the results of most existing studies. Namely, the proposed method in this paper does not change system heterogeneity when considering uncertainty.

In addition, we further analyze how strength is distributed across nodes in a similar way to the weighted edge distribution, as shown in Figure 6(b). Node strength is the total of the edge weights in the network incident to the node. In fact, node strength is the measure of total EVC proposed in Section 3.3. Similar to the result of Figure 6(a), heterogeneity exists in node strength distribution. In other words, the distribution of strength values across nodes is based on power-law; many nodes have a lower EVC while a smaller number of key nodes take the lion's share of centrality, making them as hubs that facilitate integration across the network. In the following sections, we will explore these critical nodes in detail.

5.2. Influential Nodes Identification. Figure 7 depicts the time evolution of the ranking of various countries during the study period according to nodes' total EVC. The results show that countries' rankings have changed markedly since the BRI. In 2013, for example, Russia, Thailand, and Ukraine were the top three countries according to their total EVC, while in 2018, Czechia, Slovakia, and Bulgaria were the top three. Meanwhile, affected by the development of global trade internationalization, the total influence value of each country is also changing which makes the analysis considerably more complicated.

Thus, the average EVC for all countries and standard deviation (SD) are used to present the ranking characteristics of sample countries in the evidential networks. Figure 8 scatters all OBOR countries and their corresponding total EVC, incoming EVC, and outgoing EVC values. In Figure 8(a), for instance, the abscissa represents the average EVC value for each of the 65 countries, the ordinate represents their SD during 2013–2018, and the imaginary lines represent the means of EVC and SD for all countries, respectively. Countries with larger values along the abscissa but smaller values along the ordinate indicate stabilized higher influence.

It can be seen from Figure 8(a) that Russia and Kazakhstan in the lower right corner dominate the centrality. Of course, the total EVC values for Czechia and Slovakia are larger than those for Russia and Kazakhstan. But, the SD values for Czechia and Slovakia are also higher. That is, these may only be the values in a particular year supporting this result, which cannot be considered as influencer roles. For resource exporters of Russia and Kazakhstan, their positions are firmly in the top 10 with little volatility (see Figure 7). It can be seen that resource advantages play an absolute role in the influence of a country's trade. Energy dependence is central to Russia's economic structure. According to the data of the Russian State Statistics Bureau, in the export commodity structure from 2015 to 2017, the proportion of raw materials such as oil and natural gas has always remained at about 60%. Therefore, it can be seen that energy export is still the main driving force of Russia's economic recovery. Besides, Kazakhstan, Turkmenistan, and Uzbekistan are major oil producers in Central Asia. Kazakhstan has the largest crude oil reserves (about 1.7% of the world's total). Turkmenistan has the largest natural gas reserves (about 9.9% of the world's total), but the country's natural gas resources are far from being fully exploited. This is also why Turkmenistan's centrality in the export structure is not obvious (see Figure 8(b)).

In addition, Ukraine, Egypt, Romania, China, Singapore, Qatar, Saudi Arabia, and Indonesia are also essential countries due to their lower SD and higher EVC values (Figure 8(a)). Another possible reason is that these countries have a high export influence, or a high import influence, such as China, Thailand, Egypt, and Saudi Arabia (Figures 8(b) and 8(c)).

After the BRI, China's influence rose briefly in 2014 and then increased sharply since 2017 after a period of stability. In 2017, China's oil import volume from countries along the Belt and Road Initiative was 141.71 billion US dollars, nearly ten times of its export. That is, China runs a large oil trade deficit, accounting for 77% of its total import. Although China is the proposer of the BRI, it can be seen that its influence in fossil energy trade is at a medium level. Possible reasons are as follows: first, China is focusing on developing renewable energy; second, Kazakhstan, Russia, and other countries are important fossil energy resource exporters. However, China's EVC value (0.608) surpassed Russia's (0.60) in 2017, indicating that the increasingly active oil and gas activities of Chinese companies in Central Asia will challenge Russia's energy monopoly.

5.3. Evaluating the Impact of Top-Ranked Influential Nodes. In this section, the impact of the identified top-ranked influential is evaluated by comparing their efficiency. Damage resistance is to evaluate the variation of network efficiency at risk nodes to observe the system's ability to maintain stability in the face of risk disturbance. In this paper, the value of network efficiency when a node disappears is used as a quantitative evaluation index to measure the efficiency of the removed node. Efficiency refers to the aggregation degree of paths within the system [53, 54]. It can measure the ability of

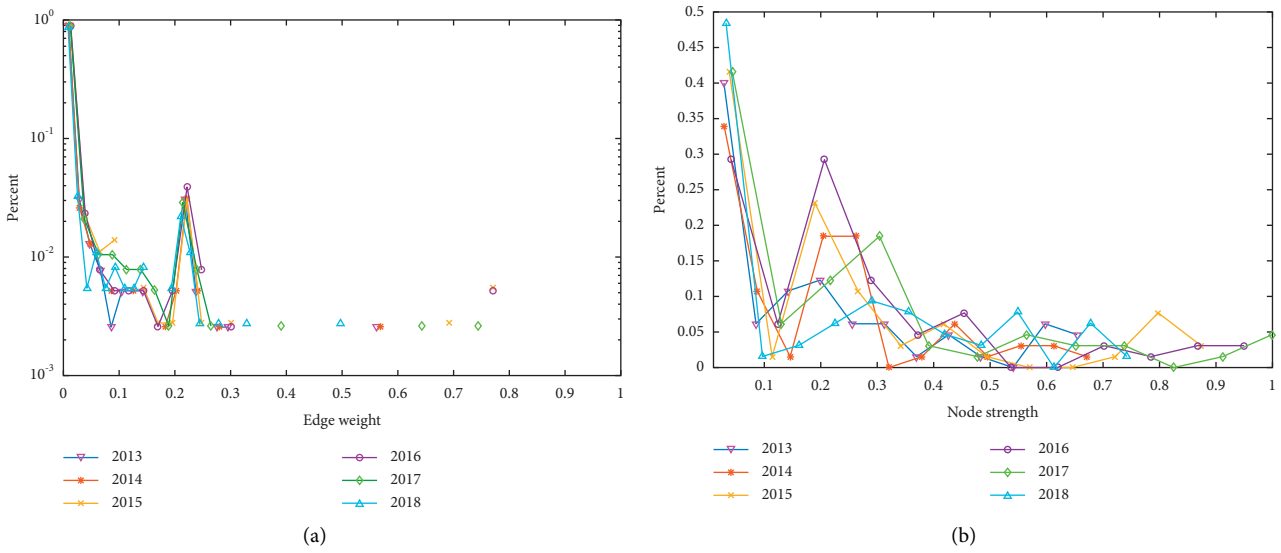


Figure 6: Weighted edge distribution and node strength distribution. (a) Edge weight. (b) Node strength.

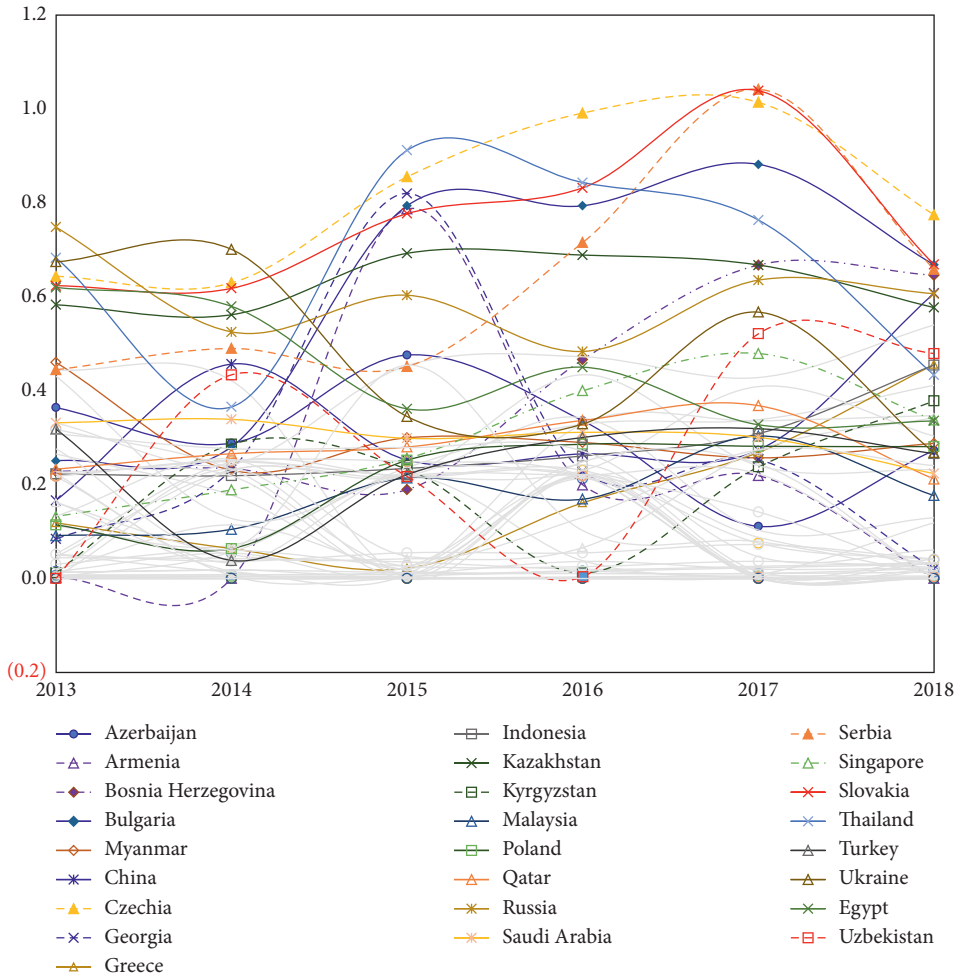


Figure 7: Evolution of the ranking of various countries by total EVC.

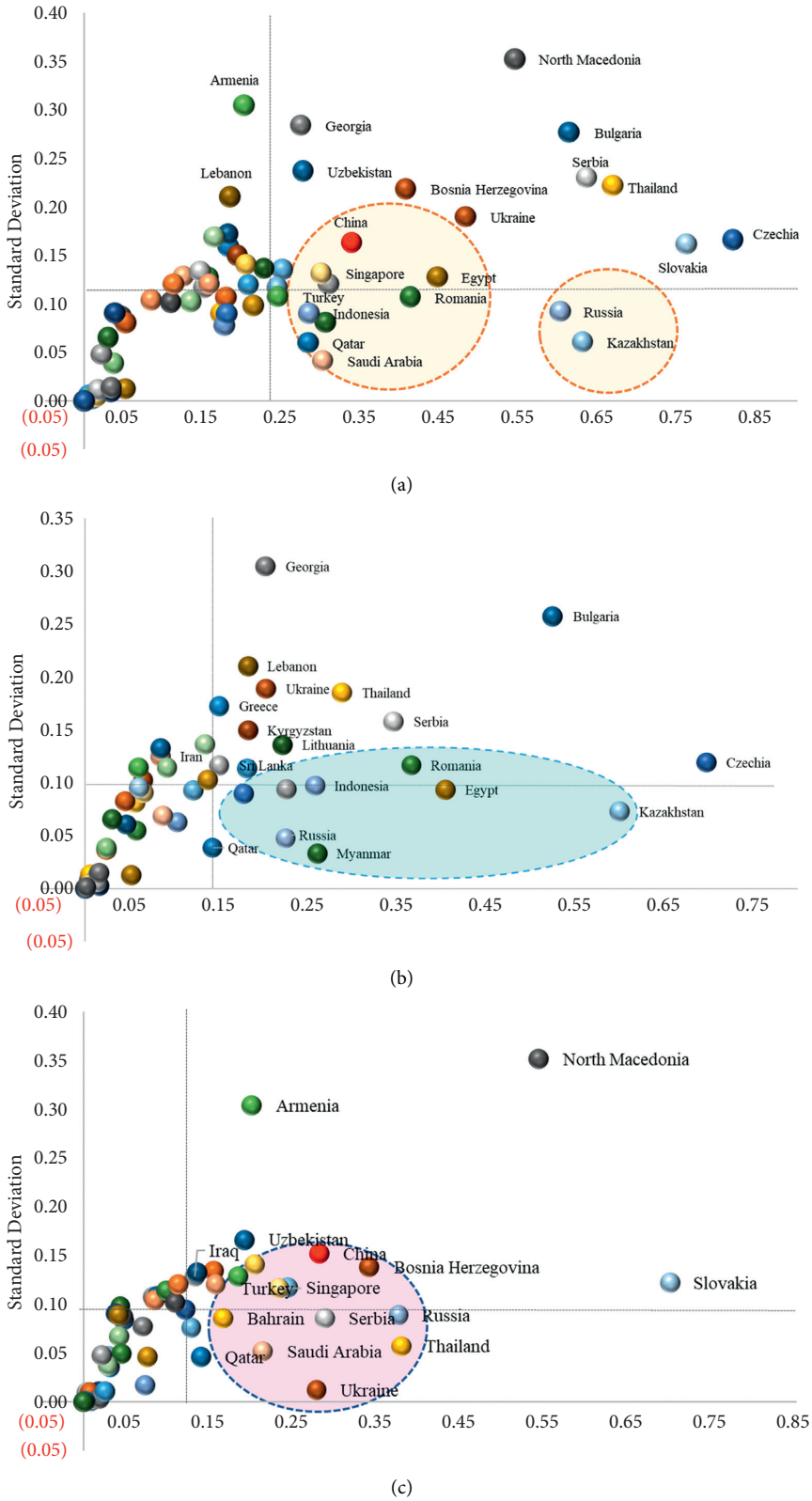


Figure 8: Scatter of EVC and SD for the Belt and Road countries. (a) Total EVC. (b) EVC_out. (c) EVC_in.

Table 2: Influences of removing the top nodes on the network efficiency when using the EVC ranking.

2013		2014		2015		2016		2017		2018	
Removed node	EffC (k)	Removed node	EffC (k)	Removed node	EffC (k)	Removed node	EffC (k)	Removed node	EffC (k)	Removed node	EffC (k)
Russia	0.134	Ukraine	−0.042	Thailand	0.052	Czechia	−0.025	Serbia	−0.048	Czechia	0.044
Thailand	0.038	Czechia	−0.030	Czechia	−0.008	North Macedonia	−0.054	Slovakia	−0.022	Slovakia	0.037
Ukraine	−0.044	Slovakia	−0.032	Georgia	−0.047	Thailand	0.036	Czechia	−0.016	Bulgaria	0.022
Czechia	−0.041	Egypt	−0.055	North Macedonia	−0.057	Slovakia	−0.022	Bulgaria	−0.037	Serbia	0.012
Slovakia	−0.036	Kazakhstan	−0.015	Bulgaria	−0.048	Bulgaria	−0.038	Thailand	0.041	Bosnia herzegovina	0.017
Egypt	−0.064	Russia	0.058	Armenia	−0.056	Serbia	−0.053	North Macedonia	−0.049	China	0.104
Kazakhstan	−0.016	Serbia	−0.060	Slovakia	−0.011	Kazakhstan	−0.032	Bosnia herzegovina	−0.049	Russia	0.185
Myanmar	−0.017	China	0.132	Kazakhstan	−0.012	Russia	0.133	Kazakhstan	−0.049	Kazakhstan	0.034
Serbia	−0.066	Uzbekistan	−0.060	Russia	0.055	Romania	−0.036	Russia	0.111	North Macedonia	0.012
Lebanon	−0.067	Lebanon	−0.061	Azerbaijan	−0.010	Bosnia herzegovina	−0.054	Ukraine	−0.025	Romania	0.032
Moldova	−0.066	Thailand	0.011	Viet Nam	−0.034	Egypt	−0.034	Uzbekistan	−0.017	Uzbekistan	0.050
Azerbaijan	−0.013	Saudi Arabia	−0.069	Romania	−0.035	Lithuania	−0.044	Singapore	0.026	Greece	0.025
Saudi Arabia	−0.081	Azerbaijan	−0.024	Serbia	−0.057	Singapore	−0.003	Romania	−0.031	Indonesia	0.072
Romania	−0.043	Kyrgyzstan	−0.060	Egypt	−0.049	Estonia	−0.054	Montenegro	−0.049	Thailand	0.088
Turkey	−0.057	Bulgaria	−0.059	Ukraine	−0.039	Qatar	−0.024	Qatar	−0.008	Croatia	0.017

nodes to transfer information in the network after the occurrence of risks, using the information theory-based network flow analysis. Here, we adopt the average mutual information to define the efficiency of weighted network G:

$$E[G] = \sum_{i,j} \frac{T_{ij}}{T_{..}} \log \frac{T_{ij} T_{..}}{T_{i.} T_{.j}}. \tag{16}$$

Here, T_{ij} represents a flow from node i to node j, $T_{i.} = \sum_j T_{ij}$ represents the total flow leaving node i, $T_{.j} = \sum_i T_{ij}$ represents the total flow entering node j, and $T_{..}$ represents the sum of all flows in the system.

The efficiency for node k in a weighted network is defined as

$$\mathrm{EffC}(k) = \frac{\Delta E}{E} = \frac{E[G] - E[G'(k)]}{E[G]},\ k = 1, \ldots, N, \tag{17}$$

where $G'(k)$ is a subgraph of G with N-1 nodes and P-Q edges obtained by removing node k and its neighbor edges. The greater the efficiency value, the higher the influence of a node.

Table 2 shows the effect of removing the top fifteen countries on the network efficiency when using the EVC ranking. According to Table 2, the influences of the top-15 listed countries are different and vary with ongoing time. To sum up, however, most of them have relatively high efficiency values, such as Russia, China, and Thailand. Without these nodes, the network efficiency will decline obviously. The efficiency value of Russia is the maximum of all nodes, although its ranking dropped significantly from 2014. In fact, it is intuitional that Russia is a pivotal node in the network because it connects to most nodes. The results show that the node evaluation index proposed in this paper has a greater impact on network performance. In other words, the failure of these nodes will make the network show more fragile damage resistance or robustness.

5.4. Hierarchical Cluster Analysis. In this section, we also conduct hierarchical clusters further, as shown in Figure 9. The clusters, depicted by the dendrogram, result from the Unweighted Pair Group Method with Arithmetic Mean (UPGMA). UPGMA is a generally used clustering technique that uses the arithmetic average approach to construct a phylogenetic tree from a distance matrix. It has been used most frequently in ecology and systematics [55] and in numerical taxonomy [56]. For detailed algorithm, please see reference [57].

According to Figure 9, the OBOR countries can be roughly divided into the following five categories: Cluster 1 contains the largest number of countries, but most of them are from underdeveloped regions although their annual GDP growth has increased in recent years. Besides, those countries are relatively concentrated geographically, such as Tajikistan, Kyrgyzstan, Kazakhstan, Afghanistan, Pakistan, and other Central Asian countries. It can be seen that geographical distance is one of the factors affecting fossil energy trade, which involves the cost of transportation. Cluster 2 includes resource countries such as Iraq, Qatar, and Saudi Arabia. This category has typical characteristics, that is, except Bahrain and Maldives, and the GDP of all countries in this category was higher than the average level

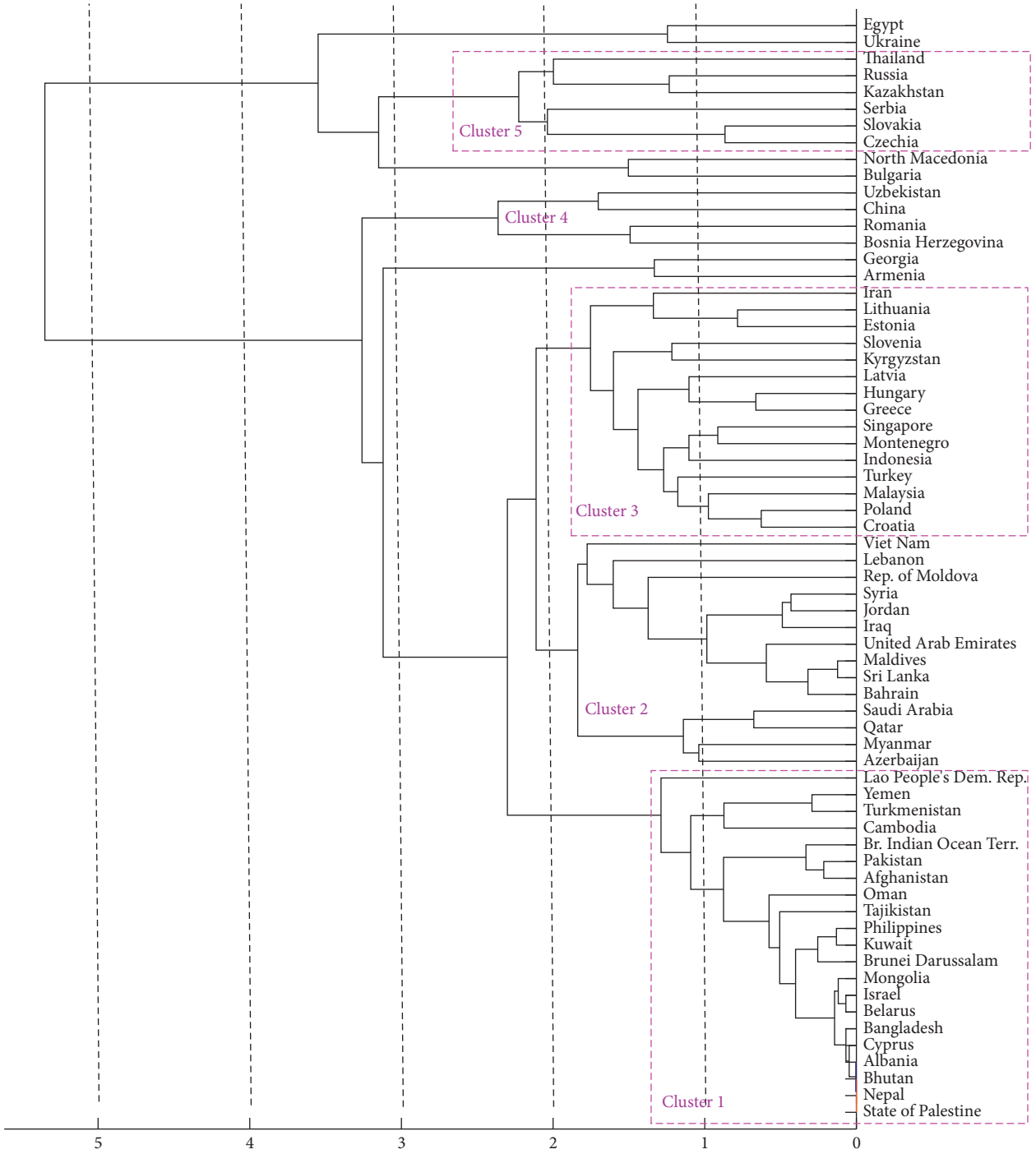


Figure 9: Dendrogram of cluster analysis.

of the OBOR countries in the year of 2018. Cluster 3 includes Turkey, Greece, Iran, Poland, Singapore, Hungary, and other emerging economies with relatively high annual GDP growth and GDP per capita simultaneously. Cluster 4, represented by China, has a high level of GDP income and annual GDP growth. However, the GDP per capita of these countries is lower than the average level. Cluster 5 includes Russia, Kazakhstan, Czechia, Slovakia, and Serbia with high GDP level, but their annual GDP growth and GDP per capita are not high.

By comparing the sorting results of the above EVC scatter plot with the classification results of cluster analysis, it can be seen that the classification of most countries is

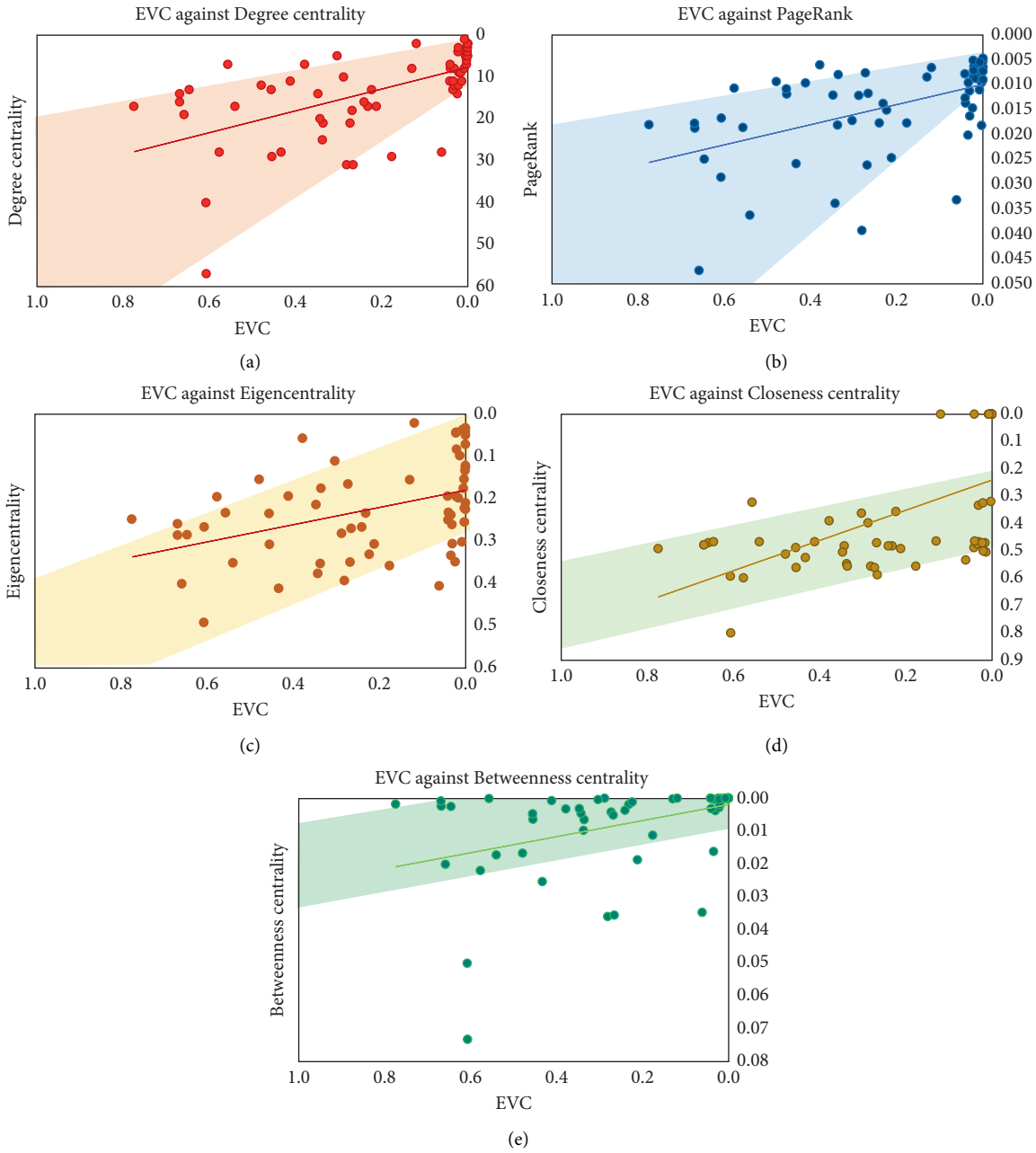


Figure 10: Scatterplots of EVC values against another centrality measure. EVC against (a) degree centrality, (b) PageRank centrality, (c) eigencentrality, (d) closeness centrality, and (e) betweenness centrality.

consistent. In all, in addition to geographical distance, economic development level, economic growth potential, and per capita national income are all potential influencing factors of fossil energy trade.

5.5. Comparison with Benchmark Centralities. Finally, in order to evaluate the performance of the proposed method, we pay attention to the correlation between EVC ranking list and five classical centrality measures, i.e., degree centrality, PageRank centrality, eigenvector centrality, closeness centrality, and betweenness centrality, as depicted in Figure 10. It is obvious that each node's EVC are highly correlated with other centrality indicators. This shows that the method proposed is reasonable and comprehensive in identifying influential nodes.

6. Conclusions

In this paper, we propose a novel evidential ranking method based on complex network analysis and evidence theory. The combination of local structure, global structure, and uncertainty is taken into consideration in the construction of evidence centrality. Then, a performance analysis is conducted on fossil energy trade along the OBOR. We utilize the network efficiency to simulate the influence of top-ranked

nodes and demonstrate the superiority of EVC. At last, various classical centrality measures are also analyzed to identify the advantages of the proposed method.

The results of node centrality based on D-S theory show that the distribution of weighted edges has obvious heterogeneity which is superior to node strength distribution. That is, only a few pairs of countries have high relevance relationships, while most pairs of countries have low relevance relationships. For influential countries, along with the increase of average EVC values, the volatility also increases. But, some countries with higher average EVC tend to be driven by one or two specific years, such as Czechia and Slovakia. Relative to these countries, Russia, Kazakhstan, and other resource-based and export-oriented countries maintain the core positions. Finally, the clustering results show that geographical distance, income level, GDP per capita, and GDP growth rate are the potential driving forces of fossil energy trade.

Data Availability

All relevant data are publicly available at https://comtrade.un.org/, https://data.worldbank.org/, and https://www.oecd.org/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was jointly supported by the National Natural Science Foundation of China (No. 71804060) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (No. PAPD-2018-87).

References

[1] S. Najm and K. i. Matsumoto, "Does renewable energy substitute LNG international trade in the energy transition?" *Energy Economics*, vol. 92, Article ID 104964, 2020.

[2] K. Kaygusuz, "Energy for sustainable development: a case of developing countries," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 2, pp. 1116–1126, 2012.

[3] Iea, *CO2 Emissions by Energy Source*, https://www.iea.org/data-and-statistics/data-browser/?country=WORLD&fuel=CO2%20emissions&indicator=CO2BySource.

[4] G. P. Peters, J. C. Minx, C. L. Weber, and O. Edenhofer, "Growth in emission transfers via international trade from 1990 to 2008," *Proceedings of the National Academy of Sciences*, vol. 108, no. 21, pp. 8903–8908, 2011.

[5] D. Cao, *BRI Hailed as Force for Sustainable Development*, China Daily, 2019, https://www.chinadaily.com.cn/a/201903/12/WS5c870266a3106c65c34ee09b.html.

[6] L. Sun, Y. Shen et al., *China's Trade and Investment Cooperation under the Belt and Road Initiative*, Shanghai Academy of Social Sciences Press, China, 2019.

[7] *BP Statistical Review of World Energy*, https://www.bp.com/, 2018.

[8] J. Zhang, "Oil and gas trade between China and countries and regions along the 'Belt and Road': a panoramic perspective," *Energy Policy*, vol. 129, pp. 1111–1120, 2019.

[9] P. Peng, F. Lu, S. Cheng, and Y. Yang, "Mapping the global liquefied natural gas trade network: a perspective of maritime transportation," *Journal of Cleaner Production*, vol. 283, Article ID 124640, 2021.

[10] Y. Bu, E. Wang, J. Bai, and Q. Shi, "Spatial pattern and driving factors for interprovincial natural gas consumption in China: based on SNA and LMDI," *Journal of Cleaner Production*, vol. 263, Article ID 121392, 2020.

[11] W. Wang and Z. Li, "The evolution of China's interregional coal trade network, 1997-2016," *Physica A: Statistical Mechanics and Its Applications*, vol. 536, Article ID 120974, 2019.

[12] W. Wang, Z. Li, and X. Cheng, "Evolution of the global coal trade network: a complex network analysis," *Resources Policy*, vol. 62, pp. 496–506, 2019.

[13] X. Xi, J. Zhou, X. Gao, D. Liu, H. Zheng, and Q. Sun, "Impact of changes in crude oil trade network patterns on national economy," *Energy Economics*, vol. 84, Article ID 104490, 2019.

[14] Q. An, L. Wang, D. Qu, and H. Zhang, "Dependency network of international oil trade before and after oil price drop," *Energy*, vol. 165, pp. 1021–1033, 2018.

[15] Q. Guan and H. An, "The exploration on the trade preferences of cooperation partners in four energy commodities' international trade: crude oil, coal, natural gas and photovoltaic," *Applied Energy*, vol. 203, pp. 154–163, 2017.

[16] W. Zhong, H. An, L. Shen et al., "Global pattern of the international fossil fuel trade: the evolution of communities," *Energy*, vol. 123, pp. 260–270, 2017.

[17] C. Gao, M. Sun, and B. Shen, "Features and evolution of international fossil energy trade relationships: a weighted multilayer network analysis," *Applied Energy*, vol. 156, pp. 542–554, 2015.

[18] Q. Ji, H.-Y. Zhang, and Y. Fan, "Identification of global oil trade patterns: an empirical research based on complex network theory," *Energy Conversion and Management*, vol. 85, pp. 856–865, 2014.

[19] W. Zhong, H. An, X. Gao, and X. Sun, "The evolution of communities in the international oil trade network," *Physica A: Statistical Mechanics and Its Applications*, vol. 413, pp. 42–52, 2014.

[20] H. An, W. Zhong, Y. Chen, H. Li, and X. Gao, "Features and evolution of international crude oil trade relationships: a trading-based network analysis," *Energy*, vol. 74, pp. 254–259, 2014.

[21] X. Ren and L. L, "Review of ranking nodes in complex networks," *Chinese Science Bulletin*, vol. 59, no. 13, pp. 1175–1197, 2014.

[22] Y.-Z. Yang, M. Hu, and T.-Y. Huang, "Influential nodes identification in complex networks based on global and local information," *Chinese Physics B*, vol. 29, no. 8, Article ID 088903, 2020.

[23] S. Gao, J. Ma, Z. Chen, G. Wang, and C. Xing, "Ranking the spreading ability of nodes in complex networks based on local structure," *Physica A: Statistical Mechanics and Its Applications*, vol. 403, pp. 130–147, 2014.

[24] J. Liu, Z. Ren, Q. Guo, and B. Wang, "Research progress of node importance ranking in complex networks (in Chinese)," *Journal of Physics*, vol. 62, no. 17, Article ID 178901, 2013.

[25] Z. Wang, C. Du, J. Fan, and Y. Xing, "Ranking influential nodes in social networks based on node position and neighborhood," *Neurocomputing*, vol. 260, pp. 466–477, 2017.

[26] D. Bucur, "Top influencers can be identified universally by combining classical centralities," *Scientific Reports*, vol. 10, Article ID 20550, 2020.

[27] R. Liu, L. He, X. Liang, X. Yang, and Y. Xia, "Is there any difference in the impact of economic policy uncertainty on the

investment of traditional and renewable energy enterprises? - a comparative study based on regulatory effects," *Journal of Cleaner Production*, vol. 255, p. 120102, 2020.

[28] F. Xiao, "GIQ: a generalized intelligent quality-based approach for fusing multisource information," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 7, pp. 2018–2031, 2021.

[29] F. Xiao, "CaFtR: A fuzzy complex event processing method," *International Journal of Fuzzy Systems*, 2021.

[30] Z. Huang, L. Yang, and W. Jiang, "Uncertainty measurement with belief entropy on the interference effect in the quantum-like Bayesian networks," *Applied Mathematics and Computation*, vol. 347, pp. 417–428, 2019.

[31] J. Zhao and Y. Deng, "Performer selection in human reliability analysis: D numbers approach," *International Journal of Computers, Communications & Control*, vol. 14, no. 4, pp. 521–536, 2019.

[32] Q. Liu, Y. Tian, and B. Kang, "Derive knowledge of Z-number from the perspective of Dempster-Shafer evidence theory," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 754–764, 2019.

[33] Z. Cao and C.-T. Lin, "Inherent fuzzy entropy for the improvement of EEG complexity evaluation," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 1032–1035, 2018.

[34] C. Zhu and F. Xiao, "A belief Hellinger distance for D-S evidence theory and its application in pattern recognition," *Engineering Applications of Artificial Intelligence*, vol. 106, Article ID 104452, 2021.

[35] F. Xiao, "A new divergence measure for belief functions in D-S evidence theory for multisensor data fusion," *Information Sciences*, vol. 514, pp. 462–483, 2020.

[36] F. Xiao, "Generalization of Dempster-Shafer theory: a complex mass function," *Applied Intelligence*, vol. 50, no. 10, pp. 3266–3275, 2020.

[37] A. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *The Annals of Mathematical Statistics*, vol. 38, no. 2, pp. 325–339, 1976.

[38] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, New Jersey, 1976.

[39] T. Bian and Y. Deng, "A new evidential methodology of identifying influential nodes in complex networks," *Chaos, Solitons & Fractals*, vol. 103, pp. 101–110, 2017.

[40] Z. J. Zhou, Y. W. Chen, C. H. Hu, and L. L. Chang, *Evidence Reasoning, Confidence Rule Base and Complex System Modeling*, Science Press, China, 2017.

[41] C. Gao, D. Wei, Y. Hu, S. Mahadevan, and Y. Deng, "A modified evidential methodology of identifying influential nodes in weighted networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 21, pp. 5490–5500, 2013.

[42] D. Wei, X. Deng, X. Zhang, Y. Deng, and S. Mahadevan, "Identifying influential nodes in weighted networks based on evidence theory," *Physica A: Statistical Mechanics and Its Applications*, vol. 392, no. 10, pp. 2564–2575, 2013.

[43] A. Ullah, B. Wang, J. Sheng, J. Long, and N. Khan, "Identification of influential nodes via effective distance-based centrality mechanism in complex networks," *Complexity*, vol. 2021, Article ID 8403738, 16 pages, 2021.

[44] Y. Wang, S. Wang, and Y. Deng, "A modified efficiency centrality to identify influential nodes in weighted networks," *Pramana*, vol. 92, no. 4, p. 68, 2019.

[45] X. Qi, G. Mei, S. Cuomo, and L. Xiao, "A network-based method with privacy-preserving for identifying influential providers in large healthcare service systems," *Future Generation Computer Systems*, vol. 109, pp. 293–305, 2020.

[46] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[47] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.

[48] G. Ren, J. Zhu, and C. Lu, "A measure of identifying influential waypoints in air route networks," *PLoS ONE*, vol. 13, no. 9, p. e0203388, Article ID e0203388, 2018.

[49] P. Bonacich and P. Lloyd, "Eigenvector-like measures of centrality for asymmetric relations," *Social Networks*, vol. 23, no. 3, pp. 191–201, 2001.

[50] S. P. Borgatti, "Centrality and network flow," *Social Networks*, vol. 27, no. 1, pp. 55–71, 2005.

[51] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford InfoLab, Stanford, California,US, 1998.

[52] M. Bianchini, M. Gori, and F. Scarselli, "Inside pagerank," *ACM Transactions on Internet Technology*, vol. 5, no. 1, pp. 92–128, 2005.

[53] A. Kharrazi, E. Rovenskaya, and B. D. Fath, "Network structure impacts global commodity trade growth and resilience," *PLOS ONE*, vol. 12, no. 2, p. e0171184, Article ID 0171184, 2017.

[54] S. Wang, Y. Du, and Y. Deng, "A new measure of identifying influential nodes: efficiency centrality," *Communications in Nonlinear Science and Numerical Simulation*, vol. 47, pp. 151–163, 2016.

[55] F. C. James and C. E. McCulloch, "Multivariate analysis in ecology and systematics: panacea or Pandora's box?" *Annual Review of Ecology and Systematics*, vol. 21, no. 1, pp. 129–166, 1990.

[56] P. H. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*, vol. 573, W. H. Freeman, San Francisco, 1973.

[57] P. Dawyndy, H. D. Meyer, and B. D. Bacts, "UPGMA clustering revisited: a weight-driven approach to transitive approximation," *International Journal of Approximate Reasoning*, vol. 42, no. 3, pp. 174–191, 2006.

Hindawi
Complexity
Volume 2021, Article ID 5337589, 12 pages
https://doi.org/10.1155/2021/5337589

Research Article

Levenberg–Marquardt Backpropagation for Numerical Treatment of Micropolar Flow in a Porous Channel with Mass Injection

Hakeem Ullah,[1] **Imran Khan**,[1] **Hussain AlSalman**,[2] **Saeed Islam**,[1] **Muhammad Asif Zahoor Raja**,[3] **Muhammad Shoaib**,[4] **Abdu Gumaei**,[5] **Mehreen Fiza**,[1] **Kashif Ullah**,[1] **Sk. Md. Mizanur Rahman**,[6] **and Muhammad Ayaz**[1]

[1]*Department of Mathematics, Abdul Wali Khan University Mardan, Mardan, KP 23200, Pakistan*
[2]*Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia*
[3]*Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan*
[4]*Department of Mathematics, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan*
[5]*Department of Computer Science Faculty of Applied Sciences, Taiz University, Taiz 6803, Yemen*
[6]*Information and Communication Engineering Technology, School of Engineering Technology and Applied Science, Centennial College, Toronto, Canada*

Correspondence should be addressed to Hakeem Ullah; hakeemullah1@gmail.com and Abdu Gumaei; abdugumaei@taiz.edu.ye

Received 8 June 2021; Accepted 19 November 2021; Published 13 December 2021

Academic Editor: Murari Andrea

In this research work, an effective Levenberg–Marquardt algorithm-based artificial neural network (LMA-BANN) model is presented to find an accurate series solution for micropolar flow in a porous channel with mass injection (MPFPCMI). The LMA is one of the fastest backpropagation methods used for solving least-squares of nonlinear problems. We create a dataset to train, test, and validate the LMA-BANN model regarding the solution obtained by optimal homotopy asymptotic (OHA) method. The proposed model is evaluated by conducting experiments on a dataset acquired from the OHA method. The experimental results are obtained by using mean square error (MSE) and absolute error (AE) metric functions. The learning process of the adjustable parameters is conducted with efficacy of the LMA-BANN model. The performance of the developed LMA-BANN for the modelled problem is confirmed by achieving the best promise numerical results of performance in the range of E-05 to E-08 and also assessed by error histogram plot (EHP) and regression plot (RP) measures.

1. Introduction

A few years ago, Eringen [1, 2] firstly presented the idea of micropolar fluids. Theories of non-Newtonian fluid are developed to describe the behavior of the fluid that does not obey Newton's law, such as micropolar fluids. This fluid summarizes specific non-Newtonian behaviors, such as liquid with polymer additives, liquid crystals, animal blood particles, suspensions, and topographic features. The governing equations of many physical problems are nonlinear in nature and cannot be solved analytically; therefore, the scientist developed some approximate and numerical techniques, such as perturbation-based methods [3, 4], homotopy perturbation-based methods [5–7], homotopy analysis-based methods [8–10], collocation-based method [11–18], and Adomian decomposition-based methods [19, 20]. Among these methods, artificial intelligence (AI)-based numerical methods have been broadly designed for solving differential equations in several diverse applications [21–23]. A few latest research works to solve the problems of

nonlinear systems include the study introduced for local fractional partial differential equations [24], fourth-order nonlinear differential equations [25], Riccati equation control of nonlinear uncertain systems [26], analytic solution of micropolar flow using the homotopy analysis method [27], and pantograph delay differential equation [28]. However, these numerical-based methods need discretization and improved linearization techniques, which only allow computing the solution for certain standards variables and required huge computer memory and time. For optimizing the results, convergence and stability should be considered to avoid divergence. The perturbative methods required the assumption of small parameters, which is itself an issue. Besides, there is no study yet has been applied a fast backpropagation method for finding an accurate series solution to micropolar flow in a porous channel with mass injection (MPFPCMI).

Therefore, the aim of this study is to develop a Levenberg–Marquardt algorithm-based artificial neural network (LMA-BANN) model to solve the nonlinear governing equation of MPFPCMI. The contributions and mechanisms of the proposed work fall under following points:

(i) The LMA-BANN model is developed to analyze MPFPCMI for diverse scenarios on variants of physical parameters.

(ii) The dataset for LMA-BANN is obtained by variations of different parameters with the help of the OHA method.

(iii) The validity and accuracy of LMA-BANN are confirmed by comparing its results at different cases and scenarios. The results of training, testing, and validation are subjected by displaying the MPFPCMI for diverse scenarios.

(iv) The MSE metric function, EHP, and RP results are obtained with the help of plots to show the performance of LMA-BANN for finding an accurate series solution of MPFPCMI successfully.

2. Problem Formulations

Consider the steady, incompressible, and laminar flow of micropolar fluid along two-dimensional channel with porous walls. The mass fluid is introduced with speed q. The walls of the channel are adjusted at $y = \pm h$, where $2h$ is the width of the channel [29, 30]. The fundamental equations governing are as follows [31, 32]:

$$\underset{\sim}{u}_x + \underset{\sim}{v}_y = 0, \tag{1}$$

$$\underset{\sim}{u}\,\underset{\sim}{u}_x + \underset{\sim}{v}\,\underset{\sim}{u}_y = -\frac{1}{\rho}p_x + \left(v + \frac{k}{\rho}\right)\left(\underset{\sim}{u}_{xx} + \underset{\sim}{u}_{yy}\right) + \frac{k}{\rho}N_x, \tag{2}$$

$$\underset{\sim}{u}\,\underset{\sim}{v}_x + \underset{\sim}{v}\,\underset{\sim}{v}_y = -\frac{1}{\rho}p_y + \left(v + \frac{k}{\rho}\right)\left(\underset{\sim}{v}_{xx} + \underset{\sim}{v}_{yy}\right) - \frac{k}{\rho}N_x, \tag{3}$$

$$\underset{\sim}{u}\,N_x + \underset{\sim}{v}\,N_y = -\frac{1}{\rho j}\left(2N + \underset{\sim}{u}_y - \underset{\sim}{v}_x\right) + \frac{v_s}{\rho j}\left(N_{xx} + N_{yy}\right). \tag{4}$$

The suitable conditions for physical boundaries are as follows [26, 27]:

$$\begin{aligned} \underset{\sim}{u}(x, \pm h) &= 0, \\ \underset{\sim}{v}(x, \pm h) &= \pm q, \\ N(x, \pm h) &= -s\underset{\sim}{u}_y|_{(x,\pm h)}. \end{aligned} \tag{5}$$

The symmetric flow (SF) is as follows [33]:

$$\begin{aligned} \underset{\sim}{u}_x(x, 0) &= 0, \\ \underset{\sim}{v}(x, 0) &= 0. \end{aligned} \tag{6}$$

Here, q greater than zero relates to suctions, q less than zero corresponds to injections, and "s" is a finite-parameters worn to model the degree to which microelements in the region of the channel walls are free to rotate, e.g., as "s" is equal to zero is the case where microelements near the boundary cannot turn around when $s = 0.5$, the situation of microrotation is identical to the velocity of the fluid at the end. Kelson et al. [29] developed the following equations:

$$\begin{aligned} \phi &= -qx\,\underset{\sim}{f}(\eta)\cdot \\ N &= \frac{qx}{h^2}\,\underset{\sim}{g}(\eta), \end{aligned} \tag{7}$$

where

$$\begin{aligned} \eta &= \frac{y}{h}, \\ \underset{\sim}{u} &= \phi_y \\ &= -\frac{qx}{h}f'(\eta), \\ \underset{\sim}{v} &= -\phi_x = \eta f(\eta). \end{aligned} \tag{8}$$

The Navier–Stokes equations (1)–(4) decrease via applying equations (7) and (8):

$$\begin{aligned} (1 + N_1)\overset{iv}{\underset{\sim}{f}} - N_1\,\underset{\sim}{g}'' - \mathrm{Re} \times \left(\underset{\sim}{f}\,\underset{\sim}{f}''' - \underset{\sim}{f}'\,\underset{\sim}{f}''\right) &= 0, \\ N_2\,\underset{\sim}{g}'' + N_1\left(\underset{\sim}{f}'' - 2\,\underset{\sim}{g}\right) - N_3\mathrm{Re}\left(\underset{\sim}{f}\,\underset{\sim}{g}' - \underset{\sim}{f}'\,\underset{\sim}{g}\right) &= 0. \end{aligned} \tag{9}$$

Dimensionless parameters are established as follows:

$$N_1 = \frac{k}{\rho v},$$
$$N_2 = \frac{v_s}{\rho v h^2},$$
$$N_3 = \frac{j}{h^2},$$
$$\mathrm{Re} = \frac{qh}{v}. \tag{10}$$

When viscosity parameter (Re) is more significant than zero used for suction and less than for injection, the BCs are

$$\underset{\sim}{f}(\pm 1) = 1,$$
$$\underset{\sim}{f}'(\pm 1) = 0, \tag{11}$$
$$\underset{\sim}{g}(\pm 1) = s \times \underset{\sim}{f}''(\pm 1).$$

The SF is as follows:

$$\underset{\sim}{f}(0) = 0,$$
$$\underset{\sim}{f}''(0) = 0,$$
$$\underset{\sim}{f}'(1) = 0, \tag{12}$$
$$\underset{\sim}{f}(1) = 1,$$
$$\underset{\sim}{g}(1) = s \times \underset{\sim}{f}''(1).$$

By applying Kelson et al. [29], we put $s = 0$, $N_1 = 1, N_2 = 1$, and $N_3 = 0.1$:

$$2\overset{iv}{\underset{\sim}{f}} - \mathrm{Re}\,\underset{\sim}{f}\,\underset{\sim}{f}''' = -\mathrm{Re}\,\underset{\sim}{f}'\,\underset{\sim}{f}'' + \underset{\sim}{g}'', \tag{13}$$

$$\underset{\sim}{g}'' + \underset{\sim}{f}'' - 0.1\mathrm{Re}\,\underset{\sim}{f}\,\underset{\sim}{g}' = 0.1\mathrm{Re}\,\underset{\sim}{f}'\,\underset{\sim}{g} + 2\,\underset{\sim}{g}. \tag{14}$$

The BCs are as follows:

$$\underset{\sim}{f}(0) = 0,$$
$$\underset{\sim}{f}''(0) = 0,$$
$$\underset{\sim}{f}'(1) = 0,$$
$$\underset{\sim}{f}(1) = 1, \tag{15}$$
$$\underset{\sim}{g}(1) = 0,$$
$$\underset{\sim}{g}(0) = 0.$$

3. Numerical Experimental Results with Discussion

A short overview of the scheme proposed for finding the proposed LMA-BANN numerical experimentation continuity and momentum equations, i.e., 2–4, based on MPFPCMI is accessible in this section. The proposed structure of stepwise flow is presented in Figure 1 by using "nftool" of the NN tool-box existed in MATLAB. LMA-BANN is implemented for two-layer structures that include single input hidden and output of feed-forward network by LMA-based backpropagation process. Figure 2 demonstrates the structural design of ANN based on ten neuron numbers with a data-sigmoid activation function.

A reference dataset for LMA-BANN for equations (14) and (15) is created between intervals [0, 1] for 201 input grids. Now, 80 percent of data are used for training as 10 percent is for testing and 10 percent is for validation in the event of a 2-layer feed-forward ANN structure fitting tool with LMA backpropagation to solve all problems of MPFPCMI. Training data are used to establish the estimated solution on the source of the MSE, validation data are used to LMA-BANN, and even as test data are used to assess the truthful input performance.

Figures 3 and 4 show the effects of LMA-BANN performance represented by error histograms and fitting of solutions for two cases of the MPFPCMI scenario, while the regression tests are shown in Figure 5 for two cases of the MPFPCMI scenario. Furthermore, the MSE, number epochs, and other convergence parameters for training, validation, and testing data are tabulated in Tables 1 and 2, for two cases of the MPFPCMI scenario. Also, in Figures 3(a) and 3(c) of the MPFPCMI scenario, the convergence of MSE for the train, validation, and test process is provided for the two cases of the MPFPCMI. The best network performance for different scenarios is $9.889E-13$ at 110 epochs and $9.889E-13$ at 110 epochs. The gradient and step-size *Mu* of backpropagation nearly $[9.8879\times10^{-08}$ and $9.8879\times10^{-08}]$ and $[10^{-11}, 10^{-11}]$ are as shown in Figures 3(b) and 3(d). The results indicate the exact and convergent output for each case of the LMA-BANN.

The result obtained by LMA-BANN is achieved with the numerical outcome of the OHA method for two cases scenario, and the outcomes are shown in Figures 4(b) and 4(d) with the input points between {0–1} and step-size 1/100. The considerable error attained for training, testing, and validation statistics by planned ANN-LMM is less than 0.9×10^{-06} and -0.9×10^{-06} for the two cases of the scheme design. For each input stage, the error dynamics are further evaluated by error histograms, and results are given in Figures 4(a) and 4(c) for two cases of the MPFPCMI scenario. The error with 20-bin is nearly $-1.4E-08$ and $-1.4E-08$ for scenarios of MPFPCMI.

The outcomes for solving the LMA-BANN for different scenarios are shown in Tables 1 and 2, respectively. The output of LMA-BANN is about 10^{-13} to 10^{-12} and 10^{-10} to 10^{-11} for scenarios of MPFPCMI. The reliable output of

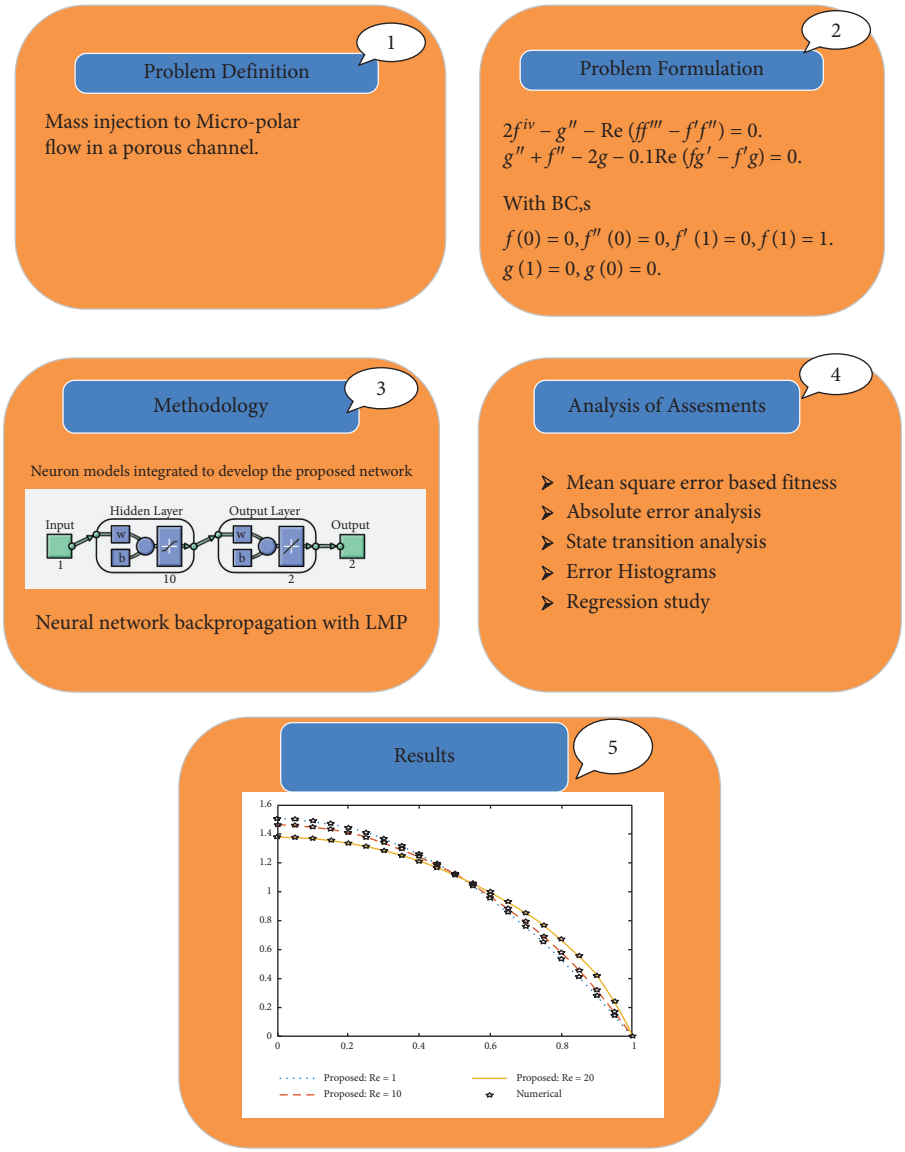


Figure 1: Process workflow of proposed LMA-BANN for MPFPCMI.

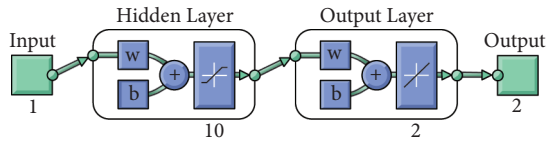


Figure 2: Neural network model architecture.

LMA-BANN for explaining MPFPCMI is illustrated in these results. The results of LMA-BANN for the velocity and the profiles for the scenarios are therefore calculated. The experimental results of LMA-BANN are obtained for the velocity profiles $f(\eta)$, $f'(\eta)$, and $g(\eta)$ of the adopted scenarios. The outcomes of $f(\eta)$, $f'(\eta)$, and $g(\eta)$ profiles are shown in Figures 6–8 of the three scenarios of the MPFPCMI. Increasing the value of Re increases the velocity in x-direction and mass injection. The order to access the correctness gauges, the outcomes of LMA-BANN matched with the OHA method solutions in situations, and absolute errors (AEs) with references are calculated, and results are displayed in Figures 6(b), 7(b), and 8(b) for the three scenarios of the MPFPCMI. The AE is about 10^{-05} to 10^{-04}, 10^{-06} to 10^{-07}, 10^{-04} to 10^{-07}, 10^{-05} to 10^{-04}, 10^{-05} to 10^{-06}, 10^{-06} to 10^{-08}, 10^{-06} to 10^{-04}, 10^{-05} to 10^{-06}, and 10^{-05} to 10^{-06} for Re = 1, Re = 6, and Re = 20 and Re = 1, Re = 10, and Re = 20 of the scenarios. All these numerical and also graphical figures validated the reliable, convergent, and effective relationship of the LMA-BANN computing

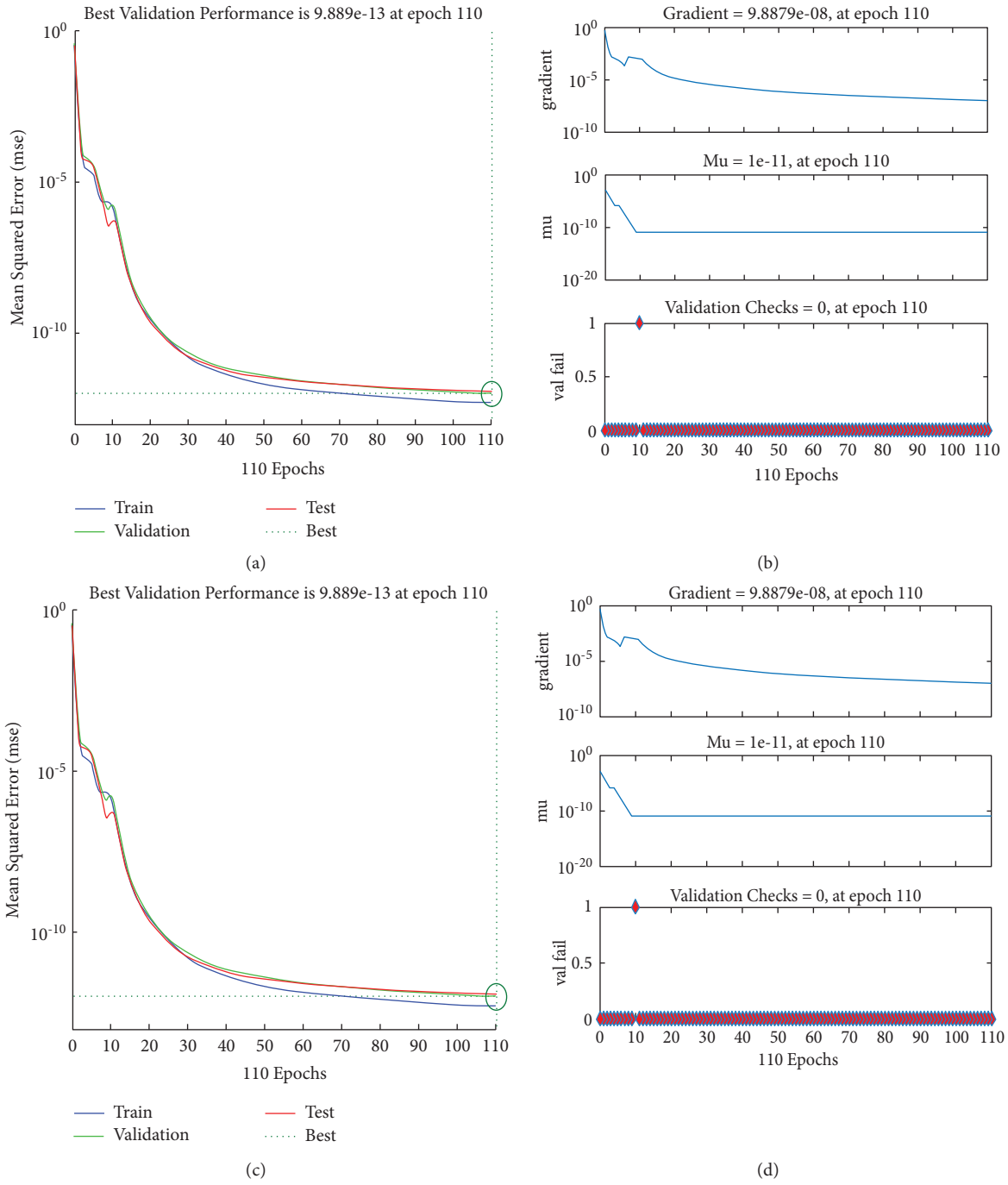


Figure 3: The performance and MSE outcomes of LMA-BANN in two cases of the MPFPCMI scenario.

approach for solving the variants of MPFPCMI. In Figure 7(a), the special effects of the Reynolds numbers on the profile of the velocity index are presented. For the velocity index $f(\eta)$, the Reynolds numbers increase with increasing the velocity index $f(\eta)$ of the injection case. The direct effect of the Reynolds number on the rotation profile of the fluid is presented in Figure 8(a). By increasing the Reynolds number, the rotational profile index of the fluid flow reduces up to $\eta = 0.6$, and then the rotation profile increases with the increase in the Reynolds number. Increasing the Reynolds number with a minimum rotation occurred does not make it move away from the origin of the channel.

According to the results in Table 1, we can see that the model achieves the values $2.07665E-12$, $4.16509E-9$, and $2.3595E-10$ of the MSE on the test data for cases 1, 2, and 3, when the number of epochs are 121, 214, and 223, respectively. We notice that all these values of the MSE are too low, confirming the effectiveness of the model. However, the lowest value is obtained for case 1 at the smallest value of

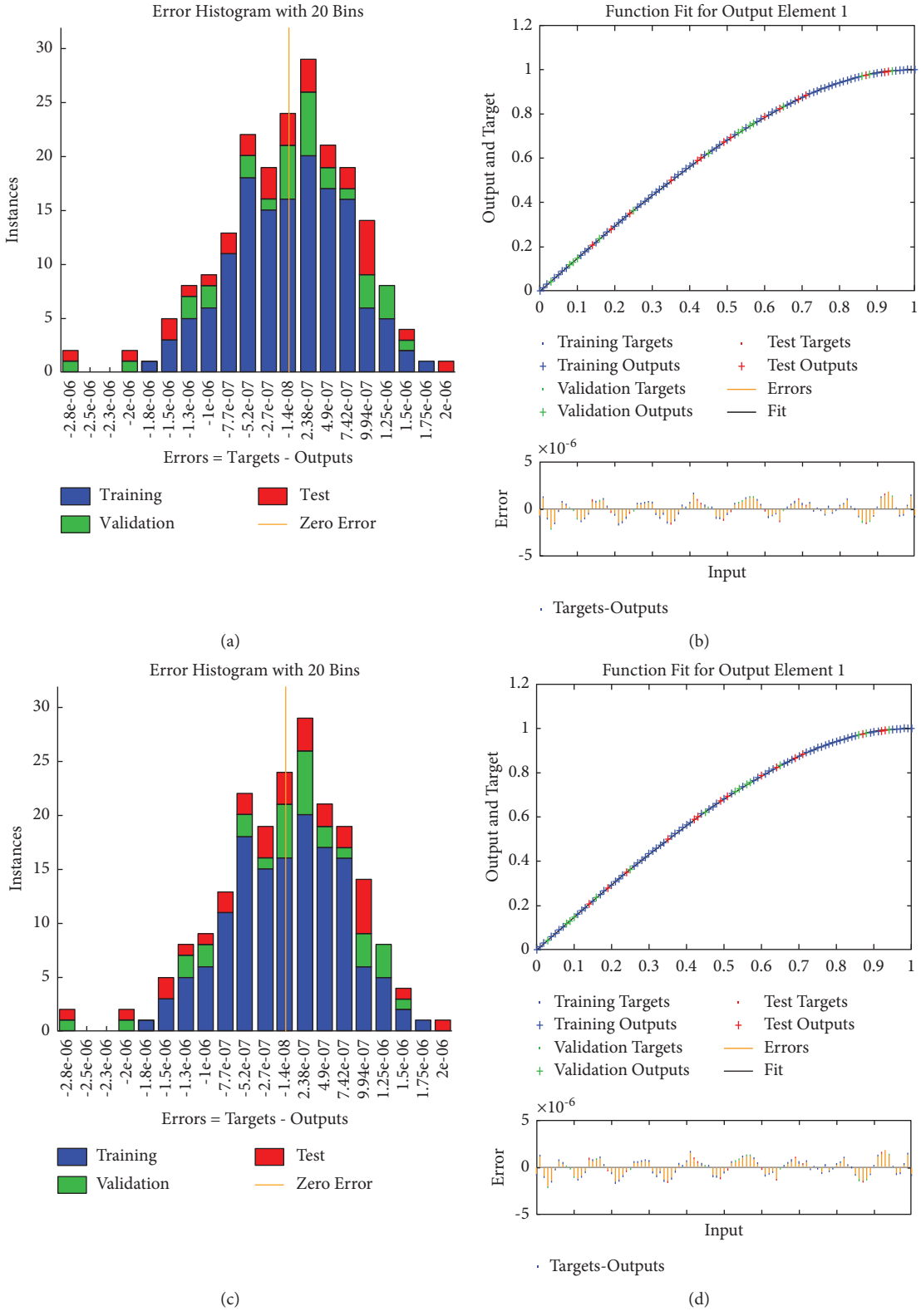
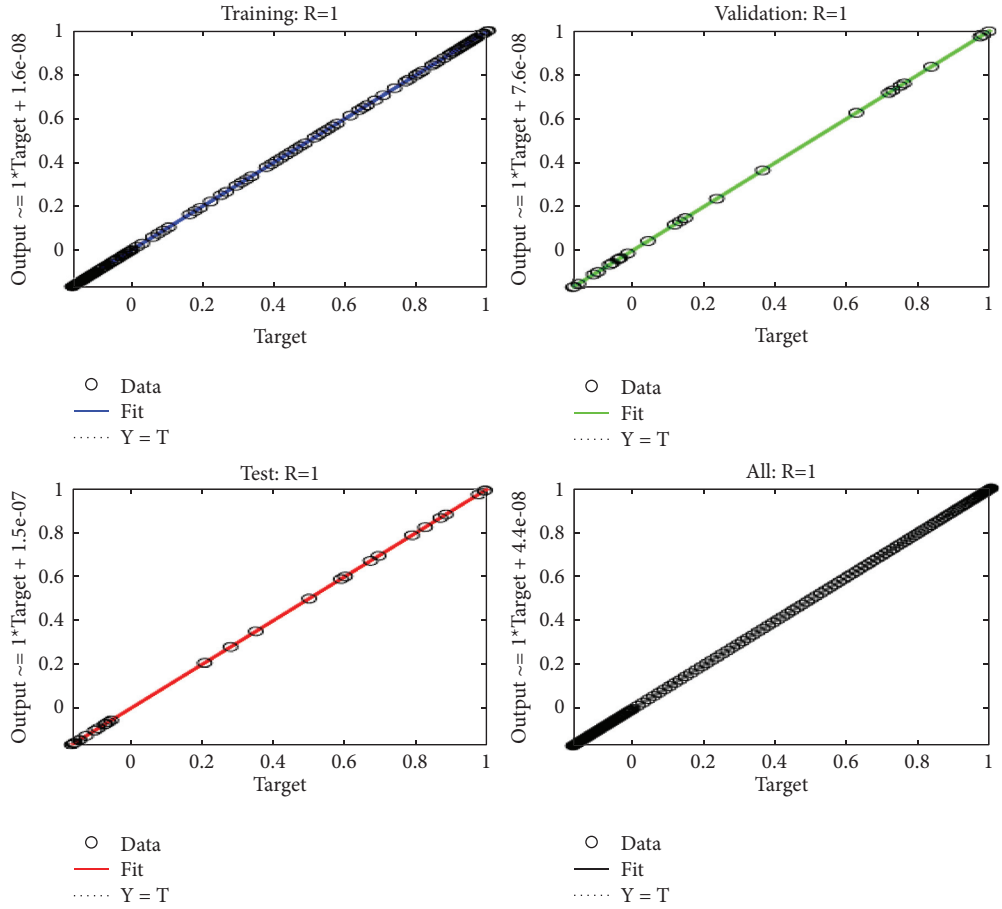
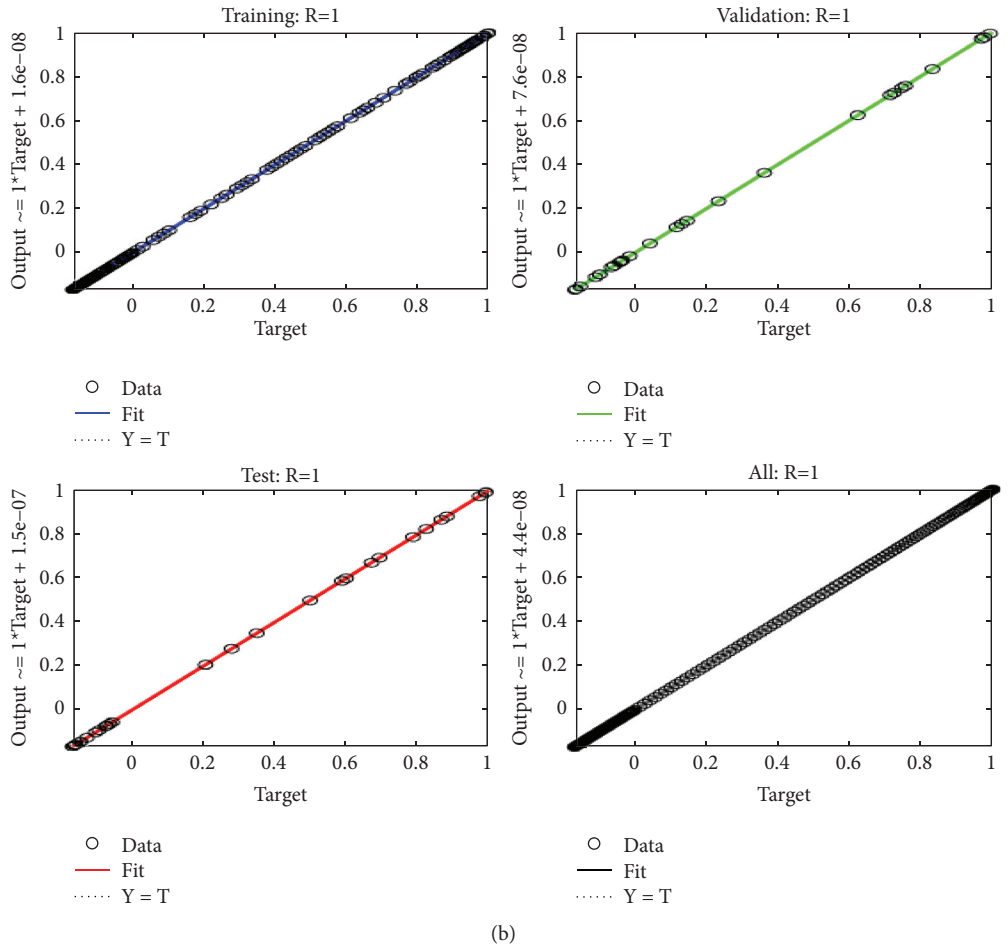


Figure 4: Outcomes of error histograms and fitting graphs of proposed LMA-BANN in two cases of the MPFPCMI scenario.



(a)

Figure 5: Continued.



(b)

Figure 5: Regression plot of LMA-BANN for two cases of the MPFPCMI scenario.

Table 1: Data of LMA-BANN for scenario 1 of the MPFPCMI.

Case	MSE			Performance	Gradient	*Mu*	Epoch
	Training	Validation	Testing				
1	$4.77038E-13$	$1.07665E-12$	$2.07665E-12$	$4.77E-13$	$9.83E-08$	$1.00E-11$	121
2	$3.31283E-11$	$1.53447E-10$	$4.16509E-9$	$3.31E-11$	$9.96E-08$	$1.00E-09$	214
3	$7.61638E-12$	$1.95429E-9$	$2.3595E-10$	$7.57E-12$	$9.84E-08$	$1.00E-10$	123

Table 2: Data of LMA-BANN for scenario 2 of the MPFPCMI.

Case	MSE			Performance	Gradient	*Mu*	Epoch
	Training	Validation	Testing				
1	$5.40353E-10$	$1.0355E-9$	$1.08132E-9$	$5.40E-10$	$9.95E-08$	$1.00E-09$	92
2	$3.93472E-10$	$7.23124E-10$	$4.7336E-10$	$3.93E-10$	$1.00E-07$	$1.00E-08$	329
3	$4.45013E-11$	$5.86838E-11$	$8.78423E-11$	$4.45E-11$	$9.90E-08$	$1.00E-09$	177

epoch's number. In addition, from the results in Table 2, it obvious that the model gets the values $1.08132E-9$, $4.7336E-10$, and $8.78423E-11$ for the test data at epoch's number 92, 329, and 177 and through the settings of case 1, case 2, and case 3. The lowest value is obtained for case 3 at epoch's number 177, which is also an acceptable number of epochs. This also confirms the efficiency of the LMA-BANN model. Finally, from Figures 6(b)–8(b), we can see that lowest values of error plot are for Re = 10 for all scenarios, which means that this value is suitable for Re parameter.

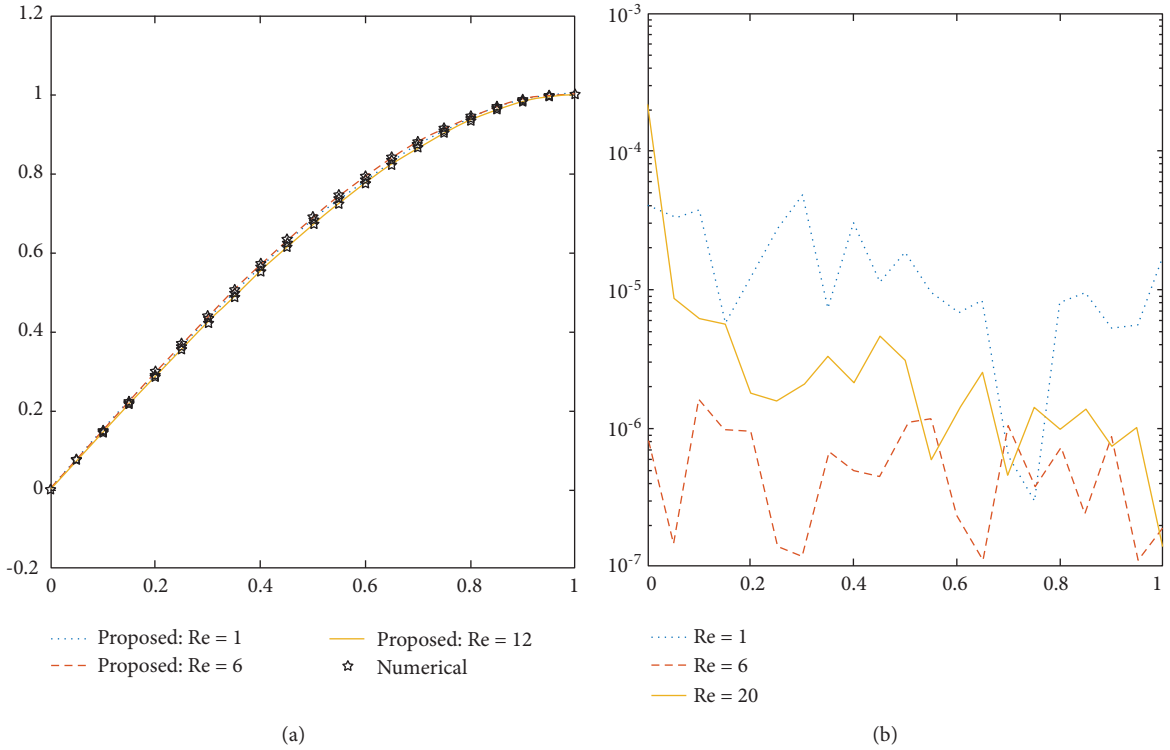


FIGURE 6: Evaluation of proposed LMA-BANN with the numerical reference results for the MPFPCMI scenario 1: (a) outcomes for different value of Re in f; (b) error plot of Re in f.

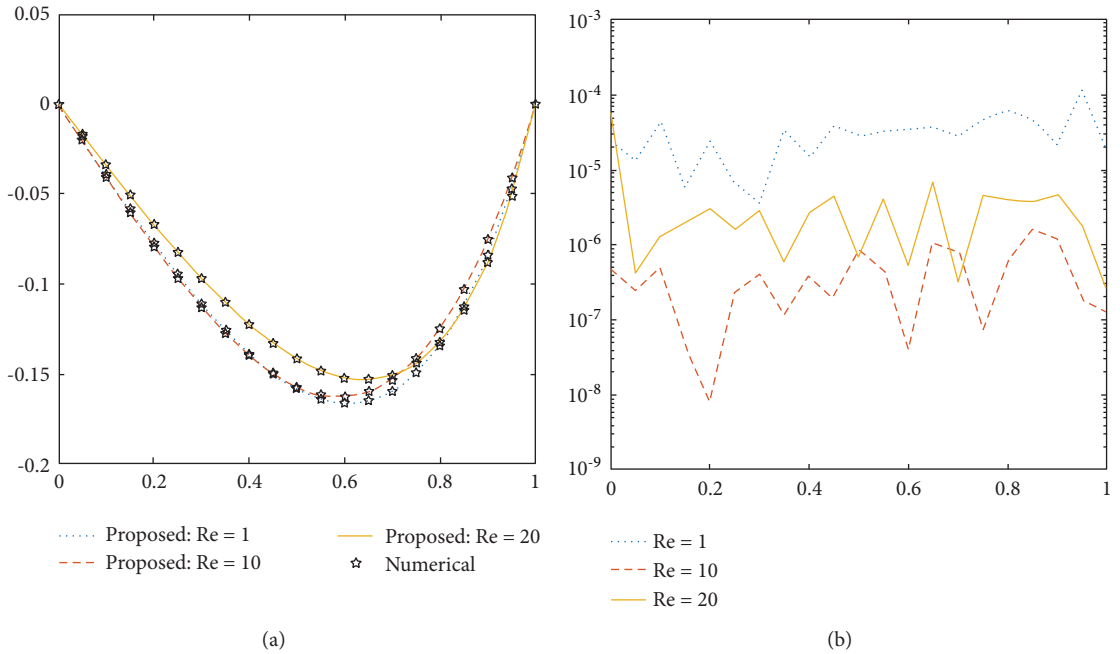


FIGURE 7: Evaluation of proposed LMA-BANN with the numerical reference results for the MPFPCMI scenario 2: (a) outcomes for different value of Re in g; (b) error plot of Re in velocity g.

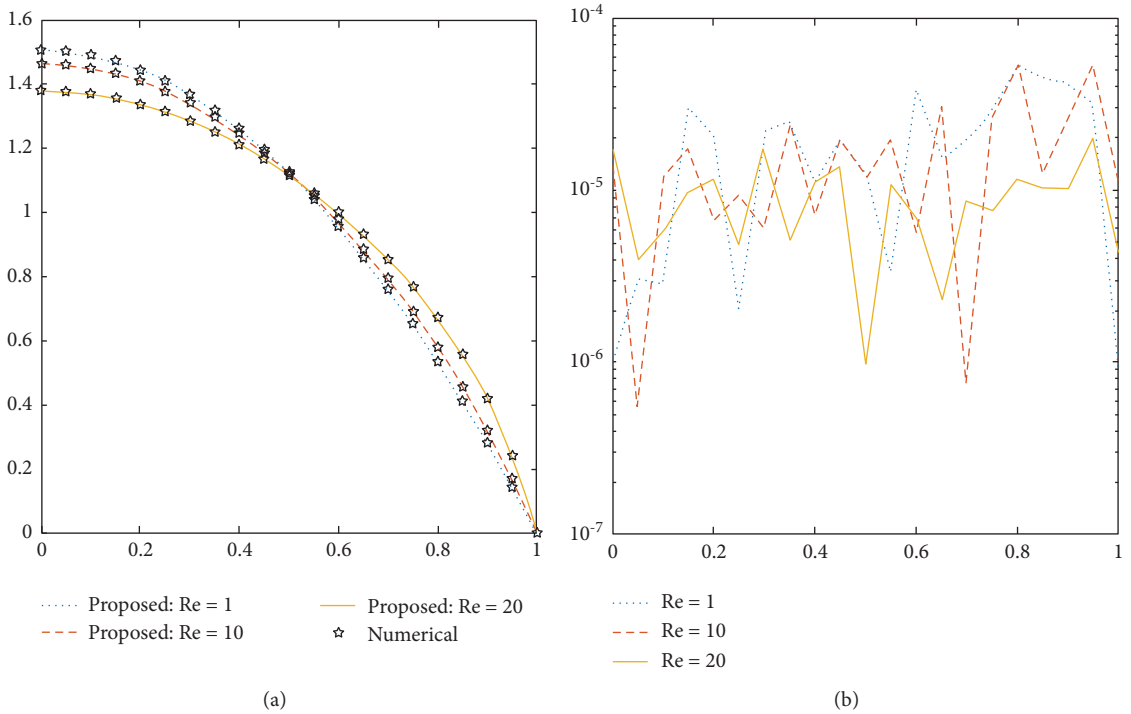


Figure 8: Evaluation of proposed LMA-BANN with the numerical reference results for the MPFPCMI scenario 3: (a) outcomes for different value of Re in f; (b) error plot of Re in velocity f.

4. Conclusion

The LMA-BANN is used as an artificial intelligence-based integrated method to find an accurate series solution for the MPFPCMI. The partial differential equations (PDEs) system of the MPFPCMI is converted to the order differential equations (ODEs) system by using the ability of similarity variables. The OHA method is used for producing the dataset of the MPFPCMI. Different measurable quantity of a set of metrics is utilized for evaluating the developed model. For training the model, a percentage equals to 80% of the data used, and a percentage of 10% from the remaining is used for testing, as well as the last 10% of the reference data is applied for validating the LMA-BANN model. The near values of both planned and reference outcomes matching of level between 10^{-05} to 10^{-07} confirm the rightness of solution, and supposed feature is additional authentic via numerical and graphical design of for the convergence of MSE, AE, error histogram plot, and regression plot measures. After this assurance, the results are demonstrated for the rotating and velocity profile when there are different values of Reynolds number and viscosity parameter (Re). From the numerical results of the problem, we get that increasing the value of Re decreases the velocity in x-direction and mass injection. Moreover, the experimental results confirmed the effectiveness of the LMA-BANN model for accurate analysis of MPFPCMI. In future work, we plan to apply the designed model for finding a solution to another nonlinear system with a large dataset and with different percentages of testing for more analyzing and investigating.

Abbreviations

$\tilde{B}$:	Magnetic field (NmA^{-1})
C:	Fluid concentration
c_p:	Specific heat (J/kgK)
β:	Non-Newtonian parameter
$\tilde{E}$:	Electric field intensity (NC^{-1})
J_w:	Mass flux
$\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3$:	Material constants
A_1, A_2, A_3:	Kinematic tensors
k:	Thermal conductivity $(Wm^{-1}K^{-1})$
M:	Magnetic parameter
n_e:	Number density of electron
O:	Origin
P:	Fluid pressure (Pa)
Pr:	Prandtl number
Q_w:	Heat flux (Wm^{-2})
q_r:	Radioactive heat flux (J)
Re:	Viscosity parameter
S:	Cauchy stress tensor
t_e:	Flow time (s)
T:	Fluid temperature (K)
u, v, w:	Velocities components (ms^{-1})
x, y, z:	Coordinates.

Greek Letters

α:	Thermal diffusivity (m^2s^{-1})
$\tilde{\kappa}$:	Vertex viscosity (mPa)
μ:	Dynamic viscosity (mPa)
υ:	Kinematic coefficient of viscosity

ρ_f: Base fluid density (Kgm^{-3})
ρ_b: Density of the particles (Kgm^{-3}).

Data Availability

All the relevant data are included within the manuscript.

Conflicts of Interest

The authors declare that there are no conflicts of interest about the publication of the research article.

Acknowledgments

This research was supported by the Researchers Supporting Project (number RSP-2021/244), King Saud University, Riyadh, Saudi Arabia.

References

[1] A. C. Eringen, "Simple microfluids," *International Journal of Engineering Science*, vol. 2, no. 2, pp. 205–217, 1964.
[2] A. Eringen, "Theory of micropolar fluids," *Indiana University Mathematics Journal*, vol. 16, no. 1, pp. 1–18, 1966.
[3] A. H. Nayfeh, "Introduction to Perturbation Techniques," *ASME Journal of Heat Transfer*, vol. 107, no. 1, pp. 248–250, 1979.
[4] R. H. Rand and D. Armbruster, *Perturbation Methods, Bifurcation Theory and Computer Algebraic*, Springer, Berlin, Germany, 1987.
[5] A. Shehzad and R. Ali, "Approximate analytic solution formagneto-hydrodynamic flow of a non-Newtonian fluid over a vertical stretching sheet," *Canadian Journal of Pure and Applied Sciences*, vol. 2, pp. 202–215, 2012.
[6] M. Turkyilmazoglu, "Some issues on HPM and HAM methods: a convergence scheme," *Mathematical and Computer Modelling*, vol. 53, pp. 1929–1936, 2011.
[7] T. Hussain, T. Hayat, S. A. Shehzad, A. Alsaedi, and B. Chen, "A model of solar radiation and joule heating in flow of third grade nanofluid," *Zeitschrift für Naturforschung A*, vol. 70, no. 3, pp. 177–184, 2015.
[8] S. J. Liao, *The proposed homotopy analysis technique for the solution of nonlinear problems*, PhD thesis, Shanghai Jiao Tong University, Shanghai, China, 1992.
[9] Z. Shah, M. Sheikholeslami, and P. Kumam, "Simulation of entropy optimization and thermal behavior of nanofluid through the porous media," *International Communications in Heat and Mass Transfer*, vol. 120, no. 8, Article ID 105039, 2021.
[10] Z Shah, E. Alzahrani, M. Jawad, and U. Khan, "Microstructure and inertial characteristics of MHD suspended SWCNTs and MWCNTs based maxwell nanofluid flow with bio-convection and entropy generation past a permeable vertical cone," *Coatings*, vol. 10, no. 10, p. 998, 2020.
[11] R. Alt and J. Vignes, "Validation of results of collocation methods for ODEs with the CADNA library," *Applied Numerical Mathematics*, vol. 21, no. 2, pp. 119–139, 1996.
[12] C. Franke and R. Schaback, "Solving partial differential equations by collocation using radial basis functions," *Applied Mathematics and Computation*, vol. 93, no. 1, pp. 73–82, 1998.
[13] C. Franke and R. Schaback, "Convergence order estimates of meshless collocation using radial basis functions," *Advances in Computational Mathematics*, vol. 8, no. 4, pp. 381–399, 1998.
[14] S. U. Siraj-ul-Islam, S. Haq, and A. Ali, "A meshfree method for the numerical solution of the RLW equation," *Journal of Computational and Applied Mathematics*, vol. 223, no. 2, pp. 997–1012, 2009.
[15] M. Uddin, S. Haq, and S. U. Siraj-ul-Islam, "Numerical solution of complex modified Korteweg-de Vries equation by mesh-free collocation method," *Computers & Mathematics with Applications*, vol. 58, no. 3, pp. 566–578, 2009.
[16] M. Uddin, S. Haq, and S. U. Siraj-ul-Islam, "A mesh-free numerical method for solution of the family of Kuramoto-Sivashinsky equations," *Applied Mathematics and Computation*, vol. 212, no. 2, pp. 458–469, 2009.
[17] S. Haq, S. U. Islam, and M. Uddin, "A meshfree method for the numerical solution of the KdVBurgers equation," *Applied Mathematical Modelling*, vol. 33, no. 8, pp. 3442–3449.
[18] S. Haq, S. U. Islam, and M. Uddin, "A meshfree interpolation method for the numerical solution of the coupled nonlinear partial differential equations," *Engineering Analysis with Boundary Elements*, vol. 33, no. 3, pp. 399–409, 2009.
[19] A.-M. Wazwaz, "Adomian decomposition method for a reliable treatment of the Bratu-type equations," *Applied Mathematics and Computation*, vol. 166, no. 3, pp. 652–663, 2005.
[20] H. Hameed, A. M. Siddiqui, B. M. Siddique, and Q. K. Ghori, "Use of adomian decomposition method in the study of parallel plate flow of a third grade fluid," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2388–2399, 2010.
[21] M. Cakmak and S. Alkan, "A numerical method for solving a class of systems of nonlinear Pantograph differential equations," *Alexandria Engineering Journal*, 2021.
[22] B. Sun, S. Wen, S. Wang, T. Huang, Y. Chen, and P. Li, "Quantized synchronization of memristive neural networks with time-varying delays via super-twisting algorithm," *Neurocomputing*, vol. 380, pp. 133–140, 2020.
[23] M. A. Zaky, "An accurate spectral collocation method for nonlinear systems of fractional differential equations and related integral equations with nonsmooth solutions," *Applied Numerical Mathematics*, vol. 154, pp. 205–222, 2020.
[24] D. Ziane, C. Mountassir Hamdi, C. Cattan, and K. Belghaba, "Yang-laplace decomposition method for nonlinear system of local fractional partial differential equations," *Applied Mathematics and Nonlinear Sciences*, vol. 4, no. 2, pp. 489–502, 2019.
[25] O. Moaaz, I. Dassios, O. Bazighifan, and A Muhib, "Oscillation theorems for nonlinear differential equations of fourth-order," *Mathematics*, vol. 8, no. 4, p. 520, 2020.
[26] S. R. Nekoo, "Model reference adaptive state-dependent Riccati equation control of nonlinear uncertain systems: regulation and tracking of free-floating space manipulators," *Aerospace Science and Technology*, vol. 84, pp. 348–360, 2019.
[27] H. Hassan and M. M. Rashidi, "An analytic solution of micropolar flow in a porous channel with mass injection using homotopy analysis method," *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 24, no. 2, 2014.
[28] I. Khan, M. A. Z. Raja, M. Shoaib et al., "Design of neural network with Levenberg-Marquardt and Bayesian regularization backpropagation for solving pantograph delay differential equations," *IEEE Access*, vol. 8, pp. 137918–137933, 2020.

[29] N. A. Kelson, A. Desseaux, and T. W. Farrell, "Micropolar flow in a porous channel with high mass transfer," *ANZIAM Journal*, vol. 44, pp. 479–495, 2003.

[30] A. Desseaux and N. A. Kelson, "Solutions for the flow of a micropolar fluid in a porous channel," in *Proceedings of the 4th Biennial Engineering Mathematics and Applications Conference*, pp. 115–118, Melbourne, Australia, September 2000.

[31] Z. Ziabakhsh and G. Domairry, "Homotopy analysis solution of micro-polar flow in a porous channel with high mass transfer," *Advances in Theoretical and Applied Mathematics*, vol. 1, pp. 79–94, 2008.

[32] M. M. Rashidi and E. Erfani, "A novel analytical method to investigate effect of radiation on flow of magneto micropolar fluid past a continuously moving plate with suction and injection," *International Journal of Modeling Simulation Scientific Computation*, vol. 1, no. 2, pp. 219–238, 2010.

[33] M. M Rashidi, S. A. M. Pour, and N Laraqi, "A semi-analytical solution of micro polar flow in a porous channel with mass injection by using differential transform method," *Nonlinear Analysis Modelling and Control*, vol. 15, no. 3, pp. 341–350, 2010.

Hindawi
Complexity
Volume 2021, Article ID 9754368, 2 pages
https://doi.org/10.1155/2021/9754368

Corrigendum

Corrigendum to "Multivariable Model Reference Adaptive Control of an Industrial Power Boiler Using Recurrent RBFN"

Jafar Tavoosi,[1] **Yavar Azarakhsh**,[1] **Ardashir Mohammadzadeh**,[2] **Saleh Mobayen**,[3] **Jihad H. Asad**,[4] **and Rabia Safdar**[5,6]

[1]*Department of Electrical Engineering, Faculty of Engineering, Ilam University, Ilam, Iran*
[2]*Department of Electrical Engineering, University of Bonab, Bonab, Iran*
[3]*Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin 64002, Douliu, Taiwan*
[4]*Department of Physics, Faculty of Applied Sciences, Palestine Technical University, P. O. Box 7, Tulkarm, State of Palestine*
[5]*Department of Mathematics, Lahore College Women University, Lahore, Pakistan*
[6]*Department of Mathematics, College of Arts and Sciences, University of Jhang, Jhang, Pakistan*

Correspondence should be addressed to Jafar Tavoosi; j.tavoosi@ilam.ac.ir and Saleh Mobayen; mobayens@yuntech.edu.tw

Received 11 November 2021; Accepted 11 November 2021; Published 11 December 2021

In the article titled "Multivariable Model Reference Adaptive Control of an Industrial Power Boiler Using Recurrent RBFN" [1], there were errors in Figures 1 and 3, which should be corrected as follows.

Additionally, "RBFN-based" in the abstract should be corrected to "radial-basis function neural network (RBFN)." On page 1 of the Introduction, "PID" should be corrected to "proportional-integral-derivative (PID)."

Figure 1: A power plant boiler.

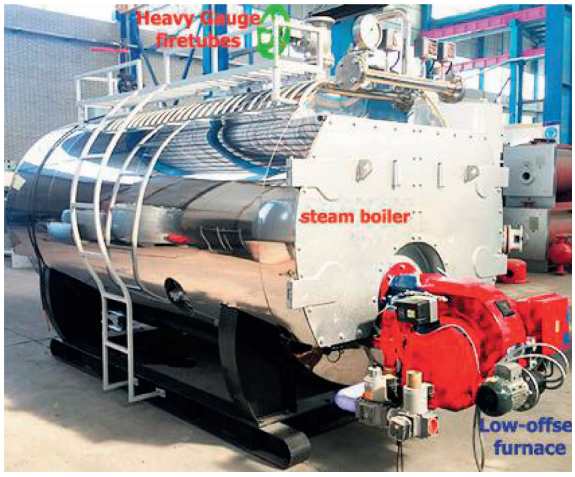


Figure 3: The industrial boiler.

References

[1] J. Tavoosi, Y. Azarakhsh, A. Mohammadzadeh, S. Mobayen, J. H. Asad, and R. Safdar, "Multivariable model reference adaptive control of an industrial power boiler using recurrent RBFN," *Complexity*, vol. 2021, Article ID 5451439, 12 pages, 2021.

Hindawi
Complexity
Volume 2021, Article ID 7179374, 31 pages
https://doi.org/10.1155/2021/7179374

Review Article

The Applicability of Reinforcement Learning Methods in the Development of Industry 4.0 Applications

Tamás Kegyes,[1] **Zoltán Süle**,[2] **and János Abonyi**[1]

[1]*MTA-PE "Lendület" Complex Systems Monitoring Research Group, University of Pannonia, Veszprém, Hungary*
[2]*University of Pannonia, Veszprém, Hungary*

Correspondence should be addressed to János Abonyi; janos@abonyilab.com

Received 9 September 2021; Accepted 25 October 2021; Published 30 November 2021

Academic Editor: Murari Andrea

Reinforcement learning (RL) methods can successfully solve complex optimization problems. Our article gives a systematic overview of major types of RL methods, their applications at the field of Industry 4.0 solutions, and it provides methodological guidelines to determine the right approach that can be fitted better to the different problems, and moreover, it can be a point of reference for R&D projects and further researches.

1. Introduction

Reinforcement learning (RL) has a significant chance to revolutionize the artificial intelligence (AI) applications by serving a novel approach of machine learning (ML) developments that lets the user to handle large-scale problems efficiently. These techniques together with widespread Internet of things tools have opened up new possibilities for optimizing complex systems, including domains of logistics, project planning, scheduling, and further industry-related domains. Extracting this potential can result in a fundamental progress of Industry 4.0 transformation [1]. During this digital transformation, the vertical and horizontal integration will be strengthened, the flexibility should be raised, and the human control and supervision need to be focused [2, 3]. Furthermore, the data produced by the integrated tools are increasing exponentially that requires a higher level of autonomous processes and decisions. Reinforcement learning can serve as a valuable tool in the development of self-optimising and organising Industry 4.0 solutions. The main challenge of developing these applications is that there are several methods and techniques and a wide range of parameters that need to be defined. As the definition of these parameters requires detailed knowledge of the nature of the RL algorithms, the main goal of this paper is to provide a comprehensive overview of RL methods from the viewpoint of Industry 4.0 and smart manufacturing.

On the basis of our best knowledge, there exists no similar overview article of reinforcement learning methods in Industry 4.0 applications. Next to the fundamental book [4], there are several overviews of reinforcement learning methods from theoretical point of view. A detailed semantic overview of Industry 4.0 frameworks [5] and a categorization of Industry 4.0 research fields are also described. An overview of key elements of Industry 4.0 researches and several application scenarios [6] highlighted the wide scope of smart manufacturing. Although many authors found that there is a lack of extensive review of Industry 4.0 revolution from different aspects, according to their persistent work nowadays, several articles are available in this topic [7]. A survey on the applications of optimal control to scheduling in production, supply chain, and Industry 4.0 systems [8] focused on maximum principle-based studies. Most of the surveys and review articles of Industry 4.0 declare the importance of optimization, but mostly only general approaches are discussed, and there are no detailed guidelines extracted. A comprehensive survey at field of Industry 4.0 and optimization [9] discussed the recent developments in data fusion and machine learning for industrial prognosis, placing an emphasis on the identification of research trends, niches of opportunity, and unexplored challenges. Even if it

considered several ML methods and algorithms, RL was mentioned only shortly without extracting its key fundamentals.

The above collected facts strengthened our motivation of preparing a detailed overview of RL applications and methods used in the field of Industry 4.0. Our main goals with this are:

Presenting a hands-on reference for researchers who are interested in RL applications

Giving compact descriptions of applicable RL methods

Serving a guideline to support them easily identify the best fitting subset of RL methods to their problems and hence letting them focus on the relevant part of the literature

Our systematic review is based on an examination of the literature available from Scopus by following the PRISMA-P (Preferred Reporting Items for Systematic Reviews and Meta-Analysis Protocols). The PRISMA-P workflow contains a 17-item checklist that supports to facilitate the preparation and reporting of a robust protocol in a standardized way for systematic reviews. The literature source list was queried in February 2021 with the following keywords: TITLE-ABS-KEY ("reinforcement learning" AND ("smart factory" OR "IOT" OR "smart manufacturing" OR "industry 4.0" OR "CPS")).

Both author keywords and index keywords were involved into the analysis. The keyword processing started with an extensive data cleansing process by:

Building up a standardized keyword unit (SKU) list and splitting complex keywords into SKUs

Assigning SKUs to one of the following keyword classification types:

(i) Principle captured
(ii) Industrial field of application
(iii) Application field of solution
(iv) Mathematical approach of application methodology

Identifying major classification groups by classification types

781 articles were involved into the analysis. Out of 14,035 original author and index keywords, 2,579 duplications were filtered out. The remaining 11,456 keywords were sliced into 45,824 SKUs. Finally, 12,017 keywords were assigned to classification types that provide the major tendencies and relations of industrial applications of reinforcement learning methods. Figure 1 shows the change of the assessed literature size over the PRISMA steps.

Our article stands for the following major parts:

First, in Section 2, we will give a short general introduction of reinforcement learning framework and summarize some major mathematical properties behind RL techniques. Furthermore, we will present a classification of RL methods that lets the reader to have a map for the further discussions.

As a next step in Sections 3.1–3.3, we will present the key findings of systematic review and a hands-on reference for further researches.

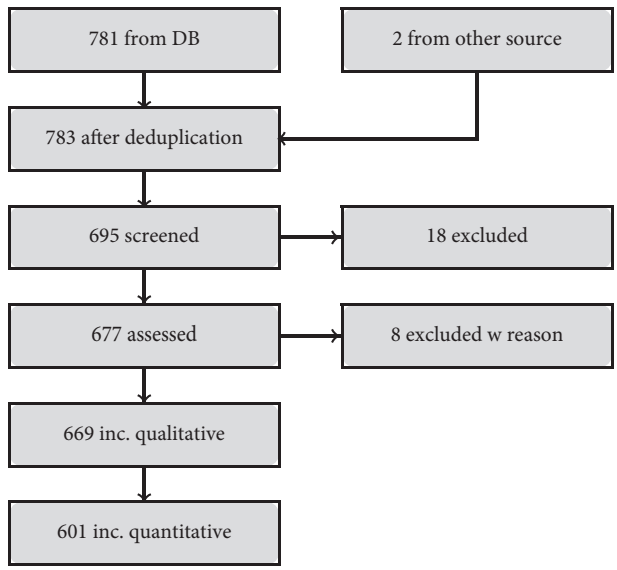


Figure 1: PRISMA processing flow.

Then, in Section 3.4 and in Section 3.5, we will discuss the conclusions and give a detailed guideline to help the reader to choose to most adequate RL method for the different problems.

Finally, in Appendices A–H, we will provide a compact overview of 18 different RL methods.

2. Theoretical Background of Reinforcement Learning

In this section, we will summarize the fundamental concept of reinforcement learning, then we will present a general classification of RL methods.

There are three main paradigms in machine learning: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, a functional relationship of a regression model of a classifier is learnt based on data that represent the input and output of the model. In unsupervised learning, the hidden structure of the data is explored, usually by clustering [9].

Reinforcement learning (RL) also refers to learning problems. As Figure 2 represents the process, an agent takes observations of the environment; then on the basis of that, it executes an action (A_t). As a result of the action in the environment, the agent will get a reward (R_t) and it can take a new observation (O_t) from the environment and the cycle is repeated. The problem is to let agent learning so as to maximize the total reward. Reinforcement learning concept was introduced in ([4], Section 3.1). While in supervised and unsupervised learning, the model fitting requires a complete set of observations; in reinforcement learning, the learning process is sequential.Reinforcement learning is based on the reward hypothesis which states that all goals can be described by the maximisation of expected cumulative rewards. Formally, the history is the sequence of observations, actions, and rewards: $H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$.

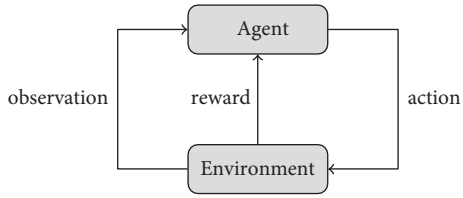


Figure 2: Reinforcement learning framework.

A state contains all the information to determine what happens next. Formally, state is a function of the history: $S_t = f(H_t)$. Let G_t denote the total discounted reward from time-step t: $G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$.

The state-value function $v(s)$ gives the expected total discounted return if starting from state s: $v(s) = \mathbb{E}[G_t \mid S_t t = ns]$. Policy covers the agent's behaviour in all possible cases, so it is essentially a map from states to actions. There are two major categories in it: (1) deterministic policy: $a = \pi(s)$, (2) stochastic policy: $\pi(a \mid s) = \mathbb{P}[A_t = a \mid S_t = s]$. The action-value function $q_\pi(s, a)$ is the expected return starting from state s, taking action a, and then following policy π: $q_\pi(s; a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$.

Practically, state-value function is a prediction of expected present values (PV) of future rewards that allows evaluating the goodness of states, so it is a map from states to scalars: $v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s]$. The optimal state-value function $v^*(s)$ is the maximum state-value function overall policies: $v^*(s) = \max_\pi v_\pi(s)$. It is easy to find that in case if an optimal state-value function is known that an optimal action-value function and an optimal policy can be derived.

Reinforcement learning concept is based on stochastic processes and on Markov chains. Markov property is fundamental of mathematical basis of reinforcement learning methods. A state is Markov if and only if the $\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, \ldots, S_t]$ condition holds. By definition, a Markov decision process (MDP) is a tuple of $\langle \mathcal{S}; \mathcal{A}; \mathrm{P}; \mathcal{R}; \gamma \rangle$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, P is a state transition probability matrix, $\mathrm{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$, $\mathcal{R}$ is a reward function, $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$, γ is a discount factor, $\gamma \in [0; 1]$, and t time-steps are discrete. The Bellman equation practically states that state-value function of an MDP can be decomposed into two parts: immediate reward and discounted value of successors states: $v(s = S_t) = R_{t+1} + \gamma v(S_{t+1})$.

Environments can be distinguished by its observability. Let us denote S^a_t as the agent's state at time-step t and S^e_t as the environment's state. Environment can be (1) fully observable if the agent directly observes all states of environment $(O_t = S^a_t = S^e_t)$, or partially observable if the agent has indirect observations $(S^a_t = (\mathbb{P}[S^e_t = s_1], \ldots, \mathbb{P}[S^e_t = s_n]))$.

Figure 3 summarizes a classification of reinforcement learning methods in tree structure. Further details of the different RL methods are described in Appendix.

3. Overview of the Industry 4.0 Relevant Applications

In this section, we will present the hands-on references in tabular format based on the results of our data cleansing process and some major results of systematic literature analysis that will highlight some general trends which is able to lead the reader to a successfully applicable RL methods by preventing the usage of inappropriate trials and hence shortening development periods. In the final part of the section, we will present a hands-on guideline to summarize the key conclusions.

3.1. Classification of Applications by Principle Captured. The main goal of this section is to give an overview what are the principal captured problem types that reinforcement learning was applied for and describe the major tools that gave an impressive performance for each and every problem category and finally to highlight some typical issues that needed to be taken care of during the implementation.

By performing SKU analysis, we identified the most relevant keywords that are assigned to a principle captured. In Table 1, the associated publications are listed by principle captured categories.

Furthermore, Figure 4 shows the principle captured classes by reinforcement learning methods. Although the related frequency table does not meet all the required criteria, in Table 2, a χ^2-test, calculation is presented, yet by principle captured classes, it makes the identification of some significant deviations from the overall distribution of RL methods possible.

In the class of prediction, forecasting, and estimation, planning value function approximation methods and Markov decision processes are over-represented. This lets us to conclude that the complex methods in the focus are less, which is fully in line with the goal to understand better the behaviour of the environment without strong optimization aims.

In the class of detection, recognition, prevention, avoidance, and protection, the policy gradient methods are over-represented, while MDPs are under-represented. This shows us that researchers are interested more in complex models with a higher predictive performance than in basic solutions.

In the classes of evaluation, assessment and allocation, assignment, and resource management, the multiagent methods are more in focus, which tells us that this field is on the way to distribute the tasks to lower level tools instead of centralized data processes. But while in the first class, the distribution of further RL methods follows the overall distribution, and in the second class, the policy gradient methods are over-represented which comes from the fact that allocation-related problems prefer to create an optimal policy.

In the classes of classification, clustering and decision making and scheduling, queuing, and planning, the situation is opposite: multiagent methods are under-represented, which means that researches of these kinds of operations are still focusing to a centralized solution.

In the class of control, the temporal-difference methods and Markov decision process contractions and multi-agent methods are over-represented, while complex approaches, like policy gradient methods, are under-represented.

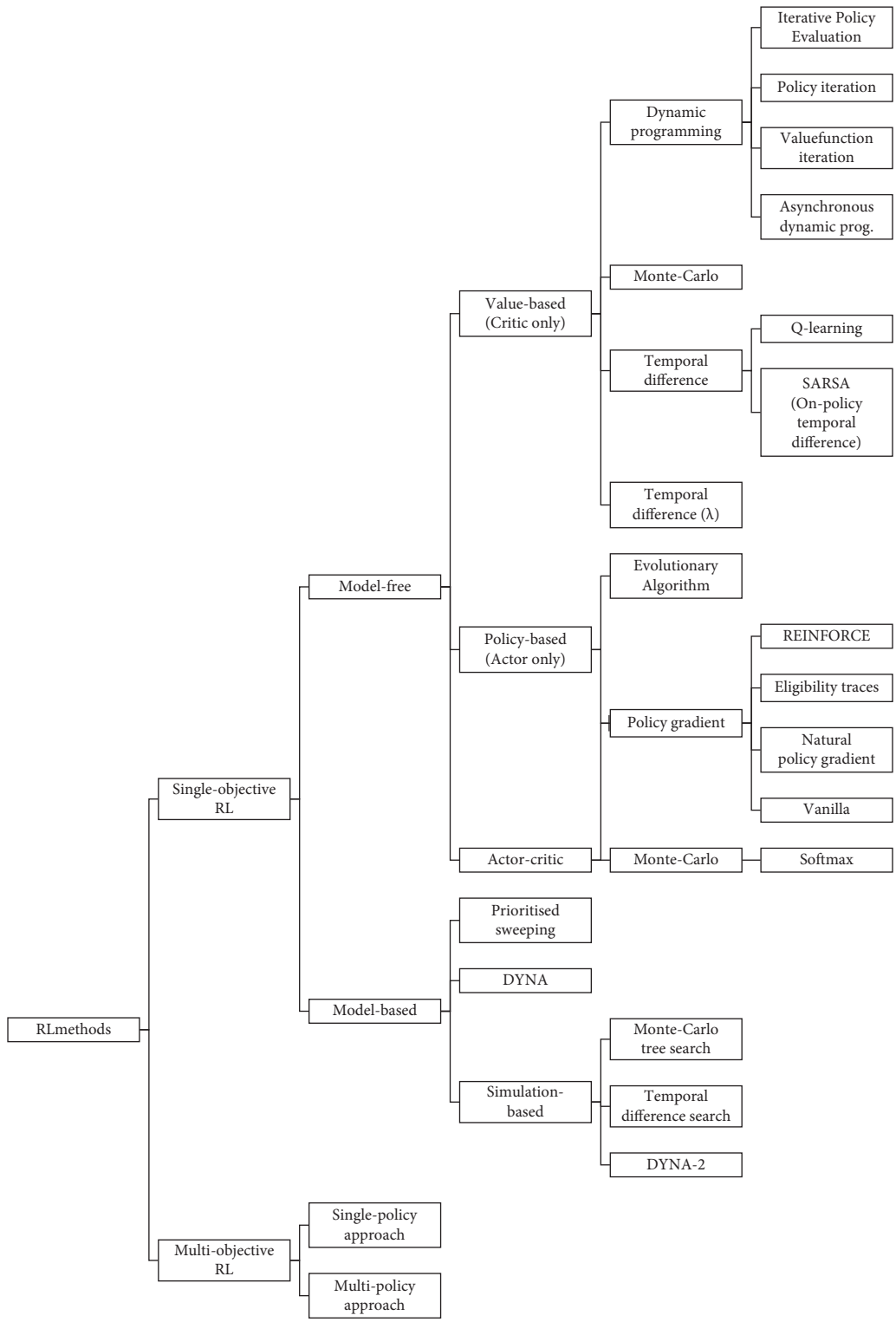


Figure 3: Classification tree of reinforcement learning methods.

Discussions of specific parts of RL solution design problems occur in smaller number of cases, but these kinds of publication demonstrate the fact that constructing an appropriate RL application is not always trivial. We can highlight state space design [12, 25, 33, 107, 144, 179, 193, 208, 217, 220, 222, 224, 227, 266, 267] and action space design [109, 220, 246, 268], reward construction [14, 76, 110, 199, 220, 226, 246, 269–273], and exploration strategy planning [86, 274] which can be determinants from the whole application point of view.

Table 1: Publication reference by principle captured.

Principle captured	Referred publications
Prediction, forecasting, estimation, planning	[10–39]
Detection, recognition, prevention, avoidance, protection	[10, 29, 39–71]
Evaluation, assessment	[18, 32, 54, 66, 72–81]
Classification, clustering	[35, 42, 66, 69, 69, 81–95]
Decision making	[11, 13, 17, 20, 21, 24, 38, 43, 61, 62, 66, 69, 82, 89, 93], [96–131]
Allocation, assignment, resource management	[20, 22, 31, 32, 39, 45, 60, 65, 67, 70, 75, 78, 83, 87, 91, 96, 97, 99, 100], [103, 104, 113, 119, 121, 121, 125, 127, 130], [130, 131, 131–154], [154–156, 156–178], [179–196], [196–202]
Scheduling, queuing, planning	[12, 19, 21, 24, 32, 72, 87, 88, 91, 93, 96, 99, 110, 113, 122, 125, 131, 150, 151, 160, 184, 188, 203–222]
Control	[12, 14, 15, 18, 23, 27, 31, 36, 37, 40, 56, 69, 70, 91, 93, 99, 101], [105–107, 111, 112, 122, 123, 129, 130], [149, 150, 167–169, 171, 180], [188, 189, 199, 200, 205, 217], [223–245], [245–265]

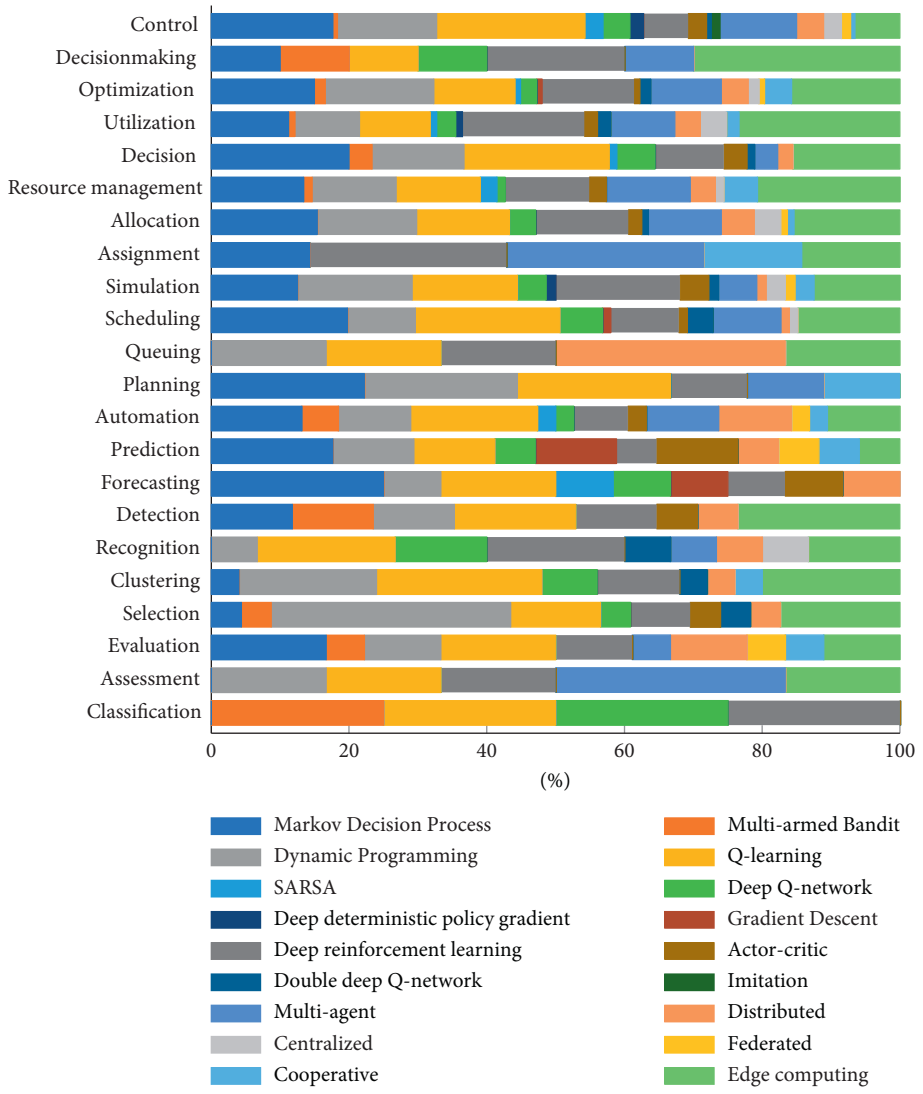


Figure 4: Distribution of RL methods by principal captured classes.

Table 2: χ^2-test table of principle captured classes by RL method types.

Principle captured	Markov decision process	Multiarmed bandit	Dynamic	Temporal difference	Value function approximation	Policy gradient	Multiagent	Edge computing
Prediction, forecasting, estimation, planning	1.96	−0.58	0.12	−0.32	3.08	0.31	−0.39	−4.17
Detection, recognition, prevention, avoidance, protection	−3.09	1.51	−1.11	−0.17	0.9988, 0.9782, 0.97850.39	0.9935, 0.8769, 0.87852.2	−1.38	1.65
Evaluation, assessment	−0.82	0.63	−0.08	−0.62	−1.21	−0.6	2.96	−0.27
Classification, clustering	−3.61	0.56	1.28	1.41	1.54	0.65	−2.88	1.05
Decision making	3.1	2.47	−0.84	1.73	0.95	0.02	−10.82	3.39
Allocation, assignment, resource management	−2.69	−1.95	0.21	−11.18	−4.74	2.08	10.54	7.74
Scheduling, queuing, planning	2.17	−1.33	−2.17	1.24	1.61	−0.04	−2.63	1.16
Control	2.67	−1.34	2.35	7.52	−1.72	−4.93	4.27	−10.82

3.2. Classification of Publications by Industrial Field of Application. Similarly, as we have shown in Section 3.1, by performing SKU analysis, we also identified the most relevant keywords that are assigned to industrial fields. In Table 3, the associated publications are listed by industrial field categories.

Similarly, as we presented categories of principal captured, we also prepared Figure 5 that shows the industrial field classes by reinforcement learning methods. Although the related frequency table does not meet all the required criteria, in Table 4, a χ^2-test, calculation is presented, yet by industrial field classes, it makes the identification of some significant deviations from the overall distribution of RL methods possible.

In the class of energy, solar, power, electric, the applications of Q-learning methods are over-represented, while more basic methods and policy gradient methods are under-represented

In the class of telecommunication, communication, networking, internet, 5G, Wi-Fi, and mobile, the policy gradient methods are over-represented and there is a strong focus on the applications of edge computing

In the class of wireless, radio, antenna, and signal, the applications of Markov decision processes are highlighted

Similarly, in the class of vehicle, unmanned aerial vehicle, drone, and aircraft, the applications of Markov decision process are over-represented together with policy gradient methods, while the multiagent solutions are less discussed

In the classes of cyber-physical system, robot and manufacturing, and factory, the basic dynamic methods and Q-learning approaches are more popular

Finally, in the class of city and building, the multiagent methods are over-represented

3.3. Classification of Publications by Mathematical Approach of Application Methodology. Similarly, as we have shown in the previous sections, we also performed the SKU analysis for the third major dimension of keywords which is the methodological approach of the solution. The most relevant keywords were identified, and then in Table 5. the associated publications are listed by methodological approach categories.

Although it is not feasible to summarize all the different methodological approaches in details, we would like to highlight some specialities of selected cases to demonstrate how widely RL approaches are used and motivate researchers to find a solution for their problems from a new perspective.

As we described in Section 2, reinforcement learning methods are based on Markov property and hence it is fundamental to model the problems as Markov decision processes (MDPs), which is far not trivial in several cases. By formulating an MDP, we need to take care about state space design, especially guaranteeing that a state representation contains all the relevant information to evaluate a situation, or with other words anytime, when the system is in the same particular action, the environment will take its response by the same characteristic for a particular action [96, 104, 191, 203, 313, 346].

Actor-critic methods are model-free learning methods that learn both the optimal policy for taking an action and the value function for most accurate evaluating of the current state. Most of the publications discuss mainly distributed autonomous IoT device networks. In these cases, the focus is shifted towards the learning and knowledge transfer solutions:

Stochastic model of cloud-based IoT for fog computing computation offload and radio resource allocation [97].

Centralized joint resource allocation solution for handling shortage of frequency resources of cellular

Table 3: Publication reference by industrial field of application.

Industrial field	Referred publications
Energy, solar, power, electric	[14, 17, 18, 23, 31, 32, 37, 45, 51, 53, 70, 72, 75, 75, 80, 81, 83, 85, 88, 100], [101, 107, 108, 118, 123, 133, 134, 136, 144, 145, 154, 159, 161, 165, 167, 168, 171], [174–177, 179, 183, 188, 192, 198, 200], [204, 208, 210, 213, 216, 221, 222, 225–227], [230–232, 234, 235, 240, 243, 246, 248, 252, 255, 257, 259], [264–266], [275–310]
Telecommunication, communication, networking, internet, 5G, Wi-Fi, mobile	[12, 17, 22, 28, 33, 35, 38, 39, 41, 45, 50, 50, 51, 65, 70, 72, 75, 75, 82, 83, 87, 88, 90], [91, 95, 100], [102, 104, 106, 109, 112, 120, 121, 133, 134, 136, 137, 141–143, 145], [146, 151, 157, 159, 160, 165–171, 174, 175], [177–179, 181, 187, 188, 190, 196, 197, 200, 201, 203, 206, 207, 211, 214, 218, 223], [234, 236, 239, 243, 245, 248, 252, 254, 258, 275, 277–279], [284, 286–292, 294, 297, 300], [305, 306, 309–343]
Wireless, radio, antenna, signal	[14, 18, 28, 32, 33, 38, 41, 45, 46, 51, 64, 68, 70, 72, 82, 83, 85, 86, 91, 94, 96, 100], [102, 104, 106–109, 128, 131, 135, 136, 140, 141, 145, 151, 158, 160], [164, 166, 168, 170, 173, 178, 179, 188, 191, 198, 200, 201, 203, 206, 207, 209, 215–218, 223, 230], [234–236, 239, 243–245, 248, 248], [251, 252, 254, 255, 257, 272, 275, 277, 279, 284, 286, 288, 293, 299, 300, 302, 304, 305, 307–309], [311, 312, 315–318, 321, 323, 328–336, 340, 341, 343–351]
Vehicle, unmanned aerial vehicle, drone, aircraft	[10, 68, 79, 90, 91, 151, 161, 170, 181, 183, 197, 217, 220, 223, 228, 229, 232, 244, 257, 259, 266], [272, 277, 282, 299, 305, 309–311, 313, 320, 325, 336, 341, 352–354]
Cyber-physical system, robot	[15, 21, 43, 50, 56, 66, 69, 73, 74, 107, 109, 112, 122, 131, 149, 188, 189, 209, 224, 225, 229, 233], [238, 241, 242, 247–249, 258, 264, 266, 268, 282, 288, 305, 350, 354–359]
Manufacturing, factory	[19, 36, 69, 79, 93, 99, 123, 128, 149, 205, 209, 228, 247, 279, 347, 359–362]
City, building	[18, 31, 45, 66, 140, 174, 201, 204, 213, 235, 244, 282, 292, 332, 340, 355, 363–368]

systems by using a neural network embedded reinforcement learning algorithm [176].

Determining optimal sampling time for IoT devices for energy harvesting by saving batteries. Hence state space contains continuous quantities, a linear function approximation was used and a set of novel features were introduced to represent the large state space [349].

A bio-inspired RL modular architecture is able to perform skill-to-skill knowledge transfer and called transfer expert RL (TERL) model. Its architecture is based on a RL actor-critic model where both the actor and critic have a hierarchical structure, inspired by the mixture-of-experts model [392].

Deep reinforcement learning-based cooperative edge caching approach [338].

Multiple IoT devices are sending data parallel, but in general, they do not provide additional information to the existing knowledge. So, it is not necessary to permanently send data. By using actor-critic method, it can be determined which data packages need to be sent to prevent redundant or irrelevant communication [221].

Mobile edge computing and energy harvesting framework of centralized training with decentralized execution by adopting MD-hybrid-AC method [120].

Asynchronous advantage actor-critic method for mobile edge computing because computation offloading cannot have good performance in many situations, but the optimal algorithm can be chosen to use on IoT side [196].

Optimization of the robustness of IoT network topology with a scale-free network model which has good performance in random attacks. A deep deterministic learning policy (DDLP) is proposed to improve the stability for large-scale IoT applications [337].

IoT devices have lack of storage capacity, therefore a jointly cache content placement and delivery policy for the cache-enabled D2D networks was constructed. [17].

A federated reinforcement learning architecture was presented where each agent working on its independent IoT device shares its learning experience (i.e., the gradient of loss function) with each other [237].

By applying multiagent methods, there are multiple ways to organize learning:

Local learning and no centralized knowledge (see Figure 6(a))

Local knowledge deployment, local learning, and central knowledge collection

Local knowledge deployment and local learning with knowledge transfer to close neighborhoods (see Figure 6(b))

Local knowledge deployment and centralized learning (see Figure 6(c))

3.3.1. Centralized and Federated Methods. As Internet of things (IoT) services and applications are growing rapidly, most of the current optimization-based methods lack a self-adaptive ability in dynamic environments. To handle these challenges, learning-based approaches are implemented generally in a centralized way. However, network resources may be over-consumed during the training and data transmission process. To solve the complex and dynamic control issues, a federated deep reinforcement learning-based

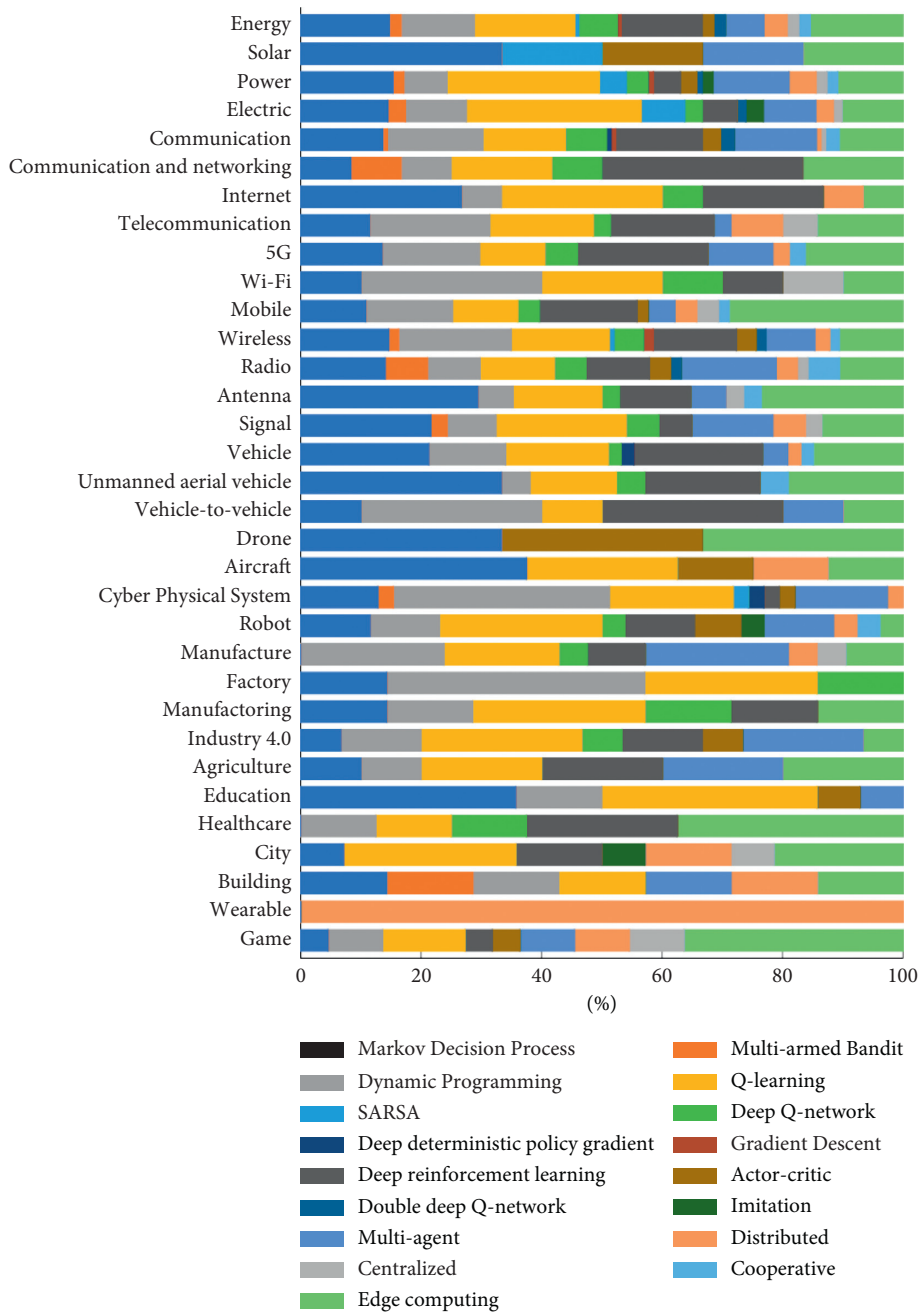


Figure 5: Distribution of RL methods by industrial field classes.

cooperative edge caching (FADE) framework is presented. FADE enables base stations (BSs) to cooperatively learn a shared predictive model by considering the first-round training parameters of the BSs as the initial input of the local training and then uploads near-optimal local parameters to the BSs to participate in the next round of global training [16].

Although the first researches have focused on designing learning algorithms with provable convergence time, but other issues, such as incentive mechanism, were explored later: a deep reinforcement learning-based incentive mechanism has been designed to determine the optimal pricing strategy for the parameter server and the optimal training strategies for edge nodes [147].

3.3.2. Hierarchical Methods. Hierarchical approaches are applied primarily to solve communication channel or information processing capacity issues. The model structure usually follows the structure of the information path. In a two-layer approach, a local IoT device needs to transfer information to a local hub and then the local hub transmits the collected information to the central decision maker. In this case, separated models can be set up for both layers to find optimal scheduling order for communication.

A new crowd sensing framework is introduced based on hierarchical structure to organize different resources and it is solved by using deep reinforcement learning-based strategy to ensure quality of service [88]. A hierarchical

Table 4: χ^2-test table of industrial field classes by RL method types.

Principle captured	Markov decision process	Multiarmed bandit	Dynamic	Temporal difference	Value function approximation	Policy gradient	Multiagent	Edge computing
Energy, solar, power, electric	0.52	1.7	−12.77	21.87	0.94	−11.54	2.05	−2.77
Communication, networking, internet, 5G, Wi-Fi, mobile	−8.61	−3.51	6.29	−18.79	2.23	13.24	−3.14	12.29
Wireless, radio, antenna, signal	5.77	3.07	−1.73	−6.62	1.33	−1.76	2.68	−2.73
Vehicle, unmanned aerial vehicle, drone, aircraft	8.44	−1.39	−2.32	−2.89	−2.49	5.9	−6.94	1.68
Cyber-physical system, robot	−1.75	0	8.14	3.86	−2.23	−2.14	1.97	−7.86
Manufacturing, factory	−3.33	−0.55	4.16	1.36	1.23	−2.54	1.52	−1.84
City, building	−1.05	0.69	−1.77	1.21	−1.01	−1.17	1.87	1.23

Table 5: Publication reference by methodological approaches.

Approach	Referred publications
Markov decision process	[12, 23, 24, 37, 64, 70, 75, 84, 96, 100, 101, 104, 127, 130, 133, 138, 144, 153, 165, 167, 170, 177, 188, 191, 199], [203, 207, 211, 212, 214, 217, 220, 231, 252, 256–259, 263, 264, 272, 274, 281, 291, 309, 313, 320, 340, 343, 346], [369–376]
Multiarmed bandit	[61, 66, 102, 198, 351, 377, 378]
Dynamic programming	[16, 19, 27, 52, 68, 70, 84, 90, 93, 107, 119, 120, 132, 135, 141, 145, 155, 156, 161, 162, 189, 191, 198, 201, 207], [209, 212, 222, 236, 242, 247, 254, 258, 259, 278, 280, 288, 289, 304, 306, 313, 321, 331, 340, 347, 357, 371, 372, 379, 380]
Q-learning	[10, 17, 24, 44, 47, 50, 64, 68, 70, 80, 81, 83, 91, 92, 94, 101, 110, 116, 124, 125, 127, 129, 133, 145, 152, 172, 179], [180, 183, 187, 201, 203, 205, 206, 208, 210, 212, 215, 219, 222–225, 227, 231, 242, 244, 246, 248, 250, 254, 262], [264, 280, 282, 283, 291, 294–296, 321, 326, 327, 343, 347, 353, 356, 366, 367, 372, 374, 381–386]
SARSA	[14, 127, 240, 246, 280, 384]
Deep Q-network	[17, 47, 83, 99, 125, 133, 190, 210, 254, 291, 294, 347, 387]
Deep deterministic policy gradient	[229, 260, 338]
Gradient descent	[26, 28, 216, 388]
Deep reinforcement learning	[19, 20, 32, 41, 47, 50, 60, 75, 77, 84, 88, 90, 95, 98, 100, 103, 117, 131, 134, 147, 154, 159, 165, 168, 176, 179, 182], [193, 199, 201, 207, 210, 220, 221, 223, 236, 241, 260, 261, 273, 275, 281, 294, 299, 301, 302, 305, 309, 311, 317, 323, 333, 338, 341, 346, 352, 355, 361, 363, 375, 380, 389–391]
Actor-critic	[15, 17, 33, 97, 120, 176, 196, 221, 237, 337, 338, 349, 392]
Double deep Q-network	[47, 83, 125, 210, 254, 294, 387, 393]
Imitation	[226, 265, 355]
Multiagent	[32, 60, 70, 77, 103, 145, 163, 168, 173, 175, 176, 188, 195, 200, 209, 218, 219, 225, 231], [245, 251, 263, 267, 280, 289, 323, 330, 338, 344, 361, 367, 377, 394, 395]
Distributed	[45, 56, 60, 73, 91, 119, 133, 145, 187, 261, 282, 348, 394]
Centralized	[60, 147, 187, 243, 296]
Cooperative	[16, 81, 170, 200, 338, 344]
Collaborative	[45, 137, 174, 196, 237, 248, 325, 381, 396]

correlated Q-learning (HCEQ) approach is presented to solve the dynamic optimization of generation command dispatch (GCD) for automatic generation control (AGC) [231]. An enhanced version of a bio-inspired reinforcement learning modular architecture is presented to perform skill-to-skill knowledge transfer and called transfer expert RL (TERL) model. TERL architecture is based on a RL actor-critic model where both the actor and critic have a hierarchical structure, inspired by the mixture-of-experts model, formed by a gating network that selects experts specializing in learning the policies or value functions of different tasks [392]. A new cloud computing model is proposed that is hierarchically composed of two layers: a cloud control layer (CCL) and a user control layer (UCL). The CCL manages cloud resource allocation, service scheduling, service profile, and service adaptation policy from a system performance point of view. Meanwhile, the UCL manages end-to-end service connection and service context from a user performance point of view. The proposed model can support nonuniform service binding and its real-time adaptation using metaobjects by intelligent service-context management using a supervised and reinforcement learning-based machine learning framework [150]. A new cooperative resource allocation algorithm is

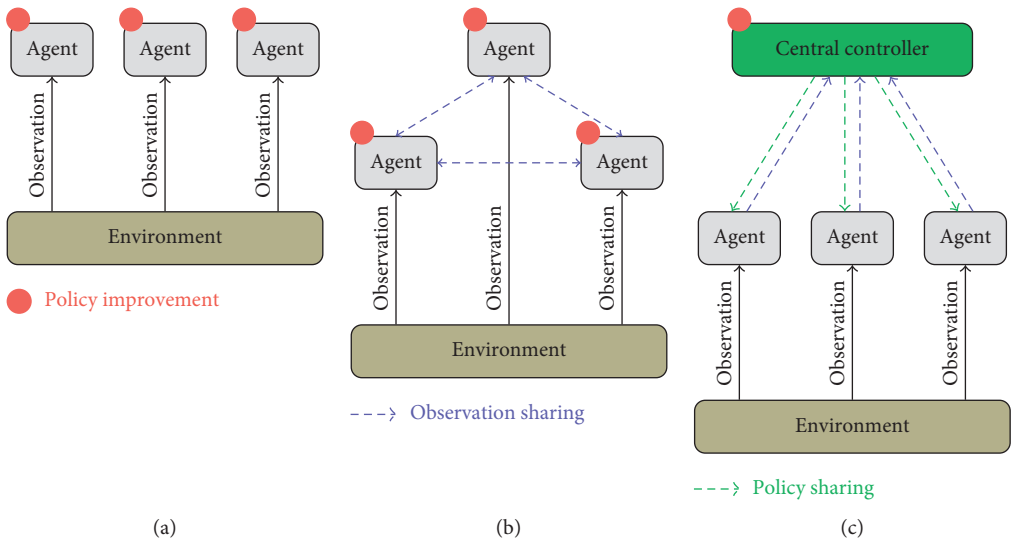


Figure 6: Cooperation concepts of multiagent learning systems. (a) Full local learning. (b) Knowledge sharing. (c) Centralized learning.

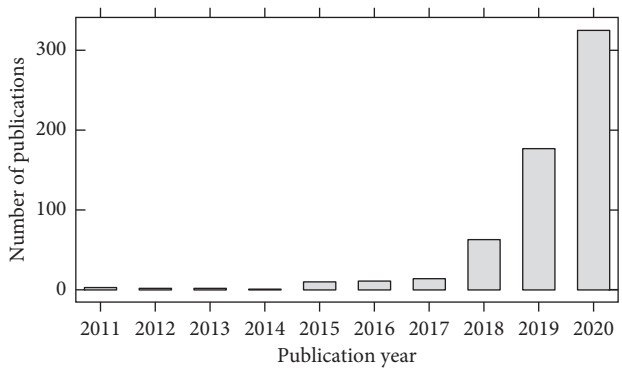


Figure 7: Distribution of publications processed in our analysis by publication years.

presented which couples reinforcement learning networks and prediction neural networks for accurate mobile targets tracking. Specifically, a hierarchical structure that performs collaborative computing is designed for alleviating computing pressure of front-end devices which are supported by edge servers [397]. A slightly different approach is applied at a resilient control problem studied for cyber-physical systems (CPSs) under the denial-of-service (DoS) attack. The term resilience is interpreted as the ability to be robust to the physical layer external disturbance and defending against cyber layer DoS attacks. The overall resilient control system is described by a hierarchical game, where the cyber security issue is modeled as a zero-sum matrix game, and physical minimax control problem is described by a zero-sum dynamic game. In virtue of the reinforcement learning method, the defense/attack policy in the cyber layer can be obtained, and additionally, the physical layer control strategy can be obtained by using the dynamical programming method [398]. Further publications in hierarchical RL topics are related to balancing timeliness and criticality when gathering data from multiple sources [116], ubiquitous user connectivity, and collaborative computation offloading for smart cities [248].

3.3.3. Distributed and Parallel Methods. It can be stated with certainty that the biggest potential of industrial applications is in intelligent devices. In this context, intelligence means some kind ability for taking autonomously decisions and furthermore being able to perform learning steps locally. There were made significant efforts to develop functional solution to reach this goal.

Computation offloading can provide a solution for the issue of the high computation requirement of resource-constrained mobile devices. The mobile cloud is the well-known existing offloading platform, which is usually far-end network solution, but this can cause other issues, such as higher latency or network delay, which negatively affects the real-time mobile Internet of things (IoT) applications. Therefore, a deep Q-learning-based autonomic management framework is proposed as a near-end network solution of computation offloading in mobile edge [133].

Another way to extend single reinforcement learning applications is to handle multiple objectives. There are two major solution practices to handle such kind of problems. The most obvious idea is to construct a mixed reward function that returns a combined result according to the different objectives [161, 259, 370]. Another possible way is to combine multiobjective ant colony optimization methods

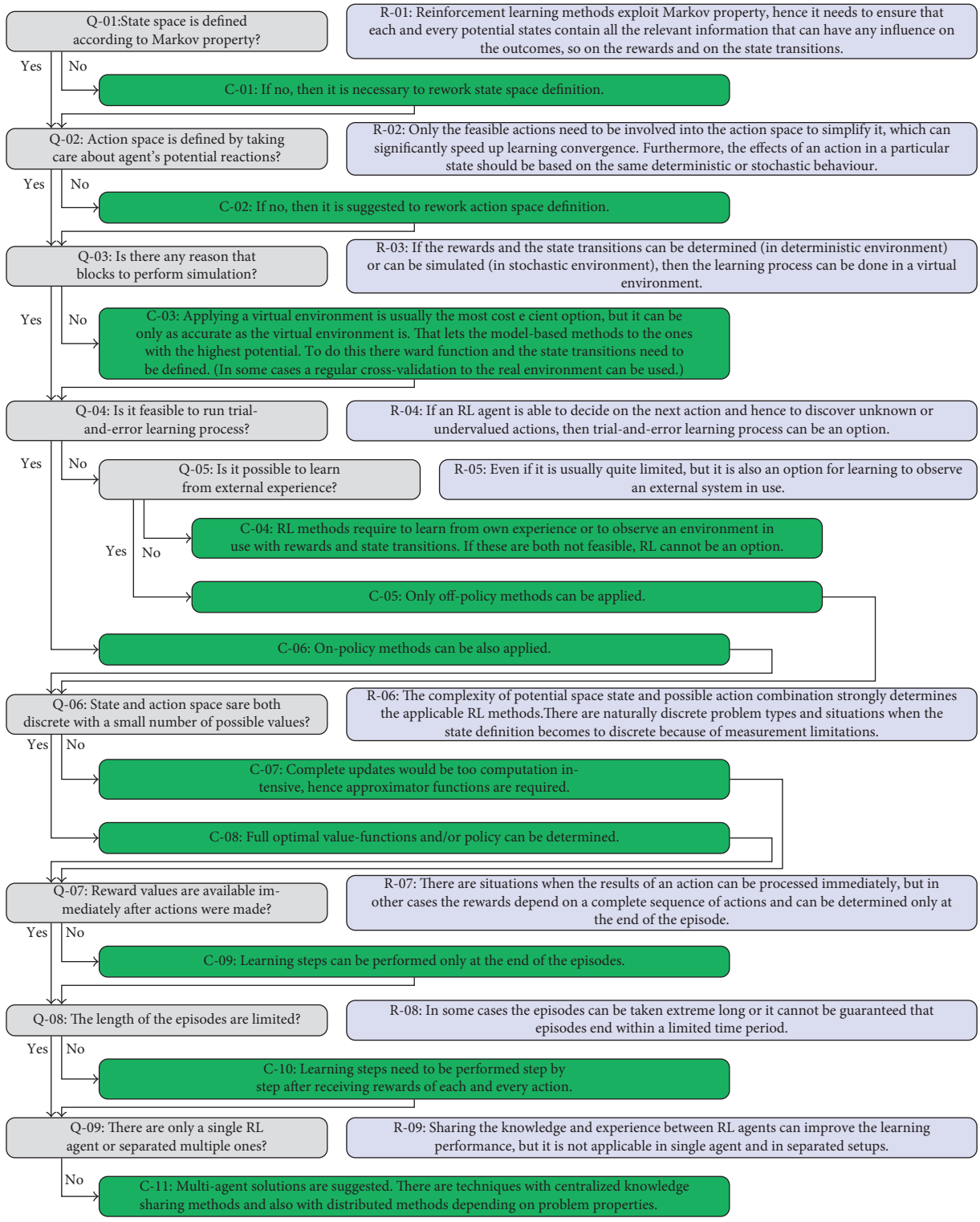


Figure 8: Guideline process to determine appropriate RL method to use.

with RL techniques like deep reinforcement learning or double Q-learning algorithms [83, 142].

3.4. General Trends of RL Applications. Before the beginning of Industry 4.0 revolution, the general methodology was based on centralized data collection, data processing, and predictive model development solutions. By spreading Internet of things (IoT) devices, it turns possible to delegate more computational task to them. This kind of potential gets being exploited by reacting to another major issue which is the lack of communication capability. On the one hand, the

communication between IoT devices and central servers or nodes are relative energy intensive processes; on the other hand, there are significant limitations on communication channels or frequencies.

By distributing computational tasks to IoT devices, a fundamental change gets required: it is not possible to assign as much human effort to data processing and predictive model development supervision as before during the centralized era. This was the major reason of appreciating RL methods because it provides a general self-learning framework that basically requires no manual or human interactions to maintain.

The early researches focused on the applicability of reinforcement learning techniques with single agents. Then, more and more complex problems were solved, and the multiagent solutions started to analyze. In the last years, the focus of the researchers is shifting to multiagent structures. The set-up of the agents and their goals or reward functions are showing very creative solutions. At a new wave of researches, the agents are defined with different roles often with attacker-defender objectives and let each of the agent to be trained an optimal strategy according to it. Then, the stability and robustness of the system can be analyzed and the weakest items can be purposefully improved.

As Figure 7 demonstrates, the number of Industry 4.0-related reinforcement learning-based researches dynamically increases, and there is no sign for expecting a slowing in it.

3.5. Discussion and Guideline Process to Determine Appropriate RL Method to Use. On the basis of the previous section, it can be highlighted that there are several ways and methods how reinforcement learning can be applied for Industry 4.0-related problems, and it is far not trivial which one can provide a successful solution.

We prepared a questionnaire and we presented it in a decision flow diagram in Figure 8. Our primary goal was to set up a method to help the readers in formulating their RL tasks. The first questions of the questionary-based process verify whether state and action spaces are appropriately defined and how the reward can be obtained. The further questions systematically narrow down the set of applicable RL methods. The possibility of using simulation or learning from own experience can determine the general learning mechanism. In contrast, the nature of reward propagation can determine a smaller subset of the RL methods that can be applicable. Even if the conclusions are soft-defined, a user with some basic knowledge of RL methods can easily interpret them, or it can be a basis of some RL methods selector wizard. We believe that researchers will have fewer failed attempts by using our guideline, and the time-to-solution can be reduced significantly.

We should keep in mind that the whole reinforcement learning concept is based on Markov decision processes. A direct conclusion is that the state space should be constructed in a way that all the potential states should contain all the relevant information that can have any influence on the outcomes. Moreover, the action space should be constructed similarly: the effects of an action in a particular state should be based on the same deterministic or stochastic behaviour. This will let the RL agent to learn the effect mechanism behind.

Once the state and action spaces are defined, it needs to be investigated whether performing simulations is an option or not. If we are able to determine the environment's behaviour when an action is made in a particular state, so deriving the reward value and the state transition, then an extensive learning process can be executed by using model-based RL methods in a cost-efficient way without significant risk of applying untrained agents. The general rule is also true in this case: the RL solution will be as adequate as the simulation is. If there is an option to validate the simulation outcomes to the real environment, then this can help to ensure the validity of the solution.

4. Conclusions

As we pointed out that reinforcement learning methods have a high potential also in Industry 4.0 applications which is a common agreement of researchers, one of the biggest reasons behind is that smart tools require a high level of optimizations which cannot be satisfied with human interventions. This continuously raises the demand of self-learning solutions, and RL techniques have been proven their efficiency at multiple fields. A major goal of our article was to give an overview of RL applications at the field of Industry 4.0. As a first step, we served a high-level overview of the general RL framework and a classification of RL methods to easily see through the possibilities, while we also presented a more detailed summary of the most widely used RL methods of Industry 4.0 applications in Appendix. Therefore, our publication can serve a starting point of further researches for RL applications.

Then, we highlighted the results of our systematic literature overview of reinforcement learning applications at the field of Industry 4.0. An extensive keyword analysis drove us to identify some typical patterns by choosing an adequate RL method for some particular combinations of principal captures and industrial fields. Although there are no unique optimal RL methods, there are RL methods that provide efficient solution for some problems. Our summary can be used as a hands-on-reference for further researches and it can help researchers to shorten the preparation time for their researches.

Furthermore, we prepared a questionnaire that provides a methodology to set up the reinforcement learning system in a proper way and to choose an appropriate method for the learning problem that the researcher is facing to. We believe that an extension of our questionnaire can be a basis of a wizard tool that enables the user to find the most fitting RL method for the learning task and guiding through the set-up processes. On the other hand, by knowing the key properties of the different RL methods, it becomes faster to adopt an existing one or to modify it to fit the specific needs and hence develop an own RL method.

We hope that our article lets the researchers strengthen to decide using RL methods for further applications as numerous successful applications show the high efficiency of them.

Appendix

In Appendix, we will describe one by one the major methods of reinforcement learning by highlighting their properties

and evolutionary stages by following David Silver's approach from the simplest ones to the more complex ones.

A. Dynamic Programming

Dynamic programming (DP) covers a decision process by breaking it down into a sequence of elementary decision steps over time. "Dynamic" refers to the sequential approach, while "programming" refers to its optimization objective.

In this section, all the methods work with the assumption that the environment is perfectly known. Iterative policy evaluation method is described for learning state-value function of a given policy Π, then value iteration method is used to determine optimal state-value function although actions are taken according to any given policy Π, and last but not least, policy iteration is presented to derive an optimal policy to the environment.

In general, there is limited usage of dynamic programming algorithms both because of its assumption to know the environment perfectly and its high computational requirements. On the other hand, dynamic programming methods provide the essence of ideas that are used in advanced methods in an easily understandable form.

Iterative Policy Evaluation. Let us assume that a policy π is given and actions are taken according to it. The goal is to determine state-value function v_π by iterative application of Bellman backup: $v_1 \longrightarrow v_2 \longrightarrow \ldots \longrightarrow v_\pi$. At each and every iteration steps, the state-value function should be updated in the following way:

$$v_{k+1}(s) = \sum_{a \in \mathscr{A}} \pi(a \mid s) \left(\mathscr{R}_s^a + \gamma \sum_{s' \in \mathscr{S}} \mathrm{P}_{ss'}^a v_k(s') \right). \quad \text{(A.1)}$$

The second term shows the cumulative rewards from state s by taking action a and applying a single Bellman decomposition while the first term provides the probability of taking action a by following policy Π. It can be proven that with weak conditions, the proposed state-value function update will converge to $v_\pi(S)$ ([4], Section 4.2).

Value Iteration. Iterative policy evaluation method can be extended to find an optimal state-value function $v^*(s)$. The main idea behind that iteration should be done by starting from the final reward and working backward. Let us assume that the solution of subproblem $v^*(s')$ is known. Then, by the solution of the next iteration step, $v^*(s)$ can be found by one-step look-ahead:

$$v^*(s) \leftarrow \max_{a \in \mathscr{A}} \left(\mathscr{R}_s^a + \gamma \sum_{s' \in \mathscr{S}} \mathrm{P}_{ss'}^a v^*(s') \right). \quad \text{(A.2)}$$

It can easily be seen that for finite state space $\mathscr{S}$, the determination of optimal state-value function for all the available states can be done in finite number of steps ([4], Section 4.4).

Policy Iteration. The iteratively learnt knowledge can be extracted by improving the policy by acting greedily with respect to v_{π^*}. This practically means to pick that action a from a particular state s which maximizes the sum of immediate reward r_s^a and discounted state-value $\gamma v_{\pi^*}(s')$ of the successor state s' ([4], Section 4.6). The learning process of policy iteration is demonstrated on Figure 9.

B. Model-Free Prediction Methods

Unlike in dynamic programming, in model-free methods, perfectly known environment is not necessary, only experience samples are required or with other words just sequences of states, actions, and rewards, no prior knowledge of the environment.

In this section, Monte-Carlo learning method is presented for learning simply by averaging the experience, and then temporal-difference learning method is discussed to let the agent learn by more frequent but smaller steps by applying bootstrapping techniques, while temporal-difference (λ) learning method is described as an extension of temporal-difference method's one-step learning to multiple-steps learning.

Monte-Carlo Learning. Monte-Carlo (MC) agent solves the reinforcement learning problem by applying average sample return, so it learns from complete episodes. Hence, it needs to be guaranteed always to terminate episodes; otherwise, the learning process cannot be performed. MC uses the simplest idea by assigning empirical mean of returns to a specific state ([4], Section 5.1). There are two major types of MC methods:

First-visit MC: only the first visit of a state will be involved into the calculation during an episode. Let us assume that state s is visited first time at time period t. Let us denote G_t as the total return from time period t and $N(s)$ the number of times that state s is visited while $S(s)$ is the sum of G_t returns up to the current episode. In this case, the state-value estimate will be the empirical mean: $V(s) = S(s)/N(s)$. As experience grows, so as $N(s) \longrightarrow \infty$, the long-term mean will converge to the state-value function: $V(s) \longrightarrow v_\pi(s)$.

Every-visit MC: all the visits of a state will be involved into the calculation during an episode. Formally, the main difference to first-visit MC is that $N(s)$ needs to be incremented at every time period t whenever state s is visited.

From computational point of view, it is important to mention that empirical mean is determined incrementally in practice. Let us denote $V^{(n)}(s)$ as the value-function estimate while $S^{(n)}(s)$ is the cumulative sum of returns after episode n, then $G_t^{(n)}$ is the total return in episode n from time period t when state s is visited and assume that state s is visited kth times overall.

$$\begin{aligned} V^{(n)}(s) &= \frac{1}{k} S^{(n)}(s) = \frac{1}{k} \sum_{i=1}^{n} S^{(i)}(s) = \frac{1}{k} \sum_{i=1}^{n-1} S^{(i)}(s) + \frac{1}{k} G_t^{(n)} \\ &== \frac{1}{k}(k-1) V^{(n-1)}(s) + \frac{1}{k} G_t^{(n)} = V^{(n-1)}(s) \\ &\quad + \frac{1}{k}\left(G_t^{(n)} - V^{(n-1)}(s) \right). \end{aligned} \quad \text{(A.3)}$$

Figure 10 demonstrates the learning process of Monte-Carlo method. As we can see, the learning step is performed at the end of an episode.

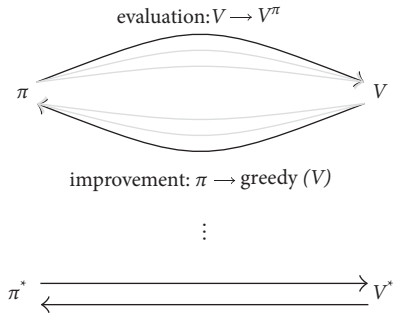


Figure 9: Learning by policy iteration method.

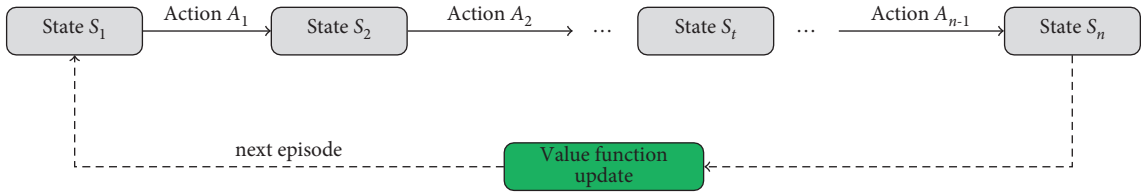


Figure 10: Monte-Carlo learning method.

Temporal-Difference Learning. Temporal-difference (TD) agent learns from incomplete episodes by applying bootstrapping. Comparing to MC learning, TD uses best guess of total return, or formally $R_{t+1} + \gamma V(S_{t+1})$ instead of episodic experience G_t to calculate value function estimates $(V(s))$. This single difference indicates that TD agent can perform a learning step after each and every actions ([4], Section 6.1), as Figure 11 shows. As a consequence, it can be applied at never ending episodes.

Temporal-Difference (λ) *Learning.* There are intermediate solutions between TD that performs VF estimate updates after 1-step return and MC that performs updates only at the end of an episode (practically ∞-step return). The main idea behind is to apply normalized geometric series $(1-\lambda)\lambda^{n-1}$ for weighting n-step returns $G_t^{(n)}$ ([4], Section 7.1). In this case, value function estimate will use a weighted total return of $G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$. It can be shown that TD(0) is equivalent to every-visit MC learning and TD(1) is equivalent to original TD learning methods. Furthermore, TD (λ) methods can be applied both forward and backward. The algorithms shown in this section can be used whether

> In offline mode: value function estimate updates are accumulated within episodes but applied only at the end of the episode, or
>
> In online mode: value function estimate updates are accumulated within episodes and can be applied immediately.

A unified view of model-free prediction techniques is shown in Figure 12. First, it was created by Richard Sutton, but this version is prepared by David Silver. It highlights the two most important dimensions of learning methods: the vertical dimension represents the depth of the updates, while the horizontal dimension represents the width of the updates.

C. Model-Free Control Methods

In the previous section, model-free prediction methods were summarized. These are methods that learn from other's experience so acting policies were managed from the external and called off-policy learning. In contrast, on-policy learning lets the algorithm to make actions on the basis of their own policy. Hence, a major objective steps to the front, to optimize policy.

In this section, ϵ-Greedy policy iteration is described to combine exploitation of the current knowledge of optimal decisions and exploration of unknown new potentials. Furthermore, on-policy temporal-difference control method known as SARSA method is presented by applying bootstrapping techniques to speed-up the learning process.

ϵ-Greedy Policy Iteration Control. ϵ-Greedy policy iteration covers a combined solution. On the one hand, MC method is applied to learn the action-value function $Q(s;a)$. On the other hand, the agent can act greedily which means that it will choose the most optimal action on the basis of the actual action-value function $Q(s;a)$. This kind of action policy exploits only the current experience and does not support to explore alternatives. With a small change in the strategy, this kind of issue can be solved: let the agent act randomly with probability ϵ and greedily with probability $(1-\epsilon)$ ([4], Section 5.4):

$$\pi(as) = \begin{cases} 1 - \epsilon + \dfrac{\epsilon}{m}, & \text{if } a = \operatorname{argmax}_{a' \in \mathcal{A}} Q(s, a'), \\ \dfrac{\epsilon}{m}, & \text{otherwise.} \end{cases} \tag{A.4}$$

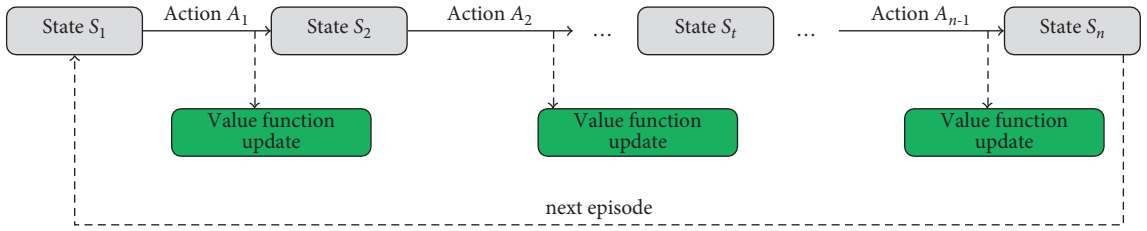


Figure 11: Temporal-difference learning method.

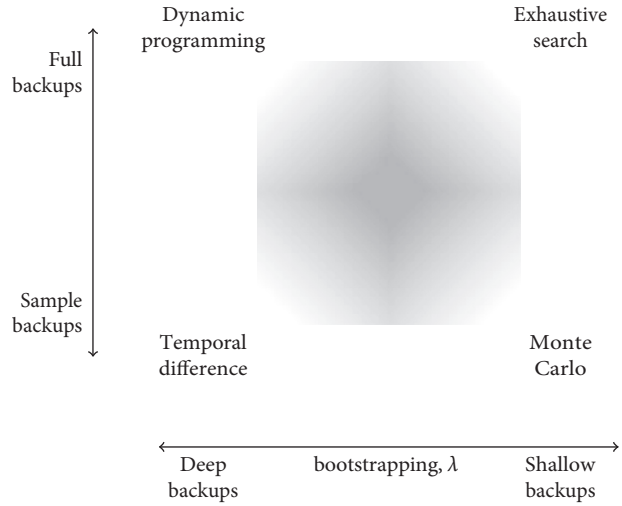


Figure 12: Unified view of model-free prediction techniques.

On-Policy Temporal-Difference Control Method, Aka SARSA Method. Similar to model-free prediction methods, there is also an algorithm to let agent learn from incomplete episodes by applying bootstrapping ([4], Section 6.4). In this case, ϵ-Greedy policy iteration method needs to be modified in the following way: instead of using MC method, TD learning should be applied for learning the action-value function $Q(s;a)$ that makes possible to perform a learning step after each and every actions and acting according to the most updated action-value function in a similar way than at ϵ-greedy policy iteration. The SARSA name comes from an acronym: state $s \longrightarrow$ action $a \longrightarrow$ reward $r \longrightarrow$ state $s' \longrightarrow$ action a'. By following SARSA method, action-value function update should look like $Q(s;a) \leftarrow Q(s;a) + \alpha(r + \gamma Q(s';a') - Q(s;a))$. It can be proved that under certain conditions, SARSA action-value function converges to optimal action-value function: $Q(s;a) \longrightarrow q^*(s;a)$.

D. Off-Policy Learning

There are several situations when the learning process is not based on just own experience. Formally, this means that target policy $\pi(a \mid s)$ or state-value function $v_\pi(s)$ or action-value function $q_\pi(s;a)$ is determined by observing results of an external behaviour policy $\mu(a \mid s)$.

In this section, importance sampling is shown to determine the most accurate of the learning objective, and then Q-learning is described as an effective alternative to get the function iteration with a lower variance.

Importance Sampling. One possible way to handle the difference of target and behaviour policy is importance sampling when a correction multiplier shall be applied by processing observations ([4], Section 5.8). If MC learning is combined with importance sampling, then value function update will look like $(S_t) \leftarrow V(S_t) + \alpha(G_t^{\pi/\mu} - V(S_t))$. But because corrections are made at the end of an episode, the product of multipliers can drive to a dramatically high variance and hence MC learning is not suitable for off-policy learning.

Therefore, TD learning seems much more adequate to combine with importance sampling, because correction multiplier should be applied for only a single step and not for a whole episode:

$$(S_t) \leftarrow V(S_t) + \alpha\left(\frac{\pi(A_t \mid S_t)}{\mu(A_t \mid S_t)}(R_{t+1} + \gamma V(S_{t+1})) - V(S_t)\right). \tag{A.5}$$

Q-Learning. Another possible way to handle the difference of target and behaviour policy is to modify the value function update logic as Q-learning does ([4], Section 6.5). Assume that in state S_t, the very next action is derived by using behaviour policy: $A_{t+1} \sim \mu(\cdot \mid S_t)$. By taking action A_{t+1}, immediate reward R_{t+1} and the next state S_{t+1} will be determined. But for value function update, let us consider an alternative successor

action on the basis of target policy: $A' \sim \pi(\cdot \mid S_t)$. Therefore, the importance sampling will be not necessary and Q-learning value function update will look like $Q(S_t; A_t) \longleftarrow Q(S_t; A_t) + \alpha(R_{t+1} + \gamma Q(S_{t+1}; A') - Q(S_t; A_t))$.

In a special case, if target policy π is chosen as a pure greedy policy and behaviour policy μ follows ϵ-greedy policy, then the so-called SARSAMAX update can be defined as follows:

$Q(S; A) \leftarrow Q(S; A) + \alpha(R + \gamma \max_{a'} Q(S'; a') - Q(S; A))$.

Last but not least, it was proven that Q-learning control converges to the optimal action-value function: $Q(s; a) \longrightarrow q_*(s; a)$.

E. Value Function Approximation

The reinforcement learning methods discussed in the previous sections represented value functions by lookup tables, but in practice, it is not feasible operating with state-level or state-action-level lookup tables. On the one hand, it would be very memory- and computation-intensive, and on the other hand, the learning process would be too slow if the state and/or action spaces are large. The solution for large problems is to estimate state-value and action-value functions with function approximation: $\widehat{v}(s; \mathbf{w}) \approx v_\pi(s)$, and similarly, $\widehat{q}(s; a; \mathbf{w}) \approx q_\pi(s; a)$.

There are many kinds of function approximation methods that can be applied: linear combination of features, neural network, decision tree, and Fourier bases. In this section, the first two types of methods are discussed. The first gradient descent method is presented that can be effectively combined with Monte-Carlo or temporal-difference methods for value function approximations, and then deep Q-network is described that serves a more sample-effective way from learning.

Value Function Approximation by Gradient Descent. A well-known tool for function approximation is gradient descent ([4], Section 9.3). Let us denote $J(\mathbf{w})$ as a differentiable function of parameter vector $\mathbf{w}$. Define the gradient of $J(\mathbf{w})$ as $\nabla_{\mathbf{w}} J(\mathbf{w}) = (\partial J(\mathbf{w})/\partial w_1, \ldots, \partial J(\mathbf{w})/\partial w_n,)^T$. To find a local minimum of $J(\mathbf{w})$, parameter $\mathbf{w}$ needs to be adjusted in the direction of negative gradient by $\Delta\mathbf{w} = -1/2\alpha\nabla_{\mathbf{w}} J(\mathbf{w})$ where α is the learning step-size parameter.

An effective solution is to use gradient descent with linear combination of features, because in this case, the formulas become much simpler. Value function representation will look like $\widehat{v}(S; \mathbf{w}) = \mathbf{x}(S)^T\mathbf{w} = \sum_{i=i}^{n} x_i(S) w_i$, while objective function to minimise mean-squared error between true value function and its approximation can be calculated by the formula of $J(\mathbf{w}) = \mathbb{E}_\pi[(v_\pi(S) - \mathbf{x}(S)^T\mathbf{w})^2]$. It is proven that stochastic gradient descent with linear combination of features converges to global optimum. Furthermore, the update rule is quite simple: $\nabla_{\mathbf{w}}\widehat{v}(S; \mathbf{w}) = \mathbf{x}(S)$, and then $\Delta\mathbf{w} = \alpha(v_\pi(S) - \widehat{v}(S; \mathbf{w}))\mathbf{x}(S)$. The result shows that parameter $\mathbf{w}$ adjustment stands for three components: learning step-size, prediction error, and feature value. In practice, the true value function is usually not known but a noisy sample of it is known at different methods:

For MC method, the target is G_t and hence parameter update $\Delta\mathbf{w} = \alpha(G_t - \widehat{v}(S_t; \mathbf{w}))\nabla_{\mathbf{w}}\widehat{v}(S_t; \mathbf{w})$.

For TD(0) method, the target is the TD target $R_{t+1} + \gamma\widehat{v}(S_{t+1}; \mathbf{w})$ while parameter update

$$\Delta\mathbf{w} = \alpha\left(R_{t+1} + \gamma\widehat{v}(S_{t+1}; \mathbf{w}) - \widehat{v}(S_t; \mathbf{w})\right)\nabla_{\mathbf{w}}\widehat{v}(S_t; \mathbf{w}). \tag{A.6}$$

For TD (λ), the target is λ-return G_t^λ and parameter update $\Delta\mathbf{w} = \alpha(G_t^\lambda - \widehat{v}(S_t; \mathbf{w}))\nabla_{\mathbf{w}}\widehat{v}(S_t; \mathbf{w})$.

Whichever method is chosen, the RL learning process needs to update the value function approximation with the same frequency than at the original method.

Deep Q-Network. Even if gradient descent-based value function approximation methods can be very calculation-effective and updates can be managed incrementally, these are less sample-effective which means that the information that could be extracted from an observation will be not necessarily exploited.

There are batch methods that are working with experience replay. Preliminary all the observed experiences should be collected. Let us denote $\mathscr{D}$ as the consisting experience of state-value pairs: $\mathscr{D} = \langle\langle s_1; v_1^\pi\rangle, \ldots, \langle s_n; v_n^\pi\rangle\rangle$. Artificial observations can be generated by random sampling from experience history: $\langle s; v^\pi\rangle \sim \mathscr{D}$. Therefore, stochastic gradient descent can be applied on it: $\Delta\mathbf{w} = \alpha(v^\pi - \widehat{v}(s; \mathbf{w}))\nabla_{\mathbf{w}}\widehat{v}(s; \mathbf{w})$. In this way, $\mathbf{w}^\pi$ converges to optimal least square solution.

One of a most commonly used RL methods was born by combining experience replay and Q-learning with periodically frozen target policy:

(1) By using behaviour policy, action a_t can be taken according to ϵ-greedy policy

(2) Transitions should be stored in replay memory $\mathscr{D}$ as $\langle s_t, a_t, t_{t+1}, s_{t+1}\rangle$

(3) There can be generated random mini-batch samples of transitions (s, a, r, s') from $\mathscr{D}$

(4) On the basis of them, Q-learning targets will be determined by using fixed parameters w^-

(5) Minimise mean-squared error between Q-network and Q-learning targets:

$$\mathscr{L}_i(w_i) = \mathbb{E}_{s,a,r,s' \sim \mathscr{D}_i}\left[\left(r + \gamma\max_{a'} Q(s', a', w_i^-) - Q(s, a, w_i)\right)^2\right]. \tag{A.7}$$

F. Policy Gradient

In contrast to value-based methods where optimal action can be determined on the basis of learnt value function in a particular state, policy gradient methods approximate directly the optimal policy: $\pi_\theta(s, a) = \mathbb{P}[a \mid s, \theta]$.

It is necessary for an objective function $J(\theta)$ to measure the goodness of fitting policy π_θ to the optimal policy. In this case, policy-based RL becomes an optimization problem to

find optimal θ according to $J(\theta)$. There are methods that use gradient as gradient descent, conjugate gradient, or quasi-Newton method and there are methods that do not use as hill climbing, simplex, or genetic algorithms. In general, these kinds of methods show better convergence properties and can work effectively with high-dimensional or continuous action spaces, and last but not least, they can learn stochastic policies. On the other hand, policy gradient methods typically converge to a local rather than global optimum. It is important to highlight that value functions can be also used to learn the optimal θ parameter, but once it is learnt, value functions are not necessary to select optimal action.

Softmax. Let $J(\theta)$ be a policy objective function. Policy gradient descent algorithms search for a local optimum in $J(\theta)$ by ascending the gradient of the policy: $\Delta\theta = \alpha\nabla_\theta J(\theta)$. By assuming that policy π_θ is differentiable and its gradient is $\nabla_\theta\pi_\theta(s,a)$, likelihood ratios can be transformed to the following form: $\nabla_\theta\pi_\theta(s,a) = \pi_\theta(s,a)\nabla_\theta\pi_\theta(s,\ a)/\pi_\theta(s,a) = \pi_\theta(s,a)\nabla_\theta\log\,\pi_\theta(s,a)$, where $\nabla_\theta\log\,\pi_\theta(s,a)$ is called score function.

Softmax policy method is based on the approach of weighting actions by using linear combinations of features $\phi(s,a)^T\theta$ ([4], Section 13.2). Therefore, the probabilities of actions are proportional to exponentiated weights: $\pi_\theta(s,a) \propto e^{\phi(s,a)^T\theta}$. The score function looks like $\nabla_\theta\log\,\pi_\theta(s,a) = \phi(s,a) - \mathbb{E}_{\pi_\theta}[\phi(s,\cdot)]$.

Gaussian/Natural Policy Gradient. In continuous action spaces, Gaussian policy is a natural option. In this case, the mean is a linear combination of features: $\mu(s) = \phi(s,a)^T\theta$. By fixing variance as σ^2, the policy will be Gaussian: $a \sim \mathcal{N}(\mu(s),\sigma^2)$. The score function will look like $\nabla_\theta\log\,\pi_\theta(s,a) = 1/\sigma^2\,(a-\mu(s))\phi(s)$.

Monte-Carlo Policy Gradient Method Aka REINFORCE. Monte-Carlo policy gradient method or with more popular name the REINFORCE algorithm updates θ parameter by using stochastic gradient ascent. It is strongly based on-policy gradient theorem that generalizes likelihood ratio approach to multistep MDPs by replacing immediate reward r with long-term values of $Q^\pi(s,a)$ with weak restrictions on $J(\theta)$. The key idea behind that the locally optimal policy can be found by gradient ascent on the objective function as follows: $\theta_{t+1}\leftarrow\theta_t + \alpha\nabla_{\theta_t}\log\,\pi_{\theta_t}(s_t,a_t)v_t$, where v_t is an unbiased sample of $Q^\pi_{\theta_t}(s_t,a_t)$.

Actor-Critic Policy Gradient. In practice, REINFORCE still has high variance. To handle it, action-value function can be also estimated: $Q_w(s,a) \approx Q^{\pi_\theta}(s,a)$. In this way, there are two sets of parameters:

Critic: it updates action-value function parameters w

Actor: it updates policy parameters θ according the actual version of critic

Updates should be done at each elementary steps as follows:

Sample reward: $r = \mathscr{R}^a_s$

Sample transition: $s' \sim \mathrm{P}^a_s$

Sample action: $a' \sim \pi_\theta(s,a')$

$\delta = r + \gamma Q_w(s',a') - Q_w(s,a)$

$\theta = \theta + \alpha\nabla_\theta\log\,\pi_\theta(s,a)Q_w(s,a)$

$w \longleftarrow w + \beta\gamma\phi(s,a)$

$s \longleftarrow s'$

$a \longleftarrow a'$

G. Model-Based Methods

Model-free methods learn value function and/or policy directly from their experience of a real environment. The accuracy of the knowledge of RL can be raised by extending the experience collection process. This can be reached either by setting up an artificial virtual environment due to defining reward and state transition functions that describes the real environment well or by building an own model that approximates the real environment by learning its history.

If it is assumed that the state space $\mathcal{S}$ and action space $\mathscr{A}$ are known, then model $\mathscr{M} = \langle \mathrm{P}_\eta; \mathscr{R}_\eta\rangle$ is a representation of MDP $\langle\mathcal{S};\mathscr{A};\mathrm{P};\mathscr{R}\rangle$ if $S_{t+1} \sim \mathrm{P}_\eta(S_{t+1} \mid S_t, A_t)$ and $R_{t+1} = \mathscr{R}_\eta(R_{t+1} \mid S_t, A_t)$. Learning model from experience is a supervised learning problem. Figure 13 presents the basic concept of model-based learning methods.

First, the model should learn and therefore an internal simulation environment can be defined. Then, using the model representation, the model-free RL methods can be used. So, model-based techniques differ from model-free techniques by using internal model representation to derive rewards and state transitions.

H. Multiagent Learning Systems

At Industry 4.0 applications, usually not a single RL agent is set up, but multiple ones. Multiagent RL topic addresses the sequential decision-making problem of multiple autonomous agents that operate in a common or quite similar environment, each of which aims to optimize its own long-term return by interacting with the environment and a central system and/or other agents.

Markov Games. One way to generalize MDPs for applying multiple agents is Markov games (MG) or also known as stochastic games. Formally, Markov game can be defined as a tuple $\langle\mathcal{N},\mathcal{S},\{\mathscr{A}^i\}_{i\in\mathcal{N}},\mathrm{P},\{R^i\}_{i\in\mathcal{N}},\gamma\rangle$, where $\mathcal{N} = \{1,\ldots,N\}$ denotes the set of $N > 1$ agents, $\mathcal{S}$ denotes the state space of all the agents, and $\mathscr{A}^i$ denotes the action space of agent $i \in \mathcal{N}$. By introducing $\mathscr{A} = \mathscr{A}^1 \times \ldots \times \mathscr{A}^N$, let P: $\mathcal{S}\times\mathscr{A} \longrightarrow \mathcal{S}$ be the transition probability function from any state $s \in \mathcal{S}$ to a particular state $s' \in \mathcal{S}$ for a joint action of $a \in \mathscr{A}$, while R^i: $\mathcal{S}\times\mathscr{A}\times\mathcal{S} \longrightarrow \mathbb{R}$ is the reward function that determines the immediate reward by starting from state s, by taking action a and by moving to state s'. Last but not least, $\gamma \in [0,1)$ is the discount factor. Figure 14 shows the general framework of Markov games.

MG problems can be classified by knowledge sharing strategies between agents and central system and their goals: whether they can learn from each other or is it worth to share observations or policies with each other or their goals are conflicting. The main categories are

Cooperative agents problem

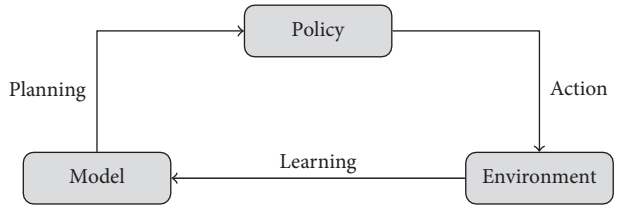


Figure 13: Model-based reinforcement learning process.

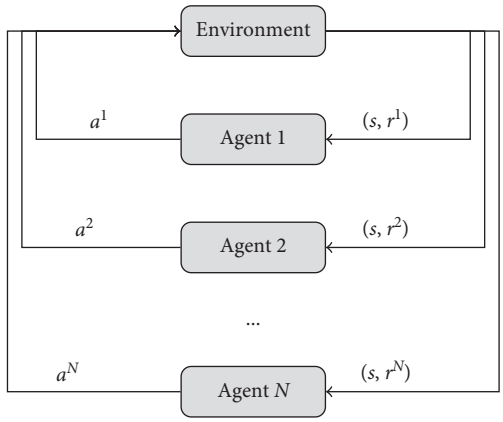


Figure 14: Schematic diagram of Markov games.

Conflicting agents problem

Mixed problem

In a fully cooperative setting, all agents have the very same or identical reward function: $R^1 = R^2 = \ldots = R^N = R$. This is also referred as multiagent MDP (MMDP). With this approach, the state- and action-value functions are identical to all agents, which thus enables the single-agent RL algorithms to be applied, if all agents are coordinated as one decision maker. The global optimum for cooperation now constitutes a Nash equilibrium of the game.

Nash equilibrium (NE) characterizes an equilibrium point π^*, from which none of the agents has any incentive to deviate. As a standard learning goal for MARL, NE always exists for discounted MGs, but may not be unique in general. Most of the MARL algorithms are contrived to converge to such an equilibrium point.

We believe that our summary of the major reinforcement learning methods gave a useful and efficient overview of the concept behind. As our literature overview shows there are numerous further modifications and extensions over the basis of the basic methods. By following our questionnaire in Figure 8, it becomes easier to determine the relevant area of RL methods that can provide an appropriate solution to be fitted to their learning problems.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the TKP2020-NKA-10 project financed under the 2020-4.1.1-TKP2020 Thematic Excellence Programme by the National Research, Development and Innovation Fund of Hungary.

References

[1] Y. Lu, "Industry 4.0: a survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

[2] V. Roblek, M. Meško, and A. Krapež, "A complex view of industry 4.0," *SAGE Open*, vol. 6, no. 2, 2016.

[3] J. Posada, C. Toro, I. Barandiaran et al., "Visual computing as a key enabling technology for industrie 4.0 and industrial internet," *IEEE Computer Graphics and Applications*, vol. 35, no. 2, pp. 26–40, 2015.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, MA, USA, 2018.

[5] R. Csalódi, Z. Süle, S. Jaskó, T. Holczinger, and J. Abonyi, "Industry 4.0-driven development of optimization algorithms: a systematic overview," *Complexity*, vol. 2021, Article ID 6621235, 22 pages, 2021.

[6] K.-D. Thoben, S. A. Wiesner, S. Wiesner, and T. Wuest, ""Industrie 4.0" and smart manufacturing - a review of research issues and application examples," *International Journal of Automation Technology*, vol. 11, no. 1, pp. 4–16, 2017.

[7] J. Mattioli, P. Perico, and P. O. Robic, *Improve Total Production Maintenance with Artificial Intelligence*, pp. 56–59, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ, USA, 2020.

[8] A. Dolgui, D. Ivanov, S. P. Sethi, and B. Sokolov, "Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications," *International Journal of Production Research*, vol. 57, no. 2, pp. 411–432, 2019.

[9] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: trends and perspectives towards Industry 4.0," *Information Fusion*, vol. 50, pp. 92–111, 2019.

[10] F. Castaño, G. Beruvides, A. Villalonga, and R. E. Haber, "Self-tuning method for increased obstacle detection reliability based on internet of things LiDAR sensor models," *Sensors*, vol. 18, no. 5, 2018.

[11] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive internet-of-things systems," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371–1387, 2019.

[12] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2009–2020, 2019.

[13] D. Nallaperuma, R. Nawaratne, T. Bandaragoda et al., "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019.

[14] S. Shresthamali, M. Kondo, and H. Nakamura, "Adaptive power management in solar energy harvesting sensor node using reinforcement learning," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 5s, 2017.

[15] Y. P. Pane, S. P. Nageshrao, J. Kober, and R. Babuška, "Reinforcement learning based compensation methods for robot manipulators," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 236–247, 2019.

[16] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9441–9455, 2020.

[17] L. Li, Y. Xu, J. Yin et al., "Deep reinforcement learning approaches for content caching in cache-enabled D2D networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 544–557, 2020.

[18] W. Hu, Y. Wen, K. Guan, G. Jin, and K. J. Tseng, "ITCM: toward learning-based thermal comfort modeling via pervasive sensing for smart buildings," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4164–4177, 2018.

[19] B. Chen, J. Wan, Y. Lan, M. Imran, D. Li, and N. Guizani, "Improving cognitive ability of edge intelligent IIoT through machine learning," *IEEE Network*, vol. 33, no. 5, pp. 61–67, 2019.

[20] C. Zhang, C. Gupta, A. Farahat, K. Ristovski, and D. Ghosh, "Equipment health indicator learning using deep reinforcement learning," *Machine Learning and Knowledge Discovery in Databases*, vol. 11053, pp. 488–504, 2019.

[21] A. Kawewong, Y. Honda, M. Tsuboyama, and O. Hasegawa, "Reasoning on the self-organizing incremental associative memory for online robot path planning," *IEICE - Transactions on Info and Systems*, vol. E93, no. 3, pp. 569–582, 2010.

[22] W. Xiong, Z. Lu, B. Li et al., "A self-adaptive approach to service deployment under mobile edge computing for autonomous driving," *Engineering Applications of Artificial Intelligence*, vol. 81, pp. 397–407, 2019.

[23] M. Chu, X. Liao, H. Li, and S. Cui, "Power control in energy harvesting multiple access system with reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9175–9186, 2019.

[24] A. Ismail and V. Cardellini, "Decentralized planning for self-adaptation in multi-cloud environment," *Communications in Computer and Information Science*, vol. 508, pp. 76–90, 2015.

[25] B. Wang, Y. Sun, T. Q. Duong, L. D. Nguyen, and L. Hanzo, "Risk-aware identification of highly suspected COVID-19 cases in social IoT: a joint graph theory and reinforcement learning approach," *IEEE Access*, vol. 8, pp. 115655–115661, 2020.

[26] F. Zhou, Q. Yang, K. Zhang, G. Trajcevski, T. Zhong, and A. Khokhar, "Reinforced spatiotemporal attentive graph neural networks for traffic forecasting," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6414–6428, 2020.

[27] L. Roveda, J. Maskani, P. Franceschi et al., "Model-based reinforcement learning variable impedance control for human-robot collaboration," *Journal of Intelligent and Robotic Systems*, vol. 100, no. 2, pp. 417–433, 2020.

[28] H. Wu, Z. Zhang, C. Jiao, C. Li, and T. Q. S. Quek, "Learn to sense: a meta-learning-based sensing and fusion framework for wireless sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8215–8227, 2019.

[29] G. Vallathan, A. John, C. Thirumalai, S. Mohan, G. Srivastava, and J. C.-W. Lin, "Suspicious activity detection using deep learning in secure assisted living IoT environments," *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3242–3260, 2021.

[30] S. Chesney, K. Roy, and S. Khorsandroo, "Machine learning algorithms for preventing IoT cybersecurity attacks," *Advances in Intelligent Systems and Computing*, vol. 1252, pp. 679–686, 2021.

[31] K. H. K. Reddy, A. K. Luhach, B. Pradhan, J. K. Dash, and D. S. Roy, "A genetic algorithm for energy efficient fog layer resource management in context-aware smart cities," *Sustainable Cities and Society*, p. 63, 2020.

[32] M. S. Munir, S. F. Abedin, N. H. Tran, Z. Han, E. Huh, and C. S. Hong, "Risk-aware energy scheduling for edge computing with microgrid: a multi-agent deep reinforcement learning approach," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3476–3497, 2021.

[33] B. Wang, Y. Sun, M. Sun, and X. Xu, "Game-Theoretic actor-critic-based intrusion response scheme (GTAC-IRS) for wireless SDN-based IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1830–1845, 2021.

[34] E. Baccour, A. Erbad, A. Mohamed et al., "RL-OPRA: reinforcement learning for online and proactive resource allocation of crowdsourced live videos," *Future Generation Computer Systems*, vol. 112, pp. 982–995, 2020.

[35] S.-G. Choi and S.-B. Cho, "Bayesian networks + reinforcement learning: controlling group emotion from sensory stimuli," *Neurocomputing*, vol. 391, pp. 355–364, 2020.

[36] K. Lepenioti, M. Pertselakis, A. Bousdekis, A. Louca, F. Lampathaki, and D. Apostolou, "Machine learning for predictive and prescriptive analytics of operational data in smart manufacturing," *Lecture Notes in Business Information Processing*, vol. 382, 2020.

[37] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, and K. Zheng, "Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7938–7953, 2021.

[38] Z.-Y. Wu, M. Ismail, E. Serpedin, and J. Wang, "Data-driven link assignment with QoS guarantee in mobile RF-optical HetNet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5088–5102, 2020.

[39] S. Sun, X. Li, M. Liu, B. Yang, and X. Guo, "DNN inference acceleration via heterogeneous IoT devices collaboration," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 57, no. 4, pp. 709–722, 2020.

[40] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: how do IoT devices use AI to enhance security?" *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[41] W. Liang, W. Huang, J. Long, K. Zhang, K.-C. Li, and D. Zhang, "Deep reinforcement learning for resource protection and real-time detection in IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6392–6401, 2020.

[42] X. Zhou, W. Liang, K. I.-K. Wang, H. Wang, L. T. Yang, and Q. Jin, "Deep-learning-enhanced human activity recognition for internet of healthcare things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

[43] F. Castaño, S. Okczak, A. Villalonga, R. E. Haber, and J. Kossakowska, "Sensor reliability in cyber-physical systems using internet-of-things data: a review and case study," *Remote Sensing*, vol. 11, no. 19, 2019.

[44] S. Tu, M. Waqas, S. U. Rehman et al., "Security in fog computing: a novel technique to tackle an impersonation attack," *IEEE Access*, vol. 6, pp. 74993–75001, 2018.

[45] B. Chatterjee, N. Cao, A. Raychowdhury, and S. Sen, "Context-aware intelligence in resource-constrained IoT

nodes: opportunities and challenges," *IEEE Design & Test*, vol. 36, no. 2, pp. 7–40, 2019.

[46] N. Aihara, K. Adachi, O. Takyu, M. Ohta, and T. Fujii, "Q-learning aided resource allocation and environment recognition in LoRaWAN with CSMA/CA," *IEEE Access*, vol. 7, pp. 152126–152137, 2019.

[47] W. Seok and C. Park, "Recognition of human motion with deep reinforcement learning," *IEIE Transactions on Smart Processing & Computing*, vol. 7, no. 3, pp. 245–250, 2018.

[48] Q. Hu, S. Lv, Z. Shi, L. Sun, and L. Xiao, "Defense against advanced persistent threats with expert system for internet of things," *Wireless Algorithms, Systems, and Applications*, vol. 10251, pp. 326–337, 2017.

[49] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the internet of things: a survey on the challenges and solutions," *Electronics (Switzerland)*, vol. 9, no. 3, 2020.

[50] R. S. Alonso, I. Sittón-Candanedo, R. Casado-Vara, J. Prieto, and J. M. Corchado, "Deep reinforcement learning for the management of software-defined networks and network function virtualization in an Edge-IoT architecture," *Sustainability*, vol. 12, no. 14, 2020.

[51] Y. Meng, S. Tu, J. Yu, and F. Huang, "Intelligent attack defense scheme based on DQL algorithm in mobile fog computing," *Journal of Visual Communication and Image Representation*, vol. 65, 2019.

[52] X. Ma and W. Shi, "AESMOTE: adversarial reinforcement learning with SMOTE for anomaly detection," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 943–956, 2020.

[53] Y. Liu, K. F. Tong, and K. K. Wong, "Reinforcement learning based routing for energy sensitive wireless mesh IoT networks," *Electronics Letters*, vol. 55, no. 17, pp. 966–968, 2019.

[54] Y. Huang, X. Guan, H. Chen, Y. Liang, S. Yuan, and T. Ohtsuki, "Risk assessment of private information inference for motion sensor embedded IoT devices," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 265–275, 2020.

[55] P. Saha and S. Mukhopadhyay, "Multispectral information fusion with reinforcement learning for object tracking in IoT edge devices," *IEEE Sensors Journal*, vol. 20, no. 8, pp. 4333–4344, 2020.

[56] P. Sun, Y. Dong, S. Yuan, and C. Wang, "Preventive control policy construction in active distribution network of cyber-physical system with reinforcement learning," *Applied Sciences*, vol. 11, no. 1, pp. 1–20, 2020.

[57] Q.-D. Ngo, H.-T. Nguyen, H.-L. Pham et al., "A graph-based approach for IoT botnet detection using reinforcement learning," *Computational Collective Intelligence*, vol. 12496, pp. 465–478, 2020.

[58] S. Sree Dharinya and E. P. Ephzibah, "Machine intelligence and automation: deep learning concepts aiding industrial applications," *EAI/Springer Innovations in Communication and Computing*, pp. 237–248, Springer, Cham, Switzerland, 2020.

[59] Y. H. Lai, Y. C. Chang, C. W. Tsai, C. H. Lin, and M. Y. Chen, "Data fusion analysis for attention-deficit hyperactivity disorder emotion recognition with thermal image and Internet of Things devices," *Software: Practice and Experience*, vol. 51, no. 3, pp. 595–606, 2021.

[60] Z. Zhang, C. Li, S. L. Peng, and X. Pei, "A new task offloading algorithm in edge computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, 2021.

[61] L. Espinosa-Leal, A. Chapman, and M. Westerlund, "Autonomous industrial management via reinforcement learning," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 6, pp. 8427–8439, 2020.

[62] S. Khan, M. Farnsworth, R. McWilliam, and J. Erkoyuncu, "On the requirements of digital twin-driven autonomous maintenance," *Annual Reviews in Control*, vol. 50, pp. 13–28, 2020.

[63] S. A. Alghamdi, "An effective strategy for fingerprint recognition based on pRAM's neural nature with data input mappings," *Proceedings of the International Congress on Information and Communication Technology*, vol. 439, pp. 623–634, 2016.

[64] H. Benaddi, K. Ibrahimi, A. Benslimane, and J. Qadir, "A deep reinforcement learning based intrusion detection system (drl-ids) for securing wireless sensor networks and internet of things," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*,vol. 317, 2020.

[65] Y. Qin, Q. Xia, Z. Xu et al., "Enabling multicast slices in edge networks," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8485–8501, 2020.

[66] R. Heartfield, G. Loukas, A. Bezemskij, and E. Panaousis, "Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1720–1735, 2021.

[67] S. Guo, Y. Qi, Y. Jin, W. Li, X. Qiu, and L. Meng, "Endogenous trusted DRL-based service function chain orchestration for IoT," *IEEE Transactions on Computers*, vol. 70, 2021.

[68] G. Wu, "UAV-Based interference source localization: a multimodal Q-learning approach," *IEEE Access*, vol. 7, pp. 137982–137991, 2019.

[69] R. K. Dhanaraj, K. Rajkumar, and U. Hariharan, "Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning," *EAI/Springer Innovations in Communication and Computing*, pp. 55–79, Springer, Cham, Switzerland, 2020.

[70] D. Yu, P. Li, Y. Chen, Y. Ma, and J. Chen, "A time-efficient multi-protocol probe scheme for fine-grain iot device identification," *Sensors*, vol. 20, no. 7, 2020.

[71] L. Chen, Y. Xu, Z. Lu, J. Wu, K. Gai, and P. C. K. Hung, "IoT microservice deployment in edge-cloud hybrid environment using reinforcement learning," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12610–12622, 2020.

[72] K.-H. Phung, B. Lemmens, M. Goossens, A. Nowe, L. Tran, and K. Steenhaut, "Schedule-based multi-channel communication in wireless sensor networks: a complete design and performance evaluation," *Ad Hoc Networks*, vol. 26, pp. 88–102, 2015.

[73] T. Akazaki, S. Liu, Y. Yamagata, Y. Duan, and J. Hao, "Falsification of cyber-physical systems using deep reinforcement learning," *Formal Methods*, vol. 10951, pp. 456–465, 2018.

[74] K. Gai, M. Qiu, M. Liu, and H. Zhao, "Smart resource allocation using reinforcement learning in content-centric cyber-physical systems," *Lecture Notes in Computer Science*, vol. 10699, 2018.

[75] F. Jazayeri, A. Shahidinejad, and M. Ghobaei-Arani, "Autonomous computation offloading and auto-scaling the in the mobile fog computing: a deep reinforcement learning-based approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 8, pp. 8265–8284, 2020.

[76] A. Pauna, I. Bica, F. Pop, and A. Castiglione, "On the rewards of self-adaptive IoT honeypots," *Annales des*

Telecommunications/Annals of Telecommunications, vol. 74, no. 7-8, pp. 501–515, 2019.

[77] D. Kwon, J. Jeon, S. Park, J. Kim, and S. Cho, "Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9895–9903, 2020.

[78] C. Shu, Z. Zhao, G. Min, J. Hu, and J. Zhang, "Deploying network functions for multiaccess edge-IoT with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9507–9516, 2020.

[79] V. Antuori, E. Hebrard, M.-J. Huguet, S. Essodaigui, and A. Nguyen, "Leveraging reinforcement learning, constraint programming and local search: a case study in car manufacturing," *Lecture Notes in Computer Science*, vol. 12333, pp. 657–672, 2020.

[80] R. Wu, J. Gong, W. Tong, and B. Fan, "Network attack path selection and evaluation based on Q-learning," *Applied Sciences*, vol. 11, no. 1, pp. 1–13, 2021.

[81] S. Redhu and R. M. Hegde, "Cooperative network model for joint mobile sink scheduling and dynamic buffer management using Q-learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1853–1864, 2020.

[82] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: the road to pareto-optimal wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1472–1514, 2020.

[83] S. Vimal, M. Khari, R. G. Crespo, L. Kalaivani, N. Dey, and M. Kaliappan, "Energy enhancement using Multiobjective Ant colony optimization with Double Q learning algorithm for IoT based cognitive radio networks," *Computer Communications*, vol. 154, pp. 481–490, 2020.

[84] X. Liu, J. Yu, J. Wang, and Y. Gao, "Resource allocation with edge computing in IoT networks via machine learning," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3415–3426, 2020.

[85] S. K. Sathya Lakshmi Preetha, R. Dhanalakshmi, and R. Kumar, "An energy efficient framework for densely distributed WSNs IoT devices based on tree based robust cluster head," *Wireless Personal Communications*, vol. 103, no. 4, pp. 3163–3180, 2018.

[86] J. Liu, D. Li, and Y. Xu, "Collaborative online edge caching with bayesian clustering in wireless networks," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1548–1560, 2020.

[87] M. K. Pandit, R. N. Mir, and M. A. Chishti, "Adaptive task scheduling in IoT using reinforcement learning," *International Journal of Intelligent Computing and Cybernetics*, vol. 13, no. 3, pp. 261–282, 2020.

[88] H. Li, K. Ota, and M. Dong, "Deep reinforcement scheduling for mobile crowdsensing in fog computing," *ACM Transactions on Internet Technology*, vol. 19, no. 2, 2019.

[89] S. A. Khowaja and P. Khuwaja, *Q-learning and LSTM Based Deep Active Learning Strategy for Malware Defense in Industrial IoT Applications*, Multimedia Tools and Applications, New York, NY, USA, 2021.

[90] A. Sharif, J. P. Li, M. A. Saleem et al., "A dynamic clustering technique based on deep reinforcement learning for Internet of vehicles," *Journal of Intelligent Manufacturing*, vol. 32, no. 3, pp. 757–768, 2021.

[91] F. Hussain, R. Hussain, A. Anpalagan, and A. Benslimane, "A new block-based reinforcement learning approach for distributed resource allocation in clustered IoT networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2891–2904, 2020.

[92] K. Nivitha, A. Solaiappan, and P. Pabitha, "Robust service selection through intelligent clustering in an uncertain environment," *Intelligence in Big Data Technologies-Beyond the Hype*, vol. 1167, pp. 325–332, 2021.

[93] T. Zhou, D. Tang, H. Zhu, and L. Wang, "Reinforcement learning with composite rewards for production scheduling in a smart factory," *IEEE Access*, vol. 9, pp. 752–766, 2021.

[94] J. H. Cho and H. Lee, "Dynamic topology model of q-learning leach using disposable sensors in autonomous things environment," *Applied Sciences*, vol. 10, no. 24, pp. 1–19, 2020.

[95] H. Qi, X. Mu, and Y. Shi, "A task unloading strategy of IoT devices using deep reinforcement learning based on mobile cloud computing environment," *Wireless Networks*, 2020.

[96] X. Cheng, F. Lyu, W. Quan et al., "Space/aerial-Assisted computing offloading for IoT applications: a learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

[97] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2061–2073, 2019.

[98] F. Bu and X. Wang, "A smart agriculture IoT system based on deep reinforcement learning," *Future Generation Computer Systems*, vol. 99, pp. 500–507, 2019.

[99] C.-C. Lin, D.-J. Deng, Y.-L. Chih, and H.-T. Chiu, "Smart manufacturing scheduling with edge computing using multiclass deep Q network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4276–4284, 2019.

[100] H. Yang, A. Alphones, W.-D. Zhong, C. Chen, and X. Xie, "Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5565–5576, 2020.

[101] T. Yu, B. Zhou, K. W. Chan, and E. Lu, "Stochastic optimal CPS relaxed control methodology for interconnected power systems using Q-learning method," *Journal of Energy Engineering*, vol. 137, no. 3, pp. 116–129, 2011.

[102] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, "Multi-Armed bandit learning in IoT networks: learning helps even in non-stationary settings," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 228, pp. 173–185, 2018.

[103] S. Shen, Y. Han, X. Wang, and Y. Wang, "Computation offloading with multiple agents in edge-computing–supported IoT," *ACM Transactions on Sensor Networks*, vol. 16, no. 1, 2019.

[104] A. Nassar and Y. Yilmaz, "Reinforcement learning for adaptive resource allocation in fog RAN for IoT with heterogeneous latency requirements," *IEEE Access*, vol. 7, pp. 128014–128025, 2019.

[105] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, "Deep reinforcement learning for autonomous internet of things: model, applications and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1722–1760, 2020.

[106] R. Ali, Y. A. Qadri, Y. Bin Zikria, T. Umer, B. S. Kim, and S. W. Kim, "Q-learning-enabled channel access in next-generation dense wireless networks for IoT-based eHealth systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[107] J. Zhang and J. Sun, "A game theoretic approach to multi-channel transmission scheduling for multiple linear systems under DoS attacks," *Systems & Control Letters*, vol. 133, 2019.

[108] M. Kwon, J. Lee, and H. Park, "Intelligent IoT connectivity: deep reinforcement learning approach," *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2782–2791, 2020.

[109] M. Camelo, M. Claeys, and S. Latre, "Parallel reinforcement learning with minimal communication overhead for IoT environments," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1387–1400, 2020.

[110] P. Farhat, H. Sami, and A. Mourad, "Reinforcement R-learning model for time scheduling of on-demand fog placement," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 388–410, 2020.

[111] X. Wei, J. Zhao, L. Zhou, and Y. Qian, "Broad reinforcement learning for supporting fast autonomous IoT," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7010–7020, 2020.

[112] S. G. Choi and S. B. Cho, "Sensor information fusion by integrated AI to control public emotion in a cyber-physical environment," *Sensors*, vol. 18, no. 11, 2018.

[113] G. Rjoub, J. Bentahar, O. Abdel Wahab, and A. Saleh Bataineh, "Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems," *Concurrency Computation*, vol. 33, no. 23, 2020.

[114] P. Loreti, L. Bracciale, and G. Bianchi, "StableSENS: sampling time decision algorithm for IoT energy harvesting devices," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9908–9918, 2019.

[115] S. Shukla, M. F. Hassan, L. T. Jung, and A. Awang, "Architecture for latency reduction in healthcare internet-of-things using reinforcement learning and fuzzy based fog computing," *Advances in Intelligent Systems and Computing*, vol. 843, pp. 372–383, 2019.

[116] H. Rashtian and S. Gopalakrishnan, "Balancing message criticality and timeliness in IoT networks," *IEEE Access*, vol. 7, pp. 145738–145745, 2019.

[117] Y. Dai, G. Wang, K. Muhammad, and S. Liu, "A closed-loop healthcare processing approach based on deep reinforcement learning," *Multimedia Tools and Applications*, p. 79, 2020.

[118] C. Lork, W. T. Li, Y. Qin, Y. Zhou, C. Yuen, and W. Tushar, "An uncertainty-aware deep reinforcement learning framework for residential air conditioning energy management," *Applied Energy*, p. 276, 2020.

[119] M. Faraji Mehmandar, S. Jabbehdari, and H. Haj Seyyed Javadi, "A dynamic fog service provisioning approach for IoT applications," *International Journal of Communication Systems*, vol. 33, no. 14, 2020.

[120] J. Zhang, J. Du, Y. Shen, and J. Wang, "Dynamic computation offloading with energy harvesting devices: a hybrid-decision-based deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9303–9317, 2020.

[121] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Collaborative edge computing and caching with deep reinforcement learning decision agents," *IEEE Access*, vol. 8, pp. 120604–120612, 2020.

[122] S. Chen, J. Wang, H. Li, Z. Wang, F. Liu, and S. Li, "Top-down human-cyber-physical data fusion based on reinforcement learning," *IEEE Access*, vol. 8, pp. 134233–134245, 2020.

[123] J. Leng, G. Ruan, Y. Song, Q. Liu, Y. Fu, and K. Ding, "A loosely-coupled deep reinforcement learning approach for order acceptance decision of mass-individualized printed circuit board manufacturing in industry 4.0," *Journal of Cleaner Production*, p. 280, 2021.

[124] M. Mobasheri, Y. Kim, and W. Kim, "Fog fragment cooperation on bandwidth management based on reinforcement learning," *Sensors*, vol. 20, no. 23, pp. 1–15, 2020.

[125] G. Rjoub, O. Abdel Wahab, J. Bentahar, and A. Bataineh, "A trust and energy-aware double deep reinforcement learning scheduling strategy for federated learning on IoT devices," *Service-Oriented Computing*, vol. 12571, pp. 319–333, 2020.

[126] J. Na, H. Zhang, X. Deng, B. Zhang, and Z. Ye, "Accelerate personalized iot service provision by cloud-aided edge reinforcement learning: a case study on smart lighting," *Service-Oriented Computing*, vol. 12571, pp. 69–84, 2020.

[127] M. Tiwari, S. Misra, P. K. Bishoyi, and L. T. Yang, "Devote: criticality-aware federated service provisioning in fog-based Iot environments," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10631–10638, 2021.

[128] A. Haldorai, A. Ramu, and M. Suriya, "Organization internet of things (IoTs): supervised, unsupervised, and reinforcement learning," *EAI/Springer Innovations in Communication and Computing*, pp. 237–248, Springer, Cham, Switzerland, 2020.

[129] A. Musaddiq, R. Ali, J.-G. Choi, B.-S. Kim, and S. Won Kim, "Collision observation-based optimization of low-power and lossy IoT network using reinforcement learning," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 799–814, 2021.

[130] X. Zhou, X. Dong, Z. Laiping, K. Li, and T. Qiu, "Learning-driven cloud resource provision policy for content providers with competitors," *IEEE Transactions on Cloud Computing*, p. 8, 2020.

[131] Y. Hao, M. Chen, H. Gharavi, Y. Zhang, and K. Hwang, "Deep reinforcement learning for edge service placement in softwarized industrial cyber-physical system," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5552–5561, 2021.

[132] T. Park, N. Abuzainab, and W. Saad, "Learning how to communicate in the internet of things: finite resources and heterogeneity," *IEEE Access*, vol. 4, pp. 7063–7073, 2016.

[133] M. G. R. Alam, M. M. Hassan, M. Z. Uddin, A. Almogren, and G. Fortino, "Autonomic computation offloading in mobile edge for IoT applications," *Future Generation Computer Systems*, vol. 90, pp. 149–157, 2019.

[134] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3516–3526, 2019.

[135] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric IoT," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 781–796, 2020.

[136] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2581–2593, 2020.

[137] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, "IRAF: a deep reinforcement learning approach for collaborative mobile edge computing IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7011–7024, 2019.

[138] S. Deng, Z. Xiang, P. Zhao et al., "Dynamical resource allocation in edge for trustable internet-of-things systems: a reinforcement learning method," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6103–6113, 2020.

[139] K. Gai and M. Qiu, "Optimal resource allocation using reinforcement learning for IoT content-centric services," *Applied Soft Computing*, vol. 70, pp. 12–21, 2018.

[140] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge

computing-supported internet of things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019.

[141] Y. Li, H. Ji, X. Li, and V. C. M. Leung, "Dynamic channel selection with reinforcement learning for cognitive WLAN over fiber," *International Journal of Communication Systems*, vol. 25, no. 8, pp. 1077–1090, 2012.

[142] S. Vimal, M. Khari, N. Dey, R. G. Crespo, and Y. Harold Robinson, "Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT," *Computer Communications*, vol. 151, pp. 355–364, 2020.

[143] A. Alsarhan, A. Itradat, A. Y. Al-Dubai, A. Y. Zomaya, and G. Min, "Adaptive resource allocation and provisioning in multi-service cloud environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 1, pp. 31–42, 2018.

[144] Z. Wei, B. Zhao, J. Su, and X. Lu, "Dynamic edge computation offloading for internet of things with energy harvesting: a learning method," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4436–4447, 2019.

[145] J. Wang, C. Jiang, K. Zhang, X. Hou, Y. Ren, and Y. Qian, "Distributed Q-learning aided heterogeneous network association for energy-efficient IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2756–2764, 2020.

[146] L. Mai, N. N. Dao, and M. Park, "Real-time task assignment approach leveraging reinforcement learning with evolution strategies for long-term latency minimization in fog computing," *Sensors*, vol. 18, no. 9, 2018.

[147] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[148] M. Khichane, P. Albert, and C. Solnon, "Strong combination of ant colony optimization with constraint programming optimization," *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, vol. 6140, pp. 232–245, 2010.

[149] A. Villalonga, G. Beruvides, F. Castano, and R. E. Haber, "Cloud-based industrial cyber-physical system for data-driven reasoning: a review and use case on an industry 4.0 pilot line," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5975–5984, 2020.

[150] T.-D. Lee, B. M. Lee, and W. Noh, "Hierarchical cloud computing architecture for context-aware IoT services," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 222–230, 2018.

[151] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-UAV-enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6898–6908, 2020.

[152] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Dynamic service function chain embedding for NFV-enabled IoT: a deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 507–519, 2020.

[153] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, 2020.

[154] A. Chowdhury, S. A. Raut, and H. S. Narman, "DA-DRLS: d," *Journal of Network and Computer Applications*, vol. 138, pp. 51–65, 2019.

[155] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Service function chain embedding for NFV-enabled IoT based on deep reinforcement learning," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 102–108, 2019.

[156] F. M. Talaat, M. S. Saraya, A. I. Saleh, H. A. Ali, and S. H. Ali, "A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4951–4966, 2020.

[157] C. Qiu, H. Yao, F. R. Yu, C. Jiang, and S. Guo, "A service-oriented permissioned blockchain for the internet of things," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 203–215, 2020.

[158] J. Yao and N. Ansari, "Task allocation in fog-aided mobile IoT by lyapunov online reinforcement learning," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 556–565, 2020.

[159] R. Zhao, X. Wang, J. Xia, and L. Fan, "Deep reinforcement learning based mobile edge computing for intelligent Internet of Things," *Physical Communication*, vol. 43, 2020.

[160] M. Nduwayezu, Q.-V. Pham, and W.-J. Hwang, "Online computation offloading in NOMA-based multi-access edge computing: a deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 99098–99109, 2020.

[161] T. Fu, C. Wang, and N. Cheng, "Deep-learning-based joint optimization of renewable energy storage and routing in vehicular energy network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6229–6241, 2020.

[162] S. Guo, Y. Dai, S. Xu, X. Qiu, and F. Qi, "Trusted cloud-edge network resource management: DRL-driven service function chain orchestration for IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6010–6022, 2020.

[163] Y. Zhang, B. Song, Y. Zhang, X. Du, and M. Guizani, "Market model for resource allocation in emerging sensor networks with reinforcement learning," *Sensors*, vol. 16, no. 12, 2016.

[164] S. Wan, J. Lu, P. Fan, and K. B. Letaief, "Toward big data processing in IoT: path planning and resource management of UAV base stations in mobile-edge computing system," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5995–6009, 2020.

[165] G. Cui, X. Li, L. Xu, and W. Wang, "Latency and energy optimization for MEC enhanced SAT-IoT networks," *IEEE Access*, vol. 8, pp. 55915–55926, 2020.

[166] H. Yang, X. Xie, and M. Kadoch, "Machine learning techniques and A case study for intelligent wireless networks," *IEEE Network*, vol. 34, no. 3, pp. 208–215, 2020.

[167] I. Khan, X. Tao, G. M. S. Rahman, W. U. Rehman, and T. Salam, "Advanced energy-efficient computation offloading using deep reinforcement learning in MTC edge computing," *IEEE Access*, vol. 8, pp. 82867–82875, 2020.

[168] H. Yang, W.-D. Zhong, C. Chen, A. Alphones, and X. Xie, "Deep-reinforcement-learning-based energy-efficient resource management for social and cognitive internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5677–5689, 2020.

[169] Q. Li, H. Yao, T. Mai, C. Jiang, and Y. Zhang, "Reinforcement-learning- and belief-learning-based double auction mechanism for edge computing resource allocation," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5976–5985, 2020.

[170] Y. Liu, S. Xie, and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12229–12239, 2020.

[171] A. Ashiquzzaman, H. Lee, T.-W. Um, and J. Kim, "Energy-efficient IoT sensor calibration with deep reinforcement learning," *IEEE Access*, vol. 8, pp. 97045–97055, 2020.

[172] J. Zhang, M. Dai, and Z. Su, "Task allocation with unmanned surface vehicles in smart ocean IoT," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9702–9713, 2020.

[173] D. Wang, W. Zhang, B. Song, X. Du, and M. Guizani, "Market-based model in CR-IoT: a Q-probabilistic multi-agent reinforcement learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 179–188, 2020.

[174] S. Xu, Q. Liu, B. Gong et al., "RJCC: reinforcement-learning-based joint communicational-and-computational resource allocation mechanism for smart city IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8059–8076, 2020.

[175] X. Chen and G. Liu, "Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10843–10856, 2021.

[176] Y. Zhao, L. Wang, S. Li, F. Zhou, X. Lin, and Q. Lu, "A visual analysis approach for understanding durability test data of automotive products," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, 2019.

[177] Y.-H. Xu, Y.-B. Tian, P. K. Searyoh, G. Yu, and Y.-T. Yong, "Deep reinforcement learning-based resource allocation strategy for energy harvesting-powered cognitive machine-to-machine networks," *Computer Communications*, vol. 160, pp. 706–717, 2020.

[178] N. N. Khumalo, O. O. Oyerinde, and L. Mfupe, "Reinforcement learning-based resource management model for fog radio access network architectures in 5G," *IEEE Access*, vol. 9, pp. 12706–12716, 2021.

[179] S. Ge, B. Lu, L. Xiao, J. Gong, X. Chen, and Y. Liu, "Mobile edge computing against smart attacks with deep reinforcement learning in cognitive MIMO IoT systems," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1851–1862, 2020.

[180] I. Alqerm and J. Pan, "Enhanced online Q-learning scheme for resource allocation with maximum utility and fairness in edge-IoT networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3074–3086, 2020.

[181] Q. Qi, L. Zhang, J. Wang et al., "Scalable parallel task scheduling for autonomous driving using multi-task deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13861–13874, 2020.

[182] G. Sun, R. Ou, and G. Liu, "Deep reinforcement learning-based resource reservation algorithm for emergency Internet-of-things slice," *Tongxin Xuebao/Journal on Communications*, vol. 41, no. 9, pp. 8–20, 2020.

[183] Y. Liao, X. Qiao, Q. Yu, and Q. Liu, "Intelligent dynamic service pricing strategy for multi-user vehicle-aided MEC networks," *Future Generation Computer Systems*, vol. 114, pp. 15–22, 2021.

[184] H. Qin, S. Zawad, Y. Zhou, S. Padhi, L. Yang, and F. Yan, "Reinforcement-learning-empowered MLaaS scheduling for serving intelligent internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6325–6337, 2020.

[185] S. Venticinque, S. Nacchia, and S. A. Maisto, "Reinforcement learning for resource allocation in cloud datacenter," *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, vol. 96, pp. 648–657, 2020.

[186] V. K. Prasad and M. D. Bhavsar, "Monitoring and prediction of sla for iot based cloud," *Scalable Computing: Practice and Experience*, vol. 21, no. 3, pp. 349–358, 2020.

[187] T. Liu, R. Luo, F. Xu, C. Fan, and C. Zhao, "Distributed learning based joint communication and computation strategy of iot devices in smart cities," *Sensors*, vol. 20, no. 4, 2020.

[188] I. Budhiraja, N. Kumar, and S. Tyagi, "Deep-reinforcement-learning-based proportional fair scheduling control scheme for underlay D2D communication," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3143–3156, 2021.

[189] S. Ramakrishna, C. Harstell, M. P. Burruss, G. Karsai, and A. Dubey, "Dynamic-weighted simplex strategy for learning enabled cyber physical systems," *Journal of Systems Architecture*, p. 111, 2020.

[190] M. Li, F. R. Yu, P. Si, W. Wu, and Y. Zhang, "Resource optimization for delay-tolerant data in blockchain-enabled IoT with edge computing: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9399–9412, 2020.

[191] A. Sapio, S. S. Bhattacharyya, and M. Wolf, "Runtime adaptation in wireless sensor nodes using structured learning," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 4, 2020.

[192] K. Priyadharshini and R. A. Canessane, "Light chain consensus reinforcement machine learning: an effective blockchain model for internet of things using for its advancement and challenges," *Computational Intelligence*, vol. 36, pp. 1–22, 2020.

[193] N. Yuan, C. Jia, J. Lu et al., "A DRL-based container placement scheme with auxiliary tasks," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1657–1671, 2020.

[194] M. Laroui, H. Ibn-Khedher, M. Ali Cherif, H. Moungla, H. Afifi, and A. E. Kamel, "So-vmec: service offloading in virtual mobile edge computing using deep reinforcement learning," *Transactions on Emerging Telecommunications Technologies*, vol. 32, 2021.

[195] J. Kim, D. Ryu, J. Kim, and J. H. Kim, "Two-stage hybrid network clustering using multi-agent reinforcement learning," *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–16, 2021.

[196] X. Shu, L. Wu, X. Qin, R. Yang, Y. Wu, and D. Wang, "Deep reinforcement learning cloud-edge-terminal computation resource allocation mechanism for IoT," *Advances in Intelligent Systems and Computing*, vol. 1274, 2021.

[197] J. Zhang, H. Guo, and J. Liu, "Adaptive task offloading in vehicular edge computing networks: a reinforcement learning based scheme," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1736–1745, 2020.

[198] A. P. Ortega, S. D. Ramchurn, L. Tran-Thanh, and G. V. Merrett, "Partner selection in self-organised wireless sensor networks for opportunistic energy negotiation: a multi-armed bandit based approach," *Ad Hoc Networks*, p. 112, 2021.

[199] H. Gao, Y. Xiao, H. Yan, Y. Tian, D. Wang, and W. Wang, "A learning-based credible participant recruitment strategy for mobile crowd sensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5302–5314, 2020.

[200] H. Yang, Z. Xiong, J. Zhao, D. Niyato, C. Yuen, and R. Deng, "Deep reinforcement learning based massive access management for ultra-reliable low-latency communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 2977–2990, 2021.

[201] Y. Liu, H. Lu, X. Li, Y. Zhang, L. Xi, and D. Zhao, "Dynamic service function chain orchestration for NFV/MEC-enabled iot networks: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7450–7465, 2020.

[202] W. Zhang, D. Yang, P. Haixia, W. Wu, W. Quan, and H. Zhang, "Deep reinforcement learning based resource management for dnn inference in industrial IoT," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7605–7618, 2021.

[203] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-q-learning-based transmission scheduling mechanism for the cognitive internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2375–2385, 2018.

[204] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent edge computing for IoT-based energy management in smart cities," *IEEE Network*, vol. 33, no. 2, pp. 111–117, 2019.

[205] Y.-R. Shiue, K.-C. Lee, and C.-T. Su, "Real-time scheduling for a smart factory using a reinforcement learning approach," *Computers & Industrial Engineering*, vol. 125, pp. 604–614, 2018.

[206] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: current issues and machine learning-assisted solutions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 426–471, 2020.

[207] L. Lei, H. Xu, X. Xiong, K. Zheng, W. Xiang, and X. Wang, "Multiuser resource control with deep reinforcement learning in IoT edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10119–10133, 2019.

[208] J. Ge, B. Liu, T. Wang, Q. Yang, A. Liu, and A. Li, *Q-learning based flexible task scheduling in a global view for the internet of things. Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 8, 2020.

[209] Q. Tan, Y. Tong, S. Wu, and D. Li, "Modeling, planning, and scheduling of shop-floor assembly process with dynamic cyber-physical interactions: a case study for CPS-based smart industrial robot production," *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3979–3989, 2019.

[210] P. Gazori, D. Rahbari, and M. Nickray, "Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach," *Future Generation Computer Systems*, vol. 110, pp. 1098–1115, 2020.

[211] B. Yin, S. Zhang, and Y. Cheng, "Application-Oriented scheduling for optimizing the age of correlated information: a deep-reinforcement-learning-based approach," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8748–8759, 2020.

[212] D. Kim, T. Lee, S. Kim, B. Lee, and H. Y. Youn, "Adaptive packet scheduling in IoT environment based on Q-learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 6, pp. 2225–2235, 2020.

[213] S. Park, S. Park, M. I. Choi et al., "Reinforcement learning-based bems architecture for energy usage optimization," *Sensors*, vol. 20, no. 17, pp. 1–33, 2020.

[214] H. He, H. Shan, A. Huang, Q. Ye, and W. Zhuang, "Edge-Aided computing and transmission scheduling for LTE-U-enabled IoT," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7881–7896, 2020.

[215] X. Fu, L. Lopez-Estrada, and J. G. Kim, "A Q-learning-based approach for enhancing energy efficiency of bluetooth low energy," *IEEE Access*, vol. 9, pp. 21286–21295, 2021.

[216] H.-S. Lee and J.-W. Lee, "Adaptive wireless power transfer beam scheduling for non-static iot devices using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 206659–206673, 2020.

[217] M. Samir, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Online altitude control and scheduling policy for minimizing AoI in UAV-assisted IoT wireless networks," *IEEE Transactions on Mobile Computing*, p. 19, 2020.

[218] H. Park, H. Kim, S.-T. Kim, and P. Mah, "Multi-agent reinforcement-learning-based time-slotted channel hopping medium access control scheduling scheme," *IEEE Access*, vol. 8, pp. 139727–139736, 2020.

[219] Y. Martínez Jiménez, J. Coto Palacio, and A. Nowé, "Multi-agent reinforcement learning tool for job shop scheduling problems," *Communications in Computer and Information Science*, vol. 1173, 2020.

[220] H. Hu, X. Jia, Q. He, S. Fu, and K. Liu, "Deep reinforcement learning based AGVs real-time scheduling with mixed rule for flexible shop floor in industry 4.0," *Computers & Industrial Engineering*, p. 149, 2020.

[221] H. Rashtian and S. Gopalakrishnan, "Using deep reinforcement learning to improve sensor selection in the internet of things," *IEEE Access*, vol. 8, pp. 95208–95222, 2020.

[222] Z. Wang, J. Wang, F. Yang, and M. Lin, "Q-learning-based energy transmission scheduling over a fading channel," *Journal of Southeast University*, vol. 36, no. 4, pp. 393–398, 2020.

[223] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[224] T. Yu, B. Zhou, K. W. Chan, L. Chen, and B. Yang, "Stochastic optimal relaxed automatic generation control in non-markov environment based on multi-step $Q(\lambda)$ learning," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1272–1282, 2011.

[225] T. Yu, H. Z. Wang, B. Zhou, K. W. Chan, and J. Tang, "Multi-agent correlated equilibrium Q(λ) learning for coordinated smart generation control of interconnected power grids," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1669–1679, 2015.

[226] T. Yu, B. Zhou, K. W. Chan, Y. Yuan, B. Yang, and Q. H. Wu, "R(λ) imitation learning for automatic generation control of interconnected power grids," *Automatica*, vol. 48, no. 9, pp. 2130–2136, 2012.

[227] J. Xia, Y. Xu, D. Deng, Q. Zhou, and L. Fan, "Intelligent secure communication for internet of things with statistical channel state information of attacker," *IEEE Access*, vol. 7, pp. 144481–144488, 2019.

[228] J. Shao, X. Zhao, J. Yang, W. Zhang, Y. Kang, and X. Zhao, "Reinforcement learning algorithm for path following control of articulated vehicle," *Nongye Jixie Xuebao/Transactions of the Chinese Society for Agricultural Machinery*, vol. 48, no. 3, pp. 376–382, 2017.

[229] H. D. Tran, F. Cai, M. Lopez Diego, P. Musau, T. T. Johnson, and X. Koutsoukos, "Safety verification of cyber-physical systems with reinforcement learning control," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 5s, 2019.

[230] J. Kang and D. S. Eom, "Offloading and transmission strategies for IoT edge devices and networks," *Sensors*, vol. 19, no. 4, 2019.

[231] X. Zhang, T. Yu, and J. Tang, "Optimal CPS command dispatch based on hierarchically correlated equilibrium reinforcement learning," *Dianli Xitong Zidonghua/Automation of Electric Power Systems*, vol. 39, no. 8, pp. 80–86, 2015.

[232] G. Faraci, A. Raciti, S. A. Rizzo, and G. Schembra, "Green wireless power transfer system for a drone fleet managed by reinforcement learning in smart industry," *Applied Energy*, vol. 259, 2020.

[233] J.-B. Kim, H.-K. Lim, C.-M. Kim, M.-S. Kim, Y.-G. Hong, and Y.-H. Han, "Imitation reinforcement learning-based remote rotary inverted pendulum control in openflow network," *IEEE Access*, vol. 7, pp. 36682–36690, 2019.

[234] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Energy-efficient orchestration in wireless

powered internet of things infrastructures," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 317–328, 2019.

[235] B.-N. Trinh, L. Murphy, and G.-M. Muntean, "A reinforcement learning-based duty cycle adjustment technique in wireless multimedia sensor networks," *IEEE Access*, vol. 8, pp. 58774–58787, 2020.

[236] D. Pacheco-Paramo, L. Tello-Oquendo, V. Pla, and J. Martinez-Bauset, "Deep reinforcement learning mechanism for dynamic access control in wireless networks handling mMTC," *Ad Hoc Networks*, vol. 94, 2019.

[237] H. K. Lim, J. B. Kim, J. S. Heo, and Y. H. Han, "Federated reinforcement learning for training control policies on multiple IoT devices," *Sensors*, vol. 20, no. 5, 2020.

[238] T. H. A. Kolobe and A. H. Fagg, "Robot reinforcement and error-based movement learning in infants with and without cerebral palsy," *Physical Therapy*, vol. 99, no. 6, pp. 677–688, 2019.

[239] R. Ali, B. Kim, S. W. Kim, H. S. Kim, and F. Ishmanov, "(ReLBT): a Reinforcement learning-enabled listen before talk mechanism for LTE-LAA and Wi-Fi coexistence in IoT," *Computer Communications*, vol. 150, pp. 498–505, 2020.

[240] T. Yu and S. P. Zhang, "Automatic control of electricity generation based on 5-component update learning algorithm SARSA (λ)," *Kongzhi Lilun Yu Yingyong/Control Theory and Applications*, vol. 30, no. 10, pp. 1246–1251, 2013.

[241] C. Liu, J. Gao, Y. Bi, X. Shi, and D. Tian, "A multitasking-oriented robot arm motion planning scheme based on deep reinforcement learning and twin synchro-control," *Sensors*, vol. 20, no. 12, pp. 1–35, 2020.

[242] L. An and G.-H. Yang, "Opacity enforcement for confidential robust control in linear cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1234–1241, 2020.

[243] G. Faraci, C. Grasso, and G. Schembra, "Fog in the clouds: UAVs to provide edge computing to IoT devices," *ACM Transactions on Internet Technology*, vol. 20, no. 3, 2020.

[244] H. Joo, S. H. Ahmed, and Y. Lim, "Traffic signal control for smart cities using reinforcement learning," *Computer Communications*, vol. 154, pp. 324–330, 2020.

[245] Q. Wu, J. Wu, J. Shen, B. Yong, and Q. Zhou, "An edge based multi-agent auto communication method for traffic light control," *Sensors*, vol. 20, no. 15, pp. 1–16, 2020.

[246] T. Yu, S. Zhang, and Y. Hong, "Dynamic optimal CPS control for interconnected power systems based on SARSA Algorithm," *Lecture Notes in Electrical Engineering*, vol. 238, pp. 269–276, 2014.

[247] H. Xu, X. Liu, W. Yu, D. Griffith, and N. Golmie, "Reinforcement learning-based control and networking Co-design for industrial internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 885–898, 2020.

[248] T. Wang, X. Shen, M. S. Obaidat, X. Liu, and S. Wan, "Edge-learning-based hierarchical prefetching for collaborative information streaming in social IoT systems," *IEEE Transactions on Computational Social Systems*, p. 7, 2020.

[249] S. Liu, S. Li, and B. Xu, "Event-triggered resilient control for cyber-physical system under denial-of-service attacks," *International Journal of Control*, vol. 93, no. 8, pp. 1907–1919, 2020.

[250] S. Souihi, M. Souidi, and A. Mellouk, "An adaptive QoE-based network interface selection for multi-homed eHealth devices," *Internet of Things. IoT Infrastructures*, vol. 169, pp. 437–442, 2016.

[251] H. Van Dong, B. Quoc Khanh, N. Tran Lich, and N. T. Ngoc Anh, "Integrating multi-agent system, geographic information system, and reinforcement learning to simulate and optimize traffic signal control," *Recent Advances in Information and Communication Technology 2018*, vol. 769, pp. 145–154, 2019.

[252] V. Hakami, S. Mostafavi, N. T. Javan, and Z. Rashidi, "An optimal policy for joint compression and transmission control in delay-constrained energy harvesting IoT devices," *Computer Communications*, vol. 160, pp. 554–566, 2020.

[253] S. Kim, "One-player game based influential maximization scheme for social cloud service networks," *EAI/Springer Innovations in Communication and Computing*, pp. 175–184, Springer, Cham, Switzerland, 2019.

[254] D. Pacheco-Paramo and L. Tello-Oquendo, "Delay-aware dynamic access control for mMTC in wireless networks using deep reinforcement learning," *Computer Networks*, p. 182, 2020.

[255] Y. Zhao, J. Hu, K. Yang, and S. Cui, "Deep reinforcement learning aided intelligent access control in energy harvesting based WLAN," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14078–14082, 2020.

[256] Y. Hadjadj-Aoul and S. Ait-Chellouche, "Access control in nb-iot networks: a deep reinforcement learning strategy," *Information*, vol. 11, no. 11, pp. 1–16, 2020.

[257] S. Khairy, P. Balaprakash, L. X. Cai, and Y. Cheng, "Constrained deep reinforcement learning for energy sustainable multi-UAV based random access IoT networks with NOMA," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1101–1115, 2021.

[258] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A gentle introduction to reinforcement learning and its application in different fields," *IEEE Access*, vol. 8, pp. 209320–209344, 2020.

[259] F. Zhang, Q. Yang, and D. An, "CDDPG: a deep-reinforcement-learning-based approach for electric vehicle charging control," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3075–3087, 2021.

[260] W. Wu, F. Zhu, Y. Fu, and Q. Liu, "Deep deterministic policy gradient with clustered prioritized sampling," *Neural Information Processing*, vol. 11302, pp. 645–654, 2018.

[261] C. Cho, S. Shin, H. Jeon, and S. Yoon, "QoS-aware workload distribution in hierarchical edge clouds: a reinforcement learning approach," *IEEE Access*, vol. 8, pp. 193297–193313, 2020.

[262] A. Musaddiq, Z. Nain, Y. Ahmad Qadri, R. Ali, and S. W. Kim, "Reinforcement learning-enabled cross-layer optimization for low-power and lossy networks under heterogeneous traffic patterns," *Sensors*, vol. 20, no. 15, pp. 1–25, 2020.

[263] R. Huang, V. W. S. Wong, and R. Schober, "Throughput optimization for grant-free multiple access with multiagent deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 228–242, 2021.

[264] T. Yu and B. Zhou, "Reinforcement learning based CPS self-tuning control methodology for interconnected power systems," *Dianli Xitong Baohu yu Kongzhi/Power System Protection and Control*, vol. 37, no. 10, pp. 33–38, 2009.

[265] T. Yu and Y. Yuan, "An average reward model based whole process R (λ)-learning for optimal CPS control," *Dianli Xitong Zidonghua/Automation of Electric Power Systems*, vol. 34, no. 21, pp. 27–33, 2010.

[266] T. Liu, B. Tian, Y. Ai, and F.-Y. Wang, "Parallel reinforcement learning-based energy efficiency improvement

for a cyber-physical system," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 617–626, 2020.

[267] M. Kiermeier, S. Feld, T. Phan, and C. Linnhoff-Popien, "Anomaly detection in spatial layer models of autonomous agents," *Intelligent Data Engineering and Automated Learning - IDEAL 2018*, vol. 11314, pp. 156–163, 2018.

[268] S. Jeong, G. Yoo, M. Yoo, I. Yeom, and H. Woo, "Resource-efficient sensor data management for autonomous systems using deep reinforcement learning," *Sensors*, vol. 19, no. 20, 2019.

[269] M. Kadohisa, J. V. Verhagen, and E. T. Rolls, "The primate amygdala: neuronal representations of the viscosity, fat texture, temperature, grittiness and taste of foods," *Neuroscience*, vol. 132, no. 1, pp. 33–48, 2005.

[270] C. Schwenck, A. Ciaramidaro, M. Selivanova, J. Tournay, C. M. Freitag, and M. Siniatchkin, "Neural correlates of affective empathy and reinforcement learning in boys with conduct problems: fMRI evidence from a gambling task," *Behavioural Brain Research*, vol. 320, pp. 75–84, 2017.

[271] M. Li, W.-j. Liu, B. Lu, Y.-h. Wang, and J.-g. Liu, "Differential expression of Arc in the mesocorticolimbic system is involved in drug and natural rewarding behavior in rats," *Acta Pharmacologica Sinica*, vol. 34, no. 8, pp. 1013–1024, 2013.

[272] C. Wang, J. Wang, J. Wang, and X. Zhang, "Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6180–6190, 2020.

[273] H.-Y. Kim and J. Kim, "A load balancing scheme for gaming server applying reinforcement learning in IOT," *Computer Science and Information Systems*, vol. 17, no. 3, pp. 891–906, 2020.

[274] A. H. Keyhanipour, B. Moshiri, M. Rahgozar, F. Oroumchian, and A. A. Ansari, "Integration of data fusion and reinforcement learning techniques for the rank-aggregation problem," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 6, pp. 1131–1145, 2016.

[275] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1930–1941, 2019.

[276] M. Min, X. Wan, L. Xiao et al., "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4307–4316, 2019.

[277] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered Public Safety IoT: a UAV-assisted adaptive-learning approach towards energy efficiency," *Journal of Network and Computer Applications*, vol. 123, pp. 69–79, 2018.

[278] Y. Cui, D. Zhang, T. Zhang, L. Chen, M. Piao, and H. Zhu, "Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices," *AEU - International Journal of Electronics and Communications*, p. 118, 2020.

[279] K. Gai, K. Xu, Z. Lu, M. Qiu, and L. Zhu, "Fusion of cognitive wireless networks and edge computing," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 69–75, 2019.

[280] S. Spanò, G. C. Cardarilli, L. Di Nunzio et al., "An efficient hardware implementation of reinforcement learning: the q-learning algorithm," *IEEE Access*, vol. 7, pp. 186340–186351, 2019.

[281] M. S. Munir, S. F. Abedin, N. H. Tran, and C. S. Hong, "When edge computing meets microgrid: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7360–7374, 2019.

[282] N. Shoeibi and N. Shoeibi, "Future of smart parking: automated valet parking using deep Q-learning," *Advances in Intelligent Systems and Computing*, vol. 1004, pp. 177–182, 2020.

[283] Z. Wang, Y. Liu, Z. Ma, X. Liu, and J. Ma, "LiPSG: lightweight privacy-preserving Q-learning-based energy management for the IoT-enabled smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3935–3947, 2020.

[284] M. Ozturk, M. Jaber, and M. A. Imran, "Energy-aware smart connectivity for IoT networks: enabling smart ports," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[285] H. Lu, X. He, M. Du, X. Ruan, Y. Sun, and K. Wang, "Edge QoE: computation offloading with deep reinforcement learning for internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9255–9265, 2020.

[286] D. Ma, G. Lan, M. Hassan, W. Hu, and S. K. Das, "Sensing, computing, and communications for energy harvesting IoTs: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1222–1250, 2020.

[287] R. Bonnefoi, C. Moy, and J. Palicot, "Improvement of the LPWAN AMI backhaul's latency thanks to reinforcement learning algorithms," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–18, 2018.

[288] X. Bao, H. Liang, and L. Han, "Transmission optimization of social and physical sensor nodes via collaborative beamforming in cyber-physical-social systems," *Sensors*, vol. 18, no. 12, 2018.

[289] M. Han, J. Duan, S. Khairy, and L. X. Cai, "Enabling sustainable underwater IoT networks with energy harvesting: a decentralized reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9953–9964, 2020.

[290] T. Mohammed, A. Albeshri, I. Katib, and R. Mehmood, "UbiPriSEQ—deep reinforcement learning to manage privacy, security, energy, and QoS in 5G IoT hetnets," *Applied Sciences*, vol. 10, no. 20, pp. 1–18, 2020.

[291] J. Tang, H. Tang, X. Zhang et al., "Energy minimization in d2d-assisted cache-enabled internet of things: a deep reinforcement learning approach," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5412–5423, 2020.

[292] G. Maselli, M. Piva, and J. A. Stankovic, "Adaptive communication for battery-free devices in smart homes," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6977–6988, 2019.

[293] H. Ke, J. Wang, H. Wang, and Y. Ge, "Joint optimization of data offloading and resource allocation with renewable energy aware for IoT devices: a deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 179349–179363, 2019.

[294] Y. Xie, Z. Xu, J. Xu, S. Gong, and Y. Wang, "Backscatter-Aided hybrid data offloading for mobile edge computing via deep reinforcement learning," *LNICST*, vol. 294, 2019.

[295] Y. Rioual, J. Laurent, and J.-P. Diguet, "Reinforcement-learning approach guidelines for energy management," *Journal of Low Power Electronics*, vol. 15, no. 3, pp. 283–293, 2019.

[296] C. Han, A. Liu, H. Wang, L. Huo, and X. Liang, "Dynamic anti-jamming coalition for satellite-enabled army IoT: a distributed game approach," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10932–10944, 2020.

[297] F. Jiang, K. Wang, L. Dong, C. Pan, and K. Yang, "Stacked autoencoder-based deep reinforcement learning for online resource scheduling in large-scale MEC networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9278–9290, 2020.

[298] X. Tu, C. Xu, S. Liu et al., "Efficient monocular depth estimation for edge devices in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2821–2832, 2021.

[299] J. Long, Y. Luo, X. Zhu, E. Luo, and M. Huang, "Computation offloading through mobile vehicles in IoT-edge-cloud network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.

[300] Y. Akbari and S. Tabatabaei, "A new method to find a high reliable route in IoT by using reinforcement learning and fuzzy logic," *Wireless Personal Communications*, vol. 112, no. 2, pp. 967–983, 2020.

[301] Y. Li, X. Zhao, and H. Liang, "Throughput maximization by deep reinforcement learning with energy cooperation for renewable ultradense IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9091–9102, 2020.

[302] J. Zheng, L. Gao, H. Wang et al., "Smart edge caching-aided partial opportunistic interference alignment in HetNets," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1842–1850, 2020.

[303] M. Peng, S. Garg, X. Wang, A. Bradai, H. Lin, and M. S. Hossain, "Learning-based IoT data aggregation for disaster scenarios," *IEEE Access*, vol. 8, pp. 128490–128497, 2020.

[304] S. Sarwar, R. Sirhindi, L. Aslam, G. Mustafa, M. M. Yousaf, and S. W. U. Q. Jaffry, "Reinforcement learning based adaptive duty cycling in LR-WPANs," *IEEE Access*, vol. 8, pp. 161157–161174, 2020.

[305] K. Wang, C. M. Chen, M. S. Hossain, G. Muhammad, S. Kumar, and S. Kumari, "Transfer reinforcement learning-based road object detection in next generation IoT domain," *Computer Networks*, p. 193, 2021.

[306] M. I. Khan, L. Reggiani, M. M. Alam et al., "Q-learning based joint energy-spectral efficiency optimization in multi-hop device-to-device communication," *Sensors*, vol. 20, no. 22, pp. 1–23, 2020.

[307] S. F. Abedin, M. S. Munir, N. H. Tran, Z. Han, and C. S. Hong, "Data freshness and energy-efficient UAV navigation optimization: a deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5994–6006, 2020.

[308] G. Kaur, P. Chanak, and M. Bhattacharya, "Energy efficient intelligent routing scheme for IoT-enabled WSNs," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11440–11449, 2021.

[309] Z. Xiong, Y. Zhang, W. Y. B. Lim et al., "UAV-assisted wireless energy and data transfer with deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 85–99, 2021.

[310] Y. Nie, J. Zhao, J. Liu, J. Jiang, and R. Ding, "Energy-efficient UAV trajectory design for backscatter communication: a deep reinforcement learning approach," *China Communications*, vol. 17, no. 10, pp. 129–141, 2020.

[311] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[312] F. M. Al-Turjman, "Information-centric sensor networks for cognitive IoT: an overview," *Annales des Telecommunications/Annals of Telecommunications*, vol. 72, no. 1-2, pp. 3–18, 2017.

[313] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11158–11168, 2019.

[314] H. Khelifi, S. Luo, B. Nour et al., "Bringing deep learning at the edge of information-centric internet of things," *IEEE Communications Letters*, vol. 23, no. 1, pp. 52–55, 2019.

[315] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the internet of things: a comprehensive survey," *Ad Hoc Networks*, vol. 93, 2019.

[316] R. M. Sandoval, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "Optimizing and updating lora communication parameters: a machine learning approach," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 884–895, 2019.

[317] H. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial intelligence enabled internet of things: network architecture and spectrum access," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 44–51, 2020.

[318] A. Foerster, A. Udugama, C. Görg, K. Kuladinithi, A. Timm-Giel, and A. Cama-Pinto, "A novel data dissemination model for organic data flows," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 158, pp. 239–252, 2015.

[319] S. Shukla, M. F. Hassan, M. K. Khan, L. T. Jung, and A. Awang, "An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment," *PLoS ONE*, vol. 14, no. 11, Article ID e0224934, 2019.

[320] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the IoT systems with blockchain-as-a-service and UAV-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1974–1993, 2020.

[321] R. Bajracharya, R. Shrestha, and S. W. Kim, "Q-learning based fair and efficient coexistence of LTE in unlicensed band," *Sensors*, vol. 19, no. 13, 2019.

[322] X. Guo, H. Lin, Z. Li, and M. Peng, "Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6242–6251, 2020.

[323] Z. Cao, P. Zhou, R. Li, S. Huang, and D. Wu, "Multiagent deep reinforcement learning for joint multichannel access and task offloading of mobile-edge computing in industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6201–6213, 2020.

[324] M. McClellan, C. Cervelló-Pastor, and S. Sallent, "Deep learning at the mobile edge: opportunities for 5G networks," *Applied Sciences*, vol. 10, no. 14, 2020.

[325] C. Wu, Z. Liu, F. Liu, T. Yoshinaga, Y. Ji, and J. Li, "Collaborative learning of communication routes in edge-enabled multi-access vehicular environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1155–1165, 2020.

[326] A. Serhani, N. Naja, and A. Jamali, "AQ-Routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET-IoT systems," *Cluster Computing*, vol. 23, no. 1, pp. 13–27, 2020.

[327] F. Jameel, U. Javaid, W. U. Khan, M. N. Aman, H. Pervaiz, and R. Jäntti, "Reinforcement learning in blockchain-enabled IIoT networks: a survey of recent advances and open challenges," *Sustainability*, vol. 12, no. 12, 2020.

[328] J. M. C. Neto, S. F. G. Neto, P. M. de Santana, and V. A. de Sousa, "Multi-cell LTE-U/Wi-Fi coexistence evaluation using a reinforcement learning framework," *Sensors*, vol. 20, no. 7, 2020.

[329] W. Ejaz, M. Basharat, S. Saadat, A. M. Khattak, M. Naeem, and A. Anpalagan, "Learning paradigms for communication and computing technologies in IoT systems," *Computer Communications*, vol. 153, pp. 11–25, 2020.

[330] J. Tang, J. Song, J. Ou, J. Luo, X. Zhang, and K.-K. Wong, "Minimum throughput maximization for multi-UAV enabled WPCN: a deep reinforcement learning method," *IEEE Access*, vol. 8, pp. 9124–9132, 2020.

[331] A. Abane, M. Daoui, S. Bouzefrane, and P. Muhlethaler, "A lightweight forwarding strategy for named data networking in low-end IoT," *Journal of Network and Computer Applications*, p. 148, 2019.

[332] R. Ali, Y. B. Zikria, B.-S. Kim, and S. W. Kim, "Deep reinforcement learning paradigm for dense wireless networks in smart cities," *EAI/Springer Innovations in Communication and Computing, Springer, Cham, Switzerland*, pp. 43–70, 2020.

[333] Q. Zhang, Y.-C. Liang, and H. V. Poor, "Intelligent user association for symbiotic radio networks using deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4535–4548, 2020.

[334] R. M. Sandoval, S. Canovas-Carrasco, A.-J. Garcia-Sanchez, and J. Garcia-Haro, "A reinforcement learning-based framework for the exploitation of multiple rats in the iot," *IEEE Access*, vol. 7, pp. 123341–123354, 2019.

[335] C. Sun, H. Ding, and X. Liu, "Multichannel spectrum access based on reinforcement learning in cognitive internet of things," *Ad Hoc Networks*, vol. 106, 2020.

[336] G. L. Santos, P. T. Endo, D. Sadok, and J. Kelner, "When 5G meets deep learning: a systematic review," *Algorithms*, vol. 13, no. 9, 2020.

[337] N. Chen, T. Qiu, C. Mu, M. Han, and P. Zhou, "Deep actor-critic learning-based robustness enhancement of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6191–6200, 2020.

[338] Y. Zhang, B. Feng, W. Quan et al., "Cooperative edge caching: a multi-agent deep learning based approach," *IEEE Access*, vol. 8, pp. 133212–133224, 2020.

[339] Y. Hao, M. Li, D. Wu, M. Chen, M. M. Hassan, and G. Fortino, "Human-like hybrid caching in software-defined edge cloud," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5806–5815, 2020.

[340] F. Dou, J. Lu, T. Xu, C.-H. Huang, and J. Bi, "A bisection reinforcement learning approach to 3-D indoor localization," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6519–6535, 2021.

[341] T.-W. Ban, "An autonomous transmission scheme using dueling DQN for D2D communication networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16348–16352, 2020.

[342] K. S. Shin, G. H. Hwang, and O. Jo, "Distributed reinforcement learning scheme for environmentally adaptive IoT network selection," *Electronics Letters*, vol. 56, no. 9, pp. 441–444, 2020.

[343] N. Garg, M. Sellathurai, V. Bhatia, and T. Ratnarajah, "Function approximation based reinforcement learning for edge caching in massive MIMO networks," *IEEE Transactions on Communications*, vol. 69, no. 4, pp. 2304–2316, 2021.

[344] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.

[345] T. S. P. Kumar and P. V. Krishna, "Power modelling of sensors for IoT using reinforcement learning," *International Journal of Advanced Intelligence Paradigms*, vol. 10, no. 1-2, pp. 3–22, 2018.

[346] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, and N. El-Sheimy, "Deep reinforcement learning (DRL): another perspective for unsupervised wireless localization," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6279–6287, 2020.

[347] M. Li, C. Chen, C. Hua, and X. Guan, "Intelligent latency-aware virtual network embedding for industrial wireless networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7484–7496, 2019.

[348] G. M. Dias, C. B. Margi, F. C. P. De Oliveira, and B. Bellalta, "Cloud-empowered, self-managing wireless sensor networks: interconnecting management operations at the application layer," *IEEE Consumer Electronics Magazine*, vol. 8, no. 1, pp. 55–60, 2019.

[349] C. Yang, K.-W. Chin, T. He, and Y. Liu, "On sampling time maximization in wireless powered internet of things," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 3, pp. 641–650, 2019.

[350] A. Kawewong, Y. Honda, M. Tsuboyama, and O. Hasegawa, "A common-neural-pattern based reasoning for mobile robot cognitive mapping," *Advances in Neuro-Information Processing*, vol. 5506, pp. 32–39, 2009.

[351] C. Moy, L. Besson, G. Delbarre, and L. Toutain, "Decentralized spectrum learning for radio collision mitigation in ultra-dense IoT networks: LoRaWAN case study and experiments," *Annales des Telecommunications/Annals of Telecommunications*, vol. 75, no. 11-12, pp. 711–727, 2020.

[352] H. Zhu, Y. Cao, X. Wei, W. Wang, T. Jiang, and S. Jin, "Caching transient data for internet of things: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2074–2083, 2019.

[353] Z. Li, Y. Lu, Y. Shi, Z. Wang, W. Qiao, and Y. Liu, "A Dyna-Q-based solution for UAV networks against smart jamming attacks," *Symmetry*, vol. 11, no. 5, 2019.

[354] J. Luo, S. Green, P. Feghali, G. Legrady, and C. K. Koç, *Reinforcement Learning and Trustworthy Autonomy*, Springer International Publishing, New York, NY, USA, 2018.

[355] Y. Liu, W. Zhang, S. Pan, Y. Li, and Y. Chen, "Analyzing the robotic behavior in a smart city with deep enforcement and imitation learning using IoRT," *Computer Communications*, vol. 150, pp. 346–356, 2020.

[356] K. Watanabe and S. Inada, "Search algorithm of the assembly sequence of products by using past learning results," *International Journal of Production Economics*, p. 226, 2020.

[357] K. Baek and I.-Y. Ko, "Effect-driven selection of web of things services in cyber-physical systems using reinforcement learning," *Lecture Notes in Computer Science*, vol. 11496, pp. 554–559, 2019.

[358] I. Verner, M. Reitman, D. Cuperman, T. Yan, E. Finkelstein, and T. Romm, "Exposing robot learning to students in augmented reality experience," *Smart Industry & Smart Education*, vol. 47, pp. 610–619, 2019.

[359] Y.-T. Tsai, C.-H. Lee, T.-Y. Liu et al., "Utilization of a reinforcement learning algorithm for the accurate alignment of a robotic arm in a complete soft fabric shoe tongues automation process," *Journal of Manufacturing Systems*, vol. 56, pp. 501–513, 2020.

[360] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver, "Review of tool condition monitoring in machining and opportunities for deep learning," *International Journal of*

Advanced Manufacturing Technology, vol. 109, no. 3-4, pp. 953–974, 2020.

[361] D. B. Noureddine, M. Krichen, S. Mechti, T. Nahhal, and W. Y. H. Adoni, "An agent-based architecture using deep reinforcement learning for the intelligent internet of things applications," *Advances on Smart and Soft Computing*, vol. 1188, pp. 273–283, 2021.

[362] R. S. Alonso, "Deep tech and artificial intelligence for worker safety in robotic manufacturing environments," *Distributed Computing and Artificial Intelligence, Special Sessions, 17th International Conference*, vol. 1242, pp. 234–240, 2021.

[363] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised deep reinforcement learning in support of IoT and smart city services," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624–635, 2018.

[364] X. Zhang, L. Yao, S. Zhang, S. Kanhere, M. Sheng, and Y. Liu, "Internet of things meets brain-computer interface: a unified deep learning framework for enabling human-thing cognitive interactivity," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2084–2092, 2019.

[365] O. A. Sianaki, A. Yousefi, A. R. Tabesh, and M. Mahdavi, "Machine learning applications: the past and current research trend in diverse industries," *Inventions*, vol. 4, no. 1, 2019.

[366] G. Neelakantam, D. D. Onthoni, and P. K. Sahoo, "Reinforcement learning based passengers assistance system for crowded public transportation in fog enabled smart city," *Electronics (Switzerland)*, vol. 9, no. 9, pp. 1–19, 2020.

[367] M. Rivas and F. Giorno, "A reinforcement learning multi-agent architecture prototype for smart homes (IoT)," *Proceedings of the Future Technologies Conference (FTC) 2018*, vol. 880, pp. 159–170, 2019.

[368] N. Magaia, R. Fonseca, K. Muhammad, A. H. F. N. Segundo, A. V. Lira Neto, and V. H. C. De Albuquerque, "Industrial internet-of-things security enhanced with deep learning approaches for smart cities," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6393–6405, 2021.

[369] S. Pan, P. Li, D. Zeng, S. Guo, and G. Hu, "A ${Q}$ -learning based framework for congested link identification," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9668–9678, 2019.

[370] H. Lin, Z. Chen, and L. Wang, "Offloading for edge computing in low power wide area networks with energy harvesting," *IEEE Access*, vol. 7, pp. 78919–78929, 2019.

[371] D. K. Sharma, J. J. P. C. Rodrigues, V. Vashishth, A. Khanna, and A. Chhabra, "RLProph: a dynamic programming based reinforcement learning approach for optimal routing in opportunistic IoT networks," *Wireless Networks*, vol. 26, no. 6, pp. 4319–4338, 2020.

[372] B. Song, J. Song, and J. Ye, "A dynamic pricing mechanism in IoT for DaaS: a reinforcement learning approach," *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, vol. 1075, pp. 604–615, 2020.

[373] D. Wang, X. Tian, H. Cui, and Z. Liu, "Reinforcement learning-based joint task offloading and migration schemes optimization in mobility-aware MEC network," *China Communications*, vol. 17, no. 8, pp. 31–44, 2020.

[374] W. Shafik, S. Mojtaba Matinkhah, P. Etemadinejad, and M. N. Sanda, "Reinforcement learning rebirth, techniques, challenges, and resolutions," *International Journal on Informatics Visualization*, vol. 4, no. 3, pp. 127–135, 2020.

[375] E. Erdemir, P. L. Dragotti, and D. Gunduz, "Privacy-aware time-series data sharing with deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 389–401, 2021.

[376] P. Wang, L. T. Yang, J. Li, X. Li, and X. Zhou, "MMDP: a mobile-IoT based multi-modal reinforcement learning service framework," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 675–684, 2020.

[377] W. Jiang, G. Feng, S. Qin, and Y. Liu, "Multi-agent reinforcement learning based cooperative content caching for mobile edge networks," *IEEE Access*, vol. 7, pp. 61856–61867, 2019.

[378] J. Ma, S. Hasegawa, S. J. Kim, and M. Hasegawa, "A reinforcement-learning-based distributed resource selection algorithm for massive IoT," *Applied Sciences*, vol. 9, no. 18, 2019.

[379] Y. Qian, L. Shi, J. Li et al., "A workflow-aided internet of things paradigm with intelligent edge computing," *IEEE Network*, vol. 34, no. 6, pp. 92–99, 2020.

[380] Z. Shi, Y. Zeng, and Z. Wu, "Service chain orchestration based on deep reinforcement learning in intent-based IoT," *Proceedings of the 9th International Conference on Computer Engineering and Networks*, vol. 1143, pp. 875–882, 2021.

[381] W.-C. Chien, H.-Y. Weng, and C.-F. Lai, "Q-learning based collaborative cache allocation in mobile edge computing," *Future Generation Computer Systems*, vol. 102, pp. 603–610, 2020.

[382] V. Vijayaraghavan and J. R. Leevinson, *Intelligent Traffic Management Systems for Next Generation IoV in Smart City Scenario*, Springer International Publishing, New York, NY, USA, 2020.

[383] T. Lee, O. Jo, and K. Shin, "CoRL: collaborative reinforcement learning-based MAC protocol for IoT networks," *Electronics (Switzerland)*, vol. 9, no. 1, 2020.

[384] C. Kim, "Deep reinforcement learning by balancing offline Monte Carlo and online temporal difference use based on environment experiences," *Symmetry*, vol. 12, no. 10, pp. 1–16, 2020.

[385] S. Misra, P. K. Deb, N. Koppala, A. Mukherjee, and S. Mao, "S-nav: safety-aware IoT navigation tool for avoiding COVID-19 hotspots," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6975–6982, 2021.

[386] D. N. Doan, D. Zaharie, and D. Petcu, "Auto-scaling for a streaming architecture with fuzzy deep reinforcement learning," *Lecture Notes in Computer Science*, vol. 11997, 2020.

[387] Y. Liu, H. Wang, M. Peng, J. Guan, and Y. Wang, "An incentive mechanism for privacy-preserving crowdsensing via deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8616–8631, 2021.

[388] H. Guo, S. Li, B. Li, Y. Ma, and X. Ren, "A new learning automata-based pruning method to train deep neural networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3263–3269, 2018.

[389] Y. Wang, X. Chen, L. Wang, and G. Min, "Effective IoT-facilitated storm surge flood modeling based on deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6338–6347, 2020.

[390] J. Yun, Y. Goh, and J.-M. Chung, "DQN-based optimization framework for secure sharded blockchain systems," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 708–722, 2021.

[391] K. E. Mwangi, S. Masupe, and J. Mandu, "Modelling malware propagation on the internet of things using an agent-based approach on complex networks," *Jordanian Journal of Computers and Information Technology*, vol. 6, no. 1, pp. 26–40, 2020.

[392] X. He, K. Wang, and W. Xu, "QoE-Driven content-centric caching with deep reinforcement learning in edge-enabled

IoT," *IEEE Computational Intelligence Magazine*, vol. 14, no. 4, pp. 12–20, 2019.

[393] T. G. Nguyen, T. V. Phan, D. T. Hoang, T. N. Nguyen, and C. So-In, "Efficient SDN-based traffic monitoring in IoT networks with double deep Q-network," *Lecture Notes in Computer Science*, vol. 12575, 2020 LNCS:26–38.

[394] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3602–3609, 2019.

[395] B. Banerjee and L. Kraemer, "Action discovery for single and multi-agent reinforcement learning," *Advances in Complex Systems*, vol. 14, no. 2, pp. 279–305, 2011.

[396] Y. Li, F. Qi, Z. Wang, X. Yu, and S. Shao, "Distributed edge computing offloading algorithm based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 85204–85215, 2020.

[397] L. Zhou, Q. Liu, F. Wu, and Y. Wei, *Deep Learning Based Sensing Resource Allocation for Mobile Target Tracking*, pp. 430–435, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ, USA, 2020.

[398] P. Zhang, Y. Yuan, Z. Wang, and C. Sun, *A Hierarchical Game Approach to the Coupled Resilient Control of CPS against Denial-Of-Service Attack*, pp. 15–20, IEEE Computer Society, Washington, DC, USA, 2019.

Hindawi
Complexity
Volume 2021, Article ID 7756299, 24 pages
https://doi.org/10.1155/2021/7756299

Research Article

A Hybrid Model for Short-Term Traffic Flow Prediction Based on Variational Mode Decomposition, Wavelet Threshold Denoising, and Long Short-Term Memory Neural Network

Yang Yu, Qiang Shang, and Tian Xie

School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo, Shandong 255049, China

Correspondence should be addressed to Qiang Shang; shangqiangv587@163.com

Received 7 September 2021; Revised 23 October 2021; Accepted 2 November 2021; Published 23 November 2021

Academic Editor: Murari Andrea

Traffic flow prediction plays an important role in intelligent transportation system (ITS). However, due to the randomness and complex periodicity of traffic flow data, traditional prediction models often fail to achieve good results. On the other hand, external disturbances or abnormal detectors will cause the collected traffic flow data to contain noise components, resulting in a decrease in prediction accuracy. In order to improve the accuracy of traffic flow prediction, this study proposes a mixed traffic flow prediction model VMD-WD-LSTM using variational mode decomposition (VMD), wavelet threshold denoising (WD), and long short-term memory (LSTM) network. Firstly, we decompose the original traffic flow sequence into K components through VMD and determine the number of components K according to the sample entropy of different K values. Then, each component is denoised by wavelet threshold to obtain the denoised subsequence. Finally, LSTM is used to predict each subsequence, and the predicted values of each subsequence are combined into the final prediction results. In addition, the performance of the proposed model and the latest traffic flow prediction model is compared on the several well-known public datasets. The empirical analysis shows that the proposed model not only has good prediction accuracy but also has superior robustness.

1. Introduction

With the rapid development of cities and the rapid increase of urban population, the number of vehicles on urban roads is also increasing. Therefore, the increased traffic pressure on urban roads has caused more and more serious problems, such as traffic accidents and traffic pollution, and road congestion has become an important factor affecting the quality of daily life of residents. Faced with this situation, the development and application of ITS has been recognized as an effective way to solve or alleviate traffic problems. Therefore, on the basis of obtaining accurate future traffic data through historical data, the intelligent transportation system can perceive future traffic conditions and traffic conditions of each section. Then, the system can formulate effective traffic organization and guidance strategies to reduce the probability of road congestion, so as to achieve the purpose of improving road traffic efficiency [1]. However, due to the complexity of road traffic or the environment, different unexpected situations often occur, resulting in the traffic flow data measured by the detector to be interfered, which will affect the regularity of daily traffic flow and thus affect the traffic flow data. Data fluctuations caused by such interference factors are called noise.

Wavelet denoising is a commonly used denoising method in the field of traffic flow prediction. The Kalman filter model based on wavelet decomposition has been used for short-term traffic flow prediction. The empirical results show that the combination of wavelet decomposition and Kalman filter can reduce the impact of noise on prediction to a certain extent [2]. Peng and Xiang proposed a traffic flow prediction method based on phase space reconstruction and wavelet denoising, in which wavelet denoising was used to preprocess the original traffic flow data [3]. In the fuzzy neural network prediction model proposed by Xiao et al., wavelet decomposition was used to smooth historical traffic flow data, and the results show that wavelet denoising can significantly improve the prediction accuracy [4]. Tang et al.

compared the denoising performance of four wavelet functions, coif (coiflet), db (daubechies), haar, and sym (symlet), on the original traffic flow data. The analysis results show that the db wavelet function has the best denoising performance [5].

In order to further improve the denoising performance, an empirical mode decomposition (EMD) denoising method was proposed that has been widely used so far [6]. EMD decomposes a complex signal into a finite number of intrinsic mode functions (IMFs), and each of the decomposed IMF components contains local characteristic signals of different time scales of the original signal. The high-frequency IMF component contains noise, and the low-frequency IMF component contains the characteristics of the original signal, that is, denoising is achieved by processing high-frequency signal. EMD has the advantages of being simple, intuitive, and efficient, but the disadvantage is that it is prone to modal aliasing. In order to make up for the disadvantages of EMD, integrated empirical mode decomposition (EEMD) was proposed [7]. Because EEMD introduces white noise on the basis of EMD to supplement the missing scale, the phenomenon of modal aliasing can be overcome to a certain extent. In 2020, Chen et al. proposed a traffic flow prediction model called EEMD-ANN using EEMD and artificial neural network (ANN) [8]. In 2021, Chen et al. compared the performance of EMD, EEMD, and wavelet in traffic flow data denoising, and the results showed that EEMD has the best performance [9].

Variational mode decomposition (VMD) is a signal processing method proposed in recent years [10]. Different from the principle of EMD, VMD uses completely nonrecursive modal variation to process the signal, and it determines the optimal center frequency and bandwidth of the component by solving the constrained variational problem, so it basically overcomes the end effect and modal aliasing of EMD. At present, VMD has been applied in many fields and achieved good results. Liu et al. proposed a wind speed prediction model using VMD and singular spectrum analysis (SSA) [10]. In this model, the original data were decomposed by VMD, and then SSA was used to extract the low-frequency components of the decomposed data for prediction. In [11], VMD was used to process the original streamflow data, and then LSTM was employed to predict the streamflow [12]. The comparison result illustrated that performance of VMD is better than that of EEMD and discrete wavelet transform (DWT). Shi et al. proposed a hybrid prediction model for network traffic based on VMD and extreme learning machine (ELM) [13], and empirical analysis results showed that VMD denoising can effectively improve prediction accuracy. Due to the good performance of VMD in other prediction fields, we have reason to believe that VMD also has great potential in traffic prediction.

After the original traffic flow data are denoised, the selection of the prediction model is very important. In order to improve the prediction accuracy, a large number of models with different data characteristics and calculation processes have been proposed for traffic flow prediction, including traditional statistical models, such as autoregressive integrated moving average (ARIMA) model [14, 15] and Kalman filter model (Kalman filter) [16, 17], and machine learning-based models, such as support vector machine (SVM) [18–20] and artificial neural network (ANN) [21, 22]. In recent years, deep learning has attracted much attention in traffic flow prediction because of its superior performance. As a variant of recurrent neural network (RNN), LSTM improves the shortcomings of gradient disappearance and gradient explosion. At present, LSTM is widely used in many prediction fields, including traffic flow prediction. Tian et al. proposed a traffic flow prediction model based on LSTM, and empirical analysis showed that the prediction accuracy of LSTM is higher than that of SVM and feedforward neural network (FFNN) [23].

In [24], convolutional neural network (CNN) was first used to extract daily features of traffic flow, and then LSTM was used to predict traffic flow. Ma et al. pointed out that bidirectional long short-term memory (BiLSTM) is more effective in short-term traffic flow prediction [25]. The empirical results in [26] show that the performance of LSTM for traffic speed prediction is better than other comparative parametric and nonparametric methods. In [27], the attention mechanism was introduced in LSTM to improve the accuracy of the model predicting traffic speed, which can properly assign weights to distinguish the importance of traffic speed time sequences. In view of the excellent performance of LSTM in traffic flow prediction, LSTM is selected as the prediction model and its parameters are optimized.

Many related studies in the field of transportation show that the subsequence obtained by signal decomposition of the original measured signal data is more conducive to showing the irregular periodic variation characteristics of the signal than the original data. In the study of predicting the missing measurement signal data of SHM systems, Li et al. [28] decomposed the original signal data into multiple subsequences by the empirical mode decomposition (EMD) method and then used ARIMA, ANN, LSTM, and SVR models to predict different subsequences. The final prediction results show that the prediction performance of the hybrid model after signal decomposition is better than that of the original data directly, which proves the superiority of signal decomposition in the field of traffic data prediction. In 2021, Huang et al. [29] used EMD to extract the intrinsic mode function (IMF) in order to make full use of the time characteristics of traffic flow. The original traffic flow data were decomposed into three components according to their own characteristics: trend component, residual component, and residual component. These three components were analyzed and predicted, respectively, and the accuracy of the prediction results was higher than that of the single method for direct prediction of the original data. This study shows that the decomposed signals are more likely to show characteristics through the prediction model.

In addition, Li et al. [30] used the ensemble empirical mode decomposition method for travel time prediction in 2018. They first decomposed the original travel time data series into multiple functions with different characteristics through the ensemble empirical mode decomposition method and then expressed these functions with the random

vector function chain network. Finally, the output results of different networks were combined to obtain the final prediction results. The results show that the effect of ensemble empirical mode decomposition is better than that of empirical mode decomposition. The above studies show that the signal decomposition method for original data can improve the prediction performance of the model to a certain extent if it can overcome the mode mixing phenomenon in the empirical mode decomposition.

To sum up, traffic flow prediction mainly faces the following two problems. One is how to reduce the influence of noise contained in the original traffic flow data on the prediction results. The other is how to accurately show the irregular periodic variation characteristics of traffic flow data. In view of the obvious improvement of traffic flow data prediction performance by the data denoising method and signal decomposition method, this paper proposes a WD-VMD-LSTM hybrid model for traffic flow prediction. This method first decomposes the complex original traffic flow data into multiple subsequences containing more prominent features through the variational mode decomposition method and then performs wavelet denoising on several subsequences, respectively. Finally, the long short-term memory network model is used to predict the denoised subsequences, respectively, and the final prediction results are obtained by combining the results of different subsequences. Compared with the empirical mode decomposition, the variational mode decomposition method can effectively avoid the phenomenon of mode mixing and boundary effect. The decomposed subsequence contains the data characteristics in the original signal. At the same time, the VMD method also has the advantages of anti-noise interference, so it is not easy to be affected by noise in the process of signal decomposition. On the other hand, the above research shows that the wavelet denoising method can effectively reduce the influence of noise on traffic flow prediction. The data characteristics contained in the original traffic flow data are difficult to identify, and the denoising processing will affect the characteristics of the original signal. Therefore, the wavelet denoising of the subsequence obtained by the variational mode decomposition can highlight the characteristics of the original signal and avoid the denoising method to suppress the useful signal. In addition, this paper compares LSTM with artificial neural network in the part of result discussion. The long short-term memory network model is more suitable for predicting complex time series data as a deep learning method.

The main contributions of this paper are summarized as follows:

(1) A denoising method combining variational mode decomposition and wavelet threshold denoising is proposed to process the original traffic flow data. At the same time, this paper compares the prediction performance of different prediction models before and after data processing.

(2) In order to avoid the phenomenon of modal aliasing and the increase of data complexity, the number of components K is determined according to the sample entropy with different K values.

(3) This study compares the denoising effects of different signal decomposition methods combined with the wavelet threshold denoising model. The advantage of variational mode decomposition in dealing with traffic flow data is discussed.

(4) Adam (its name is derived from adaptive moment estimation) optimizer is used to obtain a better model when training LSTM.

(5) In this study, two different public datasets are used to comprehensively compare different prediction models, and it is proved that the proposed model has better prediction performance than other comparison models.

The rest of this paper is organized as follows. The methods of WD, VMD, and LSTM are briefly introduced, and the process of the proposed VMD-WD-LSTM model is listed in Section 2. Section 3 demonstrates experiments where the prediction results of the proposed model and comparison models are evaluated. Finally, Section 4 provides conclusion of this research and makes the next research plan.

2. Methodology

2.1. Variational Mode Decomposition. Variational mode decomposition (VMD) is a method of signal processing using completely nonrecursive modal variation. Compared with traditional empirical mode decomposition (EMD), this technology can artificially determine the number of modal decompositions and then realize the frequency-domain decomposition and effective separation of IMF according to the best center frequency and limited bandwidth of each component after decomposition. In this way, the effective decomposition component of the target signal is obtained, and the optimal solution of the variational problem is realized. Variational mode decomposition has a solid theoretical foundation, and there is no end effect of traditional empirical mode decomposition and the problem of modal component aliasing. This method can reduce the complexity and nonstationarity of nonlinear time series and can decompose multiple stationary subsequences with different frequency scales.

The first step of variational mode decomposition is to construct the variational problem and solve the constrained variational problem:

$$\min_{\{u_k\}\{w_k\}} \left\{ \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-jw_k t} \right\|_2^2 \right\} \quad \text{s.t.} \quad \sum_k u_k(t) = f(t), \tag{1}$$

where $u_k(t)$ is input signal modal function, $\{u_k\}$ is the k-th modal component with limited bandwidth after decomposition, $\{w_k\}$ is the center frequency corresponding to the k-th modal component of the input signal, $\delta(t)$ is the Dirac function, $*$ represents the convolution operator, and $f(t)$ is the input signal.

Then, introduce the Lagrangian multiplication operator λ and the quadratic penalty factor α to rewrite formula (1) to transform the constrained variational problem into an unconstrained variational problem. The rewritten formula is as follows:

$$L(\{u_k\},\{w_k\},\lambda) = \alpha \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-jw_k t} \right\|_2^2 + \left\| f(t) - \sum_k u_k(t) \right\|_2^2 + \left[\lambda(t), f(t) - \sum_k u_k(t) \right]. \tag{2}$$

Use the alternating direction multiplication algorithm (ADMM) to solve equation (2) and obtain the optimal solution of the respective center frequencies of a group of modal components, that is, alternately update u_k, w_k, and λ to obtain the minimum point of the extended Lagrangian expression, and the formula is as follows:

$$\begin{aligned} \hat{u}_k^{n+1}(w) &= \frac{\hat{f}(w) - \sum_{i=1}^{k-1} \hat{u}_k^{n+1}(w) - \sum_{i=k+1}^{k} \hat{u}_k^n(w) + \left(\hat{\lambda}^n(w)/2 \right)}{1 + 2\alpha \left(w - w_k^n \right)^2}, \\ w_k^{n+1} &= \frac{\int_0^\infty w \left| \hat{u}_k^{n+1}(w) \right|^2 \mathrm{d}w}{\int_0^\infty \left| \hat{u}_k^{n+1}(w) \right|^2 \mathrm{d}w}, \\ \hat{\lambda}^{n+1}(w) &= \hat{\lambda}^n(w) + \tau \left(\hat{f}(w) - \sum_k \left| \hat{u}_k^{n+1}(w) \right| \right). \end{aligned} \tag{3}$$

After dividing the frequency band according to the characteristics of the original signal, continuously update the center frequency of each inherent modal component and the corresponding component, and finally, realize the adaptive decomposition of the target signal according to the constraints.

2.2. Wavelet Threshold Denoising. The noise in the original data is usually a high-frequency signal, and the useful data are regarded as a low-frequency signal. Wavelet decomposition decomposes the signal into approximate components containing low-frequency signals and detailed components containing high-frequency signals. The part containing low-frequency signals can be further decomposed, as shown in Figure 1. Figure 1 is a three-layer wavelet decomposition diagram, cA1, cA2, and cA3 represent the low-frequency signal part of the original signal, while cD1, cD2, and cD3 represent the high-frequency signal part of the original signal. The cD1, cD2, and cD3 contain noise. In this study, cD1, cD2, and cD3 are processed by wavelet threshold denoising, and then we reconstruct the signal by wavelet transform. Finally, the denoising results are obtained.

Wavelet threshold denoising uses the continuity characteristics of the original signal in the time series, and the wavelet coefficient of noise is smaller than the wavelet coefficient of the useful signal. Select an appropriate threshold through this feature, quantize the wavelet coefficients, and then reconstruct the wavelet coefficients to obtain the denoised data.

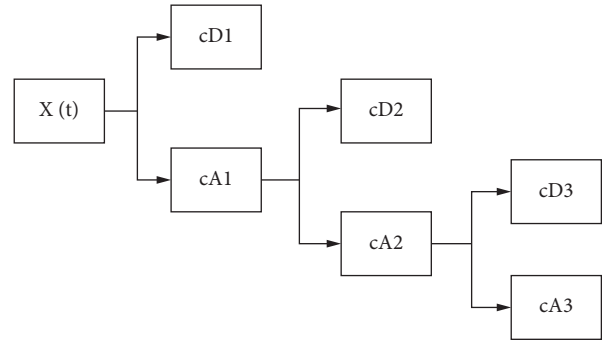


Figure 1: Wavelet decomposition process (cA is low-frequency information; cD is high-frequency information).

Wavelet threshold denoising can be divided into hard threshold denoising and soft threshold denoising in selecting threshold function. In terms of the effect of signal denoising, the signal after soft threshold denoising is smoother, but it is easy to remove some useful signals. The signal after hard threshold denoising will oscillate and there will be jumping points, but the error should be lower than the soft threshold. Denoising does not affect the degree of approximation between the denoised signal and the original signal. Therefore, from the perspective of ensuring the accuracy of the prediction result, this study chooses to perform hard threshold denoising. Hard threshold denoising is when the wavelet coefficient is greater than the threshold, it is determined to be generated by the signal, and it is retained after processing. When the wavelet coefficient is less than the threshold, it is determined to be noise generated and replaced with 0, as shown in the following equation:

$$w_\lambda = \begin{cases} w, & |w| \geq \lambda, \\ 0, & |w| < \lambda, \end{cases} \tag{4}$$

where w_λ represents the new wavelet coefficient and λ represents the set threshold.

2.3. LSTM Network. At present, the deep learning model has been widely used in the research of time series data. As a kind of neural network model, the deep learning model can extract the characteristics of the input signal and obtain the law of the complex signal. Among the deep learning models, the recurrent neural network (RNN) shows good adaptability when performing time series data analysis. Long short-term memory (LSTM) network is a variant of cyclic neural network, which improves the problems of gradient explosion and gradient disappearance in cyclic neural network and performs better in analyzing time series data.

The LSTM network is composed of an input layer, a hidden layer, and an output layer. Compared with the traditional RNN, the hidden layer of the LSTM is a unit with a unique memory mode. Figure 2 shows the hidden layer structure of the RNN and the hidden layer structure of the LSTM.

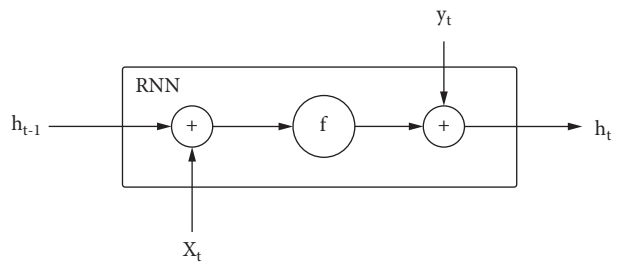


Figure 2: RNN architecture.

The memory unit is the core of the LSTM unit structure (see Figure 3). The memory unit at the current time t is marked as c_t. The memory unit can delete or add information through input gates, forget gates, and output gates. Specifically, the workflow of the LSTM unit is as follows:

(1) The LSTM unit receives the current state x_t, the hidden state h_{t-1} of the LSTM at the previous moment, and the state c_{t-1} of the internal memory unit through the input gate, forget gate, and output gate at each moment.

(2) After receiving the information, each gate performs operations on the information from different sources and decides whether to activate it.

(3) After the information received by the input gate is transformed by a nonlinear function, it is combined with the state of the internal memory unit processed by the forget gate to form a new memory unit state c_t, and then the newly formed memory unit state is formed by the dynamic control of the output gate. The output information h_t outputs the LSTM unit.

The calculation relationship between various variables is as follows:

$$\begin{aligned} i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right), \\ f_i &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right), \\ c_t &= f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right), \\ o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right), \\ h_t &= o_t \tanh\left(c_t\right). \end{aligned} \tag{5}$$

In the above equation, i, f, c, and o are input gate, forget gate, cell state, and output gate, respectively. W_{xi}, W_{xf}, W_{xc}, and W_{xo} are all weight coefficient matrices linking the input signal x_t, and W_{hi}, W_{hf}, W_{hc}, W_{ho} are the weight coefficient matrices of the input signal h_t of the link hidden layer, and W_{ci}, W_{cf}, W_{co} are the diagonal matrices to link neuron activation function of the output vector c_t with the gate function. b_i, b_c, b_f, and b_o are bias vectors, σ is the sigmoid activation function, and tanh is the hyperbolic tangent activation function.

2.4. The Proposed Model (VMD-WD-LSTM). The framework of the VMD-WD-LSTM-based traffic flow prediction model is shown in Figure 4. The main steps of the VMD-WD-LSTM model are as follows:

(1) The original traffic flow data are decomposed into multiple eigenmode functions (IMFs) by VMD, and the number of IMFs is determined by the sample entropy of the reconstructed data under different K values.

(2) Each IMF is processed by the hard threshold function denoising method of wavelet threshold denoising, and the denoised subsequences are obtained.

(3) LSTM is employed to predict each subsequence, and the predicted value of each subsequence is synthesized into the final prediction result.

3. Experiments

This section provides a concise and precise description of the experimental results, their interpretation, and the experimental conclusions that can be drawn.

3.1. Data Description. The open-source data used are selected from the PeMS database, which collects traffic data from more than 39,000 individual detectors. The sensor layout covers the highway system in all metropolitan areas in California. Specifically, the experimental data of this paper are collected from the Kumeyaay Highway in California. This paper selects three detectors from many detectors (see Figure 5). We took the complete traffic flow data for five consecutive days from Monday to Friday from these three detection points for analysis. In addition, in order not to affect the accuracy of the prediction results, the dates we selected do not include holidays. The time period is from 14 September 2020 to 18 September 2020, the time interval of traffic flow is 5 minutes, and the number of samples is 1440 data. All the data obtained are divided into training set and testing set. From Monday to Thursday, 1152 (80% of all data) data are used as training set, and 288 (20% of all data) data on Friday are used as testing set. The three detectors are represented by A, B, and C, respectively. Figure 5 gives the location of each detector, and the detailed information of each detector is shown in Table 1. In this experiment, the traffic flow data of the three detectors are all used to test and analyze the performance of the proposed model. Limited to the length of the paper, the traffic flow data of detector A are taken as an example to illustrate the specific operations and results of each step of the proposed model.

The original traffic flow data of the three detectors are illustrated in Figure 6. It can be seen that the traffic flow from Monday to Friday has obvious periodicity, and the characteristics of the daily traffic volume change are obvious, but there is obvious nonlinearity and volatility. Part of the reason is due to the presence of noise.

As mentioned in the experiment, the traffic flow data obtained from the detector are easily affected by various unexpected factors, and there is a certain degree of abnormal fluctuations. Therefore, the abnormal value of the data is suppressed through the variational mode decomposition and wavelet threshold denoising, so as to obtain reliable traffic flow data.

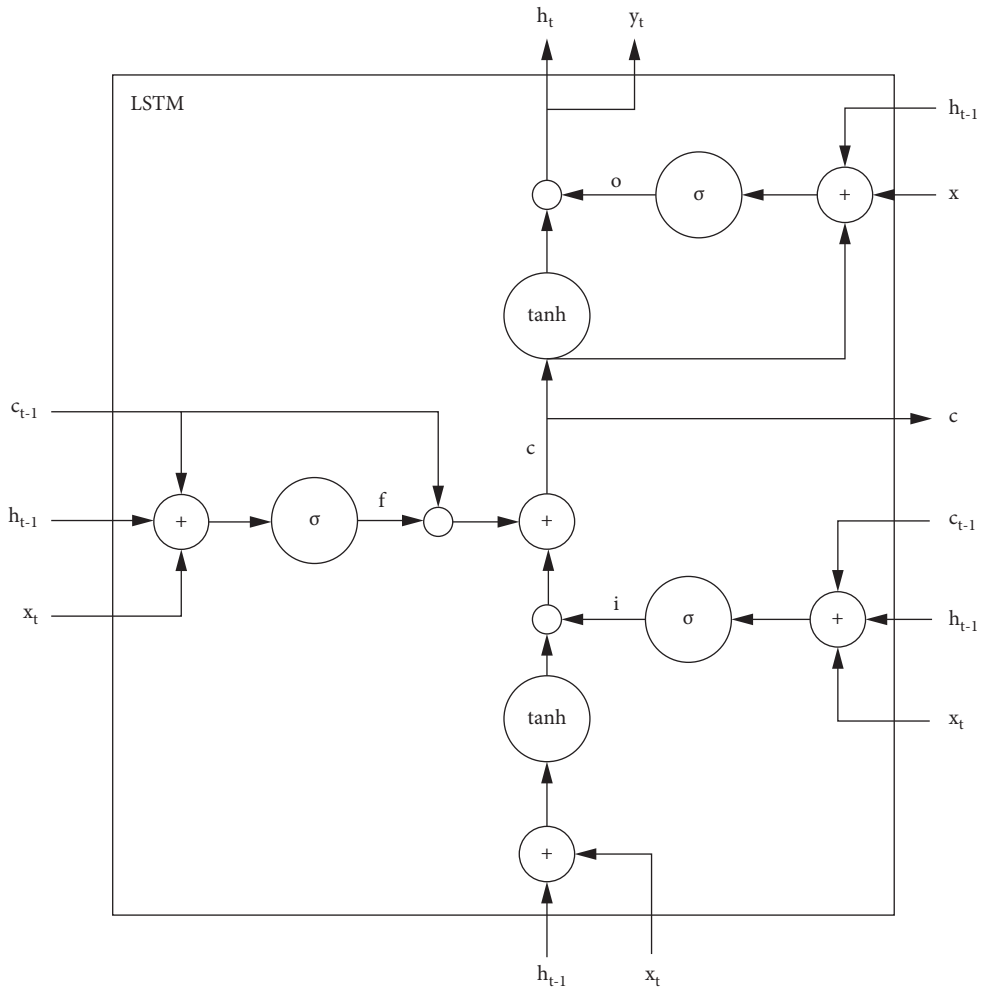


Figure 3: LSTM neural network architecture.

3.2. Evaluation Indexes. The experiment uses three commonly used standards to evaluate the advantages and disadvantages of the model. The three standards are the root mean square error (RMSE), the average absolute error (MAE), and the average absolute percentage error (MAPE), which are defined as follows:

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(f_i - \widehat{f}_i\right)^2}, \\ \mathrm{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left(f_i - \widehat{f}_i\right), \\ \mathrm{MAPE} &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{f_i - \widehat{f}_i}{f_i}\right| \times 100\%, \end{aligned} \tag{6}$$

where f_i represents the real traffic flow data, $\widehat{f}_i$ represents the predicted data, and n is the number of samples.

3.3. VMD Results of Traffic Flow Data. The original traffic flow data of detector A are decomposed using VMD. And it is important to determine that original traffic flow data should be decomposed to how many IMFs (each IMF corresponds to a reconstructed component). On the one hand, too few IMFs may not be able to extract the features hidden in the original data. On the other hand, too many IMFs may lead to a poor prediction result because of prediction error accumulation in the ensemble step. In this study, the optimal number of IMFs is determined according to the sample entropy values corresponding to the reconstructed component with different number of IMFs. The greater the sample entropy, the greater the complexity of the sample sequence, which makes data prediction more difficult. Therefore, the number of IMFs corresponding to the minimum sample entropy is the optimal number of decompositions. The greater the sample entropy, the greater the complexity of the sample sequence, which makes data prediction more difficult. Therefore, it is necessary to select the K value that minimizes the entropy of the sample as the decomposition number. The sample entropy values corresponding to the number K of IMFs are shown in Figure 7. It can be seen from Figure 7 that when $K = 3$, the obtained sample entropy is the smallest, and the optimal number of IMFs is 3. The three IMF components obtained by VMD are shown in Figure 8.

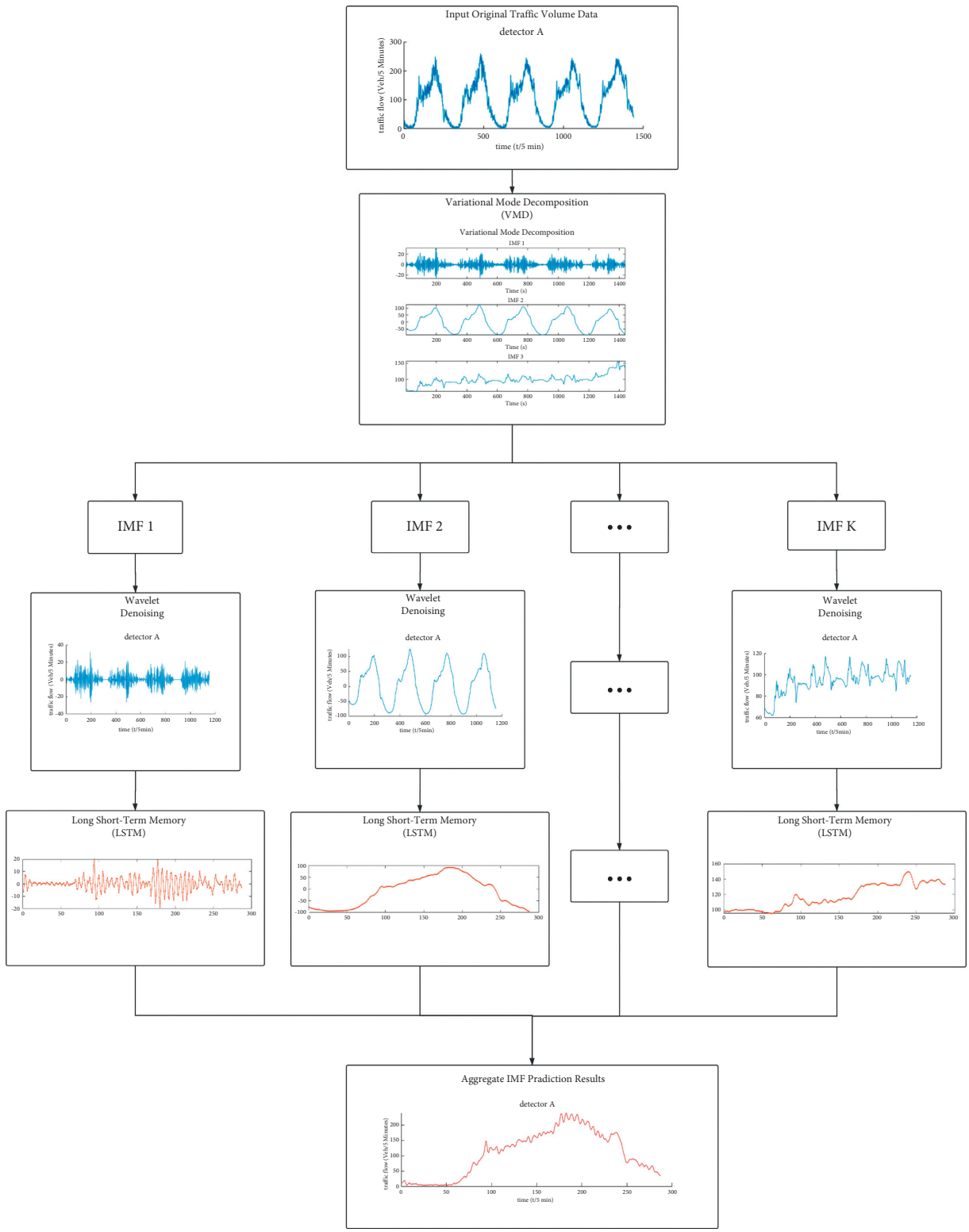


Figure 4: The framework of the VMD-WD-LSTM model.

3.4. Wavelet Denoising of Each IMF. In order to further reduce the influence of noise on the prediction results, wavelet transform is used to denoise each IMF. In order to ensure a more impressive result in terms of root mean square error, a hard threshold function is selected. The result of wavelet denoising can be seen from Figure 9. It can be seen that the data curve after wavelet denoising is smoother and the data features are more obvious.

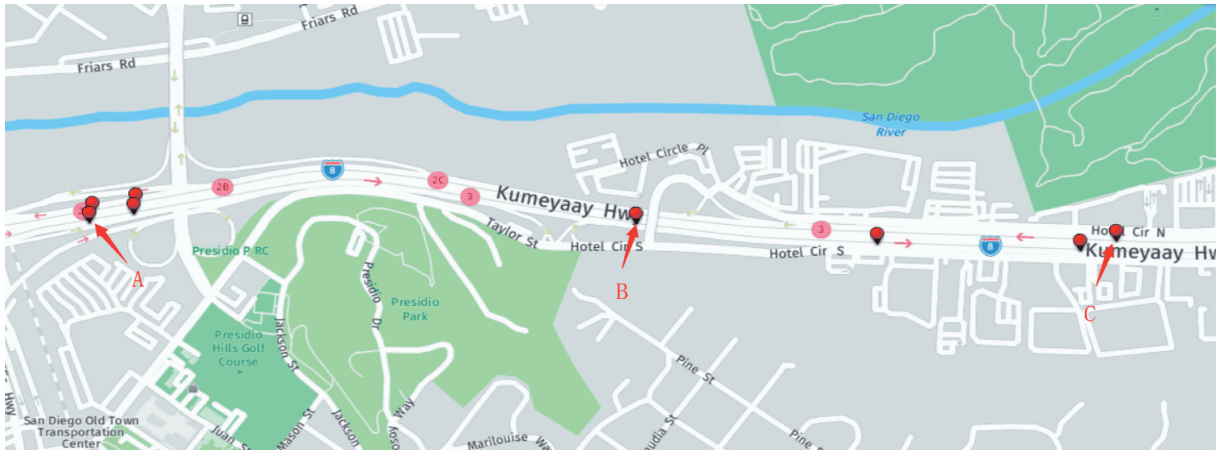


Figure 5: Location of the three detectors.

Table 1: Detailed information for the three detectors.

Location	Freeway	Detector number	Max cap	Lane point
A	I8	1113028	68.2	2
B	I8	1115407	161.0	5
C	I8	1115549	162.6	5

3.5. Results and Discussion. The LSTM model is employed to predict each subsequence obtained by VMD-WD, and then the predicted value of each subsequence is synthesized into the final prediction result. The LSTM model is composed of an input layer, a hidden layer, and an output layer. The hidden layer contains 200 neural units. The input feature dimension and the output feature dimension are both 1. In terms of options for training deep learning neural networks, the LSTM model uses the Adam optimizer. Specifically, the maximum number of training epochs is 250, and the initial learning rate is 0.005. In order to avoid the problem of gradient explosion, when the training reaches 125 epochs, the global learning rate is reduced by a multiplier factor (which is set to 0.2).

Figures 10–12 show the prediction results of IMF1 component, IMF2 component, and IMF3 component, respectively. It can be seen intuitively that the predicted curves and observed curves of IMF1, IMF2, and IMF3 components are highly fitted. Moreover, RMSE of the prediction results of the IMF1, IMF2, and IMF3 components is 1.8089 veh/5 min, 2.1161 veh/5 min, and 1.7235 veh/5 min, respectively, which also illustrate that the proposed model has superior performance. The characteristics of each IMF component can also be seen from these three figures. IMF1 represents the high-frequency component obtained from the original data, which shows the randomness of the original traffic flow data. IMF2 and IMF3 both represent low-frequency components, which show the regularity of the original traffic flow data. In summary, it is easier to capture the different features contained in the original signal by separately predicting each IMF component obtained by VMD. The final step is to accumulate the prediction results of each IMF component to get the final prediction result. Figure 13 shows the cumulative prediction results. It can also be seen intuitively that the predicted curve and the observed curve are highly fitted.

In order to prove the influence of the VMD-WD hybrid denoising method on the final prediction results and to prove the prediction performance of LSTM model, this paper sets up a comparative study. Firstly, the prediction results of detector A are predicted by the traditional differential autoregressive integrated moving average (ARIMA) model, artificial neural network (ANN) model, and LSTM model, respectively, and the root mean square error and mean absolute error of the three models are compared (see Figure 14). Then, the original traffic flow data of detector A are denoised by the VMD-WD hybrid method in this study. Finally, the denoised data are predicted by ARIMA, ANN, and LSTM models, respectively, and RMSE and MAE are calculated for comparison. The final results are shown in Figures 14 and 15. It can be seen from the image that the prediction performance of LSTM model is significantly better than that of the ARIMA model and ANN model. Similarly, from the data denoised by the VMD-WD hybrid denoising method, the prediction results of the LSTM model are still the most accurate. On the other hand, the prediction performance of the hybrid prediction model combined with VMD-WD denoising method is better than that of the single prediction model. The prediction performance of the LSTM model is obviously improved. Compared with the LSTM model, the RMSE and MAE of the VMD-WD-LSTM model are reduced by 37.8% and 38.2%. The above results show that VMD-WD denoising method can improve the prediction performance of prediction model, and the prediction accuracy of the LSTM model based on deep learning is higher than that of traditional methods.

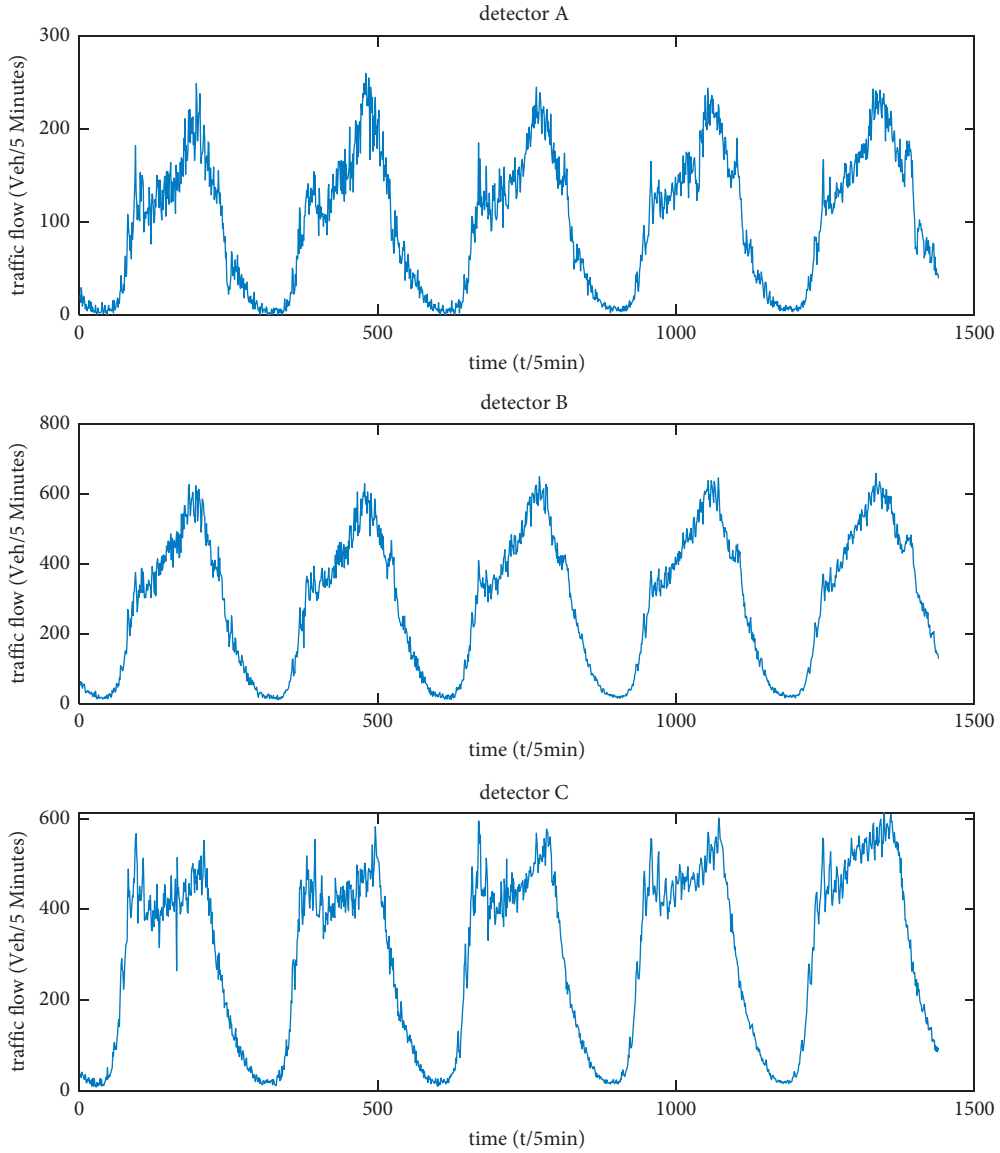


Figure 6: The original traffic flow data of the three detectors.

Figure 7: Sample entropy with different K.

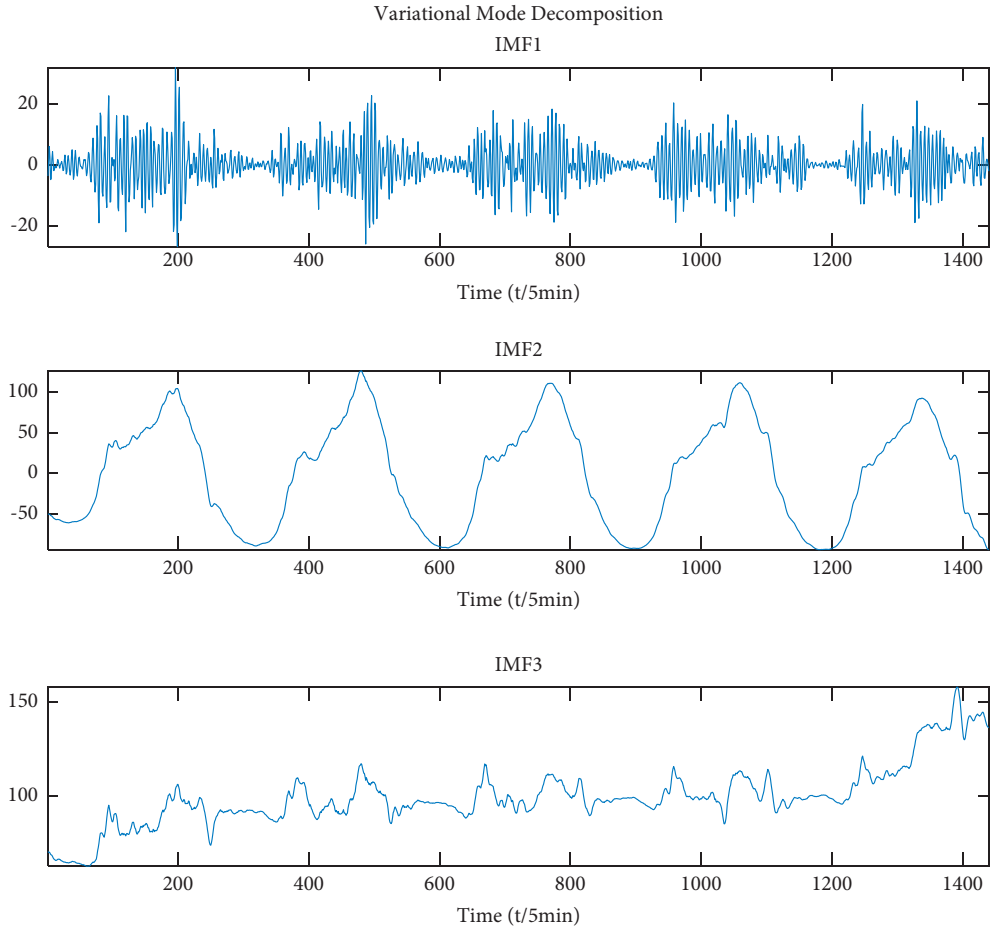


Figure 8: VMD result of detector A.

In order to further prove the prediction performance of the hybrid prediction model proposed in this study, seven different prediction models are introduced in this paper. The prediction results of the seven models are compared with those of the VMD-WD-LSTM method. The comparison methods include the prediction model without denoising steps, the model with signal decomposition by other methods, the models with different prediction methods, and the latest methods proposed in this research field in recent two years. The RMSE and MAE of the WD-LSTM method were 10.85 and 8.066, respectively, which were compared with the prediction results of the LSTM method. MAE and MAPE are reduced by 2%, 2.6%, and 28.9%, respectively. It can be seen that the prediction performance of the prediction model after wavelet threshold denoising is improved because the wavelet threshold denoising method can remove the noise signal in the original signal to a certain extent and retain the characteristics of the original signal, which reduces the interference of noise on traffic flow prediction. However, due to the complex time characteristics of the original signal, the denoising effect still has room for improvement. Then, this study compares the EMD-WD-LSTM method with the WD-LSTM method. It can be seen from the evaluation index that the RMSE and MAE of the EMD-WD-LSTM method are further reduced compared with the latter. This is because the signal after the EMD method decomposes the original signal containing complex characteristics into simpler subsequences, which is not only conducive to the wavelet threshold method to identify high-frequency noise but also conducive to the LSTM model for prediction. However, the MAPE of EMD-WD-LSTM method is higher than that of the latter. This study believes that this is due to the limitation of EMD method itself. The IMF components obtained by EMD decomposition will have the phenomenon of modal mixing. When there are abnormal events and other disturbances in the signal, each IMF will contain more than one frequency component, which will affect the prediction performance of the prediction model to a certain extent. In the third step, this paper compares the prediction results of VMD-WD-LSTM method with the LSTM method, WD-LSTM method, and EMD-WD-LSTM method. It can be seen from the prediction indicators (Table 2 and Figure 16) that the prediction performance of the VMD-WD-LSTM method proposed in this study has been significantly improved compared with the above three methods. Compared with the EMD-WD-LSTM method, RMSE, MAE, and MAPE are increased by 32.5%, 32.9%, and 34.5%, respectively. This result is consistent with the description in the first part of

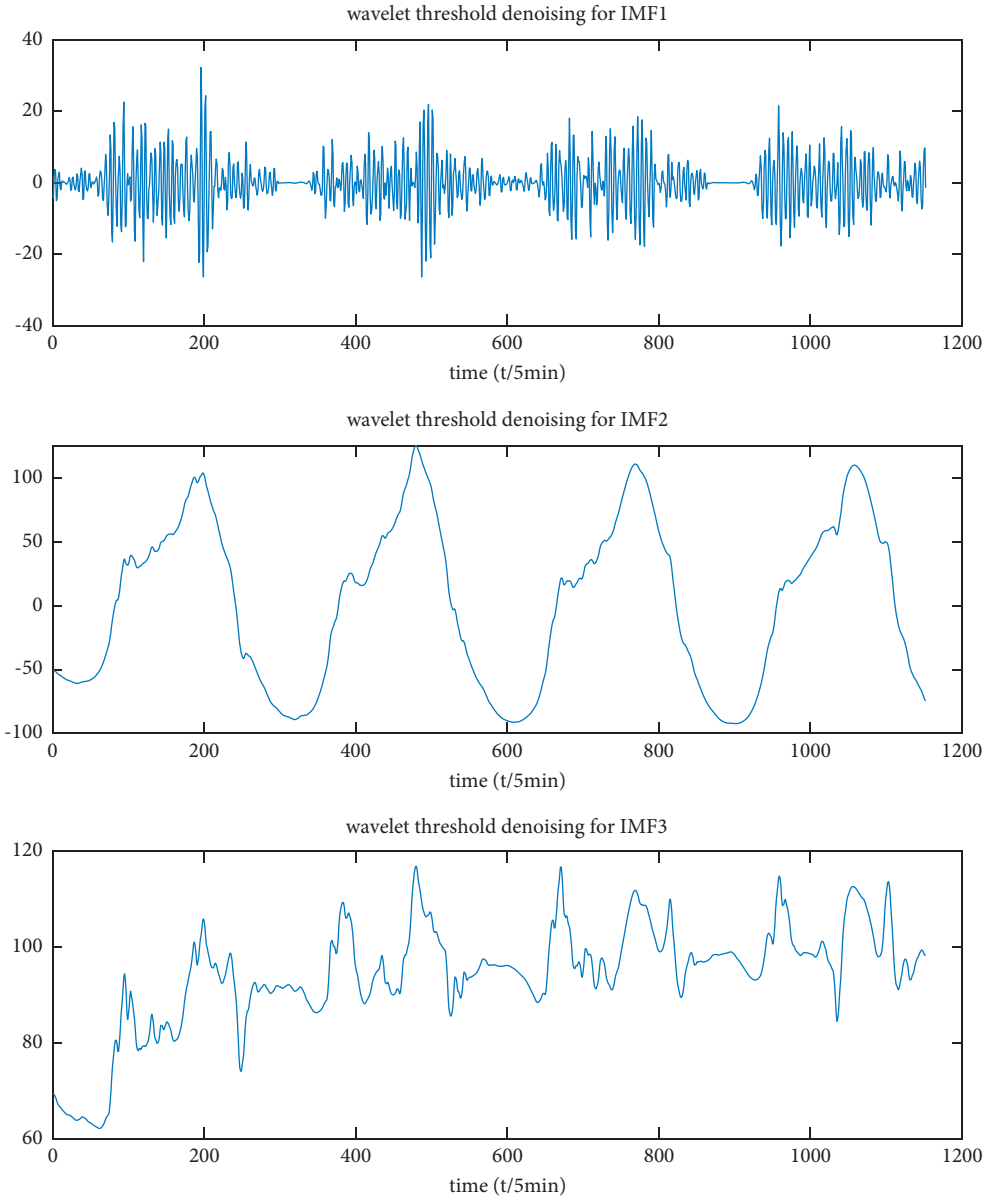


Figure 9: Curves of different IMF components based on WD processing.

this paper. Compared with EMD, the VMD method overcomes the problem of modal aliasing, so the decomposed subsequence is more conducive to denoising and prediction. On the other hand, the prediction performance of the VMD-WD-LSTM method is better than that of the VMD-WD-ANN method, which indicates that the LSTM model is more suitable for time series prediction than the traditional artificial neural network model. Finally, this paper selects the method proposed in the field of traffic flow prediction in the past two years to compare it with the VMD-WD-LSTM method proposed in this study. These methods are the EEMD-ANN model proposed by Chen et al. [8] in 2020, ARIMA-LSTM model proposed by Lu et al. [31], and TSD-BiLSTM model proposed by Huang et al. [29] in 2021. The operation steps and parameter selection of the above model are strictly consistent with the literature, and the model parameters are shown in Table 3. Comparing the prediction results of the four models for detector A traffic flow data, it can be seen that the RMSE, MAE, and MAPE of the VMD-WD-LSTM method are the lowest among the four models. As shown in Figure 16, the prediction image of the VMD-WD-LSTM model is closest to the real data. This shows that the VMD-WD-LSTM model proposed in this study still has practical value compared with the new methods in this field.

The boxplots of the absolute errors of the different models are shown in Figure 17. For each boxplot, the central mark (red line) is the median; the edges of boxes are the 25th (Q1) and 75th (Q3) percentiles, and the interquartile range (IQR = Q3−Q1) is used for evaluating the degree of

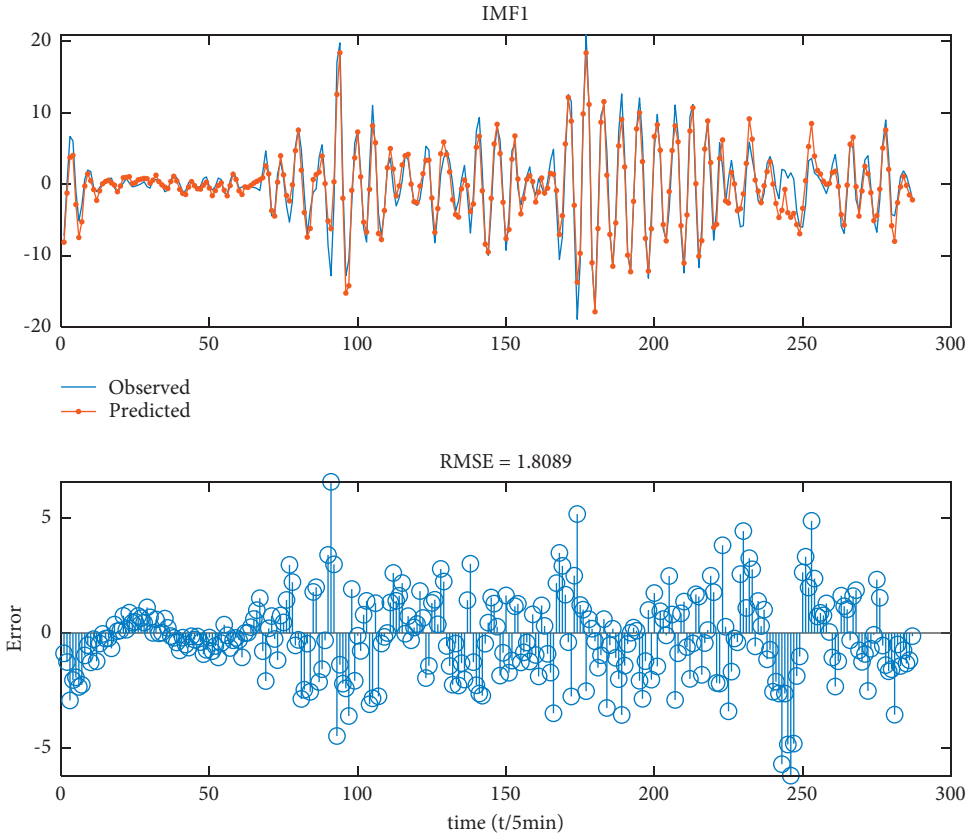


Figure 10: Prediction results of IMF1 (detector A).

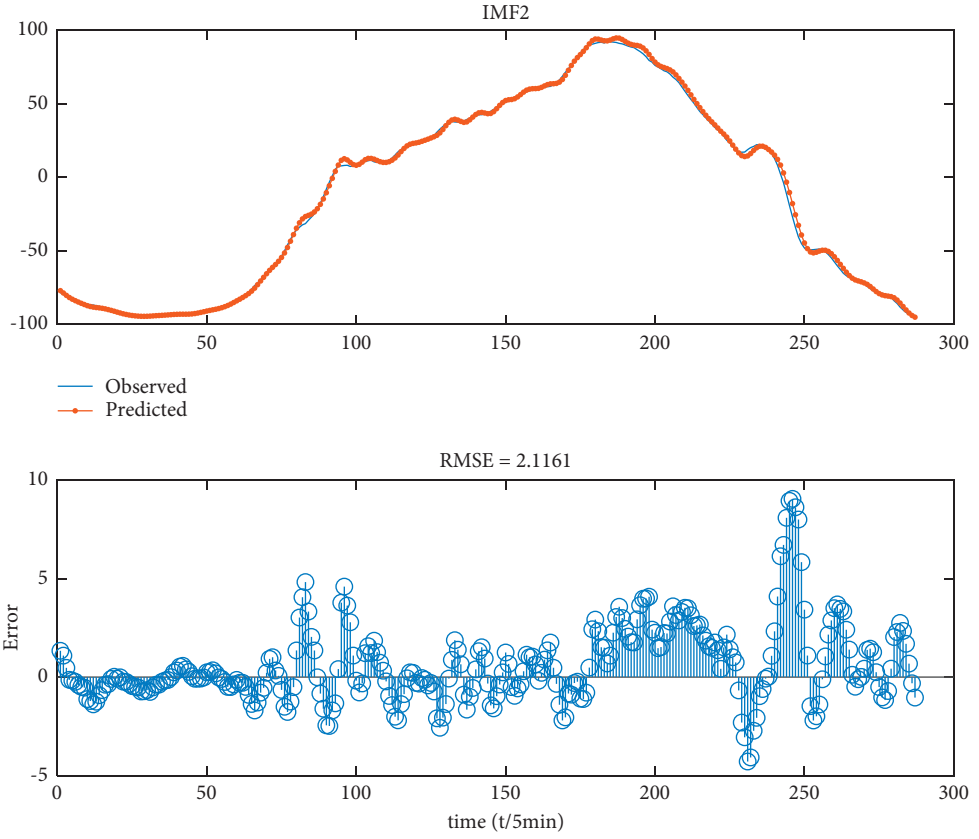


Figure 11: Prediction results of IMF2 (detector A).

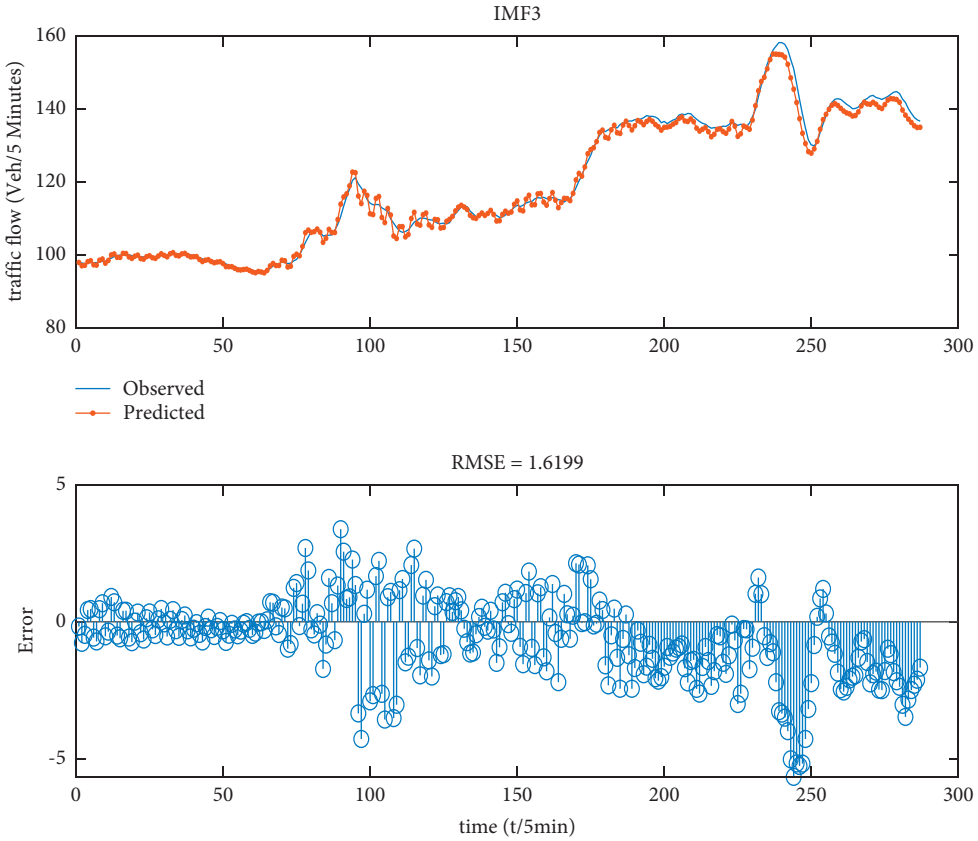


Figure 12: Prediction results of IMF3 (detector A).

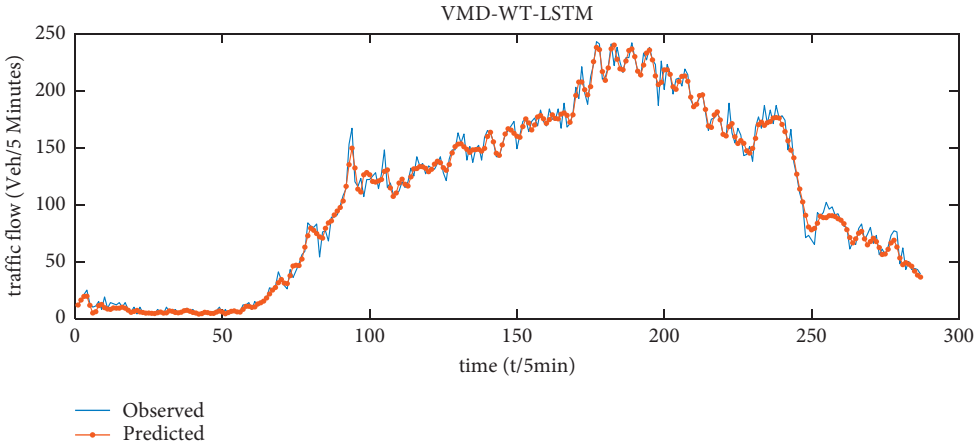


Figure 13: Prediction result of the VMD-WD-LSTM model (detector A).

concentration to median; the whiskers extending to the most extreme data points are not considered as outliers (abnormal data points). It can be seen from Figure 17 that the IQR of the absolute error of the VMD-WD-LSTM model is the smallest (that is, the fluctuation of the absolute error is the smallest), indicating the outstanding stability of this prediction model.

We continue to compare and analyze the performance of the proposed model on the traffic flow data of detectors B and C to analyze whether the model can maintain good prediction performance on different traffic flow data. Figures 18 and 19 show the prediction curves of different models for detector B and detector C data, respectively. It can be clearly seen from Figures 18 and 19 that the predicted and observed values of the VMD-WD-LSTM model have achieved good fitting results on the traffic flow data of detector B and detector C. Tables 4 and 5 illustrate RMSE, MAE, and MAPE for different models with data from detectors B and C, respectively. Specifically, the RMSE,

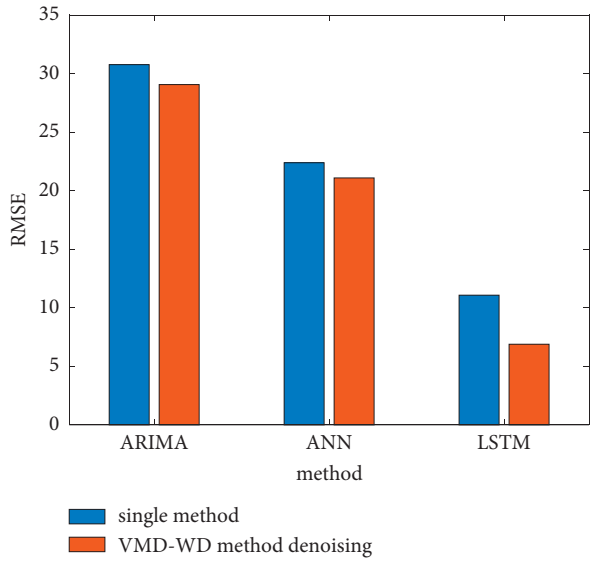


Figure 14: RMSE of the proposed method and the three single methods.

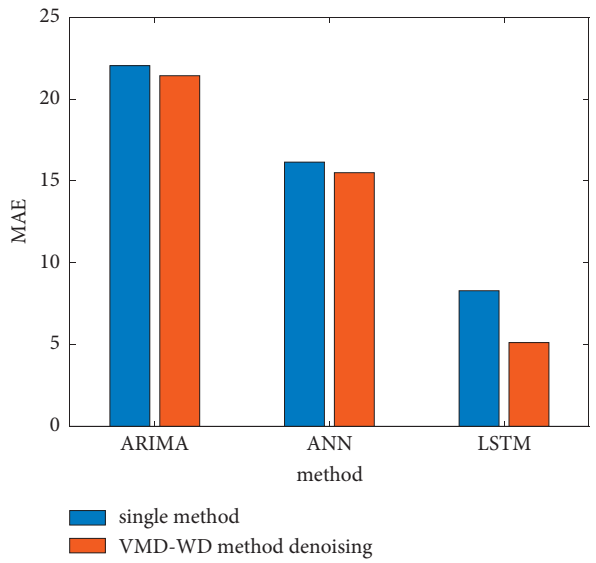


Figure 15: Comparison of hybrid VMD-WD methods.

Table 2: The prediction errors of different models using the traffic flow data of detector A.

Prediction models	RMSE (veh/5 min)	MAE (veh/5 min)	MAPE (%)
LSTM	11.077	8.285	15.360
WD-LSTM	10.850	8.066	10.912
EMD-WD-LSTM	10.217	7.626	14.227
VMD-WD-ANN	21.092	15.498	20.030
EEMD-ANN	22.305	16.936	25.391
TSD-BiLSTM	16.448	11.877	17.247
ARIMA-LSTM	12.138	8.846	12.612
VMD-WD-LSTM	**6.888**	**5.116**	**9.313**

Bold indicates the best value of each evaluation index.

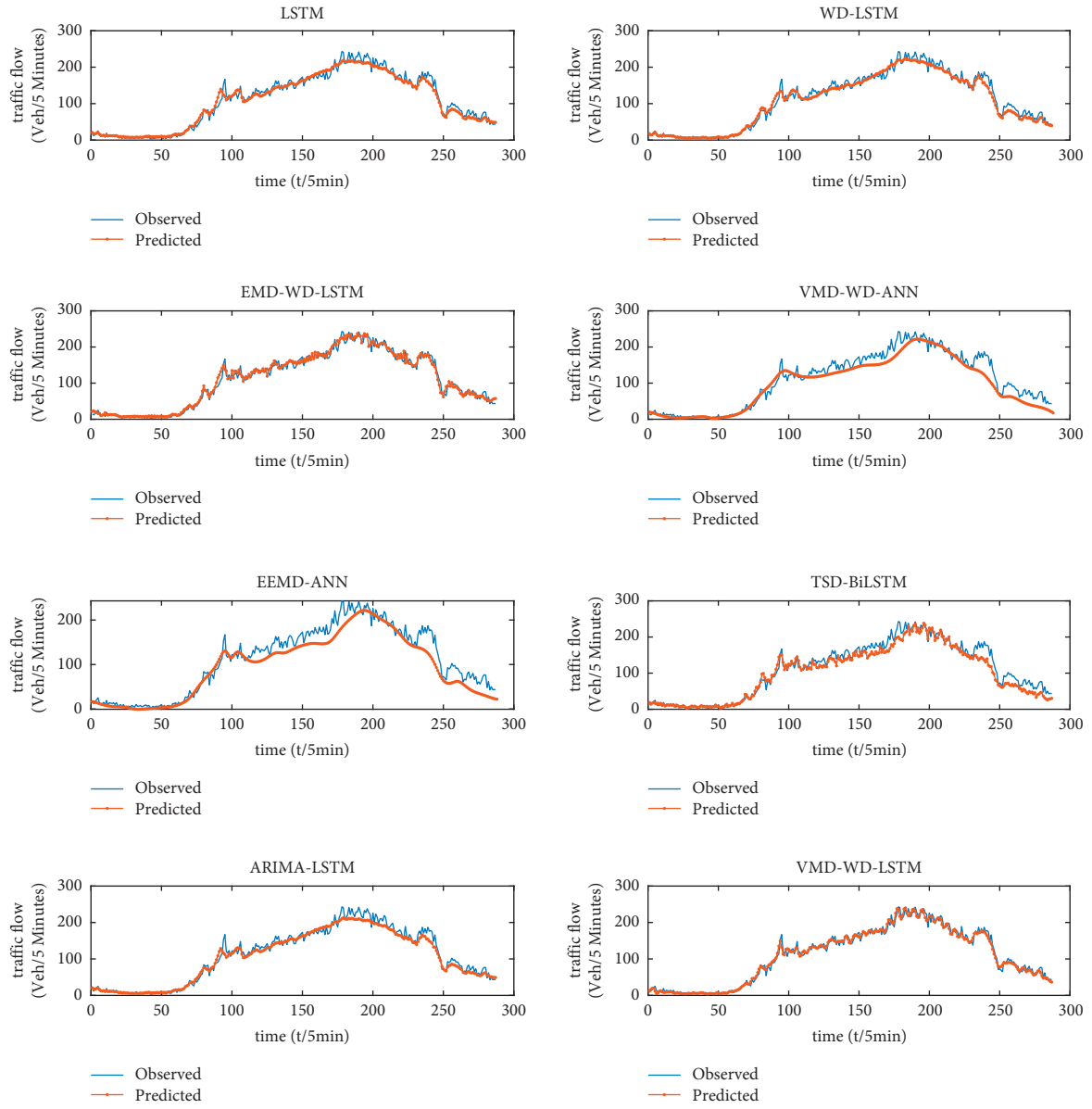


Figure 16: Predicted curves of different models using the traffic flow data of detector A.

Table 3: Parameter configuration of different models.

Model	Primary parameter	Setting value
ARIMA	p	2
	d	1
	q	4
EEMD	Added white noise	0.2
	The ensemble number	1000
LSTM	Learning rate	0.005
	The number of training epochs	250
	Number of hidden units	288
BiLSTM	Learning rate	0.005
	The number of training epochs	50
	Number of hidden units	32
ANN	Learning rate	0.005
	The number of training epochs	250
	Number of hidden units	288

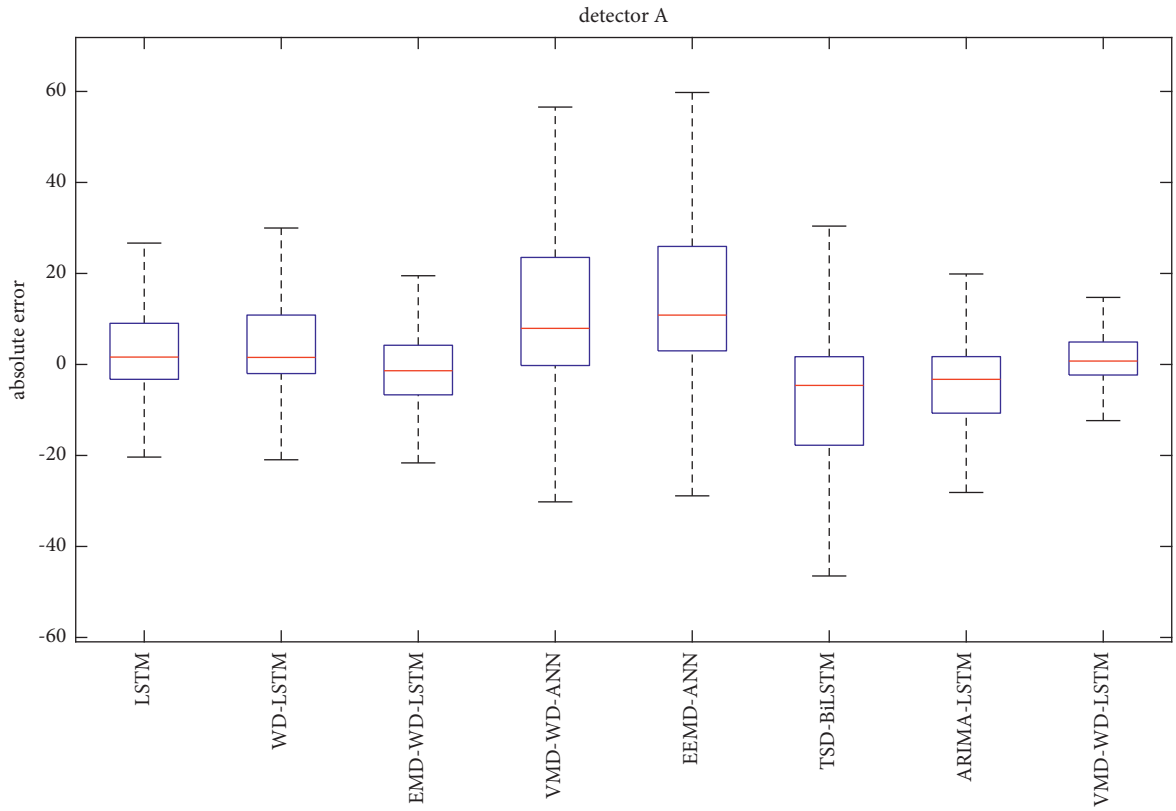


Figure 17: Boxplots of absolute error of different models (detector A).

MAE, and MAPE of the VMD-WD-LSTM model on the detector B data are 10.304, 7.984, and 5.979%; the RMSE, MAE, and MAPE of the VMD-WD-LSTM model for detector C data are 13.980,10.965, and 5.268%. It can be seen that RMSE and MAE on detector B data of the VMD-WD-LSTM model and RMSE, MAE, and MAPE on detector C data are the smallest, while detection point B and detection point C have more lanes and larger traffic flow than detection point A, indicating that the method proposed in this study can maintain excellent performance in dealing with different types of road sections and different scales of traffic flow. We further analyze the comparison methods one by one. First, from the traffic flow prediction results of the two detection points, the prediction results of the method after signal decomposition and denoising are more accurate than the results obtained by a single prediction model. From the evaluation indicators RMSE and MAE, it can be seen that the VMD-WD-LSTM method proposed in this study still shows the best prediction performance, but the MAPE of the VMD-WD-LSTM model (5.979%) is slightly higher than that of the EMD-WD-LSTM model (5.208%). This shows that the error of the results predicted by the EMD-WD-LSTM method is smaller than that of the original data. In this case, we believe that the traffic flow has obvious cyclical characteristics and is affected by various external factors. However, the external factors are difficult to predict. The difference in the error between the predicted flow and the actual flow in a small enough range does not mean that there is a significant difference in the prediction performance. On the other hand, this paper also recognizes that the VMD-WD-LSTM model still has room for improvement, and different types of external factors should be considered as important factors affecting traffic flow prediction in future research. The RMSE, MAE, and MAPE of the VMD-WD-LSTM model are much smaller than those of the VMD-WD-ANN model, which further proves that LSTM has better prediction accuracy due to its advantages in long-term dependence of capturing time series. The RMSE, MAE, and MAPE of the VMD-WD-LSTM model on the three detector data are smaller than those of the WD-LSTM model, indicating that the wavelet threshold denoising method combined with VMD can not only effectively improve the prediction accuracy but also show good robustness. In addition, the RMSE and MAE of the WD-LSTM model and the EMD-WD-LSTM model on the data of detector B and detector C are slightly smaller than those of the single LSTM model, while the MAPE is slightly larger, indicating that only the wavelet threshold denoising method cannot stably maintain the effect of reducing the noise data interference when dealing with the traffic flow data with complex characteristics. The number of IMF components obtained by EMD is uncertain. When removing high-frequency components for EMD-based denoising, the direction of the original signal will be affected. Therefore, it is difficult for the EMD-WD-LSTM model to maintain stable prediction accuracy on different datasets. The prediction errors of the VMD-WD-ANN model and EEMD-ANN model are generally large. The

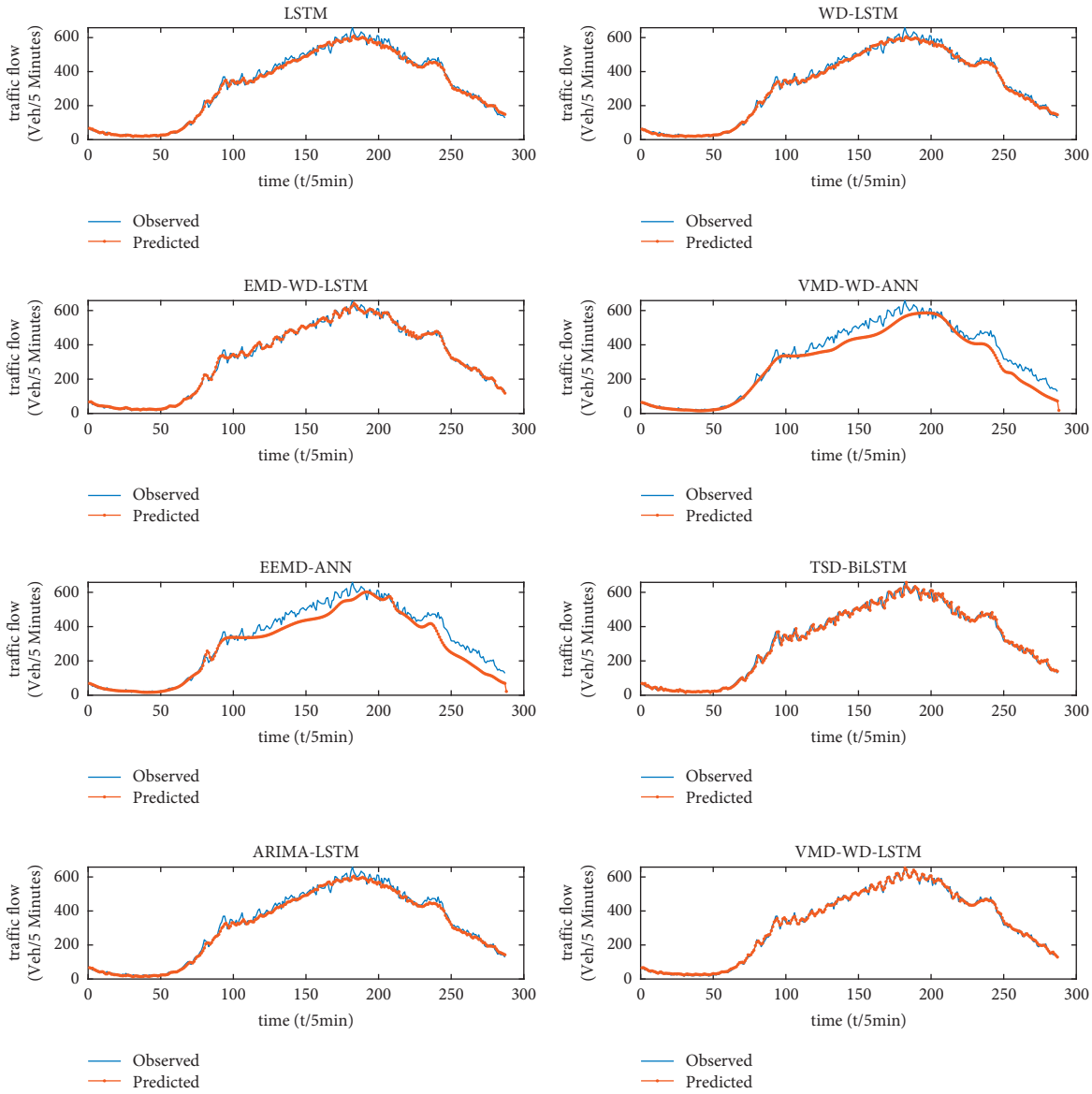


Figure 18: Predicted curves of different models using the traffic flow data of detector B.

reason may be that the ANN model can predict the trend of traffic flow sequence but cannot capture the time-varying characteristics of traffic flow (see Figures 16, 18, and 19). In addition, compared with several new methods, the VMD-WD-LSTM method has maintained a good prediction effect as always. In summary, the VMD-WD-LSTM model has the highest prediction accuracy and the strongest robustness in different comparison methods.

Figures 20 and 21 are boxplots of the absolute errors of different models using the traffic flow data of detector B and detector C, respectively. It can be seen that the VMD-WD-LSTM model has the smallest IQR on the detector B data and the detector C data, although it is not as obvious as on the detector A data. This result is sufficient to demonstrate that the VMD-WD-LSTM model not only has the smallest MAE but also has the smallest fluctuation range of the absolute error. In addition, compared with the absolute error of detector A data, the absolute error of the VMD-WD-ANN model and the EEMD-ANN model on the detector B data and detector C data has increased significantly, which also shows that the performance of the ANN model is not stable enough. The reason for this phenomenon may be caused by the large traffic flow of detector B and detector C during peak hours. On the traffic flow data of the three detectors, the absolute error of the WD-LSTM model is more concentrated than that of the single LSTM model, which also proves the effectiveness of WD-based preprocessing of the traffic flow time series.

Based on the previous analysis results, it can be seen that the VMD-WD-LSTM model is able to predict the traffic flow collected from different detectors with a smaller absolute error than that of the comparison model. The results obtained are encouraging.

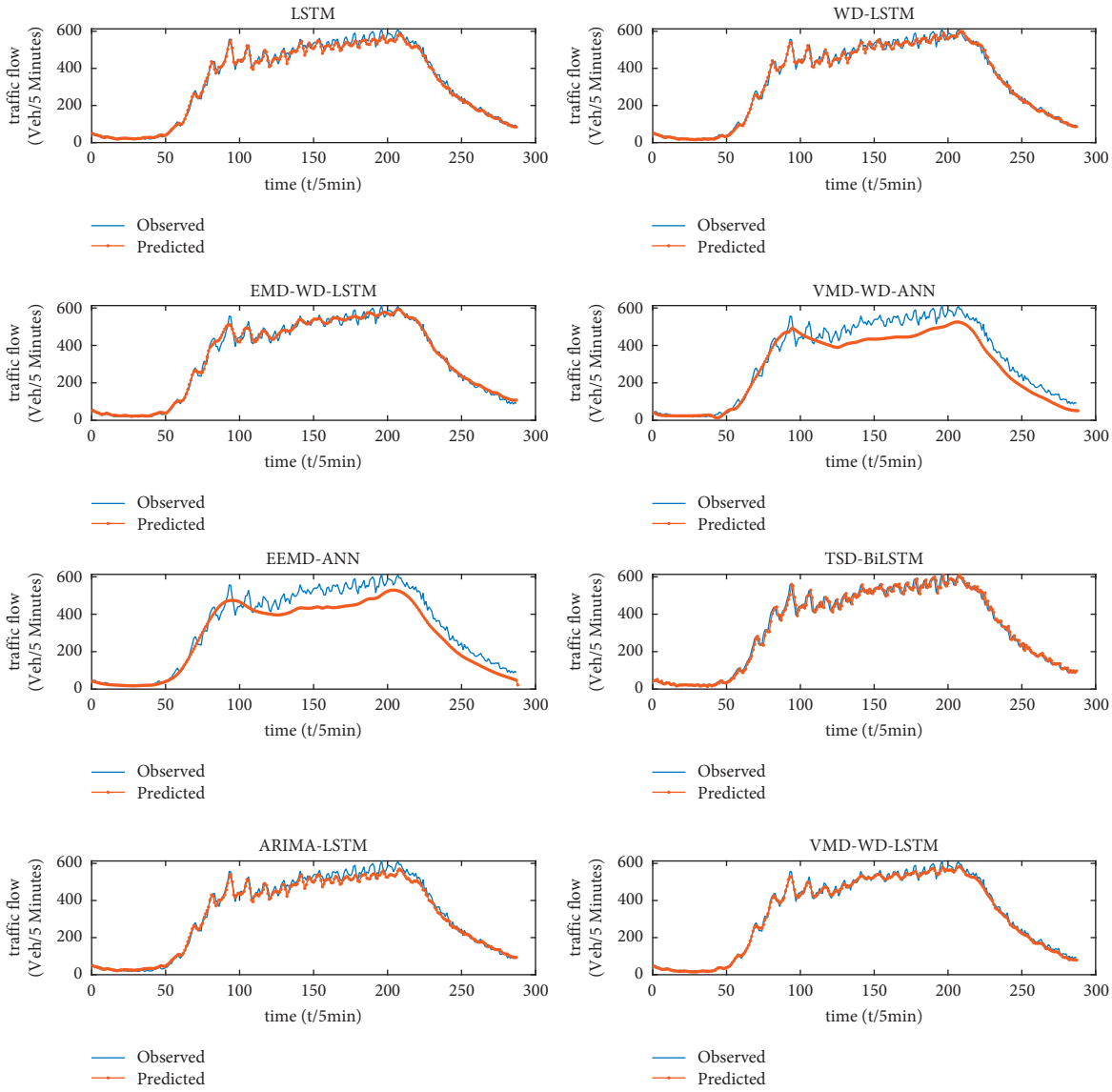


Figure 19: Predicted curves of different models using the traffic flow data of detector C.

Table 4: The prediction errors of different models using the traffic flow data of detector B.

Prediction models	RMSE (veh/5 min)	MAE (veh/5 min)	MAPE (%)
LSTM	18.711	14.094	6.320
WD-LSTM	17.969	13.538	6.457
EMD-WD-LSTM	13.220	9.817	**5.208**
VMD-WD-ANN	49.357	37.466	13.365
EEMD-ANN	51.022	38.303	13.557
TSD-BiLSTM	18.765	13.664	6.531
ARIMA-LSTM	20.234	15.220	7.173
VMD-WD-LSTM	**10.304**	**7.984**	5.979

Bold indicates the best value of each evaluation index.

Table 5: The prediction errors of different models using the traffic flow data of detector C.

Prediction models	RMSE (veh/5 min)	MAE (veh/5 min)	MAPE (%)
LSTM	18.248	13.444	6.137
WD-LSTM	16.939	12.442	5.885
EMD-WD-LSTM	17.157	13.071	6.620
VMD-WD-ANN	60.737	49.949	17.915
EEMD-ANN	61.205	51.208	15.970
TSD-BiLSTM	20.801	15.161	6.616

Table 5: Continued.

Prediction models	RMSE (veh/5 min)	MAE (veh/5 min)	MAPE (%)
ARIMA-LSTM	22.969	17.244	9.057
VMD-WD-LSTM	**13.980**	**10.965**	**5.268**

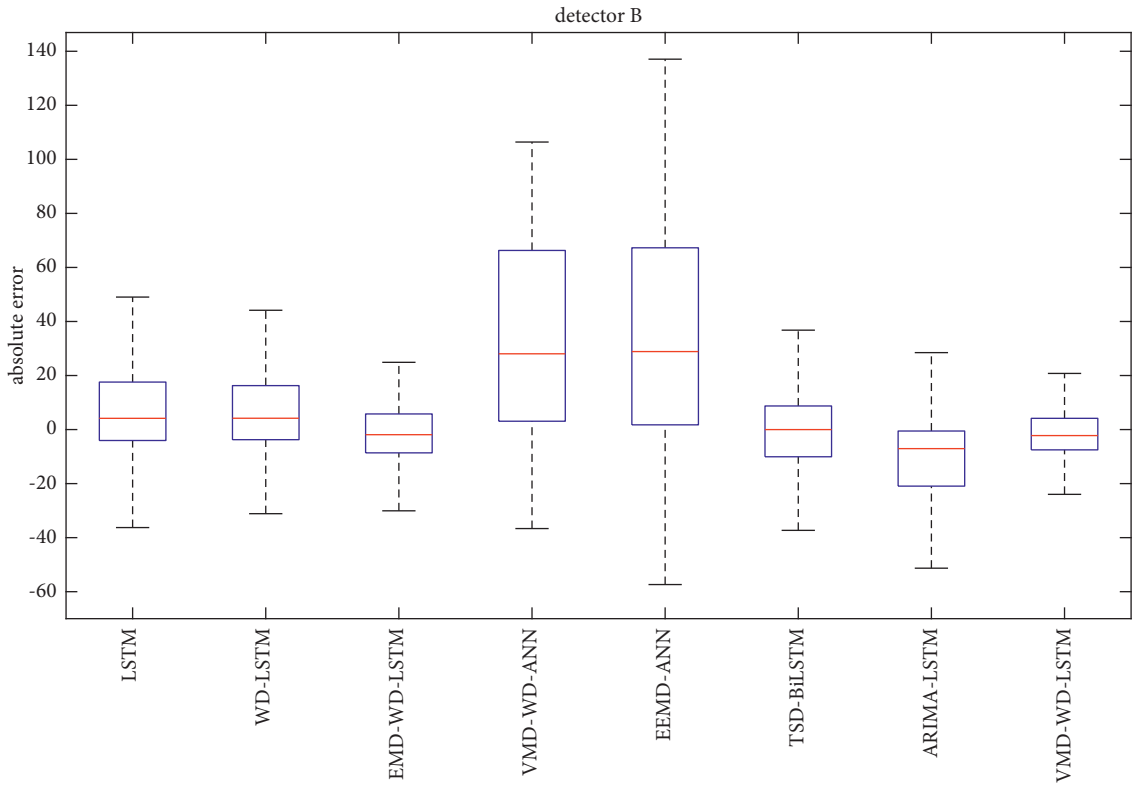


Figure 20: Boxplots of absolute error of different models (detector B).

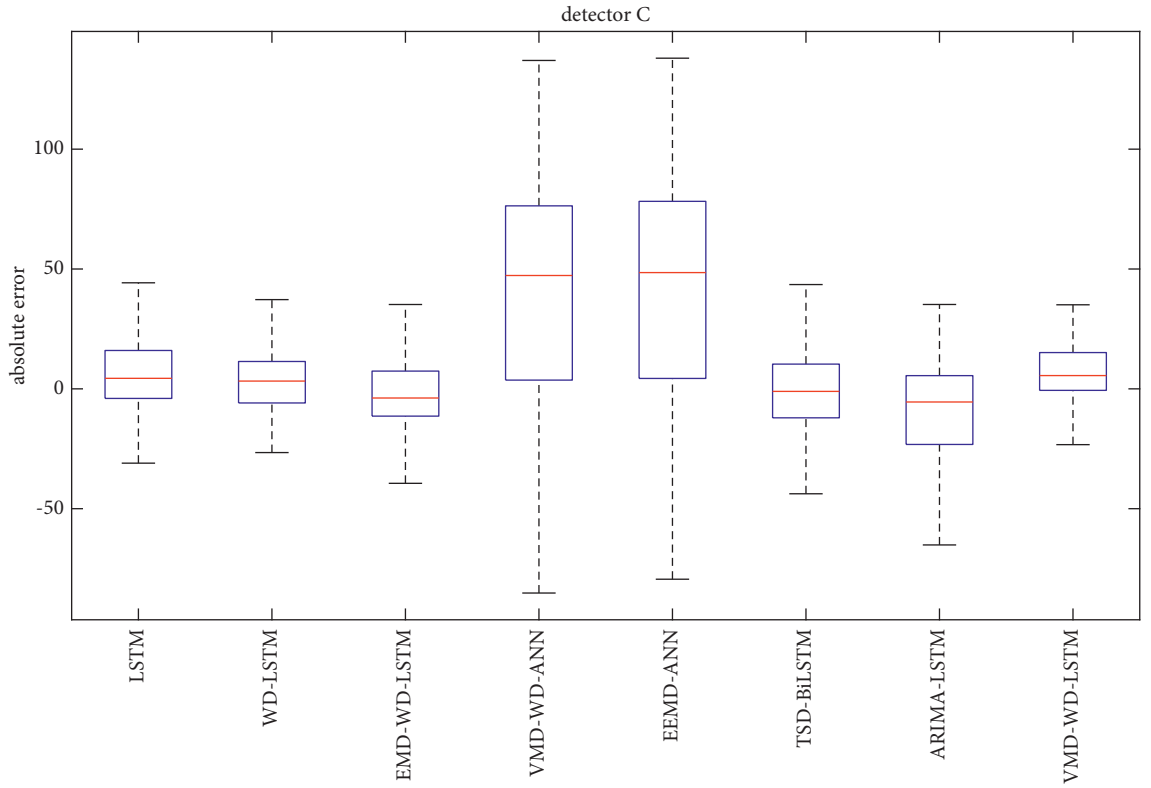


Figure 21: Boxplots of absolute error of different models (detector C).

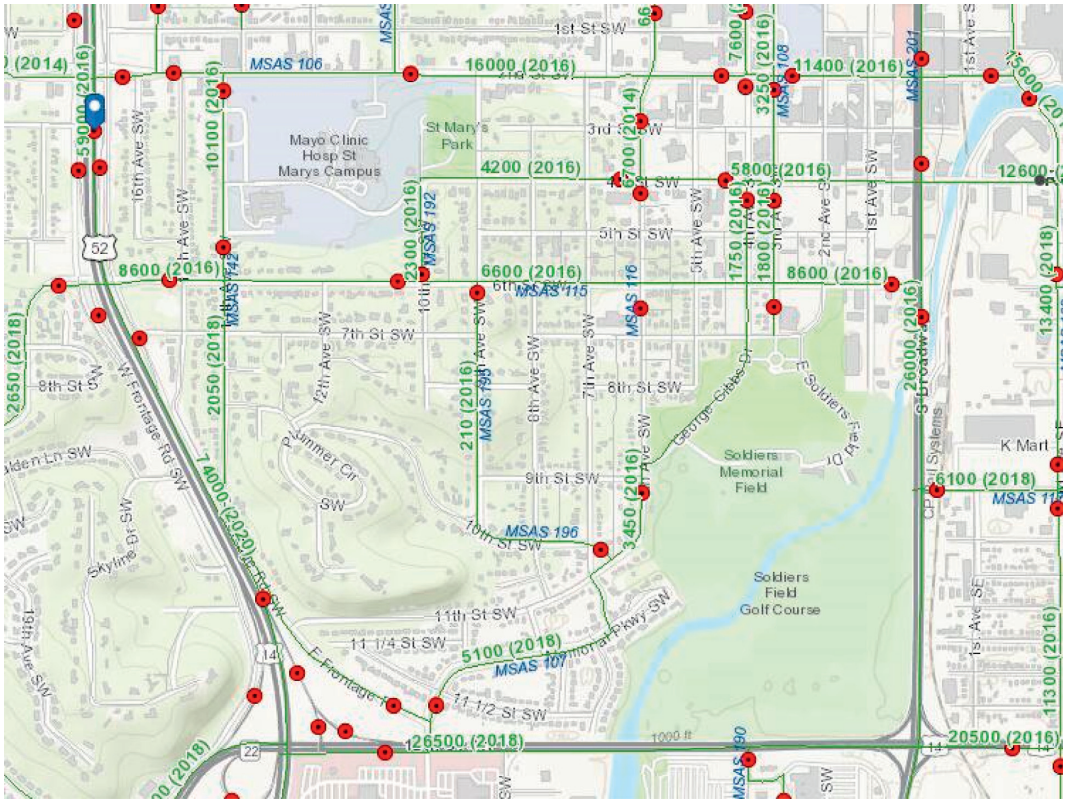


Figure 22: The location of the detection point (marked blue).

Table 6: Detailed information for detector D.

Detector	Freeway	Detector number	Number of main lanes
D	US-52	3891	3

Finally, in order to further verify the prediction performance of the proposed method in this study in the prediction of traffic flow and show the prediction effect of the VMD-WD-LSTM method on different datasets, this paper selects a section of traffic flow data from the public dataset provided by Minnesota Department of Transportation (Mn/DOT) and Transportation Research Data Lab (TDRL). The traffic flow data are obtained by a loop detector (denoted as detector D) of Minnesota Expressway. The traffic flow time interval of this data sample is also 5 minutes, and a detection point is selected in Rochester, Minnesota. The data collection period is from September 14, 2020, to September 18, 2020. The location of the detection point is shown by the blue marker in Figure 22. Table 6 shows the relevant information of the road section.

For the traffic flow data of detector D, we use the same method as detector A, detector B, and detector C to denoise and predict and then compare the prediction performance of different prediction models by evaluating RMSE, MAE, and MAPE. The prediction results of eight prediction models are shown in Figure 23. It can be seen from the predicted image that the prediction results of the EMD-WD-LSTM method and EMD-WD-LSTM method have obvious fluctuations. This is because the traffic flow of detection point D is smaller than that of the three detection points in the PeMS dataset, and the uncertainty of traffic flow is enhanced. Any traffic flow change caused by external factors will interfere with the periodic change of traffic flow to a greater extent, which will make the process of identifying noise data more difficult. However, it can be seen from Table 7 that the RMSE, MAE, and MAPE of the VMD-WD-LSTM method are still reduced by 51%, 48%, and 45% compared with the single LSTM model, and it still has better prediction performance than the seven comparison methods. Compared with the data of several other detection points, the RMSE and MAE of the traffic flow prediction results of detection point D are significantly less than those of the other three detection points. This situation is due to the differences in road types and temporal and spatial correlation between different detection points. On the other hand, it shows that the proposed method will get more accurate prediction results when the traffic flow is small. In addition, in the image shown in Figure 24, the absolute error of the VMD-WD-LSTM method is the smallest, which is consistent with the previous analysis results. This shows that the prediction model proposed in this study also has excellent performance and good robustness on other datasets.

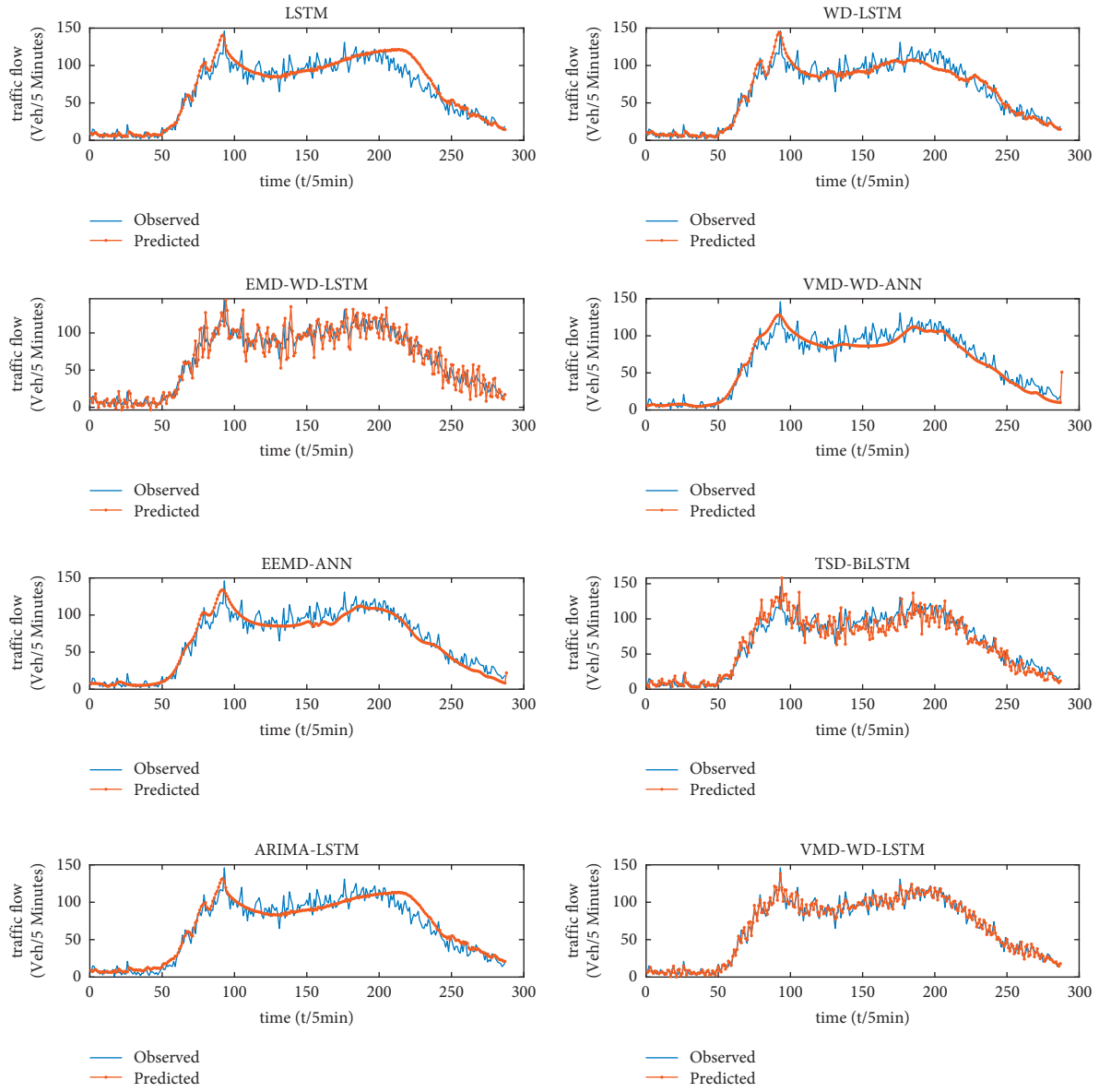


Figure 23: Predicted curves of different models using the traffic flow data of detector D.

Table 7: The prediction errors of different models using the traffic flow data of detector D.

Prediction models	RMSE (veh/5 min)	MAE (veh/5 min)	MAPE (%)
LSTM	11.462	8.290	22.69
WD-LSTM	9.795	7.382	21.316
EMD-WD-LSTM	13.976	10.824	32.833
VMD-WD-ANN	9.909	7.702	20.667
EEMD-ANN	9.688	7.546	20.461
TSD-BiLSTM	14.898	10.929	27.831
ARIMA-LSTM	10.269	7.936	28.417
VMD-WD-LSTM	**5.623**	**4.301**	**12.500**

Bold indicates the best value of each evaluation index.

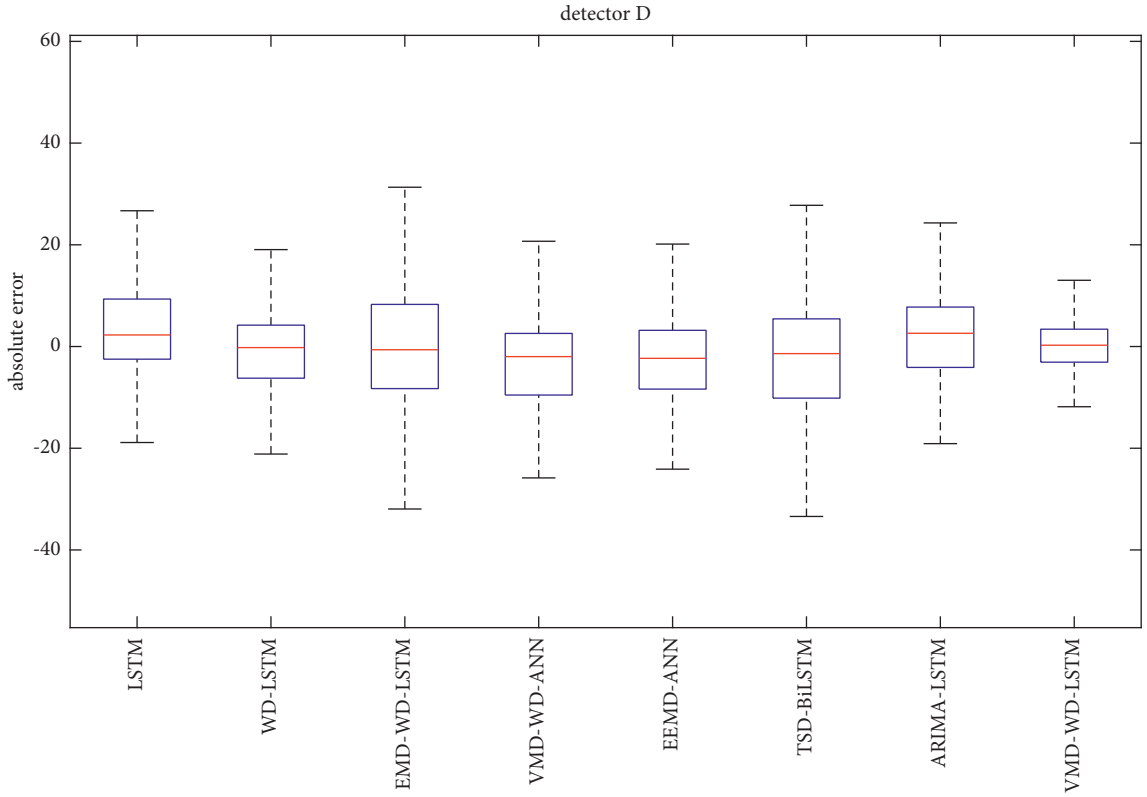


Figure 24: Boxplots of absolute error of different models (detector D).

4. Conclusions

The randomness and complex periodic characteristics of traffic flow make it difficult to predict traffic flow. To solve this problem, this study proposes a VMD-WD-LSTM prediction method, which includes data denoising, signal decomposition, and data prediction. Specifically, this model first decomposes the original traffic flow data into multiple subsequences by the variational mode decomposition (VMD) method. Since the VMD method can control the number of decompositions, we determine the appropriate number of decompositions by sample entropy. The subsequences obtained by these decompositions can reflect different characteristics of the original signal. The second step is to conduct wavelet threshold denoising for each decomposed subsequence. Compared with the denoising method for original data, denoising multiple IMFs can better reduce the impact of noise on prediction results. Finally, the denoised IMF component is predicted by the LSTM model, and the final prediction results are obtained by combining the predicted values of each component.

In order to evaluate the denoising effect and prediction performance of the VMD-WD-LSTM model, this study first compares the results obtained by direct prediction of the original data with the results predicted by the VMD-WD method after denoising. From the prediction results of the three prediction models of ARIMA, ANN, and LSTM, the denoising of the original signal by the VMD-WD method can improve the prediction performance, and the improvement effect on the LSTM model is the most obvious. In addition, the performance of the proposed model is also compared with LSTM, WD-LSTM, EMD-WD-LSTM, VMD-WD-ANN, EEMD-ANN, TSD-BiLSTM, and ARIMA-LSTM methods on four detectors in two different open-source datasets. The results show that the VMD-WD denoising method can better reduce noise pollution. On the basis of data denoising with VMD-WD method, the LSTM model can accurately predict the characteristics of traffic flow data and obtain excellent prediction results.

In summary, the VMD-WD-LSTM model proposed in this study can realize the feature decomposition of the original traffic flow data and the prediction of the traffic flow on the working day. Accurate prediction of traffic flow can effectively avoid traffic congestion. In the face of the upcoming congestion, we can make early warnings and take evacuation measures. At present, the method proposed in this paper still has some shortcomings. First, this study only analyzes the traffic flow of working days in terms of data selection and does not analyze the changes in traffic flow at weekends, holidays, and special periods. In addition, different weather conditions, road conditions, and spatial-temporal correlation will affect the prediction results. The method proposed in this study only analyzes the traffic flow changes during the nonholiday period and does not consider the special weather and road conditions. Obviously, it is not feasible to predict the traffic flow affected by other factors by traffic flow in general. Therefore, in the future work, this study will focus on the analysis of traffic flow changes in different scenarios and study the impact of different external factors on traffic flow. Another future research direction is

the analysis and prediction of traffic flow changes in specific periods of the day, especially in the morning or evening when traffic pressure increases. It is also important to predict traffic flow in this period alone.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the MOE (Ministry of Education of China) Project of Humanities and Social Sciences (grant no. 21YJC630110) and the Key Research and Development Program of Shandong Province (Soft Science) Project (grant no. 2020RKB01793).

References

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[2] Y. Xie, Y. Zhang, and Z. Ye, "Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 5, pp. 326–334, 2007.

[3] Y. Peng and W. Xiang, "Short-term traffic volume prediction using GA-BP based on wavelet denoising and phase space reconstruction," *Physica A: Statistical Mechanics and Its Applications*, vol. 549, 2020.

[4] H. Xiao, H. Sun, B. Ran, and Y. Oh, "Fuzzy-neural network traffic prediction framework with wavelet decomposition," *Transportation Research Record*, vol. 1836, 2003.

[5] J. Tang, X. Chen, Z. Hu, F. Zong, C. Han, and L. Li, "Traffic flow prediction based on combination of support vector machine and data denoising schemes," *Physica A: Statistical Mechanics and Its Applications*, vol. 534, Article ID 120642, 2019.

[6] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[7] X. Chen, H. Chen, Y. Yang et al., "Traffic flow prediction by an ensemble framework with data denoising and deep learning model," *Physica A: Statistical Mechanics and Its Applications*, vol. 565, Article ID 125574, 2021.

[8] X. Chen, J. Lu, J. Zhao, Z. Qu, Y. Yang, and J. Xian, "Traffic flow prediction at varied time scales via ensemble empirical mode decomposition and artificial neural network," *Sustainability*, vol. 12, no. 9, Article ID 3678, 2020.

[9] Z. Wu, J. Feng, F. Qiao, and Z.-M. Tan, "Fast multidimensional ensemble empirical mode decomposition for the analysis of big spatio-temporal datasets," *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, 2016.

[10] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.

[11] H. Liu, X. Mi, and Y. Li, "Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM," *Energy Conversion and Management*, vol. 159, pp. 54–64, 2018.

[12] G. Zuo, J. Luo, Ni Wang, Y. Lian, and X. He, "Decomposition ensemble model based on variational mode decomposition and long short-term memory for streamflow forecasting," *Journal of Hydrology*, vol. 585, Article ID 124776, 2020.

[13] J. Shi, Y.-B. Leau, K. Li, and H. Chen, "Optimal variational mode decomposition and integrated extreme learning machine for network traffic prediction," *IEEE Access*, vol. 9, pp. 51818–51831, 2021.

[14] M. Van Der Voort, M. Dougherty, S. Watson, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[15] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.

[16] J. Guo, W. Huang, B. M. Williams, and Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[17] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin, "A noise-immune Kalman filter for short-term traffic flow forecasting," *Physica A: Statistical Mechanics and Its Applications*, vol. 536, Article ID 122601, 2019.

[18] N. Zhang, Y. Zhang, and H. Lu, "Seasonal autoregressive integrated moving average and support vector machine models: prediction of short-term traffic flow on freeways," *Transportation Research Record*, vol. 2215, no. 1, 2011.

[19] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang, "Accurate multisteps traffic flow prediction based on SVM," *Mathematical Problems in Engineering*, vol. 2013, Article ID 418303, 8 pages, 2013.

[20] T. D. Toan and V.-H. Truong, "Support vector machine for short-term traffic flow prediction and improvement of its model training using nearest neighbor approach," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2675, no. 4, pp. 362–373, 2021.

[21] M. Özuysal, S. P. Çalışkanelli, S. Tanyel, and T. Baran, "Capacity prediction for traffic circles: applicability of ANN," *Proceedings of the Institution of Civil Engineers-Transport*, vol. 162, no. 4, pp. 195–206, 2009.

[22] K. Kumar, M. Parida, and V. Kumar Katiyar, "Short term traffic flow prediction in heterogeneous condition using artificial neural network," *Transport*, vol. 30, no. 4, pp. 397–405, 2015.

[23] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proceedings of the 2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)*, pp. 153–158, IEEE, Chengdu, China, December 2015.

[24] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2627–2636, 2021.

[25] C. Ma, G. Dai, and J. Zhou, "Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 1–10, 2021.

[26] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[27] P. Wu, Z. Huang, Y. Pian, L. Xu, J. Li, and K. Chen, "A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting," *Journal of Advanced Transportation*, vol. 2020, Article ID 8863724, 15 pages, 2020.

[28] L. Li, H. Zhou, H. Liu, C. Zhang, and J. Liu, "A hybrid method coupling empirical mode decomposition and a long short-term memory network to predict missing measured signal data of SHM systems," *Structural Health Monitoring*, vol. 20, no. 4, pp. 1778–1793, 2021.

[29] H. Huang, J. Chen, R. Sun, and S. Wang, "Short-term traffic prediction based on time series decomposition," *Physica A: Statistical Mechanics and Its Applications*, vol. 585, no. 1, Article ID 126441, 2022.

[30] L. Li, X. Qu, J. Zhang, H. Li, and B. Ran, "Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network," *Applied Soft Computing*, vol. 73, pp. 921–932, 2018.

[31] S. Lu, Q. Zhang, G. Chen, and D. Seng, "A combined method for short-term traffic flow prediction based on recurrent neural network," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 87–94, 2021.

Hindawi
Complexity
Volume 2021, Article ID 9629331, 17 pages
https://doi.org/10.1155/2021/9629331

Research Article

Agility Factors' Analyses Framework in Project-Oriented Organizations through a Sustainability Approach in Large Projects Case Study: Isfahan Municipality

Ahmadreza Tahanian,[1] Hasan Haleh ,[2] Farhad Etebari,[1] and Behnam Vahdani[1]

[1]*Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[2]*Industrial Engineering Department, Golpayegan University of Technology, Golpayegan, Iran*

Correspondence should be addressed to Hasan Haleh; hasanhaleh40@gmail.com

Received 27 June 2021; Revised 1 August 2021; Accepted 4 August 2021; Published 17 September 2021

Academic Editor: Michela Gelfusa

In recent years, the project management concept is coupled with sustainable development. In terms of profitability and adapting to environmental changes, generating value for the customer and responding to market needs is a challenging issue. To gain profit and success, a project should adhere to agility factors. With respect to the importance of project execution success, corresponding to the sustainability dimensions and by developing agility parameters, this study provides a framework for clustering and analyzing "large projects" based on agility factors in project-oriented organizations through a sustainability approach and by applying Quality Function Deployment. To this end, critical factors of project success and then agility factors of the project-oriented organizations have been identified. Thereafter, the importance of these two major components has been measured by executive managers in Isfahan Municipality and academic experts. Afterward, the agility factors' weights have been calculated based on extracted sustainability factors' weights, which have been gained according to the project's critical success factors and by applying quality function deployment. By determining the agility factors' clusters in the project-oriented organizations and calculating their importance weights, the first cluster that contains project communication management, organizational culture, and contracts management gains the maximum weight of importance.

1. Introduction

In today's business world, in many projects, considering the very high level of uncertainty in all aspects of the activities, priorities, and latencies, activity connections, and also resources correlation, applying the traditional techniques and instruments in project planning is a hard task. In such cases, mere project planning improvement would not yield satisfactory output. For any type of project, an agile model can promote quality and productivity in a specific time and resource framework [1]. The purpose of organizational agility (OA) as a major contributive dynamic ability is for realizing the environmental changes and respond effectively and efficiently to the changes therein. Being engulfed with environmental, highly dynamic, complex, and uncertain changes, organizations face serious threats as to their survival in a highly competitive market. Such business atmosphere being equipped with mechanisms which allow the organizations to respond appropriately to the identified changes will directly be connected with better performance in order to taking the new opportunities and resources in a competitive world. Thus, agility is a concern for the survival and success of the organization [2]. Applying the principles of agility to the project as an interim organization is a necessity to deal with environmental uncertainties. Agility is mostly concentrated on the project implementation phase. This approach focuses on the accuracy of planning with respect to the details of planning, better risk management, and more customer affair on its agenda [3].

Innovative and sophisticated technologies have introduced dramatic changes in industrial products and social services [4]. Project management is obligated to manage both the simple and complex activities through modern approaches that are more efficient than the traditional approaches. A project manager seeks to complete the project within time, cost, and resources framework and by generating value for the client [5]. The project management system faces serious and expanding challenges as to value generation response to changes and profitability in business environment. To overcome this challenge, the project needs performance agility. In this century, achieving sustainable development is a blanket and essential challenge for all organizations. Sustainability is a concept on which the researchers and entrepreneurs concern in order to improve and integrate it with the field of project management at different social and business levels [6]. Sustainability is defined as an innate potential through which the long-term, resource-related risks, fluctuations in energy costs, debts, product costs, and pollution are reduced and the waste management is improved. The equilibrium among economic development, supervising environmental issues, and social justice is another definition of sustainability [7], provided that the importance weights of all three fields of economy, environmental conformity, and social balance are considered as equal [8]. The objectives of promoting economic growth, social health, and rational natural resource utilization cannot be actualized without considering the effects of the mentioned components. As to organizational success, sustainability would provide balance and harmony among these three components. This concept is named *triple bottom lines* (TBL) and offers different values and perspectives to measure the success of organizations in the three areas of economics, society and the environment. The three main components of sustainability are population, world, and profits. Profit is directly related to economics, population contains society and the organizational environment, and the other component modifies the environmental consequences. The first component consists of the economic environment, and the second is the environmental effects caused by the organization's activities [9]. The increase in the projects' success depends on the managers' improvement in approaches towards the project objectives. Rational decisions drawn by management are highly contributive in project management and the outcomes therein [10]. For this purpose, project management, the core operations of a business, is not separated from sustainability concept [11]; consequently, the field of project management requires project managers to be fully responsible in project sustainability (Silvius and Brink, 2012). The products, services, technologies, projects, and organizations each on its own constitute a sustainable system. According to the project complexity, to deliver the project results subject to sustainability conditions, the necessary instruments and techniques must be properly coordinated [12].

Dynamism in today's economy requires projects that would introduce major changes in people's daily conducts. Due to the current global crisis and gradual resource depletion, as we go forward, project managers are obligated to develop new and innovative perspectives on sustainability and devise measures to achieve the appropriate levels therein. As to time, the nature of the project is periodic and to status is discrete, while the nature of sustainability is unified. To achieve sustainable development, the project activities must follow economically transparent, socially acceptable, and environmentally safe pattern [13]. Project managers often try to improve the issues related to measurement, success, and factors influencing the project performance and seek to improve them [13]. In project management perspective, the critical success factors consist of features, circumstances, and variables, the effect of which on project success are oriented towards sustainability [14].

A brief review of the related literature, in the context of modern business, would reveal the importance of agility, as a key approach as to adapting to the changes in the business world, responding to the customer and generating value therein, in project-oriented organization. Expanding sustainability, that is, observing its principles with respect to resource utility, which should correspond with the credible sources of project management, is essential. Depending on the features, a large project is a type of project that typically has a longer duration (it might be run for more than six months), larger teams of staffs (include more than 25–30 members and even may require the support staff), greater budget and resource allocation, and more tasks complexity, including many tasks having to be done concurrently. A project-oriented organization that manages and executes large projects can be a complex network with high rate of interactions. Hence, the complexity of such systems leads the organization to be agile while they are following the sustainability principles and manages the projects on the basis of critical success factors of project management.

In this study, a framework is proposed for analyzing agility factors in project-oriented organizations with a sustainability approach in big projects by applying quality function deployment (QFD). Identifying agility factors in project-oriented organizations based on the main project success and sustainability factors is a major step in evaluating the project, based on the agility and sustainability approach, which in turn would promote adaptation to the changes in business environment and customer demands. Determining the factors' weight and their clustering in determining the important clusters, which would contribute in planning and goal setting, is essential. This can be accomplished by applying the principles of agility and sustainability components.

This article is organized as follows: the literature is reviewed in Section 2; the project management is explained in Section 3; agility in project-oriented organization is introduced in Section 3.2; the sustainability is discussed in Section 3.3; QFD is applied in Section 3.4; clustering is analyzed in Section 3.5; the research method is detailed in Section 4, the framework is implemented in the subject organization in Section 5, and the article is concluded in Section 6.

2. Literature Review

To review the research literature, the connected researches have been found based on the keywords and the journals

with the related issues. The select studies run on the main factors of project success are tabulated in Table 1, the select studies run on sustainability are tabulated in Table 2, and the select studies run on organizational agility are tabulated in Table 3.

What is deduced from reviewing the related articles is that in today's competitive and changing business world, customer satisfaction is accredited to the organization credibility, quality and speed in response, and agility in project performance. Project-oriented organizations are fully aware of the need to incorporate agility dimensions in project management if adaptation to changes in the environment and response-specified time, cost, and resources are sought. Organizations are obligated to assess the resource planning and the effects of a project as to its economics and profitability, with respect to social dimension next to its effects in environmental sense to correspond with sustainability. The different aspects of organization and management approaches as to resource allocation and planning for project control, resource protection, environmental effects, economic performance, and social effects of the project are subject to direct effect of sustainability.

Identifying the agility indices and dimensions in project-oriented organizations, based on accurate infrastructure according to the critical success factors of project and the dimensions of sustainability in three areas of economic, social and environmental therein, has been somewhat neglected. In this context, while evaluating the project based on the mentioned factors, the project manager should strengthen the weak indices and generate value, respond to the customer and meet his/her satisfaction, if achieving sustainability, profitability, credibility and social prestige and effects are sought. This in turn would allow successful performance to assure environmental and social sustainability. In all these efforts, though the project was delivered successfully, still the agility aspect is missing in the run studies.

3. Theoretical Background

3.1. Project Management. A project consists of human and nonhuman resources, in a temporary organizational framework with the objective to accomplish a specific task [5]. Organizations, today, perceive a project as a manner of organizing tasks. In most industrial, commercial, and governmental organizations, project management is perceived as an instrument to accomplish objectives [30]. The project assists the organization to define and select a set of activities that would generate positive results therein. This phenomenon can be applied in economic, marketing, or technical areas. The project manager is obligated to manage the project through a set of instruments and methods without disturbing the regular routine operations of the organization. Project management activities consist of defining task requirements, planning and scheduling task implementation, allocation progress, monitoring, and keep project planning online. As to project delivery, the responsibility of the project manager is to have met all budget and desired operational and quality standards [31]. Project management can be considered as a temporary effort in separate fields of activities with a specific beginning and end time. Given the existing global economic status, projects must be managed and controlled in a periodical manner. Time, cost, and objective achievement level are considered as the indices of success or failure for organizations that should be supported per project beginning from the structure to the budget. Supervision and control in the implementation phase are also necessities [16]. Project management as an effective controlling instrument is expressed in multiactivity endeavors [32].

3.2. Agility in Project-Oriented Organizations. Today, organizations face varying, dynamic, complex, and uncertain circumstances in these highly competitive markets that pose a threat to organizations [2]. The innovative and sophisticated technologies have introduced dramatic changes in industrial products and social services [3]. Business and management models are essential to support Industry adoption and foster sustainable value creation and competitiveness [4]. The need for an efficient management strategy is evident [32]. Organizational agility enables an organization to understand the environmental changes and provide an appropriate response as to adapting it as a management mechanism [2]. In a sense, these environmental features are the challenges confronting project management [33]. The principles of project management based on traditional techniques are based on regular planning and control practices. Developing an integrated network, consisting of customers, suppliers, supervisors, authorities, and political institutions and competitors is a proper manner of responding to environmental changes, something impossible in traditional project management format.

3.3. Sustainability in Project-Oriented Organizations. The ongoing major changes at global scale have made sustainable development an important strategic measure. Sustainable development seeks to protect, develop, and balance economic, social, and environmental objectives to the available resources to assure the comfort and convenience of the generations to come. The correlation between these objectives and resources is high and important. Today, next to economic values, countries must consider the social benefits and environmental consequences as well. Climate change threatens the future significantly to a certain extent. The natural resources are depleting recklessly, that is, accessing them will be difficult if not impossible for the next generation [34]. Nowadays, organizations define their activities and business as projects and seek to improve their success. Drawing managerial decisions are contributive in project management and its outcomes [10]. Sustainable development addresses humanity's aspiration for a better life while observing the limitations imposed by nature [35]. Dynamism in today's global economy is based on the available projects with the objective to make major changes in people's lives. The current global crisis and extending resource depletion force project managers to develop innovative perspectives on sustainability and assure its continuity. The nature of any project is temporary and discrete, accomplishable through different techniques and methods. Sustainability is not an isolated concept, and it is achieved by

Table 1: Review on the related literature of critical success factors.

Row	Authors	Year	Objective	Method and tools	Description
1	Ahmadabadi, and Heravi [15]	2019	Assessing the CSF in state/private sector joint ventures, the national highways	RLS	All operations at all stages are supervised, and the project is evaluated against critical situations as to its success
2	De melo moura [16]	2016	Identifying the main factors as to support information system's projects	—	PMCSFS are contributive in optimum performance in information systems
3	Orouji [17]	2016	A review run on articles on PMCSFS	—	Assessing the articles published from 1978 to 2015 as to their CSF vs. failure in 6-sigma projects
4	Costantino et al [18]	2015	Selecting the projects for forming a portfolio by applying ANN according to CSF	DSS ANN	To design, develop, and test the decision-making support systems for predicting project functionality and applying ANN to form CSF groups according to risk rate
5	Alias et al 2014 [19]	2014	Determining the CSFs in project management in a comprehensive framework	Statistical tests	The connection rate between CSF and project functionality is measured and a comprehensive method consisting of project management, methods, human factor, external issues, and project-related features are presented
6	Abdulrahman et al [5]	2014	Determining the essential features in project success and the techniques that lead to project management success	—	Project management theory, project systems, and the CSFs are addressed
7	Thi and swierczek [20]	2010	Measuring project success based on CSFs in project management and run case studies in Vietnam	Regression analysis	Completing the project life cycle that is directly related to positive relations and success. In the project execution, the external consistency affects and organization support is compensated for
8	Naoum et al [19]	2004	Identifying the critical and noncritical factors affecting the project success in big construction projects	—	A management led by project and execution management

applying project activities run in continuous and transparent economic, social, and environmental context [13]. To deliver a project yield subject to sustainability, the required instruments and techniques should be in accordance with the complexity of the project [36]. Sustainability theme is involved in the integration of the three aspects: economic, social, and environmental aspects, which extends in project management and project delivery, followed by enhancing the project system and the management therein [37].

3.4. Quality Function Deployment. Quality function deployment is one of the qualitative tools contributive in providing services and producing products that would meet consumers' requirements. As one of the new methods in engineering, quality function deployment begins from running studies on market and identifying consumer services while identifying users' demands and seeks to apply them in the design process. The basic insight of this method is to convert the qualitative features into technical features, production processes, and requirements, by applying four matrices, in which the house of quality is the most important part. This matrix has a number of numerical inputs, which after the qualitative identification of the features, statistical methods will be applied to calculate the relative importance of the features on the basis of the consumers' opinions and experienced people ideas [38]. Hence, QFD promotes the systematic translation of customer needs and requirements into design requirements and evaluates alternatives and their impacts [39].

3.5. Clustering. Clustering and classification are among the most efficient methods in data analysis. Extracting the patterns through grouping individuals and variables is one of the main objectives consisting of many methods and patterns applicable in different scientific fields because human beings when encountering complex issues instinctively seek to compare and divide the given data as different groups. These methods are applied in different scientific fields like medicine, biology, management, planning, data mining, information discovery, and new structure in speech and images and environmental and natural sciences [40, 41].

Massive data, next to being beneficial for organizations and individuals, are problematic when it comes to their analysis. Data mining techniques is a solution in extracting beneficial information and correlations therein. Clustering, with its different methods, is one of the powerful instruments to forces these analyses. Identifying clusters or areas with dense population in a multidimensional data set through different techniques and algorithms is one of the

Table 2: Review on the related literature of sustainability factors.

Row	Authors	Year	Objective	Method and tools	Description
1	Mavi, and Standing [21]	2018	Introducing project management's CSFs in the five project, project management, organization external environment, and sustainability	Fuzzy DEMATE L-ANP	By applying fuzzy DEMATEL organization and sustainability environment are recognized as the causes, and project management is recognized as the effect and ANP is also applied for weighing the subcriterion
2	Dobrovolskienė, and Tamošiūnienė [22]	2016	Measuring construction industry projects sustainability in Lithuania	MCDM	The case study results show that from industrial experts' point of view, 15 indicators are more important than the whole
3	Martens and Carvalho [23]	2017	Assessing key factors in project management in all its dimensions	Factorial exploration analysis	Results show that innovative and sustainable business model, stockholders management, economy and competitive management, and finally, environmental policies and resources storage are paramount factors
4	Daneshpour [24]	2015	Assessing the management and the aspects of combining project management and sustainability	—	Organizations' awareness to know the value of sustainability and to achieve sustainability dimensions in practice
5	Amiril et al [25]	2014	Evaluating sustainability factors and project performance in transportation infrastructure	—	Sustainability factors and project performance can be fit in environmental, economic, resource utility, and project management categories
6	Silvius and schipper [26]	2014	Assessing sustainability as one of the project management competencies and analyzing the gaps therein among the present project managers	—	By identifying this gap, procedures are proposed to minimize it in addition to introduce new measures and standards in this context as to organization sustainability

Table 3: Review on the related literature of agility factors in project-oriented organizations.

Row	Authors	Publication year	Objective	Method and instrument	Description
1	Pocter et al [27]	2019	Providing a pattern for competition scram network in agile project management	—	Integrating agility methods and principles in multidisciplinary cooperation require high flexibility and training patterns through implementation
2	Fernandez and Fernandez [28]	2018	A review run on articles focused on agile project management and project success therein	—	The traditional and modern project management steps are compared in this context. It is found that APM is effective in project success
3	Chen et al [29]	2018	Assessing the agility and sustainability concepts as contradicting or complementary concepts	—	Generating value for customer and adopt it with the environments changes next to seeking sustainability in both the contexts

major issues in multidimensional data analysis. Finding the proper clustering algorithm or/and the most optimal clusters, in a reasonable time, is one of the major challenges in this field [42].

Clustering consists of grouping similar samples into one data mass. The essential point here is data distribution into different K groups with similar data and the same for the dissimilar. This difference is defined based on distance measurement criteria. Compared to grouping, clustering is an unsupervised grouping process, where the groups are not predefined. Clustering is an indirect data mining operation [42].

In most data mining methods, like the decision trees and neural networks, the process begins with an educational set through which it is sought to devise a model that would make the data in segments and then predicts the appropriate set for a new data. On the contrary, in the clustering, there exist no initial group and the variables are not divided into two independent and dependent groups. In clustering, the focus is on groups of objects that are similar, which when

discovered, their behaviors can be identified better, indicating better decision making [43].

In hierarchical clustering, clusters are expressed as a tree named dendrogram. These algorithms can be in up-down divider or bottom-up adder format, where, the divider must specify which two clusters are divided into two parts and how is this done, while in the additive, the algorithm begins from one cluster and then, the two clusters merge to obtain a make a throughout cluster [44].

4. Methodology

Project success depends on the delivery of the service and product, creating value for the customer while adapting to the changing world of modern business. Next to the economic approach, the sustainable development that covers the project revenue and profitability, attention must be directed on environmental consequences, resource protection, and the project performance in the social dimension. In this study, a framework is provided for analyzing the agility factors in project-oriented with respect to sustainability approach in large QFD projects. Then, the proposed method is solved as a case study as a sample for large projects. The applied algorithm and description of the phases is illustrated in Figure 1.

The research method is explained separately for each phase as follows:

Phase 1: identifying the key factors of success in project management, sustainability, and agility in project-oriented organizations

This phase is accomplished in three steps to identify project management critical success factors, agility factors in project-oriented organizations, and sustainability factors.

Step 1: a review made on the research literature to identify project management critical success factors.
In this step, project management critical success factors are identified through reviewing the research literature, and then, final indices are selected by interviewing the experts.
Step 2: a review made on the research literature to identify agility factors in project orientation organizations.
In this step, agility factors in project-oriented organizations are identified through reviewing the research literature, and then, final indices are selected by interviewing the experts.

Step 3: a review on the research literature to identify sustainability factors.
In this step, sustainability factors are identified through reviewing the research literature, and then, final indices are selected by interviewing the experts.

Phase 2: measuring the critical success factors in project management, sustainability, and agility in project-oriented organizations

In this phase, project management critical success factors' weights, agility factors' weight, and sustainability factors' weight are determined based on experts' point of view.

Step 4: measuring the indices in each approach from the experts' perspective, through academic experts' perspective.

(1) Project management critical success factors' weight determination: each factor's weight is determined based on project management and academic experts' point of view and using the questionnaire that is prepared based on Likert scale.
(2) Agility factors' in weight determination: each factor's weight is determined based on project management and academic experts' point of view and using the questionnaire that is prepared based on Likert scale.
(3) Sustainability factors' weight determination: each factor's weight is determined based on project management and academic experts' point of view and using the questionnaire that is prepared based on Likert scale.

Step 5: measuring the indices in each approach from the experts' perspective, based on the organizational experts' perspective.

(1) Determining the importance weight of critical success factors in project management field: the weight of each factor is determined based on Isfahan Municipality project management staffs' point of view and using questionnaire that is applied Likert scale.
(2) Determining the importance weight of agility factors in project-oriented organizations: the weight of each factor is determined based on Isfahan Municipality project management staffs' point of view and using questionnaire that is applied Likert scale.

Phase 3: calculating the agility weight factors in project-oriented organizations based on the critical success factors in project management with a sustainability approach in projects by applying two-stage QFD

Step 6: calculating the sustainability factors' weights based on critical success factors in project management applying QFD
For this purpose, the house of quality matrix is applied to determine the relative importance of sustainability factors based on project management critical success factors of project management. To this end, as Table 4 shows that QFD matrix is formed to rank sustainability factors based on the project management critical success factors and to signify the relative weight of sustainability factors.
Step 7: calculating the agility factors' weight in project-oriented organizations based on balanced sustainability factors by applying QFD

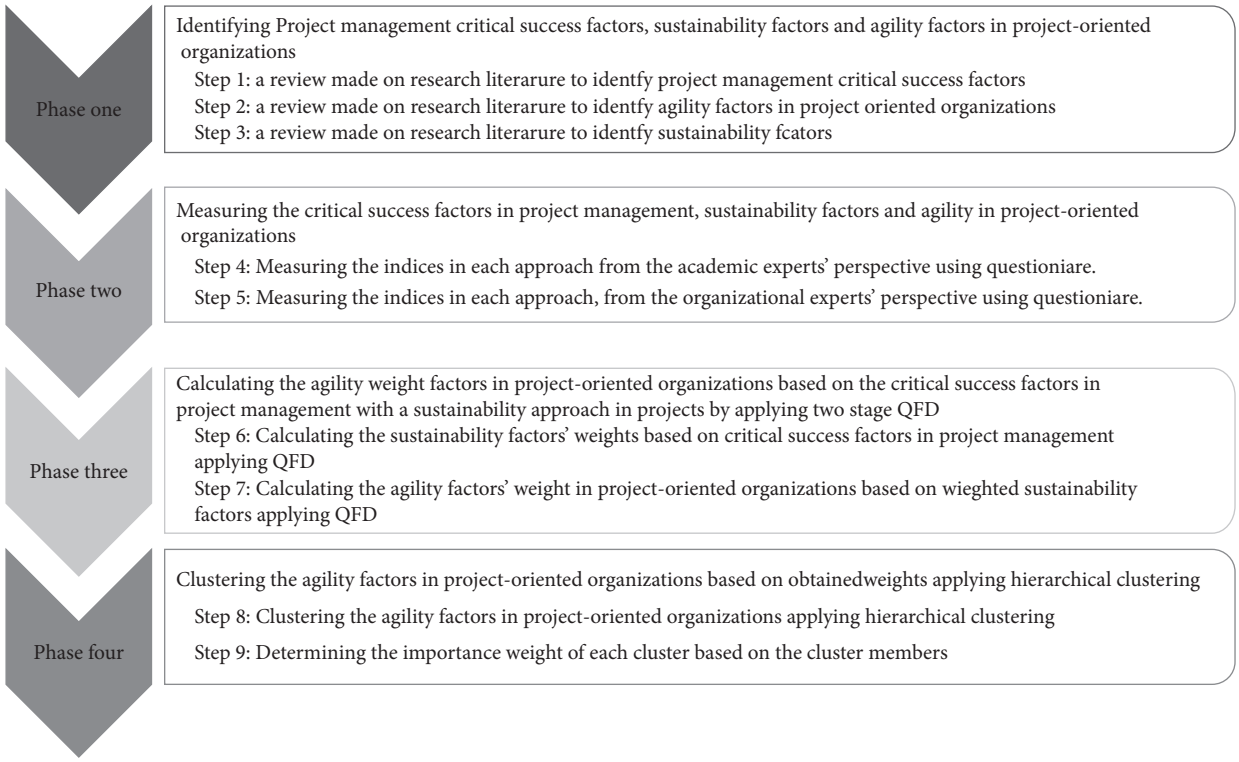


Figure 1: Research method.

Table 4: QFD matrix: PM CSFs-Sustainability factors.

Critical success factors	Sustainability factors	
	Critical success indices weight	Sustainability factors
Project management critical success factors	S_{11}. . . S 1n	C 1
	.	.
	.	.
	.	.
	S n1. . . S nn	C n
Weighted summation of the sustainability factors according to project management critical success factors		S c1. . . S cn

For this purpose, the house of quality matrix is applied to determine the relative importance of agility factors in project-oriented organizations based on sustainability factors. To this end, as Table 5 shows, QFD matrix is formed to rank sustainability factors based on the project management critical success factors and to signify the relative weight of sustainability factors.

Phase 4: clustering the agility factors in project-oriented organizations based on obtained weights applying hierarchical clustering

In this phase, agility factors in project-oriented organizations are clustered applying a hierarchical clustering model and based on importance weights that have been gained in second and third phase.

Step 8: clustering the agility factors in project-oriented organizations based on weight indices obtained by applying hierarchical clustering

In this step, the importance value of each cluster is determined according to the clustering results that have been gained in the previous step and based on the factor members' importance weight of each cluster.

Step 9: determining the importance of the agility factors clusters in project management based on the member factors of each cluster.

In this step, the importance of each cluster is determined based on the factors that are counted as a cluster member in the previous step and then the most important cluster is signified.

5. Implementation of the Proposed Framework in Large Projects of Isfahan Megacity Municipality

This research has been done with the aim of clustering and analyzing large projects based on agility factors in project-

Table 5: QFD matrix: PM CSFs-Sustainability factors.

Sustainability indices	Agility indices	
	Sustainability factors weight	Agility factors
Sustainability factors	A 11. . . A 1n	C 1
	.	.
	.	.
	.	.
	A n1. . . A nn	C n
Weighted summation of the agility indices weight according to sustainability indices		A s1. . . A sn

oriented organizations and considering the sustainability attitude and applying the Quality Function Deployment. The study has been done in the period of 2018–2019 and in the city of Isfahan as a spatial range. The proposed framework has been applied in Isfahan Municipality and for the large projects.

Phase 5: identifying the key factors of project management success, sustainability, and agility in project-oriented organizations.

Step 1: reviewing the related literature and identifying the project management critical success factors

By reviewing the literature and interviewing academic experts and project management, the main factors of success in project management are identified, and the final indices are determined according to the experts' opinions (Table 6).

Step 2: reviewing the literature and identifying the agility factors in project-oriented organizations

By reviewing the related literature and interviewing academic experts and project management, agility factors in project-oriented organizations are identified, and the final indices are determined based on the opinion of experts (Table 7).

Step 3: reviewing the related literature and identifying the sustainability factors

By reviewing the research literature and interviewing academic experts and project management, the factors of sustainability are identified and the final indices are extracted according to the opinion of experts (Table 8).

Phase 6: measuring the importance of project management critical success factors, sustainability factors, and agility factors in project-oriented organizations

Step 4: this measuring consist of three approaches where a questionnaire based on project management and academic experts' opinions is devised as the instrument.

(1) Determining the project management critical success factors' importance weight in project management: through the devised questionnaire based on the Likert scale (1 to 5, that is, a low importance to high importance), which has been distributed among 13 project managers and university professors in this field with 7 years of experience and published scientific articles.

(2) Determining the importance weight and agility factors in project-oriented organizations: the same procedure above is followed.

Step 5: measuring the importance weight of factors in all three approaches through the devised questionnaire based on project management experts' viewpoint in the field of opinions in Isfahan Municipality.

(1) Determining the importance weights of the project management critical success factors: it has been determined based on the project management staff viewpoints and through the questionnaire with Likert scale (1 to 5, that is, low importance to 5 high importance), which has been distributed among 52 staff members.

(2) Determining the importance weight of agility factors in project-oriented organizations: based on the project staff point of view and through a questionnaire with Likert scale (1 to 5, that is, a low importance to 5 high importance), which has been distributed among the employees in this field, including 52 staff. Table 9 shows the geometric mean of agility factors' weights in project-orient organizations.

Phase 7: calculating agility factors' weights in project-oriented organizations based on project management success and sustainability approach

In this phase, the agility factors' weights are calculated based on the extracted weights of sustainability factors according to the project management critical success factors, and then, agility factors' weights are calculated based on sustainability factors' weights applying QFD.

Step 6: calculating the sustainability factors' weights based on project management critical success factors by applying QFD

For this purpose, the house of quality matrix is applied to determine the relative importance of sustainability factors based on project management critical success factors. The house of quality matrix is configured to rank sustainability factors based on project management critical success factors and to determine the relative weight of sustainability factors, which are detailed in Table 10.

Step 7: calculating the agility factors' weights in relation to sustainability factors' weights in project-oriented organizations by applying QFD

The house of quality matrix is configured to rank agility factors based on stability factors and determining the relative agility weights' indices in project-oriented organizations, which are shown in Table 11. For matrix columns, agility factors are also considered in project-oriented organizations. The QFD matrix designed by the experts is completed, and its geometric mean is inserted in the matrix cells. By multiplying the stability factors' weights in each one of the columns and normalizing the

Table 6: Critical success factors of project management.

Row	Indices	The critical success factors in project management
1	Project management related	Leading and managing the project team
2		Applying innovative management techniques and innovative ideas [44]
3		Applying the project management standard [45]
4		Exercising strong project management leadership [46, 47]
5		Project management responsibility and commitment [14, 19, 31]
6		Senior and executive management support [14, 19, 31, 35]
7	Project team-related	Employing experienced experts in the project management team [42, 46, 48]
8		Ability to perform team work [31]
9		The expertise and skillfulness of the project management team [44, 46, 48]
10		Appropriate reward and penalty systems [14, 19]
11		Self-learning teams [44]
12		Confidence and consent among team members
13	Project-related	Project objectives access level
14		Safety first
15		Determining the requirements at the beginnings of the project and provide the details for each phase
16	Consumer-related	Customer participation level [20, 48]
17		Consumer satisfaction [44]
18		Ability to conduct soft management for the beneficiaries in controversies and competitions therein [20, 48]
19	Quality-related	Supervision and control of the project [14, 20, 48]
20		Transparent planned objectives and strategies [16, 44]
21		Contract articles' actualization rate [19, 44]
22		Accomplishing the agreed quality [44]
23	Time management	Orderly discourse with the beneficiaries during the project [45]
24		Conducting regular meetings between project management and the staff [44]
25		On-time project delivery management [19, 48]
26	Cost management	Proper management of the project physical resources [44, 46]
27		Project conformity level with the designated budget [18, 48]
28		Elimination of unnecessary in project phase planning

Table 7: Agility factors in project-oriented organizations [43].

Row	Field	Index
1	Communications management project	Horizontal communications vs. hierarchical
2		Communicational coherency rate in the project
3		Strong and orderly project staff communication
4		Providing the grounds for consultant communication with the beneficiaries
5		Access to valuable data on time
6	Organizational culture	Access to classified database
7		Sharing knowledge and information in the project
8		Allowing project managers and staff to draw decisions
9		Organizational perspective of the project management
10		Decentralized decision making
11	Project management	Project management transparency
12		Accurate management method execution
13		Accurate management method execution level
14		Applying new management method
15		Applying proper project management
16		Leadership through brisk managers
17	Project scheduling and control	Project structure to project requirements' ratio
18		Accurate short-term planning
19		Proper scheduling with respect to accessible resources
20		Accurate predictions of project requirements
21		Simulations execution of activities
22		Project schedule modification
23		Provisional modification upon need

Table 7: Continued.

Row	Field	Index
24		Project risk management
25		Accurate perception of project format
26		Project budget estimation
27	Project speed	Estimating project requirements in the shortest time
28		Rapid decision drawing
29		Rapid response to the client and the beneficiaries
30		Providing training in the shortest time possible
31		Implementing the simple and practical rules
32		Eliminating organization bureaucracy
33	Flexibility	Workforce flexibility
34		Adaptive management
35		Managing essential changes in the project
36		Rapid response to environmental changes in the project
37	Responsiveness	Effective response to client's inquiries
38		Rapid response to customer requests
39		Considering clients' expectations in the contract
40	Contract management	General and specific contract conditions' transparency
41		Selecting proper contract style
42		Selecting contractor with proper criteria
43	Contractor management	Reading the contractor records
44		Evaluating the contractor records
45		Establishing proper communication among contractor, consultant and client
46		Awareness of new technologies
47	Technology	Providing the proper hardware
48		Applying the proper software
49		Data security level
50		Data management
51		Provide coherent information system
52		Information electronic transaction
53		Providing the means for project agility
54		Providing motivation in the respectful teams
55	IT in project and human resources	Providing team work
56		On-time salary payment
57		Secure employee satisfaction
58		Employing skilled staff
59		Employing multifunctional staff
60		Prioritizing continuous training of the team members
61		Promoting team objective orientation
62		Proper task assignment
63		Respecting customers' ideas
64	Focus on consumer	Beneficiaries involvement in contract negotiations
65		Customer interests' consideration

Table 8: Stability features of each dimension.

Row	Dimension	Stability features of each dimension
1		Available sufficient sustainable financial sources [24, 49, 50]
2		Savings in costs and proper manpower use [22]
3		Proper resource and local facilities application in the project [51]
4		Improving socioeconomic status [51, 52]
5		Increasing efficiency and manipulation [52]
6	Economic	Reducing project time
7		Reducing indirect costs (annual operational costs and maintenance costs) [22]
8		Promoting infrastructure quality [53]
9		Savings in energy consumption [24, 52]
10		The project's economic efficiency [24, 47, 49–51]
11		Project sustainable revenues [24, 49, 50, 52, 54]
12		Reducing construction and facilities installation costs [22]

Table 8: Continued.

Row	Dimension	Stability features of each dimension
13	Social	Providing job for the local manpower [52]
14		The importance of the sociopolitical nature of the project [49]
15		Promoting business ethics and prevent corruption [50, 53, 55]
16		Improving social health [22, 49, 52]
17		Supporting social security [49]
18		Accepting the social nature of project [54]
19		Contributing to justice in social setting [24, 50]
20		Consumer cooperation rate [52]
21		Private sector investment rate [25]
22		Consumer requirements meeting [52]
23		Finished goods/service safety rate [50, 51]
24		Reducing and managing the environmental reduction [49]
25		Adopting project with the local climatic conditions [52]
26		Renewal resources efficiency rate [52]
27		Preventing water, air, social and noise pollution [24, 25]
28	Environmental	Improving environmental hygiene [25, 52]
29		Reducing nonrenewable resource consumption [25, 52]
30		Following green provision [22, 24, 51]
31		Applying new environment friendly technologies and products [24]
32		Being aware of environmental effects of the project [22, 49]
33		Industrial management and ability to run recyclable industry [24, 25, 50]

Table 9: Geometric mean of agility factors' weights in project-orient organizations.

Every agility feature in project-oriented organization	Weight	
	Experts	Municipality staff
Project communication management	4.023	3.926
Organizational culture	3.782	3.908
Project management	4.002	3.953
Project scheduling and control	4.016	4.052
Project speed	3.984	4.089
Flexibility	3.939	3.943
Response	3.712	3.940
Coherent management	3.847	3.995
Contractor/supplier management	3.996	4.005
Technology	4.198	3.807
IT in project	3.974	4.045
Project human resources	4.075	4.099

results, the agility factors in the project-oriented organizations are obtained.

Phase 8: clustering of agility factors in project-oriented organizations based on weighted factors obtained from three weighting attitudes

In this phase, the agility factors in project-oriented organizations are clustered based on the weights obtained from the second phase, according to the project management and academic experts' opinions and the weight obtained from the third phase by solving the QFD matrices. Table 12 shows the weights of agility factors based on the three considered attitudes.

Based on the weights obtained in the second phase and the results and the weights obtained from the third phase by solving QFD matrices, the analysis details for agility factors in project-oriented organizations are tabulated in Table 12.

Phase 9: clustering the agility factors in project-oriented organizations based on weight indices obtained from three weight approaches

In this phase, the agility factors in project-oriented organizations are clustered based on the weights obtained from the second phase based as to the project managements' point of views and academic experts and the weight obtained from the third phase by solving

Table 10: Calculating the weight of sustainability factors based on project management critical success factors.

Environmental sustainability												Social sustainability										Economic sustainability										Factors weight	PM CSFs
33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		
1.59	2.38	3.605	3.075	2.865	1.65	2.235	1.865	2.33	3.015	2.555	3.035	3.345	3.53	2.83	3.415	1.53	1.875	2.86	2.11	2.65	3.42	4.265	4.6	3.35	2.85	3.16	4.775	4.6	4.08	2.66	4.55	3.7	**0.040**	1
2.18	1.015	2.19	1.375	2.065	1.61	2.66	2.37	1.28	2.52	3.035	3.06	2.36	2.645	1.58	3.355	2.125	2.635	2.14	1.105	2.575	3.46	3.96	4.075	2.94	2.1	2.69	3.71	4.54	3.44	3.36	4.225	2.36	**0.035**	2
1.735	3.155	3.695	3.045	2.46	2.23	2.265	2.64	2.655	3.565	3.3	3.59	1.91	2.735	2.36	2.925	3.585	3.36	3.19	2.765	2.3	2.53	2.865	2.73	4.005	2.875	1.875	4.21	4.07	2.95	1.715	2.2	1.88	**0.039**	3
2.485	2.715	3.145	3.14	2.53	2.04	2.815	2.06	2.525	2.915	2.08	2.875	2.08	3.2	3.555	2.355	3.05	2.4	2.76	2.09	2.13	3.315	4.01	3.73	2.935	3.315	3.2	4.5	4.6	3.61	3.355	3.66	3.4	**0.032**	4
3	2.81	3.22	3.71	1.88	2.555	2.03	1.635	209.5	2.615	2.965	3.56	2.56	2.66	2.865	3.805	3.18	2.4	3.59	3.025	2.59	3.87	3.95	4.35	2.96	3.115	2.96	4.215	4.08	3.58	2.845	3.06	2.71	**0.039**	5
3.735	3.09	3.71	4.23	1.365	2.615	1.71	1.865	2.115	2.225	2.815	3.37	2.91	3.09	2.8	3.535	2.38	2.21	0.59	2.605	2.125	3.71	4.3	4.05	3.125	2.925	2.61	4.325	4.365	3.1	3.08	2.875	3.405	**0.037**	6
2.85	3.685	3.365	3.165	2.2	1.03	2.605	1.705	1.09	2.735	1.865	2.215	1.235	1.72	1.63	2.36	2.93	2.225	2.915	2.15	3.08	3.36	3.905	3.915	2.355	2.23	3.365	3.825	3.85	1.56	1.66	4.15	4.215	**0.029**	7
1.63	2.16	2.655	3.21	2.675	2.09	2.715	1.865	1.21	2.125	2.7	2.405	1.51	3.315	1.07	1.85	2.08	1.415	1.93	1.055	2.94	2.96	3.125	2.675	2.065	1.715	1.98	3.33	3.58	3.09	2.385	3.56	2.06	**0.028**	8
2.15	3.09	2.665	3.815	3.03	2.61	3.375	3.09	1.725	3.86	2.33	2.375	2.69	1.72	2.37	2.905	3.08	2.07	3.38	1.55	2.63	3.08	3.56	4.025	2.13	2.09	3.5005	4.58	4.625	2.88	3.065	4.6	3.355	**0.037**	9
1.865	2.745	1.855	2.655	2.68	3.09	2.875	2.09	1.055	1.56	1.365	1.43	2.38	2.62	2.08	2.09	2.56	1.7	1.07	1.64	1.585	2.18	2.595	3.035	2.38	1.555	3.1	3.64	4.25	2.45	2.305	2.56	3.88	**0.038**	10
1.125	1.715	1.68	1.14	1.085	1.14	1.865	2.46	1.525	2.14	1.8	1.13	3.265	3.075	2.75	2.555	1.875	2.125	2.125	1.365	1.125	2.03	3.59	3.215	1.96	1.45	2.57	3.56	3.8	2.14	1.735	3.615	2.215	**0.036**	11
1.51	1.465	2.09	3.26	1.59	1.655	3.375	2.14	1.38	1.855	2.065	2.85	3.205	3.205	2.16	2.69	1.705	3.15	3.35	1.205	2.135	2.63	3.125	2.75	1.715	1.815	3.155	3.845	4.55	3.53	2.75	3.035	2.53	**0.036**	12
1.415	2.065	1.71	1.865	1.465	1.12	2.53	1.875	1.415	2.19	1.365	2.135	2.06	2.54	1.365	2.215	2.07	1.58	1.46	1.4	2.46	2.16	3.965	3.35	3.075	1.705	2.09	1.7	4.2	2.06	2.04	1.7	3.35	**0.038**	13
2.985	1.96	2.38	1.625	2.38	2.14	3.13	1.23	2.075	3.08	3.885	1.715	1.955	2.08	1.39	1.465	1.72	1.555	1.095	1.725	2.27	2.7	3.8	4.4	2.585	1.265	2.255	2.31	4.5	3.39	1.58	2.065	3.71	**0.034**	14
1.125	1.435	1.865	3.16	2.08	1.87	2.04	1.255	1.73	1.835	3.4	3.22	2.875	3.83	2.375	2.8	3.865	3.09	3.23	2.75	2.64	2.555	3.355	3.695	3.08	2.81	3.06	3.54	4.2	2.54	1.81	2.23	2.635	**0.036**	15
2.8	3.7	2.73	3.25	2.38	2.685	3.375	2.25	1.16	3.105	1.865	3.71	3.875	2.7	2.245	2.405	3.145	2.375	2.65	2.105	3.075	3.08	2.96	3.365	2.95	1.905	2.59	3.06	4.58	3.8	3.415	3.1	2.17	**0.037**	16
1.85	3.115	2.66	3.865	2.06	1.63	2.09	1.59	1.23	1.34	2.205	3.31	2.375	2.18	2.57	3.41	3.56	2.59	1.64	1.375	2.55	3.1	3.155	3.1	2.9	1.215	1.95	2.96	4.455	2.375	2.455	2.375	2.56	**0.036**	17
1.335	1.73	2.115	2.55	2.13	1.06	2.46	1.865	1.39	1.405	1.08	3.96	4.06	3.09	2.235	2.125	2.815	1.64	2.46	1.6	2.09	2.55	2.625	2.655	2.35	1.9	2.375	2.65	3.57	1.705	2.07	2.4	3.23	**0.035**	18
1.13	1.89	2.885	2.36	3.11	2.09	3.39	2.245	1.78	1.705	1.08	3.15	2.6	2.08	2.57	2.63	2.46	1.64	1.59	2.08	2.275	3.825	4.075	3.925	2.81	2.57	3.8	4.905	3.95	3.45	3.535	3.875	3.375	**0.035**	19
2.78	3.33	3.56	3.39	3.09	3.06	3.72	3.76	3.135	3.61	1.6	3.21	3.875	3.6	2.375	2.215	1.88	2.66	2.64	2.06	3.075	3.585	3.55	3.45	2.375	2.875	3.08	3.58	4	3.375	2.13	2.575	2.305	**0.037**	20
1.63	1.12	1.09	1.745	1.62	2.44	1.94	1.755	1.03	1.215	2.21	2.395	2.61	2.2	3.1	2.875	2.825	1.82	1.365	1.465	2.08	2.2	3.05	3.11	2.46	2.11	1.6	2.725	3.08	1.51	1.67	2.35	2.71	**0.036**	21
2.08	3.275	1.64	3.865	2.125	1.08	1.195	1.375	1.87	1.89	2.06	2.96	4.105	3.15	3.6	1.8	2.38	2.58	3.075	1.63	2.36	2.13	3.91	3.815	3.08	2.57	3.36	2.06	4.7	2.58	3.035	3.05	4.415	**0.031**	22
2.57	1.875	1.145	2.655	83.5	1.58	1.59	1.71	2.03	2.135	2.555	2.545	3.4	3.03	1.335	2.83	3.365	2.21	2.64	1.56	2.58	2.215	2.95	3.06	2.875	2.405	2.91	3.25	3.725	2.375	2.06	2.35	2.4	**0.039**	23
2.09	2.74	1.59	2.695	1.86	2.12	1.43	1.365	1.195	1.625	2.8	1.7	1.375	1.325	1.7	2.705	2.08	1.645	1.865	1.125	2.06	1.56	2.56	2.7	1.555	1.75	2.025	3.05	3.365	1.815	2.625	2.35	2.3	**0.037**	24
1.63	3.115	1.875	1.73	1.385	1.035	1.24	1.495	1.14	2.125	2.11	3.25	2.855	2.865	1.465	1.53	1.09	1.865	1.56	1.08	2.14	2.14	3.08	3.365	1.96	2.165	2.08	3.365	3.54	2.405	1.72	1.6	3.72	**0.029**	25
1.125	1.38	1.34	3.19	1.23	2.375	1.095	1.885	1.37	3.215	1.87	3.815	2.965	3.03	2.71	2.365	2.605	2.85	2.415	3.08	2.625	2.57	3.58	3.875	2.95	2.215	3.255	2.52	4.205	3.265	2.655	3.445	4.055	**0.037**	26
1.63	1.56	2.13	3.875	2.38	1.09	2.36	2.01	1.08	1.425	2.46	3.035	3.365	2.54	1.98	2.405	1.34	1.615	3.105	2.06	3.06	3.13	3.61	3.75	2.125	2.85	3.405	3.06	3.965	3.805	2.38	2.95	2.75	**0.038**	27
1.375	1.13	1.865	2.12	1.625	1.185	1.635	2.08	1.04	1.375	1.815	2.645	1.88	1.8	2.65	3.2	3.315	3.36	1.045	1.66	2.44	2.36	2.865	3.355	3.25	2.75	3.065	3.74	4.58	2.655	1.7	3.225	2.21	**0.037**	28
0.021	0.024	0.025	0.030	0.055	0.020	0.024	0.021	0.102	0.024	0.024	0.029	0.028	0.028	0.024	0.027	0.026	0.023	0.024	0.019	0.025	0.029	0.036	0.036	0.028	0.024	0.029	0.036	0.043	0.030	0.025	0.031	0.031	Agility factors' weights	

Table 11: Calculating the weight of agility factors in project-oriented organizations in relation to sustainability factors' weights applying QFD.

Agility factors in project-oriented organizations														Sustainability factors
13	12	11	10	9	8	7	6	5	4	3	2	1	Factors weight	
2.070	2.125	2.625	4.560	3.590	4.200	3.560	3.800	4.125	4.020	4.465	2.750	3.000	**0.031**	1
3.400	4.375	3.090	4.465	3.635	3.125	4.100	3.160	3.965	4.590	4.410	3.915	2.580	**0.031**	2
3.135	3.930	3.125	2.375	2.625	2.865	3.365	2.860	3.725	3.590	4.200	3.215	3.365	**0.025**	3
3.310	3.460	2.215	2.570	2.625	2.640	2.125	2.300	1.570	2.625	4.160	3.580	3.085	**0.030**	4
2.660	3.070	3.450	3.950	3.165	3.540	4.000	2.715	3.800	4.640	4.210	3.215	3.625	**0.043**	5
2.460	3.350	3.275	4.060	3.615	4.125	3.610	3.115	4.060	4.210	4.110	2.125	3.540	**0.036**	6
2.570	2.625	3.835	4.125	3.360	3.635	3.625	2.360	3.440	3.625	3.965	2.875	3.390	**0.029**	7
2.060	2.580	1.610	3.815	3.060	2.440	1.375	2.570	1.510	3.075	4.625	3.610	2.850	**0.024**	8
2.160	2.625	2.375	4.475	1.640	2.660	2.590	2.080	2.610	3.000	4.055	2.580	3.580	**0.028**	9
2.570	2.365	3.580	3.500	2.375	3.250	2.590	2.160	2.365	3.610	4.140	2.865	2.360	**0.036**	10
3.085	2.125	2.885	2.625	3.580	2.875	2.000	2.600	1.875	3.570	4.090	3.840	3.385	**0.036**	11
2.030	2.570	2.125	4.060	2.070	2.140	2.625	2.965	2.525	3.580	4.125	2.610	2.165	**0.029**	12
2.375	3.110	1.815	1.950	3.165	3.100	1.875	2.100	1.125	2.590	4.000	2.950	3.880	**0.025**	13
1.100	3.075	3.250	2.500	3.060	2.510	1.900	1.900	1.140	2.975	2.360	2.125	3.350	**0.019**	14
1.415	4.080	2.450	2.125	3.850	2.340	1.800	2.200	1.615	3.865	4.080	3.885	2.730	**0.024**	15
3.115	4.600	3.150	1.640	3.100	2.900	4.200	3.000	2.195	4.105	3.800	4.125	3.575	**0.023**	16
2.340	4.200	2.800	1.560	3.160	4.365	3.600	2.555	1.580	3.300	2.665	4.210	2.700	**0.026**	17
3.265	2.580	3.530	2.375	2.940	3.365	3.760	2.375	1.860	4.080	4.410	3.750	3.405	**0.027**	18
2.440	3.365	3.865	2.065	3.080	4.400	3.950	1.900	1.465	2.840	3.125	3.085	2.355	**0.024**	19
4.165	4.625	4.160	1.530	1.915	1.300	2.855	3.095	2.030	2.950	3.275	3.815	4.165	**0.028**	20
3.350	3.925	3.625	1.865	4.605	3.830	3.310	3.100	2.530	2.850	2.865	4.080	4.025	**0.028**	21
4.150	2.625	2.365	3.550	2.580	3.570	2.650	2.350	1.600	3.055	3.910	2.960	2.565	**0.029**	22
4.500	4.150	3.950	4.100	2.460	3.100	2.375	3.075	2.190	3.200	2.735	3.840	3.140	**0.024**	23
3.250	3.410	3.400	3.805	3.065	2.400	1.300	2.070	1.365	3.000	2.925	4.150	2.815	**0.024**	24
3.090	2.090	3.725	3.175	1.645	2.850	2.055	3.375	1.575	2.750	3.250	2.210	1.850	**0.102**	25
2.355	3.375	3.365	4.025	2.725	2.360	1.375	2.080	1.640	2.355	2.665	3.825	2.460	**0.021**	26
2.460	3.665	2.570	3.960	3.365	1.950	1.225	2.060	1.660	1.955	2.360	3.375	1.805	**0.024**	27
2.885	3.960	2.175	3.580	2.960	2.220	1.500	2.225	1.375	2.085	2.460	3.965	2.875	**0.020**	28
2.140	1.850	2.135	4.350	3.035	1.805	1.975	2.125	1.815	3.400	3.035	4.590	2.965	**0.055**	29
2.625	4.275	3.825	2.950	3.725	3.385	2.080	2.425	1.955	2.955	3.350	4.625	3.105	**0.030**	30
3.165	2.640	3.620	4.265	3.115	2.225	2.500	1.900	2.000	3.100	3.580	3.905	3.580	**0.025**	31
2.465	2.480	3.440	3.075	2.960	3.810	2.000	2.210	1.700	2.655	3.365	3.400	3.750	**0.024**	32
2.040	3.365	2.965	4.090	3.515	3.115	1.875	1.850	1.660	2.500	2.095	3.780	2.350	**0.021**	33
2.764	3.075	3.081	3.297	2.916	2.979	2.608	2.591	2.205	3.270	3.575	3.366	2.967		
0.071	**0.079**	**0.080**	**0.085**	**0.075**	**0.077**	**0.067**	**0.067**	**0.057**	**0.085**	**0.092**	**0.087**	**0.077**	**Agility factors' weight**	

Table 12: Importance weights of the agility factors' in project-oriented organizations through the three attitudes.

Agility in project-oriented organizations	Agility weight importance (average)	Agility weight importance (experts)	Agility weight importance (municipality staff)	Agility weight importance (sustainability approach)
Project communication management	0.0733	0.0768	0.0764	0.0769
Project culture	0.0750	0.0751	0.0760	0.0739
Project management	0.0864	0.1074	0.0769	0.0782
Project scheduling	0.0830	0.0926	0.0788	0.0785
Project speed	0.0757	0.070	0.0795	0.0799
Flexibility	0.0739	0.0683	0.0767	0.077
Response	0.0736	0.0716	0.0766	0.0726
Coherent management	0.1658	0.0780	0.0777	0.0725
Contractor's management	0.0762	0.0726	0.0779	0.0781
Technology	0.0820	0.0908	0.074	0.0821
Project IT	0.0749	0.0686	0.0787	0.0777
Human resources	0.0756	0.0708	0.0797	0.0793
Focusing on consumer	0.0678	0.0619	0.0712	0.0708

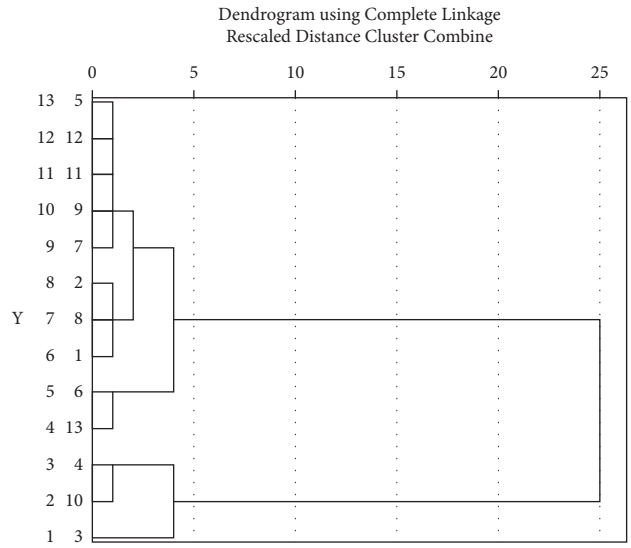


Figure 2: Dendrogram from hierarchical clustering.

Table 13: Results of clustering and the manner of agility factors placement.

Cluster #	Agility in project-oriented organizations	Factor #
1	Project communication management	1
1	Project culture	2
2	Project management	3
3	Project scheduling	4
4	Project speed	5
5	Flexibility	6
4	Response	7
1	Coherent management	8
4	Contractor's management	9
3	Technology	10
4	Project IT	11
4	Human resources	12
5	Focusing on consumer	13

Table 14: The agility clusters' importance in project-oriented organizations.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Clusters' members	(1) Communication management in project (2) Organizational culture (3) Contracts management	(1) Project management	(1) Project scheduling and control (2) Responsiveness (3) Contractors management (4) IT IN PROJECT (5) Human resources	(1) Project speed	(1) Flexibility (2) Focus on consumers
Cluster's importance weight	0.0987	0.0864	0.0825	0.0753	0.0708

QFD matrices, and therefore, they clustered through a hierarchical model.

Step 8: clustering the agility factors in project-oriented organizations based on weight indices obtained from the three approaches

At this stage, the agility factors are clustered through the hierarchical clustering method. The yield dendrogram is shown in Figure 2. Accordingly, considering the similarity coefficient is ˃95%, agility factors in project-oriented organizations are located in five

clusters, and the clustering details therein are tabulated in Table 13.

Step 9: determining the importance of the agility factors' clusters in project management based on the member factor of each cluster.

In this step, based on both the results of the previous step and the weight of clusters' factors, the importance of each cluster is determined, and each cluster's rank is determined according to their importance value. As observed in Table 13, after determining the average importance weight of the agility factors' in project-oriented organizations, for the agility factors that they are members of clusters, the weighted mean of each factor is calculated to obtain the importance weight of cluster. The cluster with the highest weight is placed at the top of the clusters' importance (Table 14).

As observed in this table, clusters one, two, and three are ranked in the first to third as to their importance, respectively.

6. Conclusion

To manage and execute a project, in project-oriented organizations, concentrating on the required infrastructures and providing the project management critical success factors, together in addition of considering the importance of environmental resources and social effects of the project and also the profitability and economic aspect which puts the organization in line with appropriate development, as to form a successful project management at macro and microlevels is of essence. By doing so, a big step is taken towards accountability and customer satisfaction due to project implementation, by considering sustainability indices. Therefore, a framework is introduced for clustering and analysis of big projects based on agility factors in project-oriented organizations with a sustainability approach, by applying quality functionality development tool. The proposed framework has been implemented and solved in Isfahan Metropolitan Municipality. For this purpose, in phase one, first, the project management critical success factors and, next, the agility factors and in project-oriented organizations and sustainability factors are identified. In phase two, the importance of the project management critical success factors and, next, agility factors in project-oriented organizations are measured from the perspective of large project managers in Isfahan Municipality and university experts. In phase three, the weights of agility factors in project-oriented organizations are calculated based on the weighted factors of sustainability according to the critical success factors in the large projects by applying quality function deployment. In phase four, the agility factors in project-oriented organizations are clustered based on the previously obtained weights. Next to guiding project managers to focus on the major success factors in project management and providing sustainability factors as to the big projects' and activities' impact on the field of environmental resources, social resources, and economic aspects, through focusing on agility factors, they can obtain the desired customer satisfaction level, timely delivery, and desired quality as the major concerns. As observed in Table 14, the first cluster, which includes project communication management, organizational culture, and contract management, is the most important at (0.0987) weight rate. Because these weights are based on sustainability and project management critical success factors, focusing on the important cluster is highly contributive in organizations' success in project management next to the principles of sustainable development and establishing the principles of agility.

7. Research Constraints

The measures taken in this study, despite the accuracy of the presented framework and application of the applied factors, can be improved by removing any of the constraints therein. Depending on the subject organization, the projects and conditions, addition, elimination, and replacing the applied factors can be contributive in closer insight therein. The findings are implemented on Isfahan Municipality's big projects. To be focused on the paramount cluster helps the organization to be successful in project management in line with setting the principles of the sustainable development and by stablising the principles of the organizational agility in project-oriented organizations.

8. Suggestions for Future Studies

Assessing the sensitivity analysis of indices and the effect on the clusters' ordering can lead to different and accurate results in the performance of the organization. Moreover, evaluating the performance of projects through the evaluation and performance analysis methods, subject to each approach in addition to the presence of combined approaches, can be effective in improving efficiency. The effect of each one of these approaches on project cost, time, and quality must be considered in project performance assessment.

Data Availability

The research data are available within the article (Tables and Figures) in detail.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. M. Nicholls, N. A. Lewis, and T. Eschenbach, "Determining when simplified agile project management is right for small teams," *Engineering Management Journal*, vol. 27, no. 1, pp. 3–10, 2015.

[2] C. M. Felipe, J. L. Roldán, and A. L. Leal-Rodríguez, "An explanatory and predictive model for organizational agility," *Journal of Business Research*, vol. 69, no. 10, pp. 4624–4631, 2016.

[3] A. Stare, "Agile project management in product development projects," *Procedia - Social and Behavioral Sciences*, vol. 119, pp. 295–304, 2014.

[4] L. Fonseca, A. Amaral, and J. Oliveira, "Quality 4.0: the EFQM 2020 model and industry 4.0 relationships and implications," *Sustainability*, vol. 13, no. 6, Article ID 3107, 2021.

[5] B. A. Abdulrahman and F. A. N. Abdallah, "Critical success factors (CSFS) in project management: critical review of secondary data," *International Journal of Scientific and Engineering Research*, vol. 5, no. 6, pp. 325–331, 2014.

[6] V. Obradović, M. Todorović, and S. Bushuyev, "Sustainability and agility in project management: contradictory or complementary?" in *Proceedings of the Conference on Computer Science and Information Technologies*, pp. 522–532, Lviv, Ukraine, September 2018.

[7] C. R. Carter and D. S. Rogers, "A framework of sustainable supply chain management: moving toward new theory," *International Journal of Physical Distribution & Logistics Management*, vol. 38, 2008.

[8] E. Gończ, U. Skirke, H. Kleizen, and M. Barber, "Increasing the rate of sustainable change: a call for a redefinition of the concept and the model for its implementation," *Journal of Cleaner Production*, vol. 15, no. 6, pp. 525–537, 2007.

[9] J. van den Brink and S. Gilbert, "Taking responsibility: the integration of sustainability and project management," HU University of Applied Sciences Utrecht, Utrecht, Netherlands, 2011.

[10] M.-M. Nahod and M. V. M. Radujković, "The impact of ICB 3.0 competences on project management success," *Procedia—Social and Behavioral Sciences*, vol. 74, pp. 244–254, 2013.

[11] A. Brent and C. Labuschagne, "Social indicators for sustainable project and technology life cycle management in the process industry (13 pp + 4)," *International Journal of Life Cycle Assessment*, vol. 11, no. 1, pp. 3–15, 2006.

[12] A. I. Gaziulusoy, "A critical review of approaches available for design and innovation teams through the perspective of sustainability science and system innovation theories," *Journal of Cleaner Production*, vol. 107, pp. 366–377, 2015.

[13] M. E. McCullins, *Sustainability and Project Management*, Athabasca University—Centre for Innovative Management, Alberta, Canada, 2007.

[14] Z. Alias, E. M. A. Zawawi, K. Yusof, and N. M. Aris, "Determining critical success factors of project management practice: a conceptual framework," *Procedia - Social and Behavioral Sciences*, vol. 153, pp. 61–69, 2014.

[15] A. A. Ahmadabadi and G. Heravi, "The effect of critical success factors on project success in Public-Private Partnership projects: a case study of highway projects in Iran," *Transport Policy*, vol. 73, pp. 152–161, 2019.

[16] T. P. Filgueira de Melo Moura, "Critical success factors for project management support information systems: SEBRAE/RN Case," *Rebrae*, vol. 9, no. 1, pp. 8–25, 2016.

[17] M. Orouji, "Critical success factors in project management," *Journal of Project Management*, vol. 1, no. 1, pp. 35–40, 2016.

[18] F. Costantino, G. Di Gravio, and F. Nonino, "Project selection in project portfolio management: an artificial neural network model based on critical success factors," *International Journal of Project Management*, vol. 33, no. 8, pp. 1744–1754, 2015.

[19] S. Naoum, D. Fong, and G. Walker, "Critical success factors of project management," in *Proceedings of the International Symposium on Globalization and Construction*, pp. 827–838, Bangkok, Thailand, November 2004.

[20] C. H. Thi and F. W. Swierczek, "Critical success factors in project management: implication from Vietnam," *Asia Pacific Business Review*, vol. 16, no. 4, pp. 567–589, 2010.

[21] R. Kiani Mavi and C. Standing, "Critical success factors of sustainable project management in construction: a fuzzy DEMATEL-ANP approach," *Journal of Cleaner Production*, vol. 194, pp. 751–765, 2018.

[22] N. Dobrovolskienė and R. Tamošiūnienė, "An index to measure sustainability of a business project in the construction industry: Lithuanian case," *Sustainability*, vol. 8, pp. 14–1, 2016.

[23] M. L. Martens and M. M. Carvalho, "Key factors of sustainability in project management context: a survey exploring the project managers' perspective," *International Journal of Project Management*, vol. 35, no. 6, pp. 1084–1102, 2017.

[24] H. Daneshpour, "Integrating sustainability into management of project," *International Journal of Environment and Sustainable Development*, vol. 6, no. 4, pp. 321–325, 2015.

[25] A. Amiril, A. H. Nawawi, R. Takim, and S. N. F. A. Latif, "Transportation infrastructure project sustainability factors and performance," *Procedia—Social and Behavioral Sciences*, vol. 153, pp. 90–98, 2014.

[26] A. J. G. Silvius and R. Schipper, "A maturity model for integrating sustainability in projects and project management," in *Proceedings of the 24th World Congress of the International Project Management Association (IPMA)*, Istanbul, Turkey, October 2010.

[27] R. Procter, M. Rouncefield, M. Poschen, Y. Lin, and A. Voss, "Agile project management: a case study of a virtual research environment development project," *Computer Supported Cooperative Work*, vol. 20, no. 3, pp. 197–225, 2011.

[28] D. J. Fernandez and J. D. Fernandez, "Agile project management—agilism versus traditional approaches," *Journal of Computer Information Systems*, vol. 49, no. 2, pp. 10–17, 2008.

[29] Q. Chen, G. Reichard, and Y. Beliveau, "Interface management-a facilitator of lean construction and agile project management," *International Group for Lean Construction*, vol. 1, no. 1, pp. 57–66, 2007.

[30] B. De Reyck, "Effective project planning: making the most of project planning tools," *Production and Inventory Management Journal*, vol. 46, pp. 10–2, 2010.

[31] A. Shahin and M. Jamshidian, "Critical success factors in project management: a comprehensive review," in *Proceedings of the 1st International Project Management Conference*, pp. 1–14, Arlington, VA, USA, November 2006.

[32] A. Munns and B. Bjeirmi, "The role of project management in achieving project success," *International Journal of Project Management*, vol. 14, no. 2, pp. 81–87, 1996.

[33] T. Bergmann and W. Karwowski, "Agile project management and project success: a literature review," in *Proceedings of the 2018 International Conference on Applied Human Factors and Ergonomics*, pp. 405–414, Orlando, FL, USA, July 2018.

[34] M. Kavacık, S. Zafer, Y. Ali, and D. Karaman, "Sustainable development in aviation industry and the case of Turkish airlines," *Journal of Animal and Veterinary Advances*, vol. 9, no. 3, pp. 547–550, 2012.

[35] L. M. Fonseca, J. P. Domingues, and A. M. Dima, "Mapping the sustainable development goals relationships," *Sustainability*, vol. 12, no. 8, Article ID 3359, 2020.

[36] L. M. Fonseca and V. M. Lima, "Countries three wise men: sustainability, innovation, and competitiveness," *Journal of Industrial Engineering and Management*, vol. 8, no. 4, pp. 1288–1302, 2015.

[37] L.-K. Chan and M.-L. Wu, "Quality function deployment: a comprehensive review of its concepts and methods," *Quality Engineering*, vol. 15, no. 1, pp. 23–35, 2002.

[38] S. Kumar, H. H. Inbarani, A. T. Azar, and A. E. Hassanien, "Rough set-based meta-heuristic clustering approach for social e-learning systems," *International Journal of Intelligent Engineering Informatics*, vol. 3, no. 1, pp. 23–41, 2015.

[39] Y. Akao and G. H. Mazur, "The leading edge in QFD: past, present and future," *International Journal of Quality and Reliability Management*, vol. 20, 2003.

[40] C.-F. Tsai, Y.-H. Hu, and Y.-H. Lu, "Customer segmentation issues and strategies for an automobile dealership with two clustering techniques," *Expert Systems*, vol. 32, no. 1, pp. 65–76, 2015.

[41] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, and T. Herawan, "Big data clustering: a review," in *Proceedings of the 14th International Conference on Computational Science and its Applications*, pp. 707–720, Guimaraes, Portugal, June 2014.

[42] G. Piatetsky-Shapiro, "The data-mining industry coming of age," *IEEE Intelligent Systems*, vol. 14, no. 6, pp. 32–34, 1999.

[43] T. Sajana, C. M. S. Rani, and K. V. Narayana, "A survey on clustering techniques for big data mining," *Indian journal of Science and Technology*, vol. 9, no. 3, pp. 1–12, 2016.

[44] S. Banihashemi, M. R. Hosseini, H. Golizadeh, and S. Sankaran, "Critical success factors (CSFs) for integration of sustainability into construction project management practices in developing countries," *International Journal of Project Management*, vol. 35, no. 6, pp. 1103–1119, 2017.

[45] D. F. Ofori, "Project management practices and critical success factors-A developing country perspective," *International Journal of Business and Management*, vol. 8, no. 21, 2013.

[46] H. Abou-Hafs and B. Hassan, "Project Management Performance: The Success Factors of Project Management for Events—Case of the Wedding Projects in Marrakech, Morocco," *Journal of Business and Economics*, vol. 4, 2013.

[47] H. Shirouyehzad, F. M. Rafiei, and E. Shahgholi, "Performance evaluation of organisational units based on key performance indicators with agility approach by using MADM, QFD; a case study in Darakar Company," *International Journal of Productivity and Quality Management*, vol. 17, no. 2, pp. 198–214, 2016.

[48] P. W. Ihuah, I. I. Kakulu, and D. Eaton, "A review of critical project management success factors (CPMSF) for sustainable social housing in Nigeria," *International Journal of Sustainable Built Environment*, vol. 3, no. 1, pp. 62–71, 2014.

[49] M. L. Martens and M. M. Carvalho, "The challenge of introducing sustainability into project management function: multiple-case studies," *Journal of Cleaner Production*, vol. 117, pp. 29–40, 2016.

[50] V. K. Chawla, A. K. Chanda, S. Angra, and G. R. Chawla, "The sustainable project management: a review and future possibilities," *Journal of Project Management*, vol. 3, no. 3, pp. 157–170, 2018.

[51] L. Shen, Y. Wu, and X. Zhang, "Key assessment indicators for the sustainability of infrastructure projects," *Journal of Construction Engineering and Management*, vol. 137, no. 6, pp. 441–451, 2011.

[52] A. Hesampour, M. Nikbakht, and H. Shirouyehzad, "A method for ranking, assessment and evaluation of sustainability factors in project," *Journal of Modern Processes in Manufacturing and Production*, vol. 5, no. 1, pp. 29–44, 2016.

[53] G. Heravi, M. Fathi, and S. Faeghi, "Evaluation of sustainability indicators of industrial buildings focused on petrochemical projects," *Journal of Cleaner Production*, vol. 109, pp. 92–107, 2015.

[54] M. Bennet, P. James, and L. Klinkers, *Sustainable Measures: Evaluation and Reporting of Environmental and Social Performance*, Greenleaf Publishing, South Yorkshire, UK, 1999.

[55] H. Maylor, "Beyond the Gantt chart: Project management moving on," *European Management Journal*, vol. 19, no. 1, pp. 92–100, 2001.

Hindawi
Complexity
Volume 2021, Article ID 5451439, 12 pages
https://doi.org/10.1155/2021/5451439

Research Article

Multivariable Model Reference Adaptive Control of an Industrial Power Boiler Using Recurrent RBFN

Jafar Tavoosi,[1] **Yavar Azarakhsh**,[1] **Ardashir Mohammadzadeh**,[2] **Saleh Mobayen**,[3] **Jihad H. Asad**,[4] **and Rabia Safdar**[5,6]

[1]*Department of Electrical Engineering, Faculty of Engineering, Ilam University, Ilam, Iran*
[2]*Department of Electrical Engineering, University of Bonab, Bonab, Iran*
[3]*Future Technology Research Center, National Yunlin University of Science and Technology, Douliu, Yunlin 64002, Taiwan*
[4]*Department of Physics, Faculty of Applied Sciences, Palestine Technical University, P. O. Box 7, Tulkarm, State of Palestine*
[5]*Department of Mathematics, Lahore College Women University, Lahore, Pakistan*
[6]*Department of Mathematics, College of Arts and Sciences, University of Jhang, Jhang, Pakistan*

Correspondence should be addressed to Jafar Tavoosi; j.tavoosi@ilam.ac.ir and Saleh Mobayen; mobayens@yuntech.edu.tw

Received 28 June 2021; Revised 28 July 2021; Accepted 19 August 2021; Published 7 September 2021

Academic Editor: Jesus Vega

In industrial steam systems, the process requires a specific pressure, and the maximum permissible operating pressure is different. If the inlet steam pressure to the steam consuming equipment exceeds the operating pressure, it may cause hazards. Therefore, the more precise control of the boiler pressure is important. Since we are dealing with a nonlinear, time-varying, and multivariable system, the control method must be designed to handle this system well. Most of the methods proposed so far are either not physically feasible or the system has considered very simple. Therefore, in this paper, while modeling the boiler and its pressure relations more precisely, we will introduce a recurrent type-2 fuzzy RBFN-based model reference adaptive control system with various uncertainties so that the uncertainty and inaccuracy of the model can be compensated. The experimental results prove the efficiency of the proposed method in boiler control.

1. Introduction

The chemical industries consume a great deal of energy to accomplish many tasks, including the process and supply of heat energy needed to heat materials. Current energy sources in the world mainly include fossil fuels, water and wind, and solar and nuclear energy. Of these, fossil fuels are used more often than any other source to supply energy carriers. Usually, in the chemical industry, carriers are electricity and steam. The best way to use fossil fuel energy to generate electricity and steam is to burn it [1]. In the burning process, the internal energy is released as heat and transferred to the environment in the form of radiation and displacement. Boilers and gas turbines are equipment commonly used to generate steam and electricity by burning fossil fuels. The boiler is a nonlinear, multivariable, and time-varying system and therefore requires an advanced control method [2]. Various articles have been presented in the field of thermal management in industries [3–5]. Here are some of the latest ones. In [6], a method for measuring the water level of the boiler is provided. The control of the boiler drum level is investigated using a feedback method. A laboratory-scale steam boiler, a small version of the power plant boiler, is intended for research in this article. In [7], fuzzy control was used to control the boiler with sugarcane fuel. This article uses a tool to find the dynamics of plant and set up control loops. In [8], a fractional order PID controller is introduced to study the temperature and pressure effects. In this paper, the AT89C55-based boiler water supply control system is designed and a simple fractional order PID algorithm is suggested. Also, the control of boiler water temperature and pressure simulation analysis has been performed. In [9], the integration of two-state model predictive control algorithms (OPM, optimal control predictive model) and generalized-

predictive control (GPC) to fit a self-control system to an advanced cylindrical boiler dynamics O is performed with nonminimal phase behavior.

Not much has been done about controlling the boiler reference model. Perhaps, this is due to the over-dependence of the method on the boiler mathematical model, and since the boiler system is highly nonlinear and time varying, model-based control may not be applicable in practice. However, some of the work done in this area is discussed below. In [10], the controller design for a boiler system with uncertainty is presented. To achieve this goal, a controller is designed for the class of multi-input nonlinear systems that correspond to the boiler equations. Computational intelligence is increasingly being developed [11–15]. Neural network-based control systems [16–20], fuzzy system [21–23], and fuzzy neural networks [24–27] have shown good performance. The following discusses some methods of boiler control based on computational intelligence. In [28], neural network-based predictive control is used to control carbon dioxide emissions from a large industrial boiler. In [29], fuzzy control is used to regulate the oxygen content of a boiler in the sugarcane industry. It has been shown in this study that fuzzy control leads to a decrease in metadata and a simultaneous increase in system response. In [30], a hierarchical fuzzy system is used to control a power boiler. It is acknowledged in the article that the proportional-integrator controller alone is not well responsive due to its complex, nonlinear, and inertial boiler nature and should therefore use with an intelligent control system. Very rare work on the neural network composition and model reference control method for boiler control is presented, which will be discussed below. In [31], the neural controller is presented. It is hard to obtain proper proficiency for the combustion controller of chain boilers due to the high latency, different quality of coal and load. A neural identification approach for time delay in nonlinear system is presented. Deficiencies [32] can be attributed to the lack of parametric and nonparametric uncertainty. In [33], a neural network radial basis function is used to model the boiler dyamics, and then, it is used in a model reference method. In other words, in this paper, the model reference block is a neural network model and it is a PID controller. The major drawback of [33] is the use of PID in the boiler system because the boiler is a nonlinear and delayed system and the PID does not provide a good response. Multivariable control means simultaneous control of parameters by considering their interaction [34, 35]. Due to the complexity of simultaneously controlling all parameters, unfortunately, in most articles, the parameters are controlled individually. Certainly, in a multivariable system such as a boiler, it is better to control all the parameters at the same time, considering their effect on each other. So, the innovations of this article are

(1) Accurate modeling of an industrial and practical boiler with all the limitations and uncertainties

(2) Present recurrent type-2 fuzzy neural network in model reference control of a boiler for the first time

(3) Consider the interaction of variables with each other and use multivariable control

In this paper, first, the nonlinear equations of a real boiler are extracted, and then, the linear state space model is calculated to obtain the reference model. In the following, the multivariable control system of the model reference based on the proposed recurrent type-2 fuzzy RBFN is presented, and then, the rules of updating the neural network are presented. Finally, by simulating a real power plant boiler, the proposed control method is evaluated.

2. Mathematical Modeling of the Boiler System

The nonlinear dynamic of boiler-turbine-generator provided by Bell and Astrom is used. Since this model is extracted from the data of a real system in Sweden, this model has been used in most research projects since 1987 [36]. As shown in Figure 1, y_1 is the steam pressure of the drum in kilograms per square centimeter (kg/cm^2), y_2 is the electrical output power in megawatts (MW), and y_3 is the drum water level in terms of meter (m) The output variables of this system are multivariable. Also, the flow rate of the input fuel, the u_2 flow rate of the boiler output steam and the turbine inlet, and the u_3 flow rate of the water entering the boiler are the input variables. So, we have a multivariable system of three inputs and three outputs. Table 1 shows the parametric values of the system.

The state-of-the-art form of this multivariable system with a power of 160 MW is as follows:

$$\begin{aligned}
\dot{x}_1 &= -0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3,\\
\dot{x}_2 &= \frac{(0.73u_2 - 0.16)x_1^{9/8} - x_2}{10},\\
\dot{x}_3 &= \frac{[141u_3 - (1.1u_2 - 0.19)x_1]}{85},\\
y_1 &= x_1,\\
y_2 &= x_2,\\
y_3 &= 0.05\left(0.13073x_3 + 100a_{cs} + q_e - 67.975\right).
\end{aligned} \tag{1}$$

In the above system, x_1 is the steam pressure of the drum, x_2 is the power output, x_3 is the steam density inside the drum in terms of kg/m^3, u_1 is the amount of flow input fuel, u_2 is the amount of steam discharge applied to the turbine from the boiler, and u_3 is the flow of water entering the drum. Also, a_{cs} and q_e are the rate of steam quality and evaporation (kg/s) and are calculated by the following equations. It should be noted that y_3 is the water level change of the drum:

$$\begin{aligned}
a_{cs} &= \frac{(-25.6 + 0.8x_1)(1 - 0.001538x_3)}{(1.0394 - 0.0012340x_1)x_3},\\
q_e &= x_1(0.854u_4 - 0.147) + 45.59u_1 - 2.514u_3 - 2.096.
\end{aligned} \tag{2}$$

To design a model reference control system, one must extract a linear model of the system that is easily controllable

Figure 1: A power plant boiler in Iran.

Table 1: System parameter values.

Nominal power	160 MW
Rate of fuel flow	14 kg/sec
Rate of vapor flow	140 kg/sec
Inlet temperature	3°C
Steam pressure	140 kg/cm^2
Steam mass	2000 kg
Steam temperature	535°C
Drama size	40 m^3
Water mass	40000 kg

and design a controller for it. In this case, since the behavior of the linear model is similar to the main nonlinear model, the main nonlinear model can also be controlled by careful control of the linear model.

3. Linearization of the Nonlinear Model of the Boiler

Boiler-turbine-generator system is one of the nonlinear and multivariable models. In this dissertation, the nonlinear model is linearized as

$$\begin{aligned} y^0 &= \left(y_1^0, y_2^0, y_3^0\right), \\ x^0 &= \left(x_1^0, x_2^0, x_3^0\right), \\ u^0 &= \left(u_1^0, u_2^0, u_3^0\right). \end{aligned} \tag{3}$$

The general form of nonlinear systems is $\dot{x} = f(x,u)$, which can be written as

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m), \\ \frac{dx_2}{dt} &= f_2(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m), \\ &\vdots \\ \frac{dx_n}{dt} &= f_n(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m). \end{aligned} \tag{4}$$

Assume that $u^0 = [\, u_1^0 \;\; u_2^0 \;\; \ldots \;\; u_m^0 \,]^T$ is entered as a constant input to the system $\dot{x} = f(x,u)$ and balance it to $x^0 = [\, x_1^0 \;\; x_2^0 \;\; \ldots \;\; x_n^0 \,]^T$. So,

$$\begin{aligned} x &= x^0 + \Delta x, \\ u &= u^0 + \Delta u. \end{aligned} \tag{5}$$

Using the Taylor extension, we will have

$$\begin{aligned} \frac{dx}{dt} &= f\left(x^0 + \Delta x, u^0 + \Delta u\right) \\ &= f\left(x^0, u^0\right) + \frac{df}{dx}\left(x^0, u^0\right)\Delta x + \frac{df}{du}\left(x^0, u^0\right)\Delta u \\ &\quad + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\left(x^0, u^0\right)(\Delta x)^2 \\ &\quad + \frac{\partial^2 f}{\partial x \partial u}\left(x^0, u^0\right)(\Delta x)(\Delta u) + \frac{1}{2}\frac{\partial^2 f}{\partial u^2}\left(x^0, u^0\right)(\Delta u)^2. \end{aligned} \tag{6}$$

So, regardless of the nonlinear terms, we will have

$$\frac{dx}{dt} = f\left(x^0, u^0\right) + \frac{df}{dx}\left(x^0, u^0\right)\Delta x + \frac{df}{du}\left(x^0, u^0\right)\Delta u, \tag{7}$$

where

$$\begin{aligned} \frac{df}{dx}\left(x^0, u^0\right) &= \begin{bmatrix} \frac{df_1}{dx_1} & \cdots & \frac{df_1}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{dx_1} & \cdots & \frac{df_n}{dx_n} \end{bmatrix}_{(x^0,u^0)}, \\ \frac{df}{du}\left(x^0, u^0\right) &= \begin{bmatrix} \frac{df_1}{du_1} & \cdots & \frac{df_1}{du_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{du_1} & \cdots & \frac{df_n}{du_n} \end{bmatrix}_{(x^0,u^0)}. \end{aligned} \tag{8}$$

We also have

$$\frac{dx}{dt} = \frac{dx^0}{dt} + \frac{d(\Delta x)}{dt} = \frac{d(\Delta x)}{dt}, \tag{9}$$

because x^0 is a fixed number. Assume

$$\begin{aligned} A &= \frac{df}{dx}\left(x^0, u^0\right), \\ B &= \frac{df}{du}\left(x^0, u^0\right). \end{aligned} \tag{10}$$

By ignoring higher-order sentences to achieve linear approximation and writing in the form of state space,

$$\frac{\mathrm{d}(\Delta x)}{\mathrm{d}t} = A\Delta x + B\Delta u. \tag{11}$$

Similarly, the output of the nonlinear system model is as follows:

$$\begin{aligned} y_1 &= h_1(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m),\\ y_2 &= h_2(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m),\\ &\vdots\\ y_n &= h_n(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m). \end{aligned} \tag{12}$$

In other words,

$$y = h(x, u). \tag{13}$$

The expansion of the Taylor series is again used as

$$y = y^0 + \Delta y. \tag{14}$$

So, we will have

$$\begin{aligned} h(x,u) &= h\left(x^0, u^0\right) + \frac{\mathrm{d}h}{\mathrm{d}x}\left(x^0, u^0\right)\Delta x + \frac{\mathrm{d}h}{\mathrm{d}u}\left(x^0, u^0\right)\Delta u,\\ y^0 &= h\left(x^0, u^0\right). \end{aligned} \tag{15}$$

So,

$$\begin{aligned} y - y^0 &= \frac{\mathrm{d}h}{\mathrm{d}x}\left(x^0, u^0\right)\Delta x + \frac{\mathrm{d}h}{\mathrm{d}u}\left(x^0, u^0\right)\Delta u,\\ \Delta y &= \overline{y} = C\overline{x} + D\overline{u}, \end{aligned} \tag{16}$$

where

$$\begin{aligned} C &= \frac{\mathrm{d}h}{\mathrm{d}x}\left(x^0, u^0\right),\\ D &= \frac{\mathrm{d}h}{\mathrm{d}u}\left(x^0, u^0\right),\\ \overline{y} &= \Delta y = \left(y - y^0\right),\\ \overline{x} &= \Delta x = \left(x - x^0\right),\\ \overline{u} &= \Delta u = \left(u - u^0\right). \end{aligned} \tag{17}$$

Now, we can write the general vector symbol of the nonlinear system and the space model mode of the linear system as

$$\begin{aligned} \dot{x} &= f(x, u),\\ y &= h(x, u), \end{aligned} \tag{18}$$

and linear form is

$$\begin{aligned} \dot{x} &= Ax + Bu,\\ y &= Cx + Du\ , \end{aligned} \tag{19}$$

where the linear matrix elements are as follows:

$$\begin{aligned} A_{ij} &= \frac{\partial f_i}{\partial x_j}\left(x^0, u^0\right),\\ B_{ij} &= \frac{\partial f_i}{\partial u_j}\left(x^0, u^0\right),\\ C_{ij} &= \frac{\partial h_i}{\partial x_j}\left(x^0, u^0\right),\\ D_{ij} &= \frac{\partial h_i}{\partial u_j}\left(x^0, u^0\right). \end{aligned} \tag{20}$$

To implement the above linearization theory on the boiler system, the following is done:

$$\begin{aligned} A_{11} &= \frac{\partial f_1}{\partial x_1}\left(x^0, u^0\right) = \frac{\partial}{\partial x_1}\left(-0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3\right)_{(x^0,u^0)}\\ &= -0.0018\frac{9}{8}u_2^0\left(x_1^0\right)^{1/8} = -0.00203u_2^0\left(x_1^0\right)^{1/8},\\ A_{12} &= \frac{\partial f_1}{\partial x_2}\left(x^0, u^0\right) = \frac{\partial}{\partial x_2}\left(-0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3\right)_{(x^0,u^0)} = 0,\\ A_{13} &= \frac{\partial f_1}{\partial x_3}\left(x^0, u^0\right) = \frac{\partial}{\partial x_3}\left(-0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3\right)_{(x^0,u^0)} = 0,\\ A_{21} &= \frac{\partial f_2}{\partial x_1}\left(x^0, u^0\right) = \frac{\partial}{\partial x_1}\left(\frac{(0.73u_2 - 0.16)x_1^{9/8} - x_2}{10}\right)_{(x^0,u^0)}\\ &= \frac{9}{8}\left((0.73u_2 - 0.16)x_1^{9/8}\right)_{(x^0,u^0)} = \left(0.08212u_2^0 - 0.018\right)\left(x_1^0\right)^{\frac{1}{8}},\\ A_{22} &= \frac{\partial f_2}{\partial x_2}\left(x^0, u^0\right) = \frac{\partial}{\partial x_2}\left(\frac{(0.73u_2 - 0.16)x_1^{9/8} - x_2}{10}\right)_{(x^0,u^0)} = -0.1,\\ A_{23} &= \frac{\partial f_2}{\partial x_3}\left(x^0, u^0\right) = \frac{\partial}{\partial x_3}\left(\frac{(0.73u_2 - 0.16)x_1^{9/8} - x_2}{10}\right)_{(x^0,u^0)} = 0,\\ A_{31} &= \frac{\partial f_3}{\partial x_1}\left(x^0, u^0\right) = \frac{\partial}{\partial x_1}\left(\frac{\left[141u_3 - (1.1u_2 - 0.19)x_1\right]}{85}\right)_{(x^0,u^0)}\\ &= \frac{\left(0.19 - 1.1u_2^0\right)}{85},\\ A_{32} &= \frac{\partial f_3}{\partial x_2}\left(x^0, u^0\right) = \frac{\partial}{\partial x_2}\left(\frac{\left[141u_3 - (1.1u_2 - 0.19)x_1\right]}{85}\right)_{(x^0,u^0)} = 0,\\ A_{33} &= \frac{\partial f_3}{\partial x_1}\left(x^0, u^0\right) = \frac{\partial}{\partial x_1}\left(\frac{\left[141u_3 - (1.1u_2 - 0.19)x_1\right]}{85}\right)_{(x^0,u^0)} = 0. \end{aligned} \tag{21}$$

Matrix B drives are also calculated as follows:

$$
\begin{aligned}
B_{11} &= \frac{\partial f_1}{\partial u_1}\left(x^0, u^0\right) = \frac{\partial}{\partial u_1}\left(-0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3\right)_{(x^0,u^0)} = 0.9,\\
B_{12} &= \frac{\partial f_1}{\partial u_2}\left(x^0, u^0\right) = \frac{\partial}{\partial u_2}\left(-0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3\right)_{(x^0,u^0)} = -0.0018\left(x_1^0\right)^{9/8},\\
B_{13} &= \frac{\partial f_1}{\partial u_3}\left(x^0, u^0\right) = \frac{\partial}{\partial u_3}\left(-0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3\right)_{(x^0,u^0)} = -0.15,\\
B_{21} &= \frac{\partial f_2}{\partial u_1}\left(x^0, u^0\right) = \frac{\partial}{\partial u_1}\left(\frac{\left(0.73u_2 - 0.16\right)x_1^{9/8} - x_2}{10}\right)_{(x^0,u^0)} = 0,\\
B_{22} &= \frac{\partial f_2}{\partial u_2}\left(x^0, u^0\right) = \frac{\partial}{\partial u_2}\left(\frac{\left(0.73u_2 - 0.16\right)x_1^{9/8} - x_2}{10}\right)_{(x^0,u^0)} = 0.073\left(x_1^0\right)^{9/8},\\
B_{23} &= \frac{\partial f_2}{\partial u_3}\left(x^0, u^0\right) = \frac{\partial}{\partial u_3}\left(\frac{\left(0.73u_2 - 0.16\right)x_1^{9/8} - x_2}{10}\right)_{(x^0,u^0)} = 0,\\
B_{31} &= \frac{\partial f_3}{\partial u_1}\left(x^0, u^0\right) = \frac{\partial}{\partial u_1}\left(\frac{\left[141u_3 - \left(1.1u_2 - 0.19\right)x_1\right]}{85}\right)_{(x^0,u^0)} = 0,\\
B_{32} &= \frac{\partial f_3}{\partial u_2}\left(x^0, u^0\right) = \frac{\partial}{\partial u_2}\left(\frac{\left[141u_3 - \left(1.1u_2 - 0.19\right)x_1\right]}{85}\right)_{(x^0,u^0)} = -\frac{1.1}{85}x_1^0,\\
B_{33} &= \frac{\partial f_3}{\partial u_3}\left(x^0, u^0\right) = \frac{\partial}{\partial u_1}\left(\frac{\left[141u_3 - \left(1.1u_2 - 0.19\right)x_1\right]}{85}\right)_{(x^0,u^0)} = \frac{141}{85}.
\end{aligned}
\tag{22}
$$

The following are also used to determine the drives of matrices C and D:

$$
\begin{aligned}
C_{11} &= \frac{\partial h_1}{\partial x_1}\left(x^0, u^0\right) = \frac{\partial}{\partial x_1}\left(x_1\right)_{(x^0,u^0)} = 1,\\
C_{12} &= \frac{\partial h_1}{\partial x_2}\left(x^0, u^0\right) = \frac{\partial}{\partial x_2}\left(x_1\right)_{(x^0,u^0)} = 0,\\
C_{13} &= \frac{\partial h_1}{\partial x_3}\left(x^0, u^0\right) = \frac{\partial}{\partial x_3}\left(x_1\right)_{(x^0,u^0)} = 0,\\
C_{21} &= \frac{\partial h_2}{\partial x_1}\left(x^0, u^0\right) = \frac{\partial}{\partial x_1}\left(x_1\right)_{(x^0,u^0)} = 0,\\
C_{22} &= \frac{\partial h_2}{\partial x_2}\left(x^0, u^0\right) = \frac{\partial}{\partial x_2}\left(x_1\right)_{(x^0,u^0)} = 1,\\
C_{23} &= \frac{\partial h_2}{\partial x_3}\left(x^0, u^0\right) = \frac{\partial}{\partial x_3}\left(x_1\right)_{(x^0,u^0)} = 0,\\
C_{31} &= \frac{\partial h_3}{\partial x_1}\left(x^0, u^0\right) = \frac{\partial}{\partial x_1}\left(0.05\left(0.13073x_3 + 100a_{cs} + q_e - 67.975\right)\right)_{(x^0,u^0)} = 5\frac{\partial a_{\mathrm{cs}}}{\partial x_1} + 0.05\frac{\partial q_e}{\partial x_1},
\end{aligned}
\tag{23}
$$

where

$$\frac{\partial a_{\mathrm{cs}}}{\partial x_1}\left(x^0,u^0\right)=\frac{\left[x_3^0\left(1.0394-0.00123404x_1^0\right)\left(0.8-0.00123404x_3^0\right)-\right]}{\left(x_3^0\left(1.0394-0.00123404x_1^0\right)\right)^2}-\frac{\left(1-0.001538x_3^0\right)\left(0.8x_1^0-25.6\right)\left(-0.00123404x_3^0\right)}{\left(x_3^0\left(1.0394-0.00123404x_1^0\right)\right)^2},$$

$$\frac{\partial q_e}{\partial x_1}\left(x^0,u^0\right)=0.854u_2^0-0.147,$$

$$C_{32}=\frac{\partial h_3}{\partial x_2}\left(x^0,u^0\right)=\frac{\partial}{\partial x_2}\left(0.05\left(0.13073x_3+100a_{cs}+q_e-67.975\right)\right)_{(x^0,u^0)}=0,$$

$$C_{33}=\frac{\partial h_3}{\partial x_3}\left(x^0,u^0\right)=\frac{\partial}{\partial x_3}\left(0.05\left(0.13073x_3+100a_{cs}+q_e-67.975\right)\right)_{(x^0,u^0)}=0.0065365+5\frac{\partial a_{\mathrm{cs}}}{\partial x_3},\tag{24}$$

where

$$\frac{\partial a_{\mathrm{cs}}}{\partial x_3}=\frac{\left[x_3^0\left(1.0394-0.00123404x_1^0\right)\left(0.00123404x_1^0+0.039372\right)\right]}{\left(x_3^0\left(1.0394-0.00123404x_1^0\right)\right)^2}-\left(1-0.001538\mathrm{x}_3^0\right)\left(0.8\mathrm{x}_1^0-25.6\right)\left(-0.00123404\mathrm{x}_1^0\right)$$

$$\cdot\left(x_3^0\left(1.0394-0.00123404x_1^0\right)\right)^2,$$

$$D_{11}=\frac{\partial h_1}{\partial u_1}\left(x^0,u^0\right)=\frac{\partial}{\partial u_1}\left(x_1\right)_{(x^0,u^0)}=0,$$

$$D_{12}=\frac{\partial h_1}{\partial u_2}\left(x^0,u^0\right)=\frac{\partial}{\partial u_2}\left(x_1\right)_{(x^0,u^0)}=0,$$

$$D_{13}=\frac{\partial h_1}{\partial u_3}\left(x^0,u^0\right)=\frac{\partial}{\partial u_3}\left(x_1\right)_{(x^0,u^0)}=0,$$

$$D_{21}=\frac{\partial h_2}{\partial u_1}\left(x^0,u^0\right)=\frac{\partial}{\partial u_1}\left(x_2\right)_{(x^0,u^0)}=0,$$

$$D_{22}=\frac{\partial h_2}{\partial u_2}\left(x^0,u^0\right)=\frac{\partial}{\partial u_2}\left(x_2\right)_{(x^0,u^0)}=0,$$

$$D_{23}=\frac{\partial h_2}{\partial u_3}\left(x^0,u^0\right)=\frac{\partial}{\partial u_3}\left(x_2\right)_{(x^0,u^0)}=0,$$

$$D_{31}=\frac{\partial h_3}{\partial u_1}\left(x^0,u^0\right)$$

$$=\frac{\partial}{\partial u_1}\left(0.05\left(0.13073x_3+100a_{\mathrm{cs}}+q_e-67.975\right)\right)_{(x^0,u^0)}=0.2279,$$

$$D_{32}=\frac{\partial h_3}{\partial u_2}\left(x^0,u^0\right)$$

$$=\frac{\partial}{\partial u_1}\left(0.05\left(0.13073x_3+100a_{\mathrm{cs}}+q_e-67.975\right)\right)_{(x^0,u^0)}=0.00427x_1^0,$$

$$D_{33}=\frac{\partial h_3}{\partial u_3}\left(x^0,u^0\right)$$

$$=\frac{\partial}{\partial u_3}\left(0.05\left(0.13073x_3+100a_{\mathrm{cs}}+q_e-67.975\right)\right)_{(x^0,u^0)}=-0.014.\tag{25}$$

Therefore, four matrices A, B, C, and D are obtained as follows.

$$A = \begin{bmatrix} -0.00203u_2^0\left(x_1^0\right)^{1/8} & 0 & 0 \\ \left(0.08212u_2^0 - 0.018\right)\left(x_1^0\right)^{1/8} & -0.1 & 0 \\ \frac{\left(0.19 - 1.1u_2^0\right)}{85} & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.9 & -0.0018\left(x_1^0\right)^{9/8} & -0.15 \\ 0 & 0.073\left(x_1^0\right)^{9/8} & 0 \\ 0 & -\frac{1.1}{85}x_1^0 & \frac{141}{85} \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 5\frac{\partial a_{cs}}{\partial x_1} + 0.05\frac{\partial q_e}{\partial x_1} & 0 & 0.0065365 + 5\frac{\partial a_{cs}}{\partial x_3} \end{bmatrix}, \quad (26)$$

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.2279 & 0.00427x_1^0 & -0.014 \end{bmatrix}.$$

Therefore, a linear state space form can be written for the boiler:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du\ . \end{aligned} \quad (27)$$

The variables y, x, and u are the output, mode, and input of the boiler system at the operating points, respectively.

4. Neural Model Reference Adaptive Control

Here, the controller structure is described (Figure 2). In the previous section, the linear model of the boiler was obtained, and it is used in the model reference block.

In Figure 2, the control system calculates the difference between the boiler output and the reference model. This error signal is then used to train the recurrent type-2 fuzzy radial basis function neural network (RT2FRBFN) [37]. We train the neural network so that the error is less and less, and as a result, the boiler output follows the output of the reference model. In this case, the system is controlled comparatively. Lyapunov's sustainability rule design approach is one of the most important methods of adaptive control [38–40]. This method tries to find the Lyapunov function and a matching mechanism so that the error between the system and the model is zero. This method also ensures the stability of the control parameters. The model reference is usually defined as a first-order system as follows:

$$\frac{dy_m}{dt} = -a_m y_m + b_m u_c, \quad (28)$$

where y_m are the output of the model reference and u_c are the input of the reference model. Suppose the system to be controlled is as follows:

$$\frac{dy}{dt} = -ay + bu, \quad (29)$$

where y is the output of the system and u is its input. The control signal with the adjustable parameters θ_1 and θ_2 is selected as follows:

$$u = \theta_1 u_c - \theta_2 y. \quad (30)$$

The error is calculated as follows:

$$e = y - y_m. \quad (31)$$

Since the above error must be zero, the error changes must be negative. So,

$$\begin{aligned} \dot{e} &= \dot{y} - \dot{y}_m \\ &= -ay + bu—a_m y_m + b_m u_c \\ &= -ay + bu + a_m y_m - b_m u_c. \end{aligned} \quad (32)$$

So, we will have

$$\begin{aligned} \dot{e} &= -ay + b\left(\theta_1 u_c - \theta_2 y\right) + a_m y_m - b_m u_c \\ &= -ay - b\theta_2 y + a_m y_m - \left(b\theta_1 - b_m\right)u_c \\ &= -a_m e - \left(b\theta_2 + a - a_m\right)y + \left(b\theta_1 - b_m\right)u_c. \end{aligned} \quad (33)$$

To find the law of conformity, consider Lyapunov's function as follows:

$$V\left(e, \theta_1, \theta_2\right) = \frac{1}{2}\left[e^2 + \frac{1}{b\gamma}\left(b\theta_2 + a - a_m\right)^2 + \frac{1}{b\gamma}\left(b\theta_1 - b_m\right)^2\right]. \quad (34)$$

Based on the above Lyapunov function, we will have

$$\begin{aligned} \dot{V} &= e\frac{de}{dt} + \frac{1}{\gamma}\left(b\theta_2 + a - a_m\right)\frac{d\theta_2}{dt} + \frac{1}{\gamma}\left(b\theta_1 - b_m\right)\frac{d\theta_1}{dt} \\ &= -a_m e^2 + \frac{1}{\gamma}\left(b\theta_2 + a - a_m\right)\left(\frac{d\theta_2}{dt} - \gamma y e\right) \\ &\quad + \frac{1}{\gamma}\left(b\theta_1 - b_m\right)\left(\frac{d\theta_1}{dt} + \gamma u_c e\right). \end{aligned} \quad (35)$$

In order for the above statement to be negative, we must have

$$\frac{d\theta_1}{dt} = -\gamma u_c e, \quad (36)$$

$$\frac{d\theta_2}{dt} = \gamma y e, \quad (37)$$

where γ is the learning rate. So,

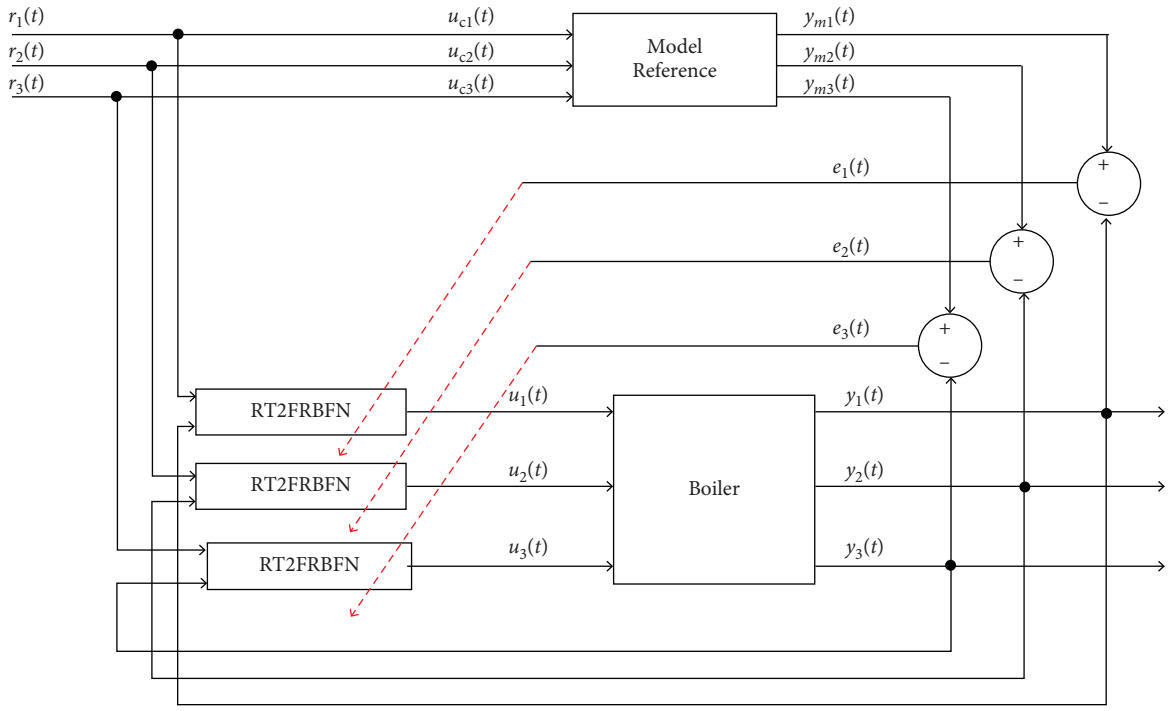


Figure 2: The suggested control structure.

$$\dot{V} = -a_m e^2. \tag{38}$$

Therefore, the system will be stable and the change of parameters will be in order to reduce the error. The adaptation law for training the neural network uses equations (36) and (37).

5. Experimental Results

In this section, the performance of the control method of the proposed model reference for boiler control is examined. The industrial boiler is shown in Figure 3. To test the control system, assume that the boiler output steam pressure in the first 100 seconds is 150 kg/cm^2, from 200 seconds to 200 seconds, it is 300 kg/cm^2, and, finally, from 200 to 300 seconds per second, it is a value of 150 kg/cm^2.

Figure 4 illustrates the performance of the proposed neural controller.

In the following, the control of the electric power output as the second component of the boiler output that must be controlled is presented. Assume that, according to the above steam pressure control scenario, the output power is to be 50 kW in the first 100 seconds, 200 kW from 100 to 100 seconds, and, finally, 300 kW from 200 seconds to 300 seconds. Figure 5 illustrates the designed control performance in electric power control. Three perceptron neural networks have been used, and the number of hidden layer is 16, 14, and 17, respectively.

The following examines the control of the water level of the drum as the third component of the boiler output that should be controlled. Assume that, according to the boiler steam pressure control scenario, the water level of the drum is supposed to be 400 cm in the first 100 seconds, 200 cm

Figure 3: The industrial boiler.

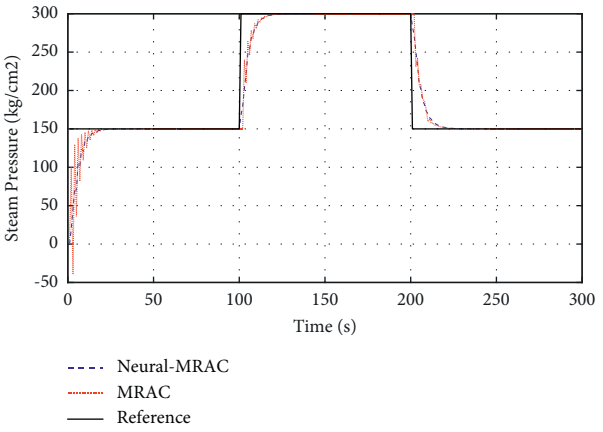


Figure 4: Control of the boiler steam pressure model reference.

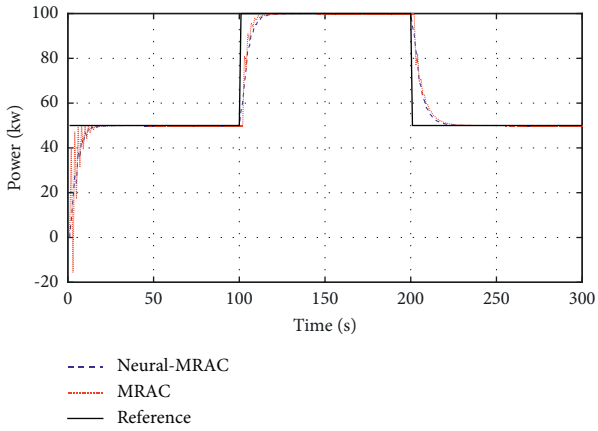


Figure 5: Control of the power reference model.

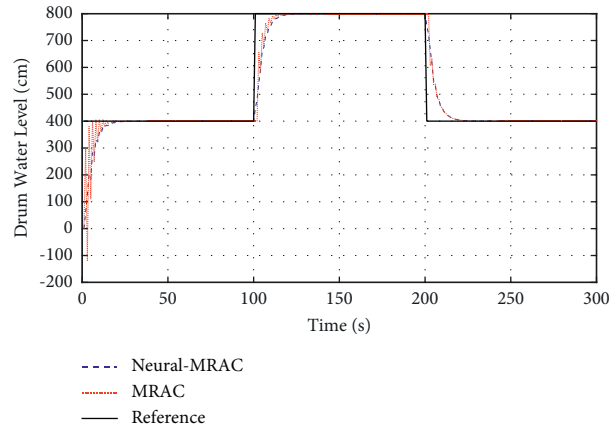


Figure 6: Droplet water model reference control.

from 100 to 100 s in the second, and 300 to 400 cm in the second from 200 to 300 s. Figure 6 presents control performance in drum water level control.

The above results are related to an ideal boiler with constant coefficients. In other words, the coefficients and parameters of the boiler are considered unchanged with time; as in practice, some coefficients may change for various reasons, such as wear and tear over time, high heat, humidity, and environmental conditions. In this case, a proper controller must be robust to parametric variations. To check the performance of the proposed model reference control system in the presence of parametric changes, assume that the coefficients of each sentence in the equations of the boiler state space change by ±10% and their nominal value randomly. In this case, the performance of the proposed reference control system model in boiler steam pressure control is shown in Figure 7.

Figure 8 shows the electrical power control outputs, despite the uncertain parameter.

In the following, the indefinite effect of the parameter in controlling the water level of the drum is examined. Figure 9 illustrates the control function of the reference water level model of the drum despite the uncertain parameter.

However, there is another uncertainty that stems from a lack of accurate modeling of the system. It is very important to examine this category of indeterminacy, especially in the model reference control method, because the basis of the model reference method is model based and is highly dependent on the exact model. To add a nonparametric indeterminacy, since x_1 is present in most of the equations of the boiler system equation, so we convert the terms from $x_1^{9/8}$ to $x_1^{9/8} + 0.2x_1^{1/8}$. It should be noted that any coefficient and power can be considered and the control system evaluated. Also, with an indefinite magnification, it is observed that the control system has gradually become weaker, and from now on, the system cannot be controlled with a large indeterminacy. Therefore, in this category, any indefinite type can be tested with trial and error. Figure 10 shows the performance of the proposed model reference control system in the boiler output steam pressure in the presence of nonparametric uncertainty.

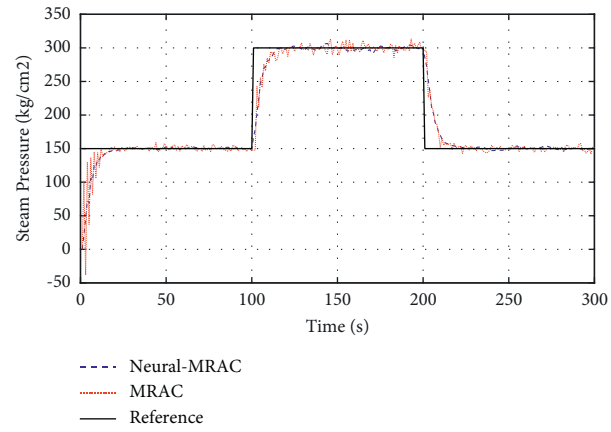


Figure 7: Steam pressure control in the presence of an uncertain parameter of ±10%.

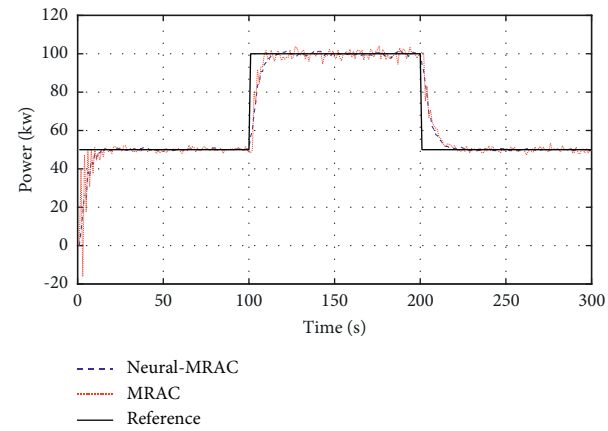


Figure 8: Electrical power model reference control in the presence of an uncertain parameter of ±10%.

Figure 11 shows the performance of the model reference control system to control the electric power in the presence of an indefinite nonparametric presence.

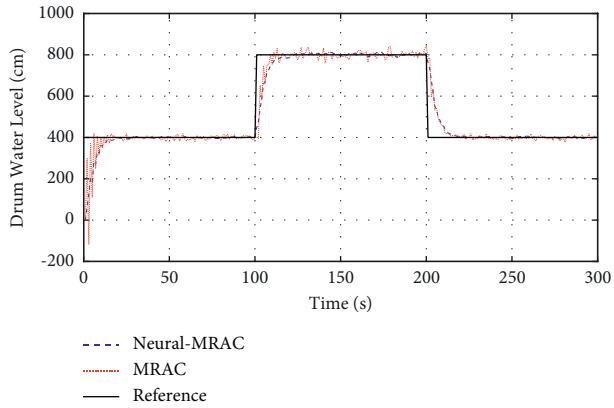


Figure 9: Control of the model reference of the drum water level in the presence of an uncertain parameter of ±10%.

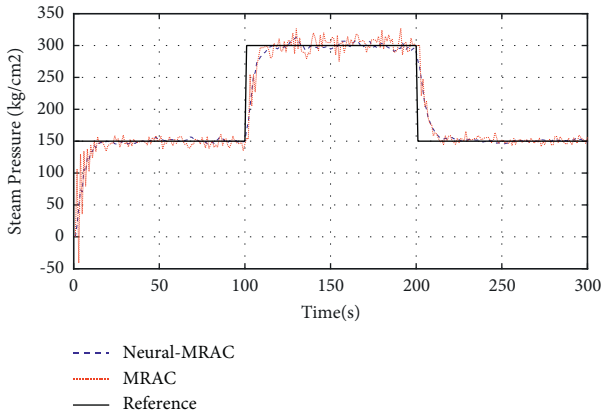


Figure 10: Control of the steam pressure model reference in the presence of nonparametric uncertainty.

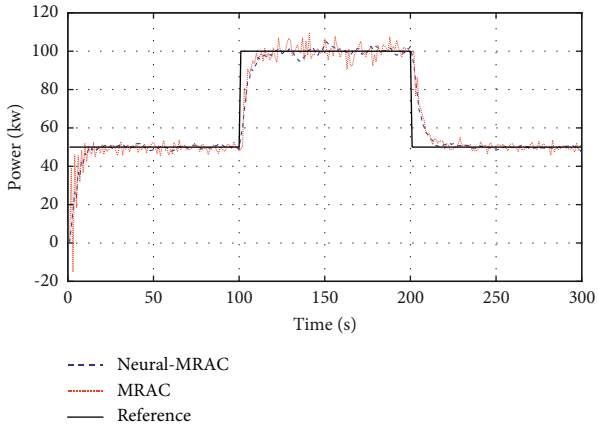


Figure 11: Control of the power model reference in the presence of nonparametric uncertainty.

Figure 12 shows the performance of the model reference control system to control the water level of the drum in the presence of nonparametric uncertainty.

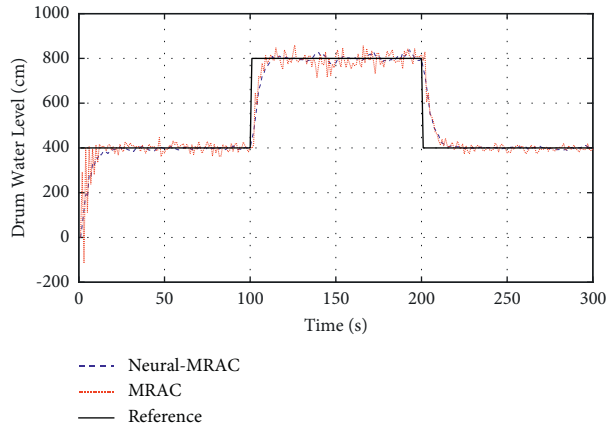


Figure 12: Control of the model reference of the drum water level in the presence of nonparametric uncertainty.

As shown in Figure 4 to 12, the use of recurrent type-2 fuzzy RBFN in the model reference control method leads to a significant reduction in fluctuations and control ripples. Also, when the reference signal has a sudden change, the recurrent type-2 fuzzy RBFN immediately directs the output of the system in the direction of the change and there is no time delay. For further comparison, Table 2 shows the performance of the proposed recurrent type-2 fuzzy neural network (RT2FRBFN) with two methods including normal (without feedback) type-2 fuzzy neural network (T2FRBFN) and type-1 fuzzy neural network (T1FRBFN).

Table 2 shows significant results. First, the number of fuzzy rules in RT2FRBFN is much less than the other two methods, especially, compared with the type-1 fuzzy neural network. Secondly, with an uncertainty increase, the number of fuzzy rules for our proposed model has only increased by 1, but in other methods, more than 1 rule has been added. The RMSE index also confirms the superiority of our proposed model. However, the training time in our proposed model is longer than the model without feedback, which is normal and related to feedback calculations.

Table 2: Comparison between our proposed method with other two methods.

	Number of fuzzy rules		Root mean squares error (RMSE)		Training time	
	Unc. 0%	Unc. 10%	Unc. 0%	Unc. 10%	Unc. 0%	Unc. 10%
T1FRBFN	26	41	0.275	0.344	51	55
T2FRBFN	9	11	0.093	0.115	33	38
RT2FRBFN	6	7	0.072	0.098	38	42

6. Conclusion

Today, boilers are widely used in the industry. Precise control and mastery of the boilers and their output parameters are very much needed. Since the nature of thermal systems in terms of control is complex systems, therefore, no control method can be used. In this paper, in order to control the boiler, the method of adaptive model reference based on recurrent type-2 fuzzy RBFN was proposed. The proposed method is resistant to parametric and nonparametric changes, and since the control system depends on the control system, so with the model changes, the control system also changes. Two indefinite parametric and nonparametric models were considered for the system. In the indefinite parameter, it was assumed that the parameters in the system model would change, and in the nonparametric indefiniteness, terms were added to the system model. The results showed that, with about ±10% increase in the amount of both indeterminate models, the model reference control system provided good capability. However, with the further increase of uncertainties (for example, about ±20%), the model reference control system does not function properly. Comparing the conventional model reference method and the recurrent type-2 fuzzy RBFN-based model reference method, it has been observed that the use of recurrent type-2 fuzzy RBFN leads to a reduction in control ripples as well as a reduction in the time delay of the control system.

Data Availability

The industrial boiler data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Md. N. Hossain, K. Ghosh, and N. K. Manna, "A multiphase model for determination of minimum circulation ratio of natural circulation boiler for a wide range of pressure," *International Journal of Heat and Mass Transfer*, vol. 150, 2020.

[2] S. Ghabraei, H. Moradi, and G. Vossoughi, "Design & application of adaptive variable structure & Hꝏ robust optimal schemes in nonlinear control of boiler-turbine unit in the presence of various uncertainties," *Energy*, vol. 142, pp. 1040–1056, 2018.

[3] J. Peng, Z. Cao, X. Yu, Y. Yu, G. Chang, and Z. Wang, "Investigation of flame evolution in heavy oil boiler bench using high-speed planar laser-induced fluorescence imaging," *Applied Sciences*, vol. 8, no. 9, p. 1691, 2018.

[4] S. Kamerling, V. Vuillerme, and S. Rodat, "Solar field output temperature optimization using a MILP algorithm and a 0D model in the case of a hybrid concentrated solar thermal power plant for SHIP applications," *Energies*, vol. 14, no. 13, p. 3731, 2021.

[5] O. Mohamed, A. Khalil, and J. Wang, "Modeling and control of supercritical and ultra-supercritical power plants: a review," *Energies*, vol. 13, no. 11, p. 2935, 2020.

[6] F. Xia, Y.-L. Quan, H. Zhang, and D.-G. Peng, "Water level deviation analysis of levels between two sides of the boiler drum," *International Journal of Automation and Computing*, vol. 13, no. 2, pp. 142–150, 2016.

[7] F. M. de Mello, A. J. G. da Cruz, and R. de Sousa, "Fuzzy control applied to combustion in sugarcane bagasse boilers," in *Computer Aided Chemical Engineering*, A. A. Kiss, Ed., Elsevier, Amsterdam, Netherlands, pp. 1135–1140, 2019.

[8] J.-L. Gao, "Research on boiler water supply control system based on AT89C55 and fractional order PID algorithm," *Procedia Computer Science*, vol. 154, pp. 173–180, 2019.

[9] S. Darwish, A. Pertew, W. Elhaweet, and A. Mokhtar, "Advanced boiler control system for steam power plants using modern control techniques," in *Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, Vancouver, Canada, June 2019.

[10] M. H. Toodeshki and J. Askari, "Model-Reference Adaptive Control for a nonlinear boiler-turbine system," in *Proceedings of the 2008 IEEE International Conference on Industrial Technology*, Chengdu, China, April 2008.

[11] J. Tavoosi, A. A. Suratgar, and M. B. Menhaj, "Stability analysis of a class of MIMO recurrent type-2 fuzzy systems," *International Journal of Fuzzy Systems*, vol. 19, no. 3, pp. 895–908, 2017.

[12] J. Tavoosi, A. A. Suratgar, and M. B. Menhaj, "Stable ANFIS2 for nonlinear system identification," *Neurocomputing*, vol. 182, pp. 235–246, 2016.

[13] M. Krawczak and G. Szkatuła, "On matching of intuitionistic fuzzy sets," *Information Sciences*, vol. 517, pp. 254–274, 2020.

[14] J. Tavoosi and B. MA, "A class of type-2 fuzzy neural networks for nonlinear dynamical system identification," *Neural Computing & Applications*, vol. 23, no. 34, pp. 707–717, 2013.

[15] J. Tavoosi, A. A. Suratgar, and M. B. Menhaj, "Stability analysis of recurrent type-2 TSK fuzzy systems with nonlinear consequent part," *Neural Computing & Applications*, vol. 28, no. 1, pp. 47–56, 2017.

[16] J. Tavoosi, "An experimental study on inverse adaptive neural fuzzy control for nonlinear systems," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 2020, In press.

[17] M. B. B. Sharifian, A. Mirlo, J. Tavoosi, and M. Sabahi, "Self-adaptive RBF neural network PID controller in linear elevator," in *Proceedings of the International Conference on Electrical Machines and Systems*, Beijing, China, August 2011.

[18] J. Tavoosi, M. Alaei, and B. Jahani, "Temperature control of water bath by using neuro-fuzzy controller," in *Proceedings of the 5th Symposium on Advance in Science and Technology*, Mashhad, Iran, May 2011.

[19] B. Hu and J. Wang, "Deep learning based hand gesture recognition and UAV flight controls," *International Journal of Automation and Computing*, vol. 17, no. 1, pp. 17–29, 2020.

[20] M. Chavoshian, M. Taghizadeh, and M. Mazare, "Hybrid dynamic neural network and PID control of pneumatic artificial muscle using the PSO algorithm," *International Journal of Automation and Computing*, vol. 17, no. 3, pp. 428–438, 2020.

[21] J. Tavoosi, "A novel recurrent type-2 fuzzy neural network for stepper motor control mechatronic systems and control," vol. 49, no. 1, 2021.

[22] L. Kong, W. He, C. Yang, Z. Li, and C. Sun, "Adaptive fuzzy control for coordinated multiple robots with constraint using impedance learning," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3052–3063, 2019.

[23] W. He, H. Gao, C. Zhou, C. Yang, and Z. Li, "Reinforcement learning control of a flexible two-link manipulator: an experimental investigation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11. In press, 2020.

[24] Y. P. Asad, A. Shamsi, and J. Tavoosi, "Backstepping-based recurrent type-2 fuzzy sliding mode control for MIMO systems (MEMS triaxial gyroscope case study)," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 2, pp. 213–233, 2017.

[25] Y. Pour Asad, A. Shamsi, H. Ivani, and J. Tavoosi, "Adaptive intelligent inverse control of nonlinear systems with regard to sensor noise and parameter uncertainty (magnetic ball levitation system case study)," *International Journal on Smart Sensing and Intelligent Systems*, vol. 9, no. 1, 2016.

[26] J. Tavoosi, A. Shamsi Jokandan, and M. A. Daneshwar, "A new method for position control of a 2-DOF robot arm using neuro-fuzzy controller," *Indian Journal of Science and Technology*, vol. 5, no. 3, 2012.

[27] J. Tavoosi, B. Ma, and S. Ghaemi, "Adaptive inverse control of nonlinear dynamical system using type-2 fuzzy neural networks," *Journal of Control*, vol. 5, no. 2, pp. 52–60, 2011.

[28] W. Yu, F. Zhao, H. Xu et al., "Predictive control of CO2 emissions from a grate boiler based on fuel nature structures using intelligent neural network and Box-Behnken design," *Energy Procedia*, vol. 158, pp. 364–369, 2019.

[29] F. M. de Mello, A. J. G. da Cruz, R. de Sousa, and R. de Sousa, "Fuzzy control applied to combustion in sugarcane bagasse boilers," in *Computer Aided Chemical Engineering*, A. A. Kiss, E. Zondervan, R. Lakerveld, and L. Özkan, Eds., vol. 46, pp. 1135–1140, Elsevier, Amsterdam, Netherlands, 2019.

[30] X. Zhuo, C. Lou, H. Zhou, J. Zhuo, and P. Fu, "Hierarchical Takagi-Sugeno fuzzy hyperbolic tangent static model control for a circulating fluidized bed boiler thermal power unit," *Energy*, vol. 162, pp. 910–917, 2018.

[31] D. Xiu-Cheng, W. Hai-Bin, and Z. Xiao-Xiao, "Model reference neural network control for boiler combustion system," in *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, August 2005.

[32] J.-Q. Li, J.-Z. Liu, Y.-G. Niu, C.-L. Niu, and W. Liu, "Application of neural network model reference adaptive control in coal-fired boiler combustion system," in *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics*, pp. 564–567, Shanghai, China, August 2004.

[33] K. Xiangsong, C. Xurui, and G. Jiansheng, "PID controller design based on radial basis function neural networks for the steam generator level control," *Cybernetics and Information Technologies*, vol. 16, no. 5, pp. 15–26, 2016.

[34] S. Yuan, B. De Schutter, and S. Baldi, "Robust adaptive tracking control of uncertain slowly switched linear systems," *Nonlinear Analysis: Hybrid Systems*, vol. 27, pp. 1–12, 2018.

[35] S. Yuan, B. De Schutter, and S. Baldi, "Adaptive asymptotic tracking control of uncertain time-driven switched linear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5802–5807, 2017.

[36] K. Åström and R. Bell, "Dynamic models for boiler-turbine alternator units: data logs and parameter estimation for a 160 MW unit," Technical Reports (TFRT-3192), 1987.

[37] J. Tavoosi, A. A. Suratgar, and M. B. Menhaj, "Nonlinear system identification based on a self-organizing type-2 fuzzy RBFN," *Engineering Applications of Artificial Intelligence*, vol. 54, 2016.

[38] B. Vaseghi, M. A. Pourmina, and S. Mobayen, "Finite-time chaos synchronization and its application in wireless sensor networks," *Transactions of the Institute of Measurement and Control*, vol. 40, no. 13, pp. 3788–3799, 2018.

[39] M. Firouzi, M. Nasiri, S. Mobayen, and G. B. Gharehpetian, "Sliding mode controller-based BFCL for fault ride-through performance enhancement of DFIG-based wind turbines," *Complexity*, vol. 2020, Article ID 1259539, 12 pages, 2020.

[40] M. Golestani, S. Mobayen, and H. Richter, "Fast robust adaptive tracker for uncertain nonlinear second-order systems with time-varying uncertainties and unknown parameters," *International Journal of Adaptive Control and Signal Processing*, vol. 32, no. 12, pp. 1764–1781, 2018.

Hindawi
Complexity
Volume 2021, Article ID 3511375, 17 pages
https://doi.org/10.1155/2021/3511375

Research Article

Computer Vision-Based Patched and Unpatched Pothole Classification Using Machine Learning Approach Optimized by Forensic-Based Investigation Metaheuristic

Nhat-Duc Hoang,[1,2] **Thanh-Canh Huynh**,[1,2] **and Van-Duc Tran**[2,3]

[1]*Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam*
[2]*Faculty of Civil Engineering, Duy Tan University, Da Nang 550000, Vietnam*
[3]*International School, Duy Tan University, Da Nang 550000, Vietnam*

Correspondence should be addressed to Nhat-Duc Hoang; hoangnhatduc@duytan.edu.vn

Received 29 April 2021; Revised 21 July 2021; Accepted 25 August 2021; Published 6 September 2021

Academic Editor: Gonzalo Farias

During the phase of periodic asphalt pavement survey, patched and unpatched potholes need to be accurately detected. This study proposes and verifies a computer vision-based approach for automatically distinguishing patched and unpatched potholes. Using two-dimensional images, patched and unpatched potholes may have similar shapes. Therefore, this study relies on image texture descriptors to delineate these two objects of interest. The texture descriptors of statistical measurement of color channels, the gray-level cooccurrence matrix, and the local ternary pattern are used to extract texture information from image samples of asphalt pavement roads. To construct a classification model based on the extracted texture-based dataset, this study proposes and validates an integration of the Support Vector Machine Classification (SVC) and the Forensic-Based Investigation (FBI) metaheuristic. The SVC is used to generalize a classification boundary that separates the input data into two class labels of patched and unpatched potholes. To optimize the SVC performance, the FBI algorithm is utilized to fine-tune the SVC hyperparameters. To establish the hybrid FBI-SVC framework, an image dataset consisting of 600 samples has been collected. The experiment supported by the Wilcoxon signed-rank test demonstrates that the proposed computer vision is highly suitable for the task of interest with a classification accuracy rate = 94.833%.

1. Introduction

The network of asphalt pavement roads is a crucial element of infrastructure in modern societies [1–4]. As pointed out by [5], asphalt pavement roads significantly support social interaction as well as economic development. In many regions in the world, economic growth is correlated with the extension of asphalt pavement networks. Nevertheless, since these networks are constantly expanded in recent decades, maintaining them becomes a costly and arduous task especially for developing countries like Vietnam. It is because the financial resources of developing countries are often restricted and central governments or provincial authorities are struggling to find a balance between the funding used to construct new road networks and the funding needed to recover deteriorated existing ones.

Poorly maintained asphalt pavements lead to a vast number of traffic accidents. According to the record of WHO [6], the number of people lives taken by traffic crashes is roughly 1.35 million per year. Moreover, traffic accidents cost most nations 3% of their gross domestic product. Particularly in Vietnam, there were 14.510 traffic accident cases in 2020 alone; these accidents caused the death of 6700 people and injured 10840 ones [7]. Thus, the proper survey on the road status and timely maintenance is crucial for identifying and restoring pavement defects. These can help to reduce the number of traffic accidents.

Among various types of asphalt pavement distress (e.g., cracks, potholes, patches, raveling, bleeding, depression, etc.), potholes are easily encountered and have been recognized as a dangerous type of defect. Potholes are typically observed as bowl-shaped holes in the pavement surface

caused by the removal of surfacing materials (refer to Figure 1). This type of defect causes a sudden change in road elevation and creates hazardous situations for drivers especially in the cases of inclement weather conditions (e.g., heavy rainfall). Therefore, pavement roads damaged by potholes must be quickly identified and maintained to recover smooth-running surfaces [5].

Patching is the common technique employed to deal with pavement potholes (refer to Figure 2). Pavement patching involves the processes of filling potholes with surfacing materials such as hot mix asphalt or asphalt emulsion mixes. To enhance the efficiency of the pavement recovering task, patching should be performed timely when potholes are early formed. If left unpatched, potholes can be the cause of serious traffic crashes and human casualties especially for motorcycle drivers. Moreover, it is beneficial to prevent water intrusion into the pavement structure as soon as possible. Water intrusion has been seen as the main factor that deepens potholes or triggers other severe failures in the asphalt pavement surface. These facts emphasize the importance of periodic pavement surveys and timely as well as the correct identification of potholes.

In many developing countries, pavement surveys based on human technicians and the manual process of pavements' visual data processing are still required and performed by local authorities or traffic management agencies. Although these processes can attain high accuracy of pavement evaluation, the manual process is painstakingly low in productivity. Moreover, a large and expanding network of asphalt pavements slows down the pavement survey process and makes timely detection of pavement distresses an impossible mission. Another issue of the manual process is that the survey outcome is considerably affected by subjective judgments of human inspectors. Therefore, scholars and practitioners are increasingly relied on automated approaches to improving the productivity and objectivity of periodic road pavement surveys.

In recent years, due to the availability of low-cost digital cameras and rapid advancement of 2-dimensional (2D) digital image processing techniques, computer vision has gained popularity and has been proven to be a feasible tool for asphalt pavement surveys. Nevertheless, computer vision-based automatic pothole detection still presents a challenging task due to the complex nature of the asphalt pavement background. Pavement distresses often coexist with noisy signals caused by stains, irregular lighting conditions, traffic marks, etc.

Accordingly, various computer vision-based methods have been proposed in the literature to deal with the problem of pavement pothole recognition. Zhou et al. [8] propose a replacement of a low-speed human-based approach using an integrated image processing system; this system mainly relies on discrete wavelet transform for asphalt pavement distress classification. A comparative study performed in [9] assesses several multiresolution texture analysis methods based on a wavelet, ridgelet, and curvelet-based texture extractors used for pothole detection. Koch and Brilakis [10] rely on histogram shape-based thresholding coupled with morphological operators to isolate pothole shape; the authors employ 70 image samples for testing and obtain a classification accuracy of 85%.

A pothole recognition and evaluation scheme based on 2D shape analysis, image coarseness comparison, and image thresholding techniques has been proposed in [11]. Buza et al. [12] put forward an unsupervised learning method based on image processing techniques and spectral clustering. Sundra Murthy and Varaprasad [13] employ histogram analysis, edge detection, and contour following to deal with the task of interest. Ryu et al. [14] attempt to improve an existing intelligent transportation system service by the utilization of a 2D image-based pothole detection with image processing techniques of segmentation, region extraction, and morphological filtering. A novel approach based on semantic texton forests used with 2D video frames has been proposed in [15]. Kamaliardakani et al. [16] employ a heuristic thresholding method for detecting sealed crack damages. Fuzzy c-means clustering algorithm and morphological reconstruction have been utilized by [17] for recognizing potholes on asphalt pavement based on 2D-color images.

Encouraged by the successes of image processing techniques applied in pothole detection, more recent works have explored the feasibility of advanced supervised machine learning models in dealing with the task at hand. Yousaf et al. [5] construct a support vector machine (SVM) trained by a set of scale-invariant feature transform (SIFT) features for recognizing potholes in labeled images. A machine learning-based approach using least squares support vector machine and neural network with steerable filter-based feature extraction has been proposed in [18]. Maeda et al. [19] and Cao et al. [20] recently put forward deep neural network-based approaches for recognizing asphalt pavement defects including potholes.

Overall, there is an increasing trend of applying image processing and machine learning methods for pothole detection using 2D digital images. This trend in the academic community has been recognized by previous reviewing works of [21–23]. In line with this trend of study and motivated by the fact that asphalt pavement roads in different regions may exhibit different surfacing features due to various factors including the use of surfacing materials, construction methods, traffic loads, and other weather/local conditions, it is necessary to investigate other advanced image processing and machine learning solutions for dealing with the task of automatic pothole detection. The reason is that because of discrepancies in characteristics of asphalt pavement roads, a data-driven method can obtain good detection accuracy for a certain study area but it may not perform well on image data collected from other study regions.

More importantly, it is necessary for road surveying systems to be able to distinguish between unpatched potholes and patched potholes. The reason is that if patched objects are correctly identified, the false positive rate of the pothole detection process can be reduced. In addition, as pointed out in previous studies [24–26], patched areas are considered pavement defects and they should be detected with high accuracy. As observed from Figures 1 and 2, it can

Figure 1: Appearances of potholes in the asphalt pavement surface.

Figure 2: Appearances of patched potholes in the asphalt pavement surface.

be argued that using 2D image samples, potholes and patched potholes can have similar shapes. Therefore, image texture analysis used for extracting the coarseness of image regions is helpful to recognize them. Texture descriptors [27–42] have been proved to be highly useful for image classification in various fields. In this study, the highly discriminative local ternary pattern is employed.

In addition, based on the current literature, it can be seen that pothole detection methods have mainly relied on individual machine learning methods. Integrations of machine learning and metaheuristic approaches have rarely been investigated for the task at hand. In various fields, the successful utilization of metaheuristics in optimizing machine learning models has been demonstrated [43–46]. Nevertheless, the applications of such hybrid scheme for pothole recognition are still limited.

Thus, the current study is an attempt to fill this gap in the literature by proposing a machine learning-metaheuristic integration for coping with the problem of interest. The employed machine learning model is SVM [47] because this machine learning method has been proven to be a highly capable tool for pattern recognition especially for asphalt pavement image data [5, 48, 49]. To further optimize the performance of the SVM model used for the task of the pothole and patched pothole detection, this study relies on a newly proposed metaheuristic of Forensic-Based Investigation (FBI) [50]. FBI is a novel metaheuristic motivated by the suspect investigation-location-pursuit process that is used by police officers and its performance has been confirmed by various optimization tasks. Accordingly, the current work proposes combining the two methods of SVM and FBI to establish an integrated data-driven model utilized for computer vision-based pothole-patched pothole recognition.

The subsequent parts of the study are organized as follows. The second section reviews the research methodology including the techniques of image texture analysis, the computational intelligence approaches, and a set of collected image samples. The next section describes the structure of the proposed integration of FBI optimized SVM used for pothole-patched pothole detection. The fourth section reports experimental results. Concluding remarks of this study are stated in the final section.

2. Research Methodology

This section of the article reviews the employed research method including image texture used for feature extraction, the machine learning approach of SVM used for pattern recognition, the FBI metaheuristic used for model optimization, and the collected pavement image dataset.

2.1. Image Texture Analysis. In the image processing field, the texture is a crucial tool used for visual perception and has been the core of many computer vision systems. Texture analysis is used to represent the degree of coarseness or fineness of objects within digital image samples. This type of analysis has been widely used in various fields of study including civil engineering [51], remote sensing [52], biomedical imaging [53], and industrial engineering [54, 55]. Based on the collected image samples, meaningful features that represent the texture properties of image regions can be computed and used for object classification [34]. In this study, statistical measurement of image pixel intensity, gray-level cooccurrence matrix's properties, and local ternary pattern are the utilized texture descriptors used for distinguishing between potholes and patched potholes.

2.1.1. Statistical Properties of Color Texture. Given an image sample I within which there are 3 color channels of red (R), green (G), and blue (B), a first-order histogram $P(I)$ can be calculated for each channel to represent its statistical distribution [56]. Based on the computed first-order histogram of the three color channels (R, G, and B), the indices of mean (μ_c), standard deviation (σ_c), the skewness (δ_c), kurtosis

(η_c), entropy (ρ_c), and range (Δ_c) are computed separately for each channel. Since each channel yields 6 color texture-based indices, the total number of extracted features representing the statistical properties of color texture is $6 \times 3 = 18$. The equations used for computing the 6 color texture-based indices for each color channel c are presented as follows [51]:

$$\mu_c = \sum_{i=0}^{NL-1} I_{i,c} \times P_c(I), \tag{1}$$

$$\sigma_c = \sqrt{\sum_{i=0}^{NL-1} \left(I_{i,c} - \mu_c\right)^2 \times P_c(I)}, \tag{2}$$

$$\delta_c = \frac{\sum_{i=0}^{NL-1} \left(I_{i,c} - \mu_c\right)^3 \times P_c(I)}{{\sigma_c}^3}, \tag{3}$$

$$\eta_c = \frac{\sum_{i=0}^{NL-1} \left(I_{i,c} - \mu_c\right)^4 \times P_c(I)}{{\sigma_c}^4}, \tag{4}$$

$$\rho_c = -\sum_{i=0}^{NL-1} P_c(I) \times \log_2\left(P_c(I)\right), \tag{5}$$

$$\Delta_c = \mathrm{Max}\left(I_c\right) - \mathrm{Min}\left(I_c\right), \tag{6}$$

where $NL = 256$ denotes the number of discrete intensity values with an 8-bit image sample.

2.1.2. Gray-Level Cooccurrence Matrix (GLCM). GLCM [57, 58] is a useful image processing tool used for representing properties regarding the repeated occurrence of certain gray-level patterns. Herein, the first is the analysis to convert an RGB image to a grayscale one [59]. Subsequently, the relationship between 2 pixels within this grayscale image sample is modeled using two parameters: r-the distance relationship and θ-the rotation relationship. Accordingly, a matrix denoting P_δ is used to represent a probability of the two gray levels of i and j with respect to the relationship dictated by r and θ [60]. Based on the normalized P_δ denoted as P_δ^N, the four indices of the angular second moment (AM), contrast (CO), correlation (CR), and entropy (ET) can be calculated as follows [58]:

$$\mathrm{AM} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_\delta^N(i,j)^2, \tag{7}$$

$$\mathrm{CO} = \sum_{k=0}^{N_g-1} k^2 \sum_{\substack{i=1 \\ |i-j|=k}}^{N_g} \sum_{j=1}^{N_g} P_\delta^N(i,j), \tag{8}$$

$$\mathrm{CR} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i \times j \times P_\delta^N(i,j) - \mu_X \mu_Y}{\sigma_X \sigma_Y}, \tag{9}$$

$$\mathrm{ET} = -\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_\delta^N(i,j) \log\left(P_\delta^N(i,j)\right), \tag{10}$$

where N_g is the number of gray-level values. μ_X, μ_Y, σ_X, and σ_Y are the means and standard deviations of the marginal distribution associated with $P_\delta^N(i,j)$ [51, 57].

2.1.3. Local Ternary Pattern (LTP). Putting forward in [61], LTP is an extension of the well-defined Local Binary Pattern (LBP). LBP [62] is an effective tool for characterizing local gray intensity patterns. This method considers a local neighborhood around a certain pixel and thresholds the neighboring pixels based on the value of the central pixels. Accordingly, a local texture descriptor consisting of a 3×3 binary matrix is used to describe the local gray intensity structure of a certain region within an image sample.

Tan and Triggs [61] propose using an additional thresholding value to construct a ternary structure which can be more discriminative and less sensitive to noise in uniform regions. Therefore, the LTP texture descriptor is highly suitable for modeling gray intensity patterns of asphalt pavement roads. Using the LTP method, the value of each thresholded pixel within a local region is either −1, 0, or 1. Therefore, the computation of LTP is split into two separated matrices called a lower pattern and an upper pattern (refer to Figure 3). Similar to the standard LBP, histograms of the two aforementioned matrices can be calculated and concatenated to establish an integrated texture discriminator.

Mathematically speaking, the LTP texture descriptor can be described as follows:

$$s'(u, i_c, t) = \begin{cases} 1 \text{ if } u \geq i_c + t \\ 0 \text{ if } \left|u - i_c\right| < t \\ -1 \text{ if } u \leq i_c - t \end{cases}, \tag{11}$$

where i_c is the gray intensity of the central pixel. $t = 5$ denotes an additional threshold. u is the grayscale value of a neighboring pixel.

2.2. Support Vector Machine Classification (SVC). SVC, introduced by [47], is a powerful supervised learning method used for multivariate data analysis especially for pattern classification. The model structure of this machine learning method is trained via the framework of a structural risk minimization scheme [63]. This feature helps an SVC model to be resilient to noisy data samples and to better guard against overfitting. Given a training dataset set $D = \{(x, y) | x \in S \text{ and } y = f(x)\}$ and a target function f: $X \longrightarrow \{-1, +1\}$ (−1 = "patched pothole" and +1 = "pothole"), the SVM can be used to learn $f(x)$ by constructing an approximated function $\widehat{f}(x)$: $X \longrightarrow \{-1, +1\}$.

The overall learning phase of SVC is illustrated in Figure 4. The first step of the learning process is to construct a function to map the data in the original input space into a high-dimensional feature space. It is noted that the original input space consists of numerical data generated from the

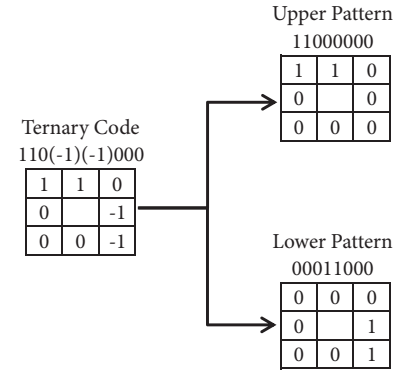


Figure 3: Demonstration of the LTP texture descriptor.

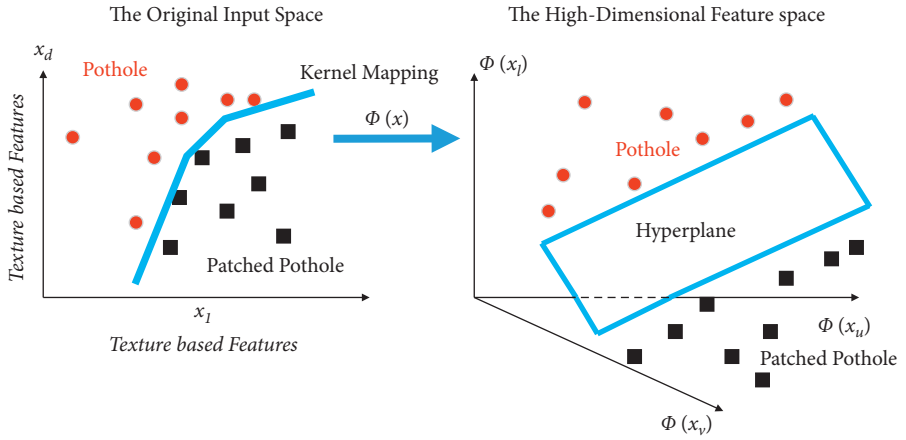


Figure 4: Demonstration of the SVC learning phase.

aforementioned texture descriptors. In such high-dimensional feature space, a hyperplane can be learnt to act as a decision boundary to separate the input data into two regions of "pothole" and "patched pothole". To identify this hyperplane, it is required to solve the following constrained optimization problem:

$$\text{Minimize } J_p(w,e) = \frac{1}{2}w^T w + C\frac{1}{2}\sum_{k=1}^{N} e_k^2, \tag{12}$$

subjected to $y_k(w^T\varphi(x_k)+b) \geq 1-e_k$, $k=1,\ldots,N$, $e_k \geq 0$. $w \in R^n$ and $b \in R$ denote the parameters of the hyperplane. e represents the vector of slack variables. C is the penalty coefficient. $\varphi(x)$ is the aforementioned nonlinear data mapping function.

A notable advantage of the SVC machine learning method is that it does not require the explicit form of the mapping function $\varphi(x)$. The quantity of interest is the dot product of $\varphi(x)$ denoting a kernel function $K(x_k, x_l)$. For nonlinear and multivariate data classification, the commonly used kernel function is the radial basis kernel function (RBKF) shown as follows:

$$K(x_k, x_l) = \exp\left(-\frac{\|x_k - x_l\|^2}{2\sigma^2}\right), \tag{13}$$

where σ is a tuning parameter of the kernel function.

2.3. Forensic-Based Investigation (FBI). As described in the previous section, the training phase of an SVC model necessitates a proper setting of the penalty coefficient C and the RBKF's tuning parameter σ. These are two crucial hyperparameters of SVC. The former expresses the amount of penalty imposing on misclassified data samples. The latter specifies the locality of the employed kernel function which affects the generalization of a trained SVC model. Since the task of hyperparameter setting can be formulated as an optimization problem, this study relies on the FBI metaheuristic to assist the SVC model's training phase.

FBI [50] is a stochastic search engine motivated by the forensic investigation process. The searching process of this stochastic search engine mimics the real operation of forensic investigation which contains five steps: open a case, interpretation of findings, the direction of inquiry, action, and prosecution [64]. The searching process of the FBI can be broken down into two phases: the investigation phase (called phase A) and the pursuit phase (called phase B). The former is carried out by a team of investigators. The latter is performed by a team of police agents. These two phases are repeated until a stopping condition (i.e., a maximum number of iterations) is met.

In phase A, based on the trace recorded at the current location X_{ij}, a new investigation location $X_{ij,\text{new}}$ is computed as follows:

$$X_{ij,\text{new}} = X_{ij} + r \times \left(X_{ij} - \frac{X_{ki} + X_{hj}}{2} \right), \tag{14}$$

where i, k, $h \in\in\{1, 2, \ldots, \text{NS}\}$ denote three investigation locations. NS is the total number of search agents. $j = 1,2,\ldots,$ D where D is the number of decision variables. r represents a random number $\in\in[-1, 1]$.

To determine where we should receive more investigation, the FBI relies on a probability featuring the quality of traces collected at a location. This probability value is computed via

$$P\left(X_{ij}\right) = \frac{\left(p_{\max} - p_{Xi}\right)}{\left(p_{\max} - p_{\min}\right)}, \tag{15}$$

where p_{X_i} denotes the objective function value of the location X_i. $p_{\max}$ and $p_{\min}$ are the largest and smallest value of the objective function in the current population, respectively.

Based on the computed probability associated with each available location, the updated suspected location is given by

$$X_{A2,i} = X_{\text{Best}} + \sum_{b=1}^{a2} \alpha_b \times X_{A,bj}, \tag{16}$$

$$X_{A2,ij} = X_{\text{Best}} + X_{A,dj} + rn \times \left(X_{A,ej} - X_{A,fi} \right), \tag{17}$$

where X_{Best} denotes the best-found location. $a2$ is the number of locations that influence $X_{A2,i}$. α_b denotes the effectiveness coefficient ranging from −1 to 1. rn is a random number generated uniformly within [0, 1]. d, e, f, and i are the suspected locations and the first three indices are chosen randomly.

In phase B, after receiving the report from the investigation team, each agent B_i moves close to the location that is associated with the highest possibility. This movement is presented as follows:

$$X_{B1,ij} = rn_0 X_{B,ij} + X_{A,dj} + rn_1 \times \left(X_{\text{Best}} - X_{B,ij} \right), \tag{18}$$

where rn_0 and rn_1 are random numbers generated uniformly within [0, 1].

Subsequently, each agent B_i exchanges information with other agents to enhance the searching reliability. Thus, the newly updated suspected location is given by

$$X_{B2,ij} = X_{B,rj} + rn_2 \times \left(X_{B,rj} - X_{B,ij} \right) + rn_3 \times \left(X_{\text{Best}} - X_{B,rj} \right), \tag{19}$$

$$X_{B2,ij} = X_{B,ij} + rn_4 \times \left(X_{B,ij} - X_{B,rj} \right) + rn_5 \times \left(X_{\text{Best}} - X_{B,ij} \right), \tag{20}$$

where rn_2, rn_3, rn_4, and rn_5 are random numbers generated uniformly within [0, 1].

2.4. The Collected Image Dataset. As mentioned earlier, the main objective of this study is to process digital image samples of asphalt pavements for detecting patched potholes and unpatched potholes. The employed pattern recognition method is SVC. Since SVC is a supervised machine learning method, a set of labeled image data must be prepared to train its model structure. This study has collected 600 asphalt pavement image samples within which the number of data samples in each class ("potholes" and "patched potholes") is 300. This selection of sample size is to guarantee a balanced classification. The label of each image sample has been assigned by human inspectors. The collected image samples are demonstrated in Figure 5.

The image dataset has been collected during field trips of the asphalt pavement survey in Da Nang city (Vietnam). The employed digital cameras are the 18-megapixel resolution Canon EOS M10 and the Gopro Hero 9 with 23.6-megapixel resolution. The images were manually taken by human inspectors. The camera is positioned at a distance of approximately 1 meter above the road surface. The image samples have obtained in cloudy weather conditions to ensure relatively consistent lighting conditions.

3. The Proposed Computer Vision-Based Forensic-Based Investigation Metaheuristic Optimized Machine Learning for Patched and Unpatched Pothole Detection

This section describes the overall structure of the proposed model used for automatic detection of the pothole and patched pothole. The model is an integration of image processing techniques, supervised machine learning-based data classification, and metaheuristic optimization. In detail, the image texture descriptors including statistical measurements of color channels, GLCM, and LTP are used to compute numerical features of image samples. Based on such extracted features, SVC incorporated with the FBI metaheuristic is used to generalize a decision boundary that separates the collected dataset into two distinctive classes of "patched pothole" and "pothole". The FBI is utilized to assist the SVC training phase by optimizing its tuning parameters (the penalty coefficient and the RBKF parameter). The overall structure of the proposed hybrid model is presented in Figure 6.

The operational flow of the model can be divided into three main steps:

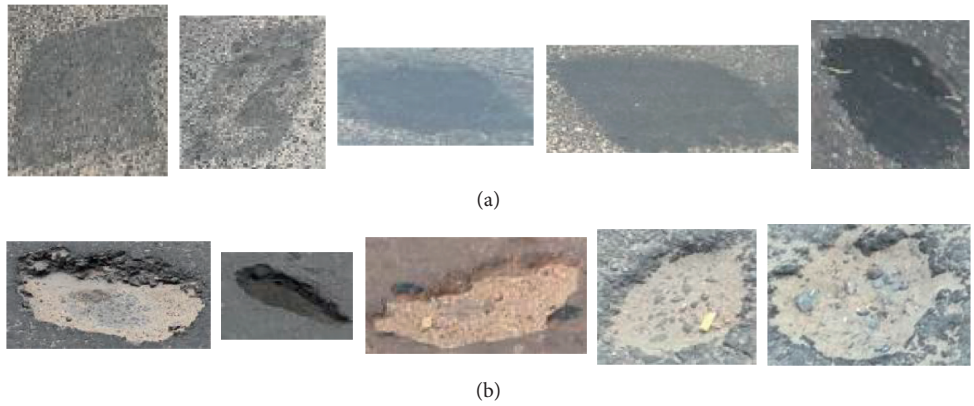
(i) Feature extraction

(ii) Model optimization

(iii) Model prediction and evaluation

The feature computation module has been programmed by the authors in Visual C# .NET. Additionally, the FBI optimized SVC module is constructed in a MATLAB environment with the built-in functions provided in the Statistics and Machine Learning Toolbox [65]. The source code of the FBI is provided by [50,66]. The final prediction model based on image texture computation and the optimized SVC model is constructed and compiled in Visual C# .NET with the assistance of the Accord.NET Framework [67].

3.1. Feature Extraction. In this step, texture descriptors including statistical measurements of color channels, GLCM, and LTP are employed to calculate features of the



(a)

(b)

Figure 5: Demonstration of the collected image samples: (a) Class label −1 (patched pothole) and (b) Class label +1 (unpatched pothole).

collected image samples. The statistical properties of image color, including three channels (R, G, and B), yield 18 features which are the mean, standard deviation, skewness, kurtosis, entropy, and range. To compute the GLCM-based texture, four GLCM with $r = 1$ and $\theta = 0^o$, 45^o, 90^o, and 135^o are computed as suggested by [57]. Since each matrix yields four properties of AM, CO, CR, and ET, the number of features generated by the GLCM texture descriptor is $4 \times 4 = 16$. In addition, the LBP feature descriptor produces 118 features that represent the local structure of grayscale image samples [61]. According to the suggestion of previous works, the threshold t of the LBP is set to be 5 [61, 68]. Accordingly, there are 152 texture-based features employed for categorizing patched potholes and potholes. Demonstrations of the feature extraction step are provided in Figure 7.

Subsequently, the extracted dataset has been randomly divided into a training set (70%) and a testing set (30%). The first set is employed for the model construction phase, and the second set is reserved for inspecting the model generalization capability. Moreover, to standardize the input data range, the Z-score equation is employed:

$$X_Z = \frac{X_D - M_X}{\mathrm{STD}_X}, \tag{21}$$

where X_Z and X_D are the normalized and the original texture features, respectively. M_X and STD_X are to the mean value and the standard deviation of the texture features, respectively.

3.2. Model Optimization. The FBI metaheuristic with the number of searching agents of 20 and the maximum number of searching iterations of 100 is used to optimize the SVM model. FBI aims at finding an appropriate set of the SVC hyperparameters including the penalty coefficient and the RBKF parameter. These hyperparameters strongly affect the learning and the predictive capability of the SVC model. The inappropriate setting of the penalty coefficient and the RBKF parameter may lead to either an overfitted or underfitted model. Hence, the objective of the FBI optimization process is to identify a set of hyperparameters that demonstrates a good training accuracy as well as generalization property. Moreover, the lower and the upper bounds of the parameters searched by the FBI metaheuristic are set to be 0.001 and 1000, respectively.

In order to optimize the SVC model, this study employs a 5-fold cross-validation-based objective function. This objective function is mathematically described as follows [69]:

$$\mathrm{OF} = \frac{\sum_{k=1}^{5} \left(\mathrm{FNR}_k + \mathrm{FPR}_k\right)}{5}, \tag{22}$$

where FNR_k and FPR_k are false negative rate (FNR) and false positive rate (FPR) calculated in the kth run, respectively.

The FNR and FPR indices are calculated according to the following equations [69]:

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}}, \tag{23}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{24}$$

where FN, FP, TP, and TN denote the false negative, false positive, true positive, and true negative data samples, respectively.

3.3. Model Prediction and Evaluation. In this step, the SVM model with the optimized hyperparameters (the penalty coefficient and the RBFK parameter) is constructed in Visual C# .NET framework 4.7.2 classify data instances in the testing set into two categories of "patched pothole" and "pothole. Moreover, it is noted that the developed computer program has been implemented on the ASUS FX705GE-EW165T (Core i7 8750H and 8 GB Ram) platform.

To evaluate the model prediction capability, a set of performance measurement metrics is used. The employed performance measurement metric consists of classification accuracy rate (CAR), precision, recall, negative predictive value (NPV), F1 score, and area under the receiver operating characteristic curve (AUC) [70, 71]. The construction of the receiver operating characteristic curve is provided in [71]. The indices of CAR, precision, recall, NPV, and F1 score are calculated according to the following formulas:

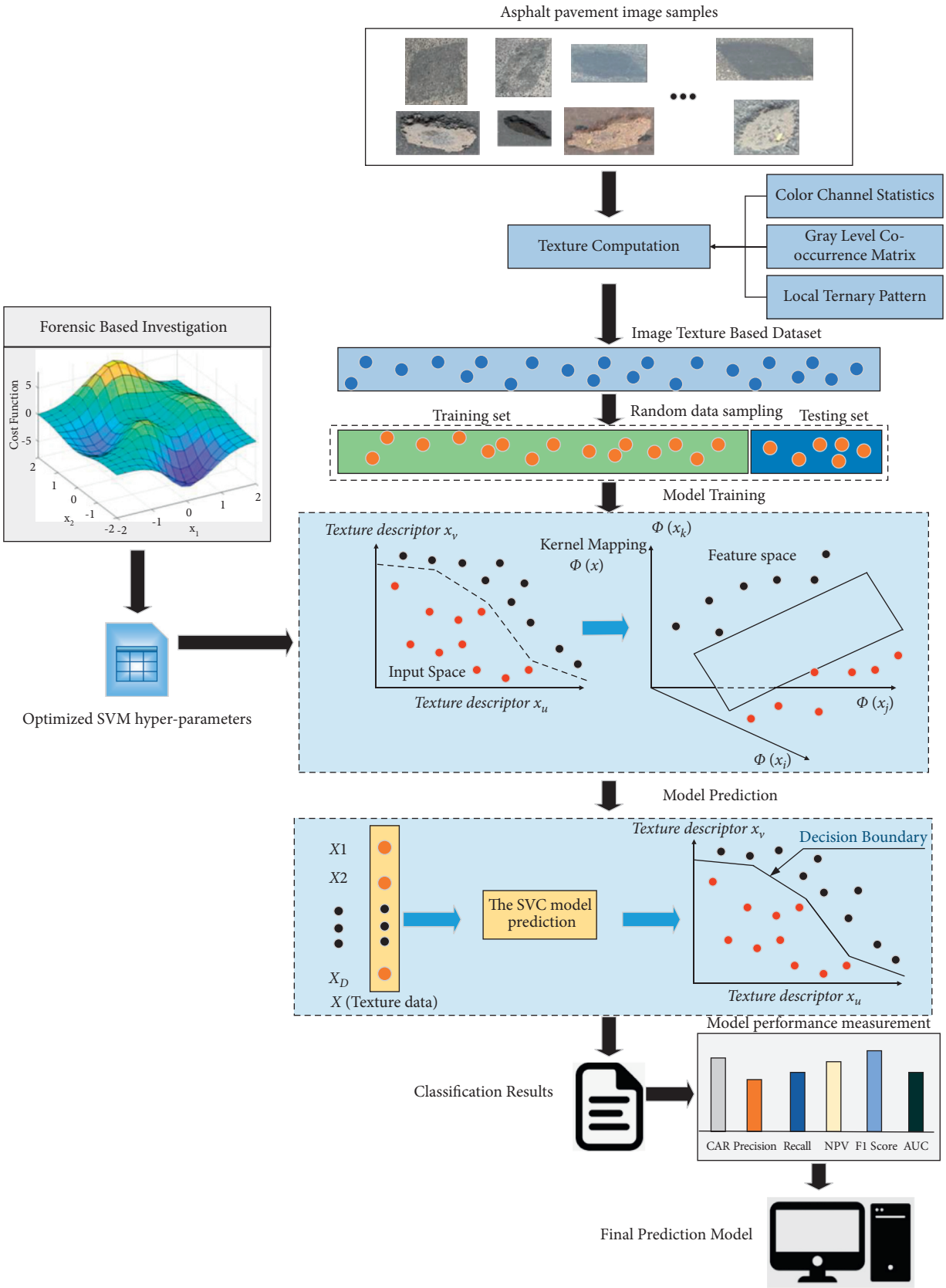


Figure 6: The proposed image processing-based Forensic-Based Investigation optimized machine learning for patched and unpatched pothole detection.

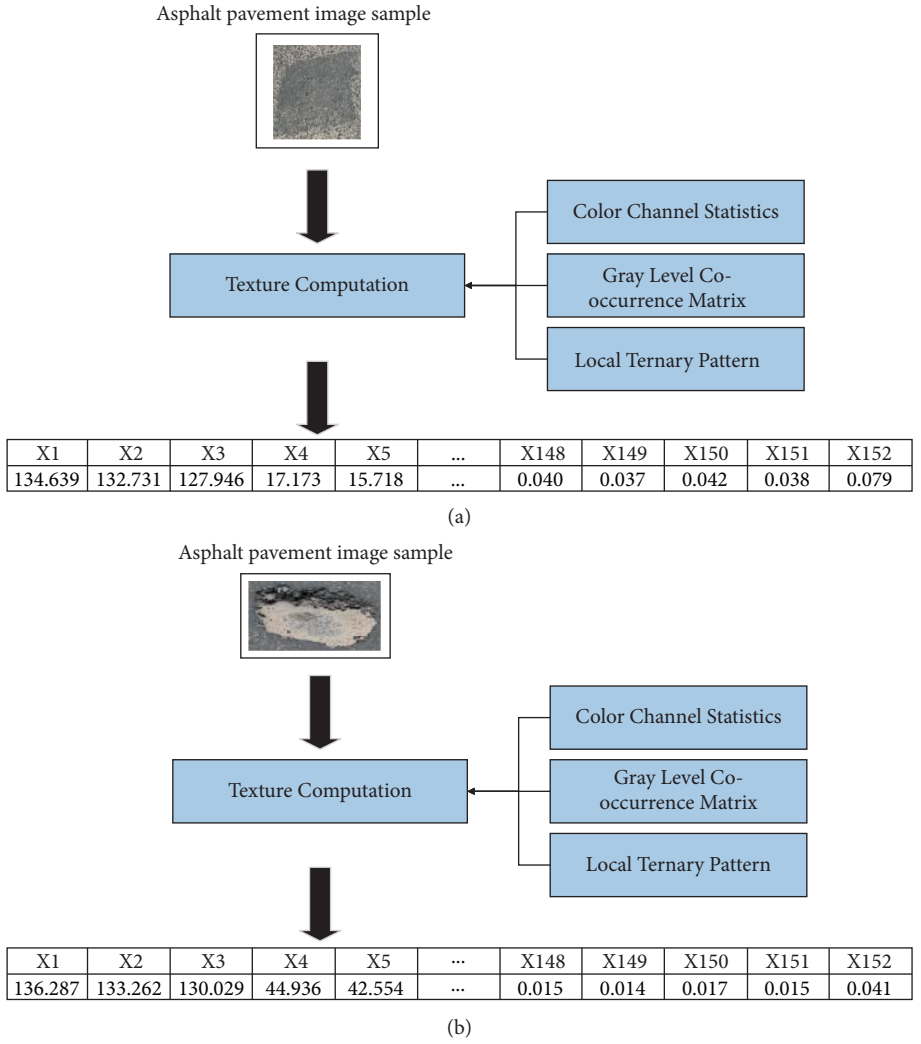


Figure 7: Demonstration of the feature extraction process: (a) a sample of the "patched pothole" class and (b) a sample of the "pothole" class.

$$\text{CAR} = \frac{N_C}{N_A} 100\%, \tag{25}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{26}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{27}$$

$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}}, \tag{28}$$

$$\text{F1 score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{29}$$

where N_C and N_A are the numbers of correctly predicted data and the total number of data, respectively. As mentioned earlier, FN, FP, TP, and TN are the false negative, false positive, true positive, and true negative data samples, respectively.

4. Experimental Results and Discussions

As aforementioned, the FBI is employed to optimize the SVM-based pothole-patched pothole detection model. Using 20 searching agents and after 100 iterations, the meta-heuristic algorithm has found the best penalty coefficient = 118.941 and the RBKF parameter = 20.300. The best-found cost function value which is described in equation (22) is 1.047. The FBI algorithm is capable of locating the most desired values of the searched variables after 41 iterations. The searching progress of the FIB meta-heuristic is illustrated in Figure 8. It is noted that, in Figure 8, the model performance is computed according to equation (22) which is stated in the previous section.

As stated earlier, the collected dataset which includes 600 instances has been randomly divided into a training set (70%) and a testing set (30%). The former is employed for model construction and the latter is served as novel data instances for verifying the model's predictive capability. Additionally, to reliably assess the predictive performance,

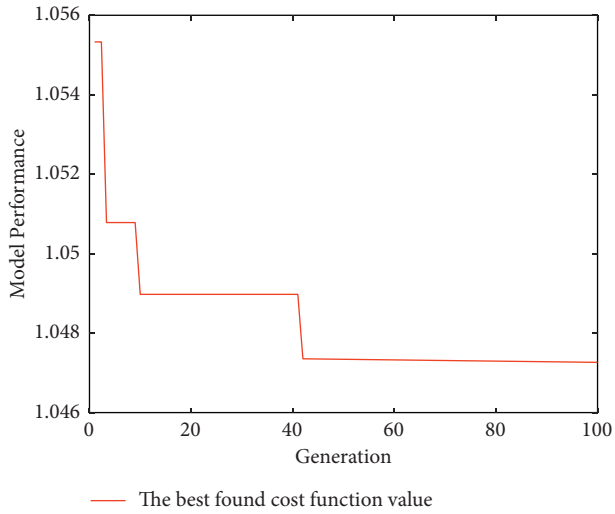


Figure 8: The FBI-based optimization progress.

Table 1: Experimental result comparison.

Metrics	FBI-SVC		RVM		RFM		CTM	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
CAR (%)	94.833	1.917	90.833	2.152	89.889	2.020	85.889	2.196
Precision	0.935	0.027	0.904	0.026	0.881	0.023	0.869	0.033
Recall	0.964	0.019	0.914	0.028	0.923	0.033	0.847	0.027
NPV	0.963	0.017	0.914	0.026	0.920	0.031	0.851	0.022
F1 score	0.949	0.019	0.909	0.022	0.901	0.021	0.857	0.021
AUC	0.989	0.006	0.962	0.015	0.962	0.010	0.881	0.037

this study has repeated the model training and verification processes 20 independent times. The statistical measurements attained from this multiple model training and verification phases are utilized for model evaluation. This process can be helpful in diminishing the variation caused by the randomness in data sampling.

In addition, to demonstrate the superiority of the newly constructed model, its performance is compared to those of other capable benchmark models including Relevance Vector Machine (RVM) [72, 73], Random Forest Model (RFM) [74], and Classification Tree Model (CTM) [75, 76]. RVM is constructed with the source code provided in [73]. The RFM and CTM approaches are trained with MATLAB's Statistics and Machine Learning Toolbox [65]. It is noted that grid search procedures with the fivefold cross-validation process [77] are employed to set the tuning parameters of the benchmark models. The RVM's parameter of basis width is selected to be 0.01. In addition, the number of individual decision trees used in the RFM is 50 and the parameter of minimum leaf size is 1.

The model prediction results are reported in Table 1 that shows the mean and standard deviation (Std) of each performance measurement index obtained from the testing process. As seen from this table, the proposed integration of FBI and SVC (denoted as FBI-SVC) has achieved the most desired predictive performance in terms of the CAR = 94.833%, the precision = 0.935, the recall = 0.964, the NPV = 0.963, and the F1 score = 0.949. The RVM is the second-best model with the CAR = 90.833%, the precision = 0.904, the recall = 0.914, the NPV = 0.914, and the F1 score = and 0.909, followed by the RFM (with the CAR = 89.889%, the precision = 0.881, the recall = 0.923, the NPV = 0.920, and the F1 score = 0.901) and CTM (with the CAR = 85.889%, the precision = 0.869, the recall = 0.847, the NPV = 0.851, and the F1 score = 0.857). The model comparison in terms of the CAR is illustrated in Figure 9. The results of the proposed model and the benchmark approach with respect to the precision, recall, NPV, and F1 score are provided in Figures 10 and 11 to ease the visual comparison.

Particularly for the AUC index, the FBI-SVC is the best model (AUC = 0.989), followed by RVM (AUC = 0.962), RFM (AUC = 0.962), and CTM (AUC = 0.881). The receiver operating characteristic curves of the proposed model and the benchmark model employed for patched pothole–pothole classification are demonstrated in Figure 12. A receiver operating characteristic curve (ROC) is a graphical plot widely used for diagnosing the predictive capability of classification models. ROC and AUC are crucial metrics because they include important predictive outcomes in terms of FPR and TPR in one plot. As can be observed from Figure 12, the AUC of FBI-SVC is very close to 1 meaning that it has achieved a highly accurate predictive result. Furthermore, the box plots of CAR, F1 score, and AUC are provided in Figures 13–15.

Particularly, Figure 14 displays the models' performance in terms of the F1-score. This index is an important and

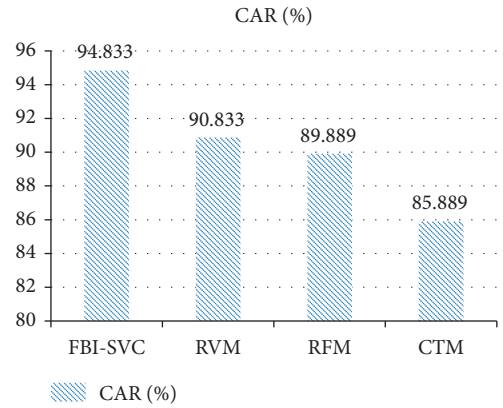


Figure 9: Performance comparison with respect to the CAR.

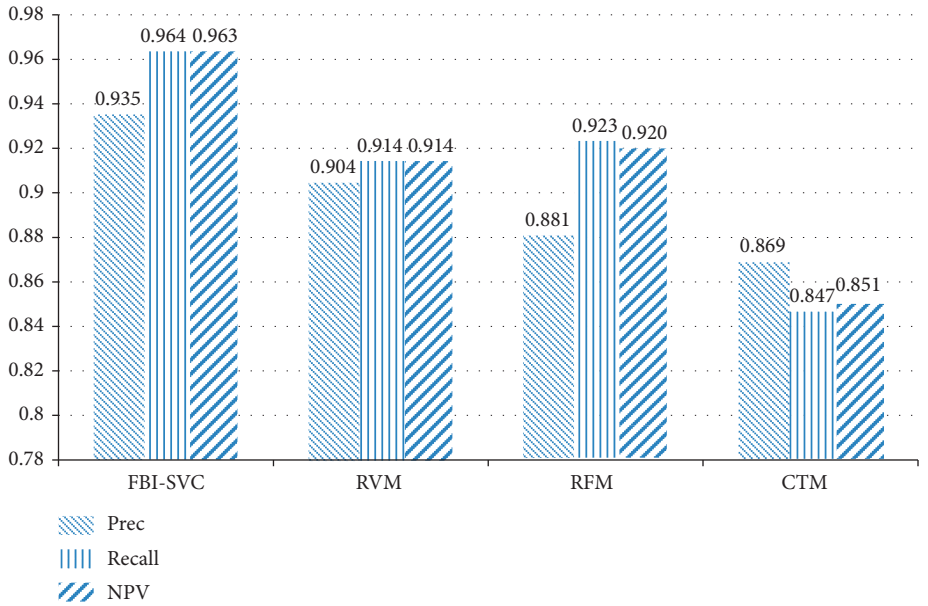


Figure 10: Performance comparison with respect to precision, recall, and NPV.

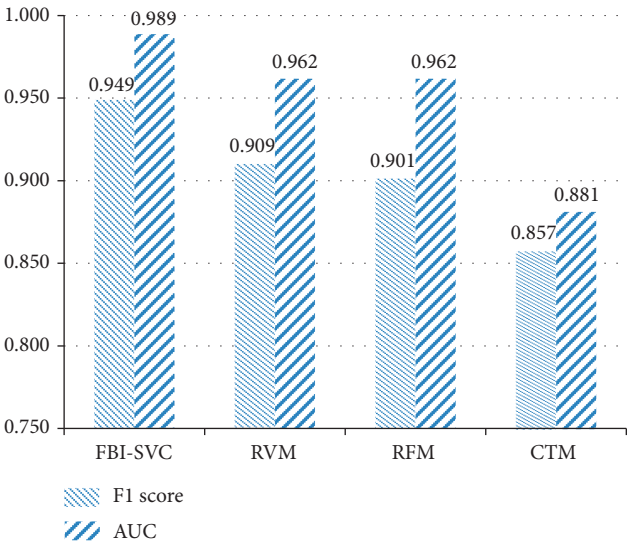


Figure 11: Performance comparison with respect to the F1 score and AUC.

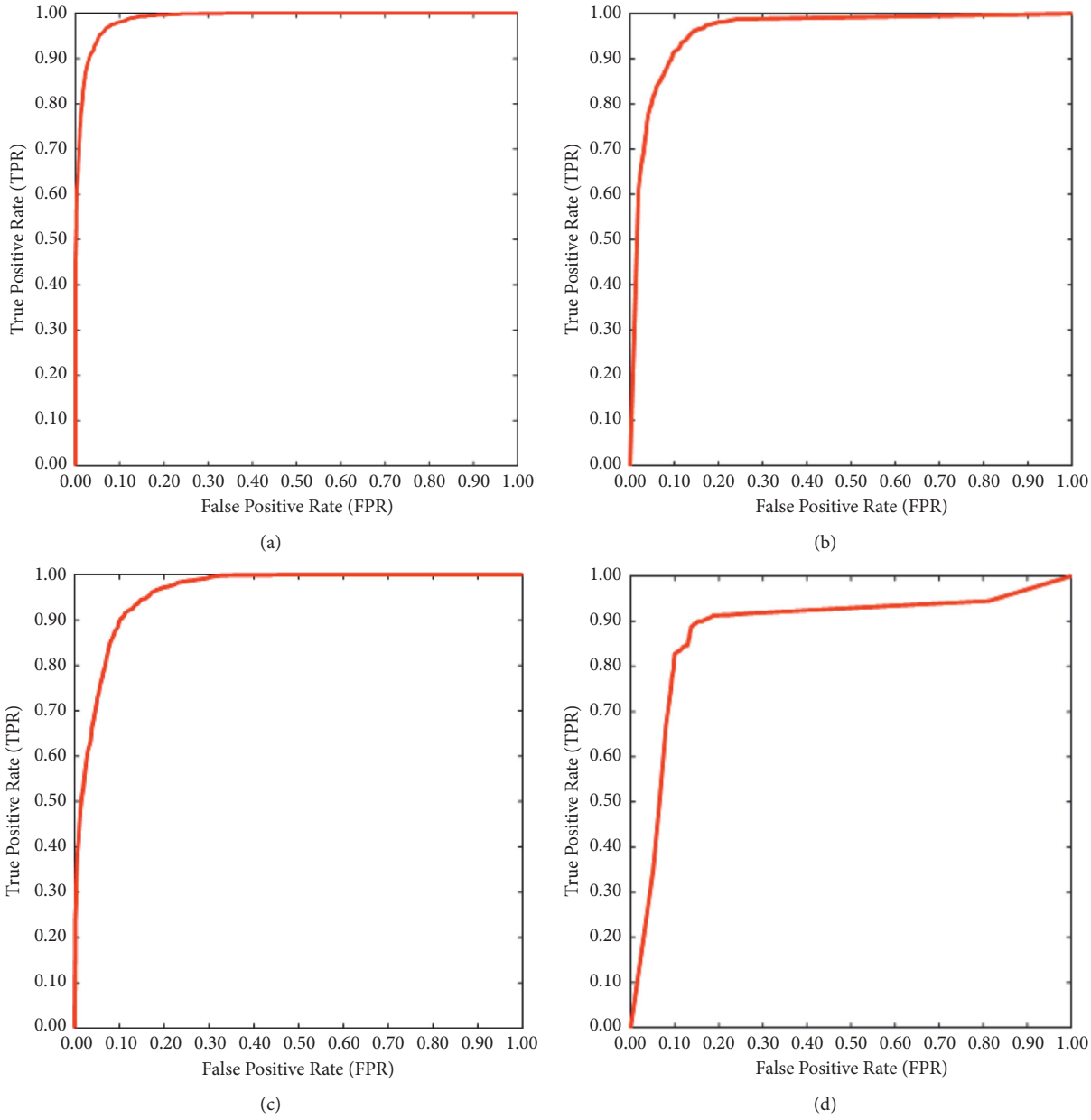


Figure 12: ROCs of the prediction models: (a) FBI-SVC, (b) RVM, (c) RFM, and (d) CTM.

representative measurement of predictive accuracy. The F1-score is a combination of the precision and recall indices. In other words, this performance measurement metric is the harmonic mean of precision and recall. The possible value of the F1-score ranges from 0 and 1; 1 indicates a perfect classification model. The higher the value of this index is, the better the predictive accuracy is. The value of the F1-score = 0.989 obtained from the FBI-SVC indicates a highly capable model used for patched/nonpatched pothole detection. Moreover, based on the boxplots shown in Figure 14, it can be observed that the proposed model's performance in terms of F1-score is significantly better than those of the benchmark methods.

Moreover, to confirm the superiority of the proposed FBI-SVC, this study has relied on the nonparametric Wilcoxon signed-rank test [78] with the significant level (p value) = 0.05. The crucial indices of CAR, F1 score, and AUC are the subjects of the Wilcoxon signed-rank test. The test outcomes of pairwise model comparison are reported in Tables 2–4, respectively. Evidently, with p values <0.05, it is able to reject the null hypotheses of equal model performances and confirm the superiority of the proposed FBI-SVC.

Although FBI-SVC has achieved the most desired predictive performance in distinguishing between patched potholes and potholes. This model has also committed several misclassifications as demonstrated in Figure 16. Inspecting the misclassified cases, it is revealed that a patched area surrounded by raveling and partly covered by traffic marks can lead to falsely predicted potholes (refer to Figures 16(a) and 16(b)). Moreover, the

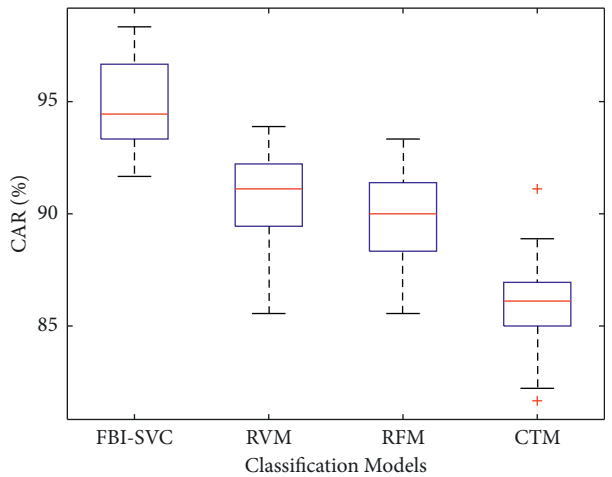


Figure 13: Box plot of CAR values obtained from the employed machine learning models.

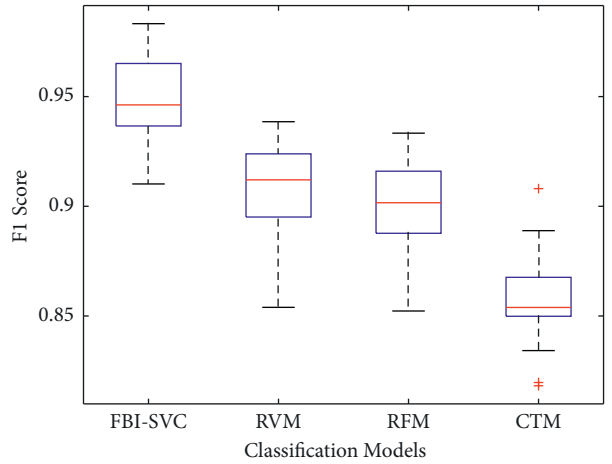


Figure 14: Box plot of F1 score values obtained from the employed machine learning models.

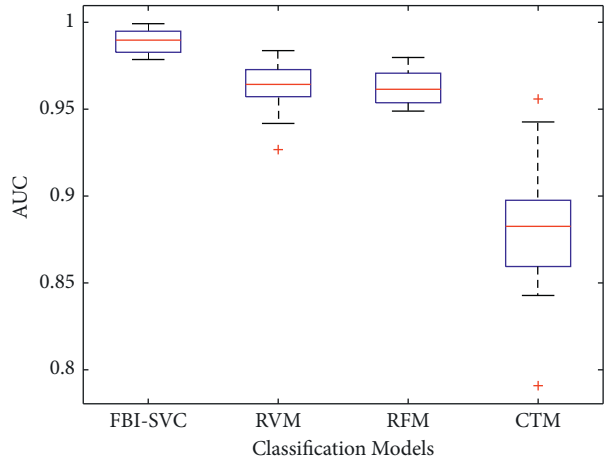


Figure 15: Box plot of AUC values obtained from the employed machine learning models.

Table 2: p values obtained from the Wilcoxon signed-rank test results with CAR index.

Models	FBI-SVC	RVM	RFM	CTM
FBI-SVC	X	0.00011	0.00009	0.00009
RVM	0.00011	X	0.21156	0.00014
RFM	0.00009	0.21156	X	0.00023
CTM	0.00009	0.00014	0.00023	X

Table 3: p values obtained from Wilcoxon signed-rank test results with F1 score index.

Models	FBI-SVC	RVM	RFM	CTM
FBI-SVC	X	0.00010	0.00009	0.00009
RVM	0.00010	X	0.29588	0.00014
RFM	0.00009	0.29588	X	0.00019
CTM	0.00009	0.00014	0.00019	X

Table 4: p values obtained from Wilcoxon signed-rank test results with AUC index.

Models	FBI-SVC	RVM	RFM	CTM
FBI-SVC	X	0.00009	0.00010	0.00009
RVM	0.00009	X	0.76520	0.00009
RFM	0.00010	0.76520	X	0.00013
CTM	0.00009	0.00009	0.00013	X

Image Samples	Actual Class	Predicted Class
 (a)	Patched Pothole	Pothole
 (b)	Patched Pothole	Pothole
 (c)	Pothole	Patched Pothole
 (d)	Pothole	Patched Pothole

Figure 16: Illustrations of misclassified cases.

cases of irregular lighting conditions and a pothole cover by a layer of dirt have caused false predictions of patched potholes as shown in Figures 16(c) and 16(d). This indicates that more efforts should be needed to deal with such irregular scenarios and improve the model prediction accuracy.

5. Concluding Remarks

Asphalt pavements play a very important role in the national and local transportation networks. Timely information regarding the distresses appearing on pavement surface is crucial for pavement management agencies to make a cost-effective decision on maintenance method and schedule. To obtain accurate and up-to-date information on the serviceability of asphalt pavement road sections, periodic surveys need to be performed regularly and the visual information attained by such surveys needs to be processed in a timely manner. To expedite this visual information processing, this study proposes an intelligent model for automatically identifying patched potholes and unpatched potholes. Potholes are widely encountered pavement distress and may cause serious damages to vehicles as well as human casualties. However, based on 2D digital images, potholes and patched potholes can have similar shapes. Therefore, this study has proposed employing image texture information to delineate these two objects.

With such research motivations, this study has relied on image texture descriptors of statistical information of color channels, GLCM, and LTP. These three texture descriptors are employed to extract useful information regarding the coarseness/fineness of image regions containing patched and unpatched potholes. An image set consisting of 600 asphalt pavement image samples has been collected and used to

construct a texture-based dataset. Within this dataset, the number of data samples in each category is 300. Using the established texture-based dataset, a novel machine learning approach that hybridizes the SVC pattern classifier and the FBI metaheuristic has been proposed to construct a decision boundary that can deliver prediction of the class labels.

The experimental outcomes supported by the Wilcoxon signed-rank test show that the FBI-SVC model is superior to those of the benchmark approaches. Good classification results with CAR = 94.833%, precision = 0.935, recall- = 0.964, NPV = 0.963, F1 score = 0.949, and AUC = 0.989 show that the proposed FBI-SVC is well suited for the task of patched-unpatched pothole classification. Further extensions of the current research work may include the following:

(i) The application of the FBI-SVC to detect other forms of pavement distresses (e.g., alligator crack, raveling, blurred traffic marks, etc.)

(ii) The investigation of other advanced texture descriptors for characterizing local structure or pattern of image regions containing pavement distresses

(iii) Collecting more image samples to enhance the generalization and applicability of the current computer vision method

Data Availability

The dataset used to support the findings of this study has been deposited in the repository of GitHub at https://github.com/ndhoangdtu/patch_unpatchpothole_fbi_svc. The first 152 columns of the data are texture-based features extracted from image samples. The last column is the ground-truth label of the data instances with −1 = "patched potholes" and 1 = "unpatched potholes."

Conflicts of Interest

The authors confirm that there are no conflicts of interest regarding the publication.

Acknowledgments

This research was funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 107.01-2019.332.

References

[1] T. Dettenborn, A. Hartikainen, and L. Korkiala-Tanttu, "Pavement maintenance threshold detection and network-level rutting prediction model based on Finnish road data," *Journal of Infrastructure Systems*, vol. 26, no. 2, Article ID 04020016, 2020.

[2] G. M. Hadjidemetriou, J. Masino, S. E. Christodoulou, F. Gauterin, and I. Brilakis, "Comprehensive decision support system for managing asphalt pavements," *Journal of Transportation Engineering, Part B: Pavements*, vol. 146, no. 3, Article ID 06020001, 2020.

[3] J. Li, G. Liu, T. Yang, J. Zhou, and Y. Zhao, "Research on relationships among different distress types of asphalt pavements with semi-rigid bases in China using association rule mining: a statistical point of view," *Advances in Civil Engineering*, vol. 2019, Article ID 5369532, 15 pages, 2019a.

[4] W. Li, R. Deng, Y. Zhang, Z. Sun, X. Hao, and J. Huyan, "Three-dimensional asphalt pavement crack detection based on fruit fly optimisation density peak clustering," *mathematical problems in engineering*, vol. 2019, no. 5, 15 pages, Article ID 4302805, 2019b.

[5] M. H. Yousaf, K. Azhar, F. Murtaza, and F. Hussain, "Visual analysis of asphalt pavement for detection and localization of potholes," *Advanced Engineering Informatics*, vol. 38, pp. 527–537, 2018.

[6] WHO, *Road Traffic Injuries*, World Health Organization, Geneva, Switzerland, 2020, https://wwweuroint.

[7] VNN, *Traffic Accidents Claim 6,700 Lives this Year*, Viet Nam News Society, Hanoi, Vietnam, 2020, https://vietnamnewsvn/society/835944/traffic-accidents-claim-6700-lives-this-yearhtml.

[8] J. Zhou, P. Huang, and F.-P. Chiang, "Wavelet-based pavement distress detection and evaluation," *Optical Engineering*, vol. 45, no. 2, Article ID 027007, 2006.

[9] F. Moghadas Nejad and H. Zakeri, "A comparison of multi-resolution methods for detection and isolation of pavement distress," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2857–2872, 2011.

[10] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics*, vol. 25, no. 3, pp. 507–515, 2011.

[11] G. M. Jog, C. Koch, M. Golparvar-Fard, and I. Brilakis, "Pothole properties measurement through visual 2D recognition and 3D reconstruction," in *Computing in Civil Engineering*American Society of Civil Engineers, Reston, VA, USA, 2012.

[12] E. Buza, S. Omanovic, and A. Huseinovic, "Pothole detection with image processing and spectral clustering," in *Proceedings of the 2nd International Conference on Information Technology and Computer Networks, 2013*, Antalya, Turkey, 2013.

[13] S. Bharadwaj Sundra Murthy and G. Varaprasad, "Detection of potholes in autonomous vehicle," *IET Intelligent Transport Systems*, vol. 8, no. 6, pp. 543–549, 2014.

[14] S.-K. Ryu, T. Kim, and Y.-R. Kim, "Image-based pothole detection system for ITS service and road management system," *Mathematical Problems in Engineering*, vol. 2015, Article ID 9683361, 10 pages, 2015.

[15] S. C. Radopoulou, I. Brilakis, K. Doycheva, and C. Koch, "A framework for automated pavement condition monitoring," *Construction Research Congress 2016*, vol. 2016, 2016.

[16] M. Kamaliardakani, L. Sun, and M. K. Ardakani, "Sealed-crack detection algorithm using heuristic thresholding approach," *Journal of Computing in Civil Engineering*, vol. 30, no. 1, Article ID 04014110, 2016.

[17] Y. O. Ouma and M. Hahn, "Pothole detection on asphalt pavements from 2D-colour pothole images using fuzzy c-means clustering and morphological reconstruction," *Automation in Construction*, vol. 83, pp. 196–211, 2017.

[18] N.-D. Hoang, "An artificial intelligence method for asphalt pavement pothole detection using least squares support vector machine and neural network with steerable filter-based feature extraction," *Advances in Civil Engineering*, vol. 2018, Article ID 7419058, 12 pages, 2018.

[19] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil*

and Infrastructure Engineering, vol. 33, no. 12, pp. 1127–1141, 2018.

[20] M.-T. Cao, Q.-V. Tran, N.-M. Nguyen, and K.-T. Chang, "Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources," *Advanced Engineering Informatics*, vol. 46, Article ID 101182, 2020a.

[21] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14531–14544, 2020b.

[22] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.

[23] B. F. Spencer Jr., V. Hoskere, and Y. Narazaki, "Advances in computer vision-based civil infrastructure inspection and monitoring," *Engineering*, vol. 5, no. 2, pp. 199–222, 2019.

[24] G. M. Hadjidemetriou, P. A. Vela, and S. E. Christodoulou, "Automated pavement patch detection and quantification using support vector machines," *Journal of Computing in Civil Engineering*, vol. 32, no. 1, Article ID 04017073, 2018.

[25] N.-D. Hoang, "Image processing based automatic recognition of asphalt pavement patch using a metaheuristic optimized machine learning approach," *Advanced Engineering Informatics*, vol. 40, pp. 110–120, 2019b.

[26] S. C. Radopoulou and I. Brilakis, "Patch detection for pavement assessment," *Automation in Construction*, vol. 53, pp. 95–104, 2015.

[27] L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods - a review," arXiv:190406554 [csCV], 2019.

[28] I. El khadiri, A. Chahi, Y. El merabet, Y. Ruichek, and R. Touahni, "Local directional ternary pattern: a new texture descriptor for texture classification," *Computer Vision and Image Understanding*, vol. 169, pp. 14–27, 2018.

[29] Y. El merabet, Y. Ruichek, and A. El idrissi, "Attractive-and-repulsive center-symmetric local binary patterns for texture classification," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 158–172, 2019.

[30] S. Fekri-Ershad, "Cell phenotype classification using multi threshold uniform local ternary patterns in fluorescence microscope images," *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 12103–12116, 2021.

[31] S. Fekri-Ershad and F. Tajeripour, "Multi-resolution and noise-resistant surface defect detection approach using new version of local binary patterns," *Applied Artificial Intelligence*, vol. 31, no. 5-6, pp. 395–410, 2017.

[32] N.-D. Hoang, "Image processing-based spall object detection using gabor filter, texture analysis, and adaptive moment estimation (adam) optimized logistic regression models," *Advances in Civil Engineering*, vol. 2020, Article ID 8829715, 16 pages, 2020.

[33] N.-D. Hoang and V.-D. Tran, "Image processing-based detection of pipe corrosion using texture analysis and metaheuristic-optimized machine learning approach," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8097213, 13 pages, 2019.

[34] A. Humeau-Heurtier, "Texture feature extraction methods: a survey," *IEEE Access*, vol. 7, pp. 8975–9000, 2019.

[35] L. Ji, Y. Ren, X. Pu, and G. Liu, "Median local ternary patterns optimized with rotation-invariant uniform-three mapping for noisy texture classification," *Pattern Recognition*, vol. 79, no. 6, pp. 387–401, 2018.

[36] A. Ledoux, O. Losson, and L. Macaire, "Color local binary patterns: compact descriptors for texture classification," *Journal of Electronic Imaging*, vol. 25, no. 6, p. 12, Article ID 061404, 2016.

[37] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "From BoW to CNN: two decades of texture representation for texture classification," *International Journal of Computer Vision*, vol. 127, no. 1, pp. 74–109, 2019.

[38] M. M. P. Petrou and S.-I. Kamata, *Image Processing Dealing with Texture*, Wiley, Hoboken, NJ, USA, 2021.

[39] S. Ranjbar, F. M. Nejad, and H. Zakeri, "An image-based system for asphalt pavement bleeding inspection," *International Journal of Pavement Engineering*, pp. 1–17, 2021.

[40] F. Tajeripour and S. Fekri-Ershad, "Developing a novel approach for stone porosity computing using modified local binary patterns and single scale retinex," *Arabian Journal for Science and Engineering*, vol. 39, no. 2, pp. 875–889, 2014.

[41] J.-H. Yuan, H.-D. Zhu, Y. Gan, and L. Shang, "Enhanced local ternary pattern for texture classification," in *Intelligent Computing Theory*, D.-S. Huang, V. Bevilacqua, and P. Premaratne, Eds., Springer International Publishing, Cham, Switzerland, pp. 443–448, 2014.

[42] X. Zhao, L. Xue, and F. Xu, "Asphalt pavement paving segregation detection method using more efficiency and quality texture features extract algorithm," *Construction and Building Materials*, vol. 277, Article ID 122302, 2021.

[43] M. E. A. Ben Seghier, H. Ouaer, M. A. Ghriga, N. A. Menad, and D.-K. Thai, "Hybrid soft computational approaches for modeling the maximum ultimate bond strength between the corroded steel reinforcement and surrounding concrete," *Neural Computing and Applications*, vol. 33, no. 12, pp. 6905–6920, 2021.

[44] J.-S. Chou and N.-M. Nguyen, "Metaheuristics-optimized ensemble system for predicting mechanical strength of reinforced concrete materials," *Structural Control and Health Monitoring*, vol. 28, no. 5, Article ID e2706, 2021.

[45] X. Li, Y. Li, Y. Zhang, F. Liu, and Y. Fang, "Fault diagnosis of belt conveyor based on support vector machine and grey wolf optimization," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, Article ID 1367078, 2020.

[46] Z. M. Yaseen, H. Faris, N. Al-Ansari, and J. Vega, "Hybridized extreme learning machine model with salp swarm algorithm: a novel predictive model for hydrological application," *Complexity*, vol. 2020, Article ID 8206245, 14 pages, 2020.

[47] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 1998.

[48] G. M. Hadjidemetriou, S. E. Christodoulou, and P. A. Vela, "Automated detection of pavement patches utilizing support vector machine classification," in *Proceedings of the 18th Mediterranean Electrotechnical Conf (MELECON)*, pp. 1–5, IEEE, Lemesos, Cyprus, April 2016.

[49] N.-D. Hoang, Q.-L. Nguyen, and D. Tien Bui, "Image processing-based classification of asphalt pavement cracks using support vector machine optimized by artificial bee colony," *Journal of Computing in Civil Engineering*, vol. 32, no. 5, Article ID 04018037, 2018.

[50] J.-S. Chou and N.-M. Nguyen, "FBI inspired meta-optimization," *Applied Soft Computing*, vol. 93, Article ID 106339, 2020.

[51] N.-D. Hoang, "Automatic detection of asphalt pavement raveling using image texture based feature extraction and stochastic gradient descent logistic regression," *Automation in Construction*, vol. 105, Article ID 102843, 2019a.

[52] Z. Zhang, F. Gao, B. Ma, and Z. Zhang, "Extraction of earth surface texture features from multispectral remote sensing data," *Journal of Electrical and Computer Engineering*, vol. 2018, Article ID 9684629, 9 pages, 2018.
[53] B. Abraham and M. S. Nair, "Computer-aided classification of prostate cancer grade groups from MRI images using texture features and stacked sparse autoencoder," *Computerized Medical Imaging and Graphics*, vol. 69, pp. 60–68, 2018.
[54] F. F. Feliciano, F. R. Leta, and F. B. Mainier, "Texture digital analysis for corrosion monitoring," *Corrosion Science*, vol. 93, pp. 138–147, 2015.
[55] S. Mathavan, A. Kumar, K. Kamal, M. Nieminen, H. Shah, and M. Rahman, "Fast segmentation of industrial quality pavement images using laws texture energy measures and k-means clustering," *Journal of Electronic Imaging*, vol. 25, no. 5, p. 11, 2016.
[56] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*Academic Press, USA, 2009.
[57] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.
[58] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
[59] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*Gatesmark Publishing, USA, 2009.
[60] F. Tomita and S. Tsuji, *Computer Analysis of Visual Textures*, Springer Science+ Business Media, New York, NY, USA, 1990.
[61] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.
[62] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.
[63] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics*, Springer, NY, USA, 2011.
[64] R. Salet, "Framing in criminal investigation," *The Police Journal: Theory, Practice and Principles*, vol. 90, no. 2, pp. 128–142, 2017.
[65] MathWorks, *Statistics and Machine Learning Toolbox User's Guide*, MathWork Inc., Natick, MA, USA, 2017, https://www.mathworks.com/help/pdf_doc/stats/stats.pdf.
[66] N.-M. Nguyen, "Forensic-based investigation algorithm (FBI) MATLAB central," 2020, https://wwwmathworkscom/matlabcentral/fileexchange/76299-forensic-based-investigation-algorithm-fbi.
[67] Accord, "Accord. NET framework," 2019, http://accord-frameworknet/.
[68] M.-T. Cao, N.-M. Nguyen, K.-T. Chang, X.-L. Tran, and N.-D. Hoang, "Automatic recognition of concrete spall using image processing and metaheuristic optimized LogitBoost classification tree," *Advances in Engineering Software*, vol. 159, Article ID 103031, 2021.
[69] N.-D. Hoang and Q.-L. Nguyen, "A novel approach for automatic detection of concrete surface voids using image texture analysis and history-based adaptive differential evolution optimized support vector machine," *Advances in Civil Engineering*, vol. 2020, Article ID 4190682, 15 pages, 2020.
[70] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.
[71] A. R. Van Erkel and P. M. T. Pattynama, "Receiver operating characteristic (ROC) analysis: basic principles and applications in radiology," *European Journal of Radiology*, vol. 27, no. 2, pp. 88–94, 1998.
[72] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
[73] M. E. Tipping, "Sparse Bayes software release for MATLAB," 2009, http://wwwmiketippingcom/downloadshtm.
[74] H. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
[75] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth and Brooks, Montery, CA, USA, 1984.
[76] L. Rokach and O. Z. Maimon, "Data mining with decision trees: theory and applications," *Series in Machine Perception and Artificial Intelligence*, World Scientific Publishing Company, Singapore, 2008.
[77] N.-D. Hoang and D. T. Bui, "Predicting earthquake-induced soil liquefaction based on a hybridization of kernel Fisher discriminant analysis and a least squares support vector machine: a multi-dataset study," *Bulletin of Engineering Geology and the Environment*, vol. 77, no. 1, pp. 191–204, 2018.
[78] M. Hollander and D. A. Wolfe, *Nonparametric Statistical Methods*, John Wiley & Sons, Hoboken, NJ, USA, 1999.

Hindawi
Complexity
Volume 2021, Article ID 8500572, 13 pages
https://doi.org/10.1155/2021/8500572

Research Article

Presentation of a Novel Method for Prediction of Traffic with Climate Condition Based on Ensemble Learning of Neural Architecture Search (NAS) and Linear Regression

Javad Artin,[1] Amin Valizadeh,[2] Mohsen Ahmadi,[3] Sathish A. P. Kumar,[4] and Abbas Sharifi[5]

[1]*Department of Computer Engineering and Information Technology, Payame Noor University, Tehran, Iran*
[2]*Department of Mechanical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*
[3]*Department of Industrial Engineering, Urmia University of Technology, P.O. Box: 57166-419, Urmia, Iran*
[4]*Department of Electrical Engineering and Computer Science, Cleveland State University, Cleveland, OH 44115, USA*
[5]*Department of Mechanical Engineering, Urmia University of Technology, Urmia, Iran*

Correspondence should be addressed to Mohsen Ahmadi; mohsen.ahmadi@ine.uut.ac.ir

Received 30 April 2021; Revised 17 July 2021; Accepted 17 August 2021; Published 31 August 2021

Academic Editor: Teddy Craciunescu

Traffic prediction is critical to expanding a smart city and country because it improves urban planning and traffic management. This prediction is very challenging due to the multifactorial and random nature of traffic. This study presented a method based on ensemble learning to predict urban traffic congestion based on weather criteria. We used the NAS algorithm, which in the output based on heuristic methods creates an optimal model concerning input data. We had 400 data, which included the characteristics of the day's weather, including six features: absolute humidity, dew point, visibility, wind speed, cloud height, and temperature, which in the final column is the urban traffic congestion target. We have analyzed linear regression with the results obtained in the project; this method was more efficient than other regression models. This method had an error of 0.00002 in terms of MSE criteria and SVR, random forest, and MLP methods, which have error values of 0.01033, 0.00003, and 0.0011, respectively. According to the MAE criterion, this method has a value of 0.0039. The other methods have obtained values of 0.0850, 0.0045, and 0.027, respectively, which show that our proposed model has a minor error than other methods and has been able to outpace the other models.

1. Introduction

Predicting and reducing traffic congestion is a critical priority for all cities in the world. Recently, this issue has received much attention from research projects to improve forecasting methods [1]. Traffic forecasting is critical to expanding a smart city and country because it improves urban planning and traffic management [2]. Traffic forecasting is an interdisciplinary research field that includes mathematics, computer science, and engineering. This prediction is very challenging due to the multifactorial and random nature of traffic. This complexity is due to the limitations that exist under the physical infrastructure of traffic, including road network capacity, traffic regulations, demeanor, days of the week, weather, and accidents [3]. Studies in the field of urban traffic in the past were based on the location of each person. It was a metropolitan area. Today, knowledge of traffic data has dramatically expanded the use of predictive model development. Many studies have examined driver behavior, travel mode, road conditions, and the importance of weather data. For example, in a study [4], rainfall intensity was associated with a 4–9% reduction in traffic. It is also found that traffic congestion has a significant relationship with temperature intensity. However, studies in

this area ignore important environmental data sources and thus lose the accurate evaluation of the traffic network [5]. The effects of weather conditions on traffic are undeniable. However, little research has been done to quantify the impact of rain and snow. Besides, existing studies have not distinguished between urban and rural freeways. Therefore, it can be said that the inclusion of nontraffic input data sets has a significant effect on predicting the ground traffic parameter. Such studies have provided accurate predictions [6, 7].

In recent years, methods based on deep learning have received much attention. Deep learning (DL) has advanced speech, natural language processing (NLP). The deep learning approach is efficient in the field of short-term traffic forecasting [7, 8]. Interestingly, the use of social media data in DL approaches helps predict traffic. Social media such as Twitter and Instagram are increasingly used for communication, news reports, and promotional events. These media are forced to compensate for high-speed data and timely dissemination. For example, in Twitter data, due to the high level of users, they have been used for various data mining purposes such as stock exchange [9] and traffic forecasting [10, 11]. Climate variables are an influential factor in road traffic. Climate variables (x variables and y variables are traffic compensation variables) include absolute humidity, dew point, visibility, wind speed, cloud height, and temperature. These variables lead to reduced visibility, friction, and road blockage, thus increasing accidents. 22% of road accidents have been due to weather conditions, especially rainfall, in the last ten years. Hence, the importance of these variables is known [12]. It should be borne in mind that rain is only one of the parameters of difficult weather conditions. We will review the studies traffic forecasting by considering the weather conditions using deep learning.

In this paper, we presented a method based on ensemble learning. Based on this approach, we were able to predict urban traffic congestion based on weather criteria. It is based on a deep learning method. We used the neural architecture search algorithm, which, based on heuristic methods, creates an optimal model concerning input data. In addition to the deep learning model, we used a linear regression model. We combined the two models based on the weighted average, and the prediction output was based on the ensemble model. The results are illustrated in the following section.

2. Literature Review

An LSTM model was investigated in the study [13]. It has studied two parameters of temperature and rainfall in addition to traffic characteristics. The results of this study show a more accurate prediction. Despite all the advances in studies in this field, even the most accurate prediction models do not have robust predictions in unusual situations that occur specifically, such as accidents, crashes, and sporting events [7]. Recently, some studies have been beneficial in predicting these events on social media data. For example, the study [14] presents a linear regression-based method for predicting road traffic. This model uses the California Performance Measurement System (PeMS) in the United States. A similar model [15] has been used that uses social media data to predict short-term traffic. This model uses Twitter data to predict incoming traffic compensation before the start of sports games. This method has been evaluated using four models: ARIMA, neural network, support vector regression, and k-nearest neighbor (k-NN). Short-term traffic forecasting techniques are mainly classified into two parts: parametric and nonparametric approaches.

Parametric models are models that recognize input data as a function and summarize the data. For example, the autoregressive integrated moving average (ARIMA) model is one of the parametric prediction models [16]. However, in the nonparametric model, we can say that the algorithms are trained on the data. That is, they select a function that fits the data set. The k-nearest neighbor (k-NN) method is one of the simple nonparametric methods [15]. Many studies have been studied as a method of traffic forecasting [17, 18]. Regarding forecasting based on virtual social page data, a study [18] shows that the inclusion of Twitter data in addition to rainfall and weather data reduced the MAE from 8 to 5.5 and created a more accurate forecasting model.

In general, many techniques and models are presented in time series predictions, such as ARIMA [1], SVM [3], and deep nonlinear algorithms [19]. Neural networks such as RNNs [5], LSTMs [13], Kernel Extreme Learning [20], and CNN's [13] are used for financial [9, 21] and traffic forecasts [14, 15]. Mainly, three main methods of deep learning have been studied in previous studies: convolutional neural networks (CNNs) [2], deep belief networks (DBNs) [22], and SAEs [23], but recent research has led to algorithms. The new one is anticipated and provides an opportunity for further investigation. In recent years, studies have shown that rainy or stormy weather significantly impacts traffic on drivers' behavior, travel demand, safety, and so on [24, 25]. Mashros et al. [26] have found that the driving speed decreases by about 14% when it rains. Another study [27] with regression analysis investigated that topical rain reduces 1.1 mph in operating speed, and storm conditions cause 4 to 8 mph. Another report shows that adverse weather conditions can affect traffic. Many other studies show that rainfall intensity hurts traffic speed [28, 29]. Identifying the impact of weather on-peak and off-peak traffic times provides policymakers with invaluable information. A summary of the results of studies that have measured weather conditions and rainfall is given in Table 1. Almost all studies have stated that light rain has little effect on traffic, i.e., the parameter of rain intensity is essential in examining traffic forecasts. Temperature also has little impact on traffic [30]. Research by Smith et al. [31] has shown the importance of rainfall intensity values at moderate speeds. Like most other articles, they categorized the rain as none, light, and heavy. They used the Scheffé method for statistical comparisons. Still, a study by Neter [32] explained that this method might be unsuitable for climatic conditions because of using an equation of equal variance.

Table 1: Studies that have worked on weather conditions and traffic forecasting.

Authors	Road type	Speed limit (kilometers per hour)	Intensity	Speed (%)	
				Rainfall	Temperature
Kyte [33]	Freeway	100	Light	−13.6	—
			Heavy	−13.6	
Smith [31]	Freeway	100	Light	−6.5	—
			Heavy	−6.5	
Agarwal [5]	Freeway	112	Light/warm	−2.5	−1.5
			Heavy/cold	−7	−3.6
Billiot [34]	Motorway	128	Light	−8	—
			Heavy	−12.5	
Tsapakis [35]	Urban	112	Light	−2.1	−1.9
			Medium	−6	−3.2
Essien [4]	Urban	48	Light/cold	−4.9	−4.12
			Medium/normal	−5.5	−18.7
			Heavy/warm	−9.7	−28.2

Deep learning models today have been able to demonstrate their ability to predict traffic. New traffic forecasting models based on deep learning and a combination of existing methods have been proposed in recent years. For example, the two-way LSTM-based model in study [36] and the deep LSTM in study [37] are designed to predict traffic. Sequence-to-sequence models are also used to predict traffic sequences [38, 39]. Some other studies also described multistream deep learning models [40, 41] as more accurate prediction methods because these models use auxiliary data such as accidents [37] and weather conditions [42]. Some prediction models [43] use spatial feature extraction to obtain spatial relationships in traffic networks, but the images become noisy. Recent studies [44, 45] attempt to use a three-dimensional convolutional network to extract data. Lin et al. [46] proposed geographical and temporal criteria to extract features from traffic data, followed by the random forest method to order the relevance of factors. Additionally, generative adversarial networks are used to produce some fresh event examples. Five types of tests are used to see if the suggested framework can handle the incident detection system's limited sample size concern, unbalanced sample problem, and timeliness issue. A stacked autoencoder is also used to identify temporal and geographical correlations of traffic flow and detect events in other research. Similarly, the sample selection approach increased the detection's real-time capability [47]. The studies that used deep learning are summarized in Table 2. Also, there are some optimization methods based on metaheuristic algorithms such as Harris hawk's optimization [48], multiswarm whale [49], moth-flame optimizer [50–52], grey wolf [53, 54], fruit fly [55, 56], bacterial foraging optimization [57], boosted binary Harris hawk's optimizer [58], ant colony [59, 60], biogeography-based whale optimization [61], and grasshopper optimizer [62]. Moreover, optimization methods use machine learning in biological studies [63–66].

3. Research Methodology

3.1. AutoML. Auto-machine learning or AutoML is an automated way to find the best data preparation, models, and hyperparameters for a predictive modeling problem. One of the advantages of AutoML is working with small amounts of data and preparing the model for better accuracy. Also, the purpose of this method is for people with machine learning knowledge to build optimal models.

3.2. Neural Architecture Search. NAS is one of the AutoML methods, in which we try to choose the architecture of the neural network in a way that is appropriate to our data and increase the accuracy of the work. In these methods, work is grouped based on the following:

(i) Search space that includes the types of neural networks we can build

(ii) Search strategy includes achieving the goal, defining the objective function, and the method of searching

(iii) Performance estimation strategy that describes the method of estimating the performance of neural network models after being defined and obtained

This method has been used in different fields of different problems. Wang et al., for example, used the NAS method in an article to optimize the deep U-Net network for segmenting medical images and other models in this field. Figure 1 shows the NAS recursive controller. In this controller, each layer counts the number of filters, filter height, filter width, stride width, and stride height backward; each prediction is done by a SoftMax and is also given as input to the next step.

This method has not been used much in the detection of urban traffic and regression data. Therefore, in this work, we used one of the NAS methods for our deep learning model and optimized this model based on "our available data" to give higher accuracy compared with other deep learning models.

3.3. Ensemble Learning. According to articles and experimental evidence, some linear regression methods are more accurate in regression problems than deep learning methods. Also, in a small amount of data, these methods can easily fit their graph on the problem data, and their

Table 2: Studies used deep learning for forecasting.

Author	Methods	Outcome
Zhang [67]	STGI-residual	Short-term traffic flow forecasting is the most important
Lu [68]	GAM with diffusion convolution	Suggest a novel spatial-temporal deep learning network for traffic forecasting
Zhang [69]	RGCLSTM model	The RGCLSTM network is a more accurate model during rush hour
Cui [36]	TGC-LSTM	The proposed model of this research has better interpretability
Diao [70]	Dynamic graph convolution neural network (DGCNN)	The proposed model of this research has 10 to 25% higher accuracy

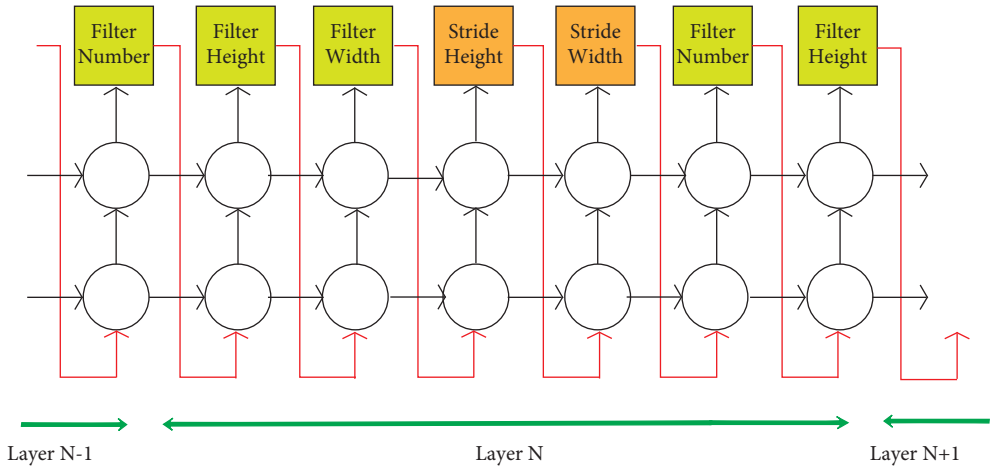


Figure 1: NAS method controller.

problems, in this case, are minor. Therefore, we used the ensemble learning method in this issue and tried to use two experts to predict traffic congestion. In ensemble learning methods, several expert learning algorithms are used to increase prediction accuracy compared with the prediction by an expert. Figure 2 shows a diagram of this model of methods.

In classification, the voting method is one of the standard techniques in this subject. However, in regression methods, the average (or the weighted average) approach can be used.

3.4. The Suggested Method. We first seek to find the best architecture for the deep learning method to fit our data in this method. First, we put the NAS module initially, and the NAS algorithm selects the best architecture. In the next step, we train this architecture with train data and build our deep learning model. This model is given in Figure 3 according to our work training. In the model analysis, we concluded that a complex model was not required due to the low amount of data. The NAS algorithm is given as the best model in the output given in the dense two-layer model. Also, ReLUx activator functions are given for each layer, and a normalization layer is provided to increase accuracy and prevent overfitting.

3.5. Linear Regression Method as the Second Expert. Since linear regression methods have high accuracy in regression problems, the second expert in our ensemble learning module was the linear regression method. This method gave the training data to a linear regression model for training and used this model for testing. In our method of testing and predicting data, the general form of work is similar to Figure 4.

In the ensemble module section, we used the weighted average of the two sections, deep learning and linear regression, and tested different numbers as weights between zero and one, which reached the amount of 0.7 and 0.3 (Algorithm 1).

4. Results and Simulations

4.1. Data Collection. Estimating urban traffic is one of the most critical issues in helping municipalities control traffic in large cities. For this reason, this issue has been studied in many articles, and attempts have been made to solve this issue by machine learning methods. In this issue, several data from the urban traffic flow based on six variables are available. These six variables are as follows: absolute humidity, dew point, visibility, wind speed, cloud height, and temperature. Also, in the last column, the urban traffic congestion per day with the conditions of the values of the above variables is given. This work has presented a method based on deep learning and regression that estimates the urban traffic congestion with six relevant variables. In this paper, we designed a deep learning model for the urban traffic control system. The model predicts traffic congestion,

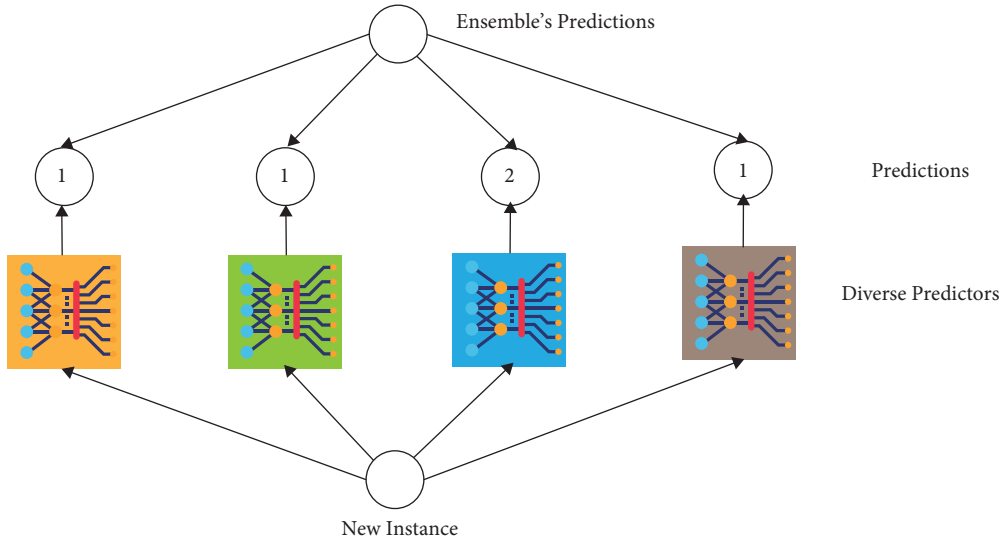


Figure 2: Ensemble learning method.

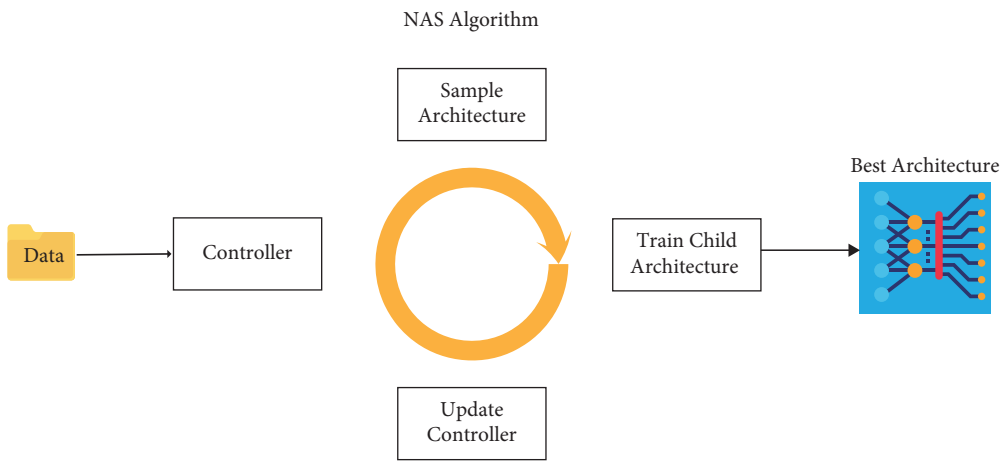


Figure 3: The proposed NAS architecture method.

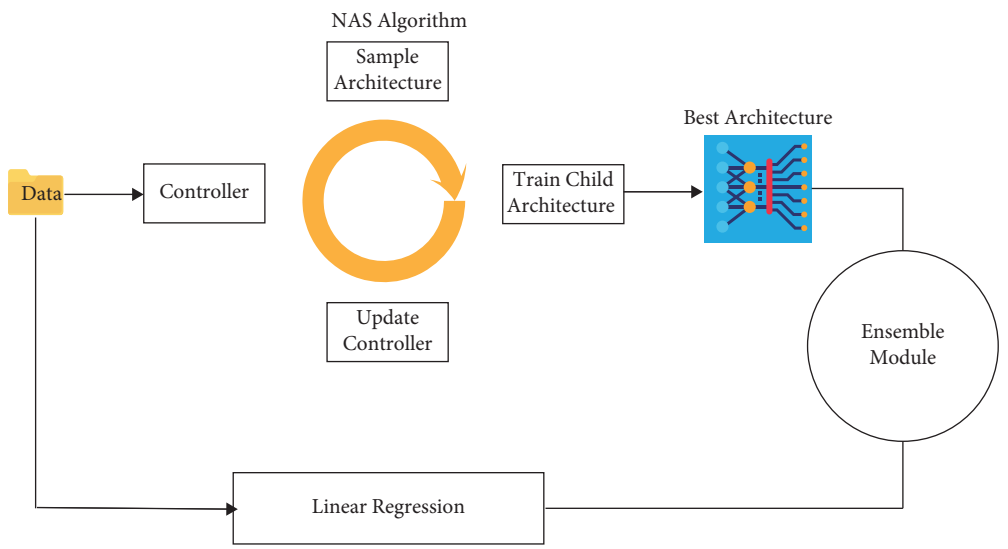


Figure 4: The proposed process for the ensemble learning method.

Input: Labeled Data
Main Scope:
Split_train_test ()
NAS_algorithm ()
While (models_count = = 30):
Best_model = NAS_algorithm ()
Train_LinearRegression ()
Make Ensemble Module (deep architecture + linear regression)
predict (test_set)
Output: Prediction for Test Data

Algorithm 1: Pseudocode of the presented method.

and it can prevent heavy traffic. In this work, we used Python programming to build models and final estimates. Python libraries have also been used to build deep networks and regression models, including *Keras*, *NumPy*, *sklearn*, and *scipy*.

4.2. Descriptive Statistics of Data. Traffic flow is recorded by sensors every thirty seconds, and these data are collected in 60 minutes. Our database has been collected over thirty days with different types of variables. The shape of the pattern expresses this traffic flow based on the variables and their changes. These variables include available meteorological information, including absolute humidity, dew point, visibility, wind speed, cloud height, and temperature. The table of detailed statistics of traffic flow, including the average and standard deviation of variables and data and the variable of traffic flow forecast, is given. The amount of traffic flow per vehicle number is expressed in 10 seconds (see Table 3).

4.3. The Implementation of the Proposed Method. In this paper, it was assumed that the climatic characteristics are closely related to the traffic flow. Therefore, we presented a model based on neural networks and regression, which we explained in detail in the previous section. This section shows the architecture obtained from the method and compares our method with the famous regression method. Figure 5 shows the diagram of meteorological variables.

We used Python 3.6 to implement this project. We used Google Colaboratory to use the appropriate GPU and hardware and wrote our code in the Jupyter Notebook. Out of 400 available data due to lack of data for deep learning, we used only 50 data for testing. Moreover, in the next step, we implemented the NAS method. In this method, following our panel data, 30 different models are based on the available search space such as number of layers, type of activation function, number of neurons, and presence or absence of normalization layer, and each model is taught 30 epochs. In this training, out of a total of 350 training data, 20% of the data have been used as validation data to evaluate the models' performance. This model results from 0.0003 in terms of mean absolute error, which is optimal compared with other methods. Figure 6 is an overview of the architecture selected by the NAS method for our train data.

Table 3: Standard regression coefficients and standard regression rank coefficients for the cases.

	Temp.	Cld. H.	Wnd. S.	Vis.	Dew P.	Abs. Hum.	Trf. F
Mean	1.032	0.204	7.170	23.184	4.662	1.026	3.019
STD	0.539	0.111	3.701	8.074	1.630	0.546	1.142

After obtaining the optimal architecture, we trained this architecture with 250 epochs on all training data and improved the model based on training and testing data (see Figure 7). We also plotted the corresponding loss and MSE, which can be seen in the figure. The training process is such that the model reduces the error in the MSE criterion with the progress of the epochs, which shows the correct training process in deep learning.

Also, our training line is parallel to the axis of the horizon, which indicates the optimal fit of the model, which may be overfitting if the training process continues and the epochs increase. Finally, the loss line for train and test data is matched, indicating the model's actual performance on both categories (see Figure 8).

After assembling the deep learning model with regression (explained in the previous section), we prepared the general model to process the test data. We gave the test data to this model for evaluation. The amount of data loss in our model is almost zero. This trend can be seen in Figure 9, where all the projected labels are close to the mainline on which the data fit, indicating the model's superiority. In this figure, the blue line is the fitted line on the predicted data, close to the 45-degree angle, which shows that our targets are predicted with high accuracy. The results are also compared with the traffic flow trend in the figure. According to Figure 10, the closer we are to the red line, the higher the model's accuracy. Our model is almost on the red line. This figure illustrates the concept that our model predicts almost all test data with equal accuracy equal to the data label and that the trend of this line is the same as that of the actual label line.

Also, the comparison results with the traffic flow trend for educational data are shown in Figure 11. The closer we get to the red line, the more accurate the model is. Our model is almost entirely on the red line for training data. This figure

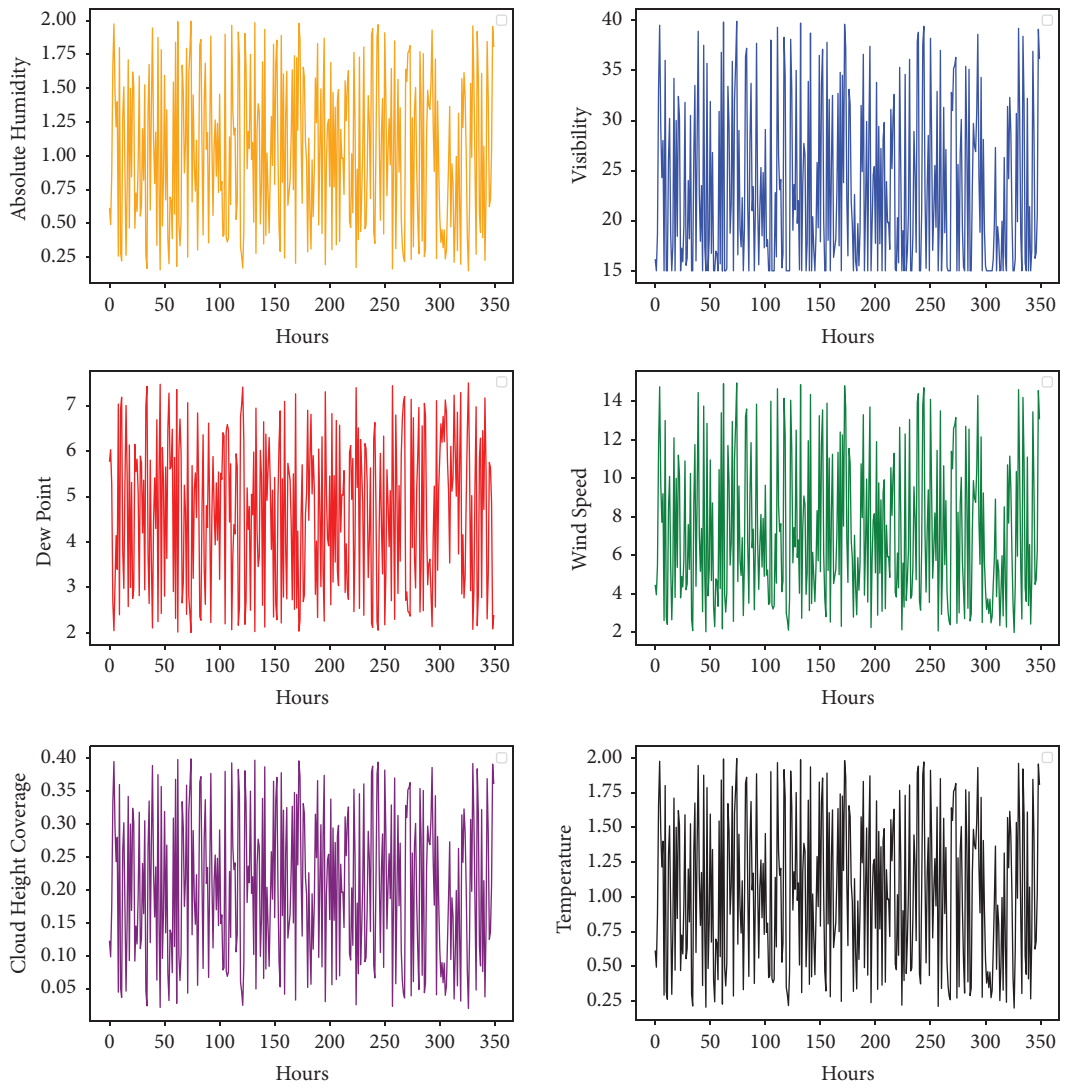


Figure 5: Descriptive diagram of meteorological variables.

shows the concept that our model predicts almost all training data with equal accuracy equal to the data label, and the trend of this line is the same as the trend of actual labels. To compare and evaluate our method, we evaluated the support vector regression method. We trained this model with the same train data with which we trained the NAS model.

Furthermore, we validated this model on the same test data that we used for our method. The rate of data loss in this model was higher than in our model. In the figure, it can be seen that the items predicted by the model on the desired line do not fit well. In this figure, there are red circles that are 45° away from our line, which should be the fitted line for the test data. The distance from this line to these circles can also be directly related to the absolute error relation, the average of which was one of the evaluation points of this project. Figure 11 shows the diagram for the train data, a model estimate of the training data. In this method, the red line does not fit well on the blue line, which is the baseline that should be predicted.

It indicates that the SVR model has not been well trained in our training data. The results are also compared with the traffic flow trend for the SVR model. Due to this shape, the closer we are to the red line, the higher the model's accuracy. This model did not have a reasonable estimate in more points than our model and was not well placed on the red line. In this figure, in some parts of the model, the data values are predicted to be more or less than the original value, so the test data trend line is not well aligned with our prediction trend line. Figure 12 shows the processing diagram for training data, which does not match the blue lines to the orange lines, which shows the weakness of the SVR model in regression training on training data.

Also, Table 4 compares the amount of MSE obtained with our proposed method with the SVR method. Our method can outpace the SVR method in terms of this criterion. The MSE criterion is calculated by the following equation:

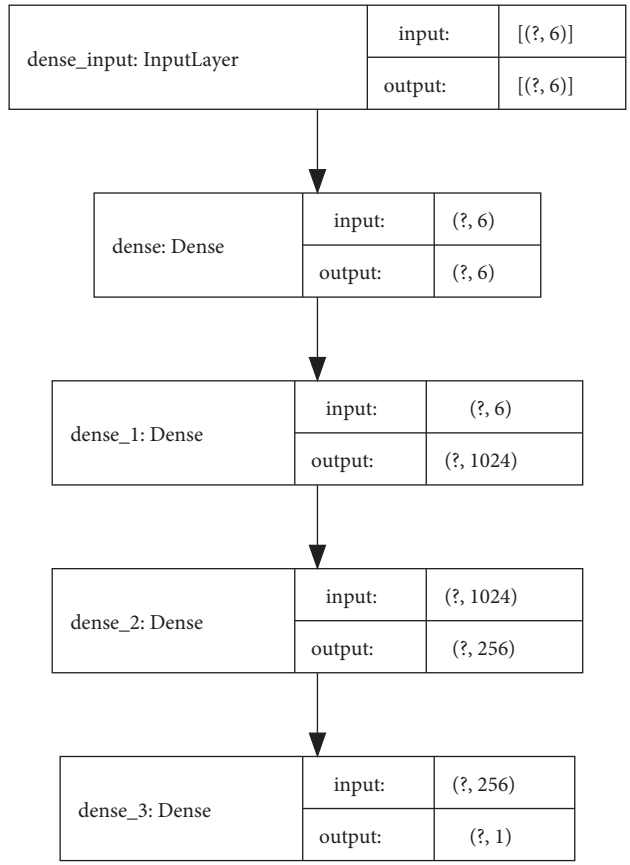


Figure 6: NAS method architecture for training data.

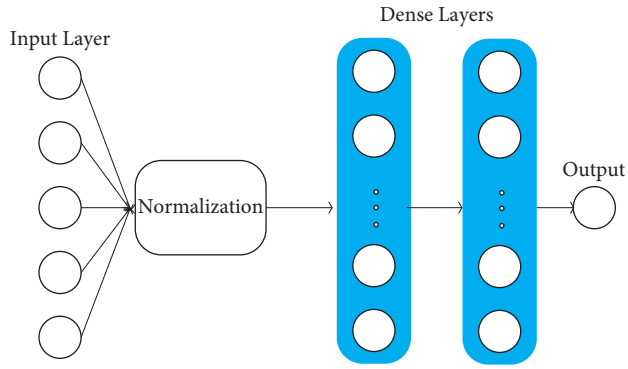


Figure 7: Model architecture extracted from the NAS method.

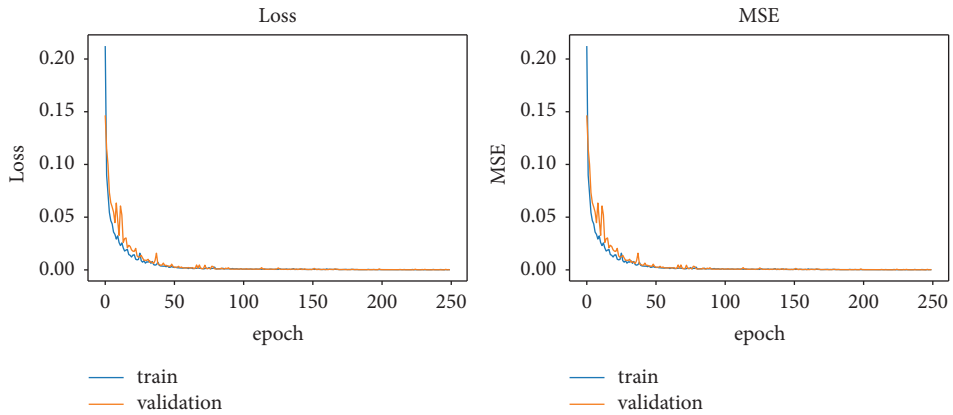


Figure 8: Criteria for errors in the deep learning method.

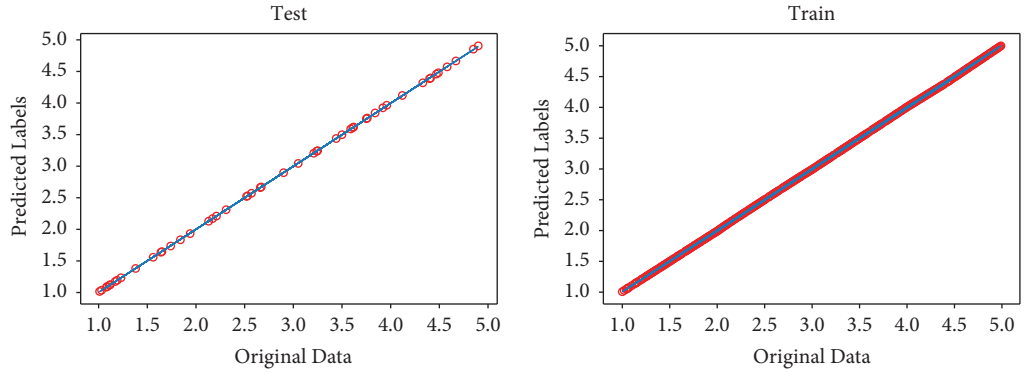


Figure 9: Dispersion of test data (a) and training (b) in the amount predicted in the proposed method.

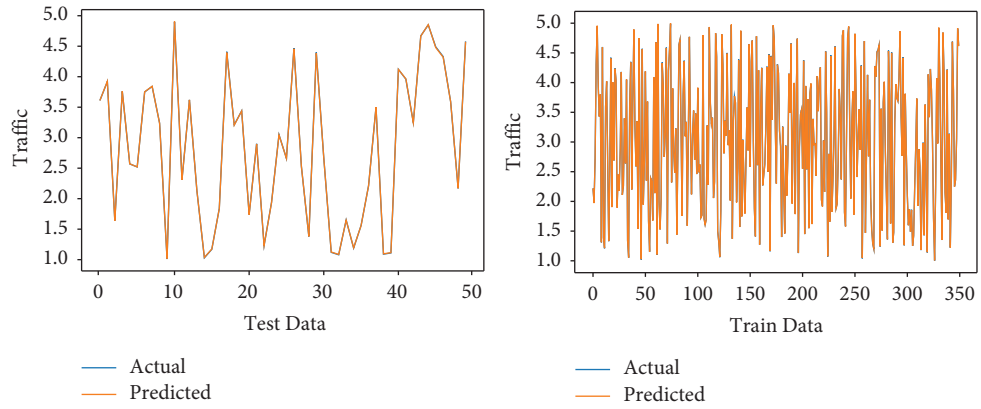


Figure 10: Processing for test data (a) and training (b) in the proposed method.

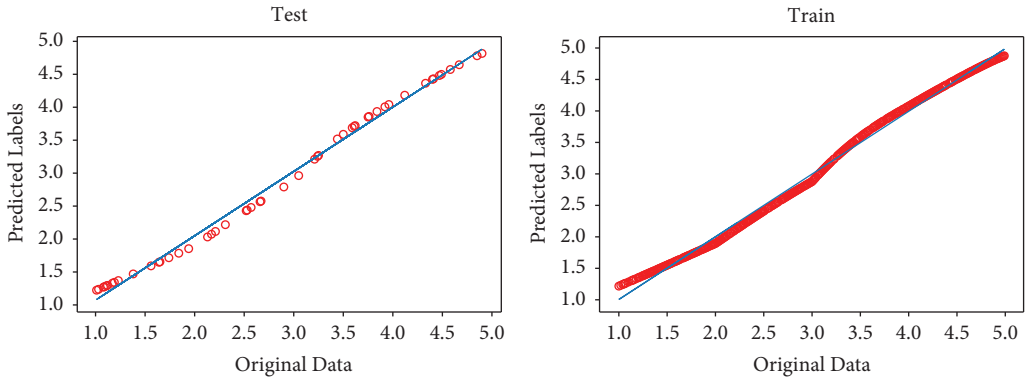


Figure 11: Dispersion of test data (a) and training (b) in the predicted SVR value.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2. \quad (1)$$

In this regard, the expressions in parentheses are the predicted values and the central target values for the test data. Our model has reached 0.00002 in terms of MSE criteria, which is an absolute advantage over the SVR method, which in terms of MSE has a value of 0.01033. Also, random forest regression and MLP methods have obtained values of 0.00003 and 0.0011 for MSE, respectively, which are overcome by this criterion concerning our method.

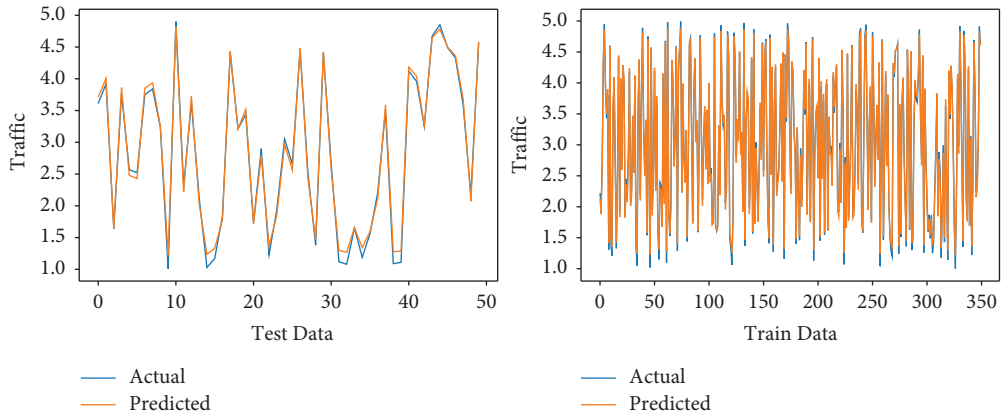


Figure 12: Processing for test data in the SVR method.

Table 4: Quantitative comparison of methods.

Method	MSE	MAE	RMSE
Proposed method	0.00002	0.0039	0.0044
Support vector regression	0.01033	0.0850	0.1016
Random forest regression	0.00003	0.0045	0.0054
Multilayer perceptron	0.0011	0.027	0.00331

5. Conclusion

This study presented a method based on ensemble learning to predict urban traffic congestion based on weather criteria. It is based on a deep learning method in which we used the NAS algorithm, which in the output based on heuristic methods creates an optimal model concerning input data. We had 400 data, which included the characteristics of the day's weather, including six features: absolute humidity, dew point, visibility, wind speed, cloud height, and temperature, which in the final column is the urban traffic congestion as the target. We trained the training data with the deep learning model, which was obtained through the NAS method. These data contain 87.5% of our data. In the end, the loss for train and test data in this model overlapped, which indicates that the network is completely fitted on the data. In addition to the deep learning model, we used a linear regression model and combined the two models based on the weighted average, and the prediction output was based on the ensemble model. The linear regression model was also trained with train data. Our analysis was that due to the problems of neural networks for regression problems, combining the deep learning model with a neural network model increases our accuracy and problem and reduces the MSE error. One of the best and simplest regression models was the linear regression model used for this work and the ensemble learning method.

Moreover, we analyzed them with the results obtained in the project; this method was more efficient than other regression models. This method had an error of 0.00002 in terms of MSE criteria and SVR, random forest, and MLP methods, which have error values of 0.01033, 0.00003, and 0.0011, respectively. According to the MAE criterion, this method has a value of 0.0039. The other methods have obtained values of 0.0850, 0.0045, and 0.027, respectively, which show that our proposed model has more minor errors than other methods that can outpace the other models. This issue is defeated with these data. Future studies suggest that the NAS method can be modified, which can also be optimized based on the convolutional method and can also depict data like a photo and use the images for work. Also, the combination of deep learning methods and machine learning is a suitable method that can be used by several other experts in the work and make predictions. Collecting more data in this field can be one of the future works of this project because the data in this project was small, and with more data, deep learning models work better.

Data Availability

The data are extracted from reference [70] for traffic flow prediction in the paper.

Disclosure

The funding sources had no involvement in the study design, collection, analysis, or interpretation of data, writing of the manuscript, or in the decision to submit the manuscript for publication.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: recent advances and new challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018.

[2] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, pp. 1–55, 2014.

[3] A. E. Essien, I. Petrounias, S. Pedro, and S. Sampaio, "Deep-PRESIMM: integrating deep learning with microsimulation for traffic prediction," in *Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 4257–4262, IEEE, 2019.

[4] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "The impact of rainfall and temperature on peak and off-peak urban traffic," *Lecture Notes in Computer Science*, vol. 11030 LNCS, pp. 399–407, 2018.

[5] M. Agarwal and R. R. Souleyrette, "Impacts of weather on urban freeway traffic flow characteristics and facility capacity," in *Proceedings of the 2005 Mid-Continent Transportation Research Symposium*, Ames, IA, USA, August 2005.

[6] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "Improving urban traffic speed prediction using data source fusion and deep learning," in *Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Kyoto, Japan, March 2019.

[7] A. Essien, G. Chukwkelu, and C. Giannetti, "A scalable deep convolutional LSTM neural network for large-scale urban traffic flow prediction using recurrence plots," in *Proceedings of the 2019 IEEE AFRICON*, Accra, Ghana, October 2020.

[8] A. Essien and C. Giannetti, "A deep learning framework for univariate time series prediction using convolutional LSTM stacked autoencoders," in *Proceedings of the 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, Sofia, Bulgaria, October 2020.

[9] M. Ahmadi and R. Taghizadeh, "A gene expression programming model for economy growth using knowledge-based economy indicators: a comparison of GEP model and ARDL bounds testing approach," *Journal of Modelling in Management*, 2019.

[10] G. Goh, J. Koh, and Y. Zhang, "Twitter-informed crowd flow prediction," in *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Singapore, October 2020.

[11] S. Wongcharoen and T. Senivongse, "Twitter analysis of road traffic congestion severity estimation," in *Proceedings of the 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Khon Kaen, Thailand, October 2013.

[12] S. Manoharan, "Short term traffic flow prediction using deep learning approach," Ph.D. diss., National College of Ireland, Dublin, Ireland, 2016.

[13] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "Improving urban traffic speed prediction using data source fusion and deep learning," in *Proceedings of the 2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Busan, Korea, October 2020.

[14] J. He, W. Shen, P. Divakaruni, and L. Wynter, "Improving traffic prediction with tweet semantics," in *Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence*, Beijing, China, October 2020.

[15] M. Ni and Q. He, "Using social media to predict traffic flow under special event conditions," in *Proceedings of the 93rd Annual Meeting of Transportation Research Board*, Washington, DC, USA, October 2020.

[16] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and non-parametric models for traffic flow forecasting,"October 2020.

[17] X. Wang, K. An, and L. Tang, *Short Term Prediction of Freeway Exiting Volume Based on SVM and KNN*, Elsevier, Amsterdam, Netherlands, 2015.

[18] L. Zhang, Q. Liu, and W. Yang, *An Improved K-Nearest Neighbor Model for Short-Term Traffic Flow Prediction*, Elsevier, Amsterdam, Netherlands, 2013.

[19] M. Ahmadi, "A computational approach to uncovering economic growth factors," *Computational Economics*, pp. 1–26, 2020.

[20] Y. Zhang, "Towards augmented kernel extreme learning models for bankruptcy prediction: algorithmic behavior and comprehensive analysis," *Neurocomputing*, vol. 430, pp. 185–212, 2021.

[21] M. Ahmadi, S. Jafarzadeh-Ghoushchi, R. Taghizadeh, and A. Sharifi, "Presentation of a new hybrid approach for forecasting economic growth using artificial intelligence approaches," *Neural Computing & Applications*, vol. 31, no. 12, pp. 8661–8680, 2019.

[22] Z. Fan, X. Song, and R. Shibasaki, "CityMomentum: an online approach for crowd behavior prediction at a citywide level," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Osaka, Japan, September 2015.

[23] X. Song and Q. Zhang, "Prediction of human emergency behavior and their mobility following large-scale disaster," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, October 2020.

[24] S. C. Hillmer and G. C. Tiao, "An ARIMA-model-based approach to seasonal adjustment," *Journal of the American Statistical Association*, vol. 77, no. 377, pp. 63–70, 1982.

[25] C. Wu and J. Ho, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[26] N. Mashros, J. Ben-Edigbe, S. A. Hassan, N. Abdul Hassan, and N. Z. Mohd Yunus, "Impact of rainfall condition on traffic flow and speed: a case study in johor and terengganu," *Jurnal Teknologi*, vol. 70, no. 4, 2014.

[27] M. L. Angel, T. Sando, D. Chimba, and V. Kwigizile, "Effects of rain on traffic operations on Florida freeways," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2440, no. 1, 2014.

[28] J. Andrey, "Long-term trends in weather-related crash risks," *Journal of Transport Geography*, vol. 18, no. 2, pp. 247–258, 2010.

[29] A. M. Ali and K. Willett, "What is the effect of the weather on trauma workload? A systematic review of the literature," *Injury*, vol. 46, no. 6, pp. 945–953, 2015.

[30] S. Datla and S. Sharma, "Variation of impact of cold temperature and snowfall and their interaction on traffic volume," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2169, no. 1, pp. 107–115, 2010.

[31] B. Smith, K. G. Byrne, and R. B. Copperman, "An investigation into the impact of rainfall on freeway traffic flow," in *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2004.

[32] J. Neter and W. Wasserman, *Applied Linear Statistical Models*, Richard D. Irwin, Homewood, IL, USA, 1974.

[33] M. Kyte, Z. Khatib, P. Shannon, and F. Kitchener, "Effect of weather on free-flow speed," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1776, no. 1, pp. 60–68, 2001.

[34] R. Billot, "Multilevel assessment of the impact of rain on drivers' behavior: standardized methodology and empirical

analysis," *Transportation Research Record*, vol. 2107, no. 1, pp. 134–142, 2009.

[35] I. Tsapakis and T. Cheng, *Impact of Weather Conditions on Macroscopic Urban Travel Times*, Elsevier, Amsterdam, Netherlands, 2013.

[36] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.

[37] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: a generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 777–785, Houston, TX, USA, April 2017.

[38] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," *Machine Learning*, 2017, https://arxiv.org/abs/1707.01926.

[39] Y. Liang, Z. Cui, Y. Tian, M. student, H. Chen, and Y. Wang, "A deep generative adversarial architecture for network-wide spatial-temporal traffic state estimation," *Transportation Research Record*, vol. 2672, no. 45, pp. 87–105, 2018.

[40] Y. Wu, H. Tan, L. Qin, and B. Ran, *A Hybrid Deep Learning Based Traffic Flow Prediction Method and its Understanding*, Elsevier, Amsterdam, Netherlands, 2018.

[41] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Chennai, India, July 2017.

[42] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: a GRU-based deep learning approach," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 578–585, 2018.

[43] W. Jin, Y. Lin, Z. Wu, and H. Wan, "Spatio-temporal recurrent convolutional networks for citywide short-term crowd flows prediction," in *Proceedings of the 2nd ACM International Conference on Compute and Data Analysis–ICCDA 2018*, pp. 28–35, DeKalb, IL, USA, March 2018.

[44] C. Chen, K. Li, S. G. Teo, and G. Chen, "Exploiting spatio-temporal correlations with multiple 3d convolutional neural networks for citywide vehicle flow prediction," in *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*, Singapore, November 2018.

[45] S. Guo, Y. Lin, S. Li, and Z. Chen, "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3913–3926, 2019.

[46] Y. Lin, L. Li, H. Jing, B. Ran, and D. Sun, "Automated traffic incident detection with a smaller dataset based on generative adversarial networks," *Accident Analysis & Prevention*, vol. 144, Article ID 105628, 2020.

[47] L. Li, Y. Lin, B. Du, F. Yang, and B. Ran, "Real-time traffic incident detection based on a hybrid deep learning model," *Transportmetrica: Transport Science*, pp. 1–21, 2020.

[48] H. Chen, A. A. Heidari, H. Chen, M. Wang, Z. Pan, and A. H. Gandomi, "Multi-population differential evolution-assisted Harris hawks optimization: framework and case studies," *Future Generation Computer Systems*, vol. 111, pp. 175–198, 2020.

[49] M. Wang and H. Chen, "Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis," *Applied Soft Computing*, vol. 88, Article ID 105946, 2020.

[50] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, and X. Zhang, "Enhanced Moth-flame optimizer with mutation strategy for global optimization," *Information Sciences*, vol. 492, pp. 181–203, 2019.

[51] X. Zhao, X. Zhang, Z. Cai et al., "Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients," *Computational Biology and Chemistry*, vol. 78, pp. 481–490, 2019.

[52] C. Yu, "SGOA: annealing-behaved grasshopper optimizer for global tasks," *Engineering with Computers*, 2021.

[53] M. Wang, H. Chen, B. Yang et al., "Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses," *Neurocomputing*, vol. 267, pp. 69–84, 2017.

[54] J. Xia, H. Chen, Q. Q. Li et al., "Ultrasound-based differentiation of malignant and benign thyroid Nodules: an extreme learning machine approach," *Computer Methods and Programs in Biomedicine*, vol. 147, pp. 37–49, 2017.

[55] H.-L. Chen, G. Wang, C. Ma, Z.-N. Cai, W.-B. Liu, and S.-J. Wang, "An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease," *Neurocomputing*, vol. 184, pp. 131–144, 2016.

[56] L. Hu, G. Hong, J. Ma, X. Wang, and H. Chen, "An efficient machine learning approach for diagnosis of paraquat-poisoned patients," *Computers in Biology and Medicine*, vol. 59, pp. 116–124, 2015.

[57] Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li, "Boosted binary Harris hawks optimizer and feature selection," *Engineering with Computers*, pp. 1–30, 2020.

[58] D. Zhao, "Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy," *Knowledge-Based Systems*, vol. 216, Article ID 106510, 2020.

[59] X. Zhao, D. Li, B. Yang, C. Ma, Y. Zhu, and H. Chen, "Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton," *Applied Soft Computing*, vol. 24, pp. 585–596, 2014.

[60] L. Shen, H. Chen, Z. Z. Yu et al., "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.

[61] J. Tu, "Evolutionary biogeography-based Whale optimization methods with communication structure: towards measuring the balance," *Knowledge-Based Systems*, vol. 212, Article ID 106642, 2020.

[62] C. Li, L. Hou, B. Y. B. Sharma et al., "Developing a new intelligent system for the diagnosis of tuberculous pleural effusion," *Computer Methods and Programs in Biomedicine*, vol. 153, pp. 211–225, 2018.

[63] W. Shan, "Double adaptive weights for stabilization of moth flame optimizer: balance analysis, engineering cases, and medical diagnosis," *Knowledge-Based Systems*, vol. 214, Article ID 106728, 2020.

[64] J. Hu, H. Chen, A. A. Heidari et al., "Orthogonal learning covariance matrix for defects of grey wolf optimizer: insights, balance, diversity, and feature selection," *Knowledge-Based Systems*, vol. 213, Article ID 106684, 2021.

[65] H. Yu, W. Li, C. Chen et al., "Dynamic Gaussian bare-bones fruit fly optimizers with abandonment mechanism: method and analysis," *Engineering with Computers*, pp. 1–29, 2020.

[66] X. Xu and H.-l. Chen, "Adaptive computational chemotaxis based on field in bacterial foraging optimization," *Soft Computing*, vol. 18, no. 4, pp. 797–807, 2014.

[67] Y. Zhang, T. Cheng, and Y. Ren, "A graph deep learning method for short-term traffic forecasting on large road networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 10, pp. 877–896, 2019.

[68] H. Lu, D. Huang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "ST-TrafficNet: a spatial-temporal deep learning network for traffic forecasting," *Electronics*, vol. 9, no. 9, p. 1474, 2020.

[69] Y. Zhang, T. Cheng, Y. Ren, and K. Xie, "A novel residual graph convolution deep learning model for short-term network-based traffic forecasting," *International Journal of Geographical Information Science*, vol. 34, no. 5, pp. 969–995, 2020.

[70] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, vol. 33, pp. 890–897, 2019.

Hindawi
Complexity
Volume 2021, Article ID 7887159, 24 pages
https://doi.org/10.1155/2021/7887159

Research Article

Event-Tree Based Sequence Mining Using LSTM Deep-Learning Model

János Abonyi, Richárd Károly, and Gyula Dörgö

MTA-PE Lendület Complex Systems Monitoring Research Group, Department of Process Engineering, University of Pannonia, Egyetem u. 10, Veszprém H-8200, Hungary

Correspondence should be addressed to Gyula Dörgö; gydorgo@gmail.com

Received 10 June 2021; Accepted 31 July 2021; Published 16 August 2021

Academic Editor: Gonzalo Farias

During the operation of modern technical systems, the use of the LSTM model for the prediction of process variable values and system states is commonly widespread. The goal of this paper is to expand the application of the LSTM-based models upon obtaining information based on prediction. In this method, by predicting transition probabilities, the output layer is interpreted as a probability model by creating a prediction tree of sequences instead of just a single sequence. By further analyzing the prediction tree, we can take risk considerations into account, extract more complex prediction, and analyze what event trees are yielded from different input sequences, that is, with a given state or input sequence, the upcoming events and the probability of their occurrence are considered. In the case of online application, by utilizing a series of input events and the probability trees, it is possible to predetermine subsequent event sequences. The applicability and performance of the approach are demonstrated via a dataset in which the occurrence of events is predetermined, and further datasets are generated with a higher-order decision tree-based model. The case studies simply and effectively validate the performance of the created tool as the structure of the generated tree, and the determined probabilities reflect the original dataset.

1. Introduction

Nowadays, uncovering possible frequent event sequence scenarios has been a critical task across many disciplines. In the age of big data, when an immense amount of data is recorded into logs in the scope of the industry 4.0 trend, it is important for engineers to acquire as much knowledge about the industrial processes as possible [1, 2]. By using frequent pattern mining algorithms on event logs, we are able to identify sequences that can lead to given system states. This particular method has already proved its capability across numerous applications and industries. Taub et al. use sequence mining to distinguish efficient and nonefficient action patterns among their subjects in a game-based learning environment [3]. A similar frequent pattern identification method was used to give insight into successful learning patterns using Betty's brain computer-based learning environment [4]. A universal (language independent) algorithm was proposed for linguistical pattern discovery, where special attention was paid to a clear, easily understandable output [5]. Kant et al. proposed a new algorithm (*MC*PRISM) to mine min-closed sequences to identify comment section spam content on websites [6]. A new framework called malicious sequential pattern-based malware detection was developed by using a novel sequential pattern mining algorithm (MSPE) to recognize new, unseen malicious executables in computer systems [7]. Weiss uses a genetic algorithm for analyzing the temporal patterns in the alarm data of telecommunication systems to identify equipment failure [8]. Sequential pattern mining has been also used for event prediction in numerous applications [9, 10].

Although these examples are perfectly capable of fulfilling the sequential pattern mining task, traditional algorithms suffer greatly with runtime and accuracy when dealing with massive datasets [11]. Another drawback of the frequent pattern mining solutions is that their output data are proved to be challenging to interpret and

handle—especially when the number of the mined sequences is high—often introducing a new problem to solve [12]. To represent the yielded information, the frequent pattern tree proved to be a much more compact and workable data structure [13].

Machine learning techniques are excellent tools for processing massive datasets. Learning patterns from exemplary training sequences is a similar task as in the case of learning languages and the identification of frequent event sequences, where the use of long-short term memory (LSTM) yields better results compared to that of traditional recurrent neural networks (RNNs) [14]. The reason why LSTM is suitable for this application is the use of the forget gate in its cell, which is able to reset the internal state of the network [15]. The algorithm known as the seq2seq learning method was developed in 2014 by Sutskever et al. at Google for frequent sequence learning using LSTM to improve machine translation [16]. Ever since this method has been used in numerous applications. Karatzoglou et al. used it to improve location-based services by learning human semantic trajectories and better predicting their upcoming location [17]. The method's capabilities have also been demonstrated in finances by Rebane et al. who analyzed the performance for cryptocurrency price prediction [18]. A seq2seq model-based approach was used to improve query focused summarization performance [19]. Wu et al. described a novel method to create, store, and convert logs of Internet of Things big data systems to be later processed through their proposed seq2seq algorithm [20]. The method has also been applied in manufacturing systems. Hwang et al. used the algorithm to predict a furnace temperature based on other process variables with a very high accuracy [21]. The general application of this structure for event prediction has been described in detail by Dörgő et al. [22, 23].

Fundamentally, the output of a seq2seq approach is a single sequence, which consists of the items that have been found as the most probable at each prediction step. By using a heuristic search algorithm during inference, further information can be retained from each prediction step. This information can aid us to understand better the black-box model of the prediction [24]. This optimization is done using beam search, which retains several best items—the number usually referred to as beam width. Cohen and Beck studied the performance degradation in neural sequence models when an inappropriate beam width is chosen [25]. In recent years, the use of beam search instead of the traditional greedy search was favored because it usually provides much better results, although it is taxing on runtime [26].Li et al. used a seq2seq model with beam search decoder to realize a dependency parser with a direct head prediction with promising performance [27]. Williams et al. proposed the use of beam search to build an end-to-end speech recognition system, which is capable of adapting the inference process based on contextual signals at each prediction step [28]. Several different pruning strategies have been explored to be used with beam search to improve runtime [29]. A seq2seq model using the dynamic beam width was applied by Jahier Pagliari et al. to an embedded translation system in order to improve its efficiency [30]. A known drawback of the beam search algorithm is that it produces pretty similar output sequences in certain use cases. A solution for this phenomenon was proposed for image captioning [31].

This paper aims to create own implementation of the seq2seq learning method with a beam search decoder, which is referred to as the seq2probTree method later. This method will be realized in the Python environment, and it is able to create a probability tree that describes the alternative network of events based on a given input. The implemented tool is capable of displaying the output an easy to interpret, structured probability tree, thus giving a visualization of the prediction and aiding the debugging of seq2seq models, as the fault analysis of deep neural networks is a task with enormous importance, especially in the case of safety-critical application [32].

First, in Section 2, the methodology will be explained. Definitions will be given to the necessary expressions and the prediction task at hand. The LSTM deep-learning model will be described along with the tree creation process. The metrics used for the evaluation will also be defined in this section. In Section 3, the implementation process and the used toolboxes will be presented briefly. Then, the seq2probTree method will be put to the test by applying it on a first-order Markov chain model and later on a higher-order tree-based system, where the extent to which the method is able to reconstruct the tree is checked, and the necessary comparison score is defined. Finally, the real-life practical applicability is confirmed by using it on the alarm logs of a hydrofluoric acid alkylation production unit. Last, in Section 4, the findings and experiences of using the developed method will be summarized, and further steps in the subject will be proposed.

2. Methodology

In this section, the previously defined task will be explained in detail. The definition will be given to an event sequence and how its probability is calculated. The peculiarity of the seq2probTree method is explained, creating a whole sequence tree instead of only predicting the most likely scenario. Here, in addition to the theory of prediction, its extension to tree-based event-scenario generation is also provided. The metrics used for the evaluation of the predicted event scenarios are also explained in detail.

2.1. Sequences and the Prediction Task. Industrial processes frequently generate *event logs* those are logically consisting of *events* (denoted as e_i) related to production, safety, transportation, storage, sales, financial transactions, marketing, etc. An event log defined as D_T database is an ordered list of these events, where the events are arranged according to their start time in the ascending order. The D_T dataset can be segmented into *sequences* (denoted as Φ_n), which are the chronologically ordered lists of events $\Phi_k := e_1 \Rightarrow e_2 \Rightarrow \cdots \Rightarrow e_k$. According to different aspects, this segmentation can be carried out: causal connection of states, temporal segmentation, periodicity, etc. Therefore, a

sequence of k events is referred to as a k-length sequence and is denoted by Φ_k. These events represent the occurrence of n different states (type of events) of the set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$. The sequence Φ_k: $= e_1 \Rightarrow e_2 \Rightarrow \cdots \Rightarrow e_k$ can be divided chronologically at any part as $\Phi_k = (\Phi_{k'} \Rightarrow \Phi_{k''})$, where $\Phi_{k'}$ and $\Phi_{k''}$ are the antecedent and future sequence of states, respectively (naturally, $k = k' + k''$). Hereinafter, the "$'$" and "$''$" symbols denote the past and future sequences or states, respectively.

As single or multiple connected processes usually generate the data analyzed here, a causal flow connects the individual temporal instances of states (regardless of the type of the dataset, e.g., events, items, transactions, etc.), and the number of occurrences of different states is not independent of each other. Therefore, the probability of the occurrence of the Φ_k sequence $P(e_1 \Rightarrow e_2 \Rightarrow e_3 \Rightarrow \cdots \Rightarrow e_k)$ can be calculated by the chain rule and the conditional probabilities of transitions between the events according to the following equation:

$$P(\Phi_k) = P(e_1) \times P(e_2 \mid e_1) \times P(e_3 | e_1 \Rightarrow e_2) \times \cdots \cdots \times P(e_k \mid e_1 \Rightarrow e_2 \Rightarrow \cdots \Rightarrow e_{k-1}). \quad (1)$$

Therefore, according to the chain rule, the probability of a k-length sequence can be calculated as the product of the conditional probabilities of the step-by-step transition from the sequence of antecedent events to the present one. A conditional probability is the ratio of the number of occurrences of the more extended sequence and the shorter one, denoted by the supp value of the sequence, according to the following equation:

$$P(e_k \mid \Phi_{k-1}) = \frac{P(\Phi_k)}{P(\Phi_{k-1})} = \frac{\mathrm{sup}p(\Phi_k)}{\mathrm{sup}p(\Phi_{k-1})}. \quad (2)$$

This probability of transition reflects how confident is the next state knowing the previous sequence of states in Φ_{k-1}.

2.2. The Network of Alternative Events: Sequence Trees. The methodology where the prediction of the following state with the highest conditional probability is accepted was described by Dörgő and Abonyi [22]. However, the underlying processes and, hence, the resultant datasets can be highly complex. The ultimate goal of this method is being able to create an event sequence tree that describes the possible courses (all highly probable $\Phi_{k''}$) of events based on a given input sequence ($\Phi_{k'}$). Figure 1 shows the idea in detail. The horizontal axis indicates the time and illustrates how the possible future scenarios after $k\prime$ past events are ordered in a tree structure. The red branch of the tree indicates the scenario if the predictions of the highest probability are accepted in every prediction step, namely, by using the greedy search algorithm. The EOS tag indicates the end-of-sequence prediction.

So far, only the scenario with the highest probability has been predicted, ignoring the possibility of the occurrence of a less likely, however, highly informative and essential subsequence, which can indicate a different scenario of upcoming events. The added feature of this method is to uncover the information that these highly probable sequences may yield.

Therefore, accepting that the conditional probability-based prediction model often predicts several events with similar probability, here, the implemented beam search algorithm is described, thus not just the future sequence with the highest probability is accepted, but a scenario tree is formalized accepting all the predicted events above a certain probability threshold (P_{thr}). Therefore, after the occurrence of the first k' events, the prediction of the first future event e_1'' is accepted if its confidence of transition is above a specific P_{thr} limit as follows:

$$\left\{ e_1'' | P\left(e_1'' | \Phi_{k\prime}\right) > P_{\mathrm{thr}} \right\}. \quad (3)$$

Applying equation (3) in every prediction step, not a single future sequence but multiple sequences or possible future scenarios are predicted as depicted in Figure 1. Thus, as it is described by the prediction task, the $P(\Phi_{k''} | \Phi_{k\prime})$ conditional probability is to be determined among all possible future $\Phi_{k''}$ sequences.

In order to annotate the scenarios as well, a hierarchical annotation was introduced in the superscript of the predicted event: the numbers divided by commas after the "$\prime\prime$" mark indicate the likeliness of the predicted event in the prediction step as the number in order of the likeliness of the prediction, where 1 indicates the most likely future state. For instance, the tag $e^{\prime\prime,1,3,1}$ shows that this is the third predicted future event (three numbers are present after the "$\prime\prime$" mark), and this was the event with the highest probability for the first predicted state $e^{\prime\prime,1}$; then accepting this prediction, the second predicted event has the third highest probability $e^{\prime\prime,1,3}$ and accepting the first two predictions, the third predicted event had the highest probability in the given prediction step. Similarly, the $e^{\prime\prime,2,1}$ future state is the prediction with the second highest probability ($e^{\prime\prime,2}$) for the first future event and accepting this prediction, this is the prediction with the highest probability in the second prediction step. Therefore, continuously accepting the most likely predictions, the sequence $e^{\prime\prime,1} \Rightarrow e^{\prime\prime,1,1} \Rightarrow e^{\prime\prime,1,1,1} \Rightarrow \cdots$ is predicted, highlighted by the red arrow in Figure 1. However, in this sequence, the predictions with the highest probability are accepted in every step, the overall probability of the sequence is not maximal in every situation, since after the acceptance of a less likely prediction in a prediction step, the following predicted events could be of a high probability and then the overall probability of the occurrence of the sequence can be relatively high (the overall probability of the occurrence of a sequence is the product of the transition probabilities according to equation (1)).

By repeating the prediction task at each node, the sequence tree explained in Figure 1 may be created. After each prediction step, by meeting the confidence of all the possible events to the previously defined P_{thr} probability limit, we can make sure that we keep the complexity of the tree as low as necessary for the given task.

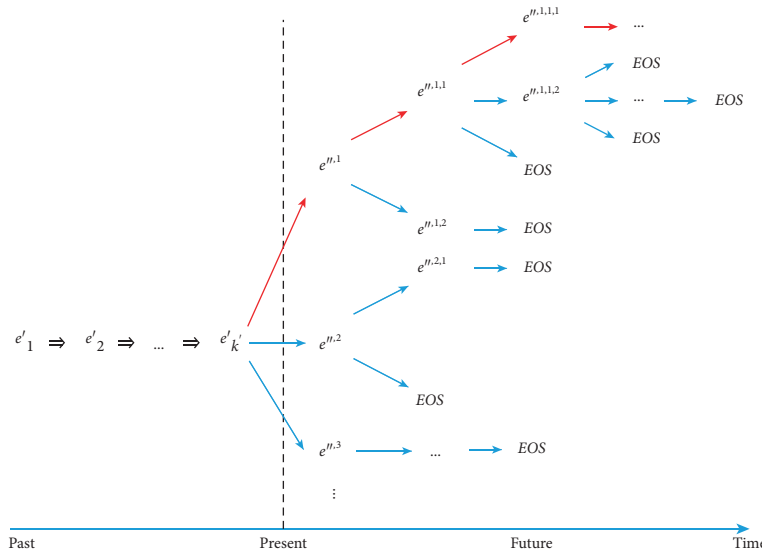


Figure 1: The predicted scenarios ordered in a tree structure (the EOS tag indicates the end-of-sequence prediction).

2.3. The LSTM Deep-learning Model. In the seq2seq machine learning method, the so-called long-short term memory is utilized as a recurrent neural network of choice. This network was specifically developed to deal with the problem of vanishing gradients with the least possible computational cost increase [33]. The LSTM network is well-known for its capability of classification, processing, and prediction making on time series data due to its relative insensitivity to gap length (lag) between discreet events, which property is welcome in the given use case. The LSTM structure is depicted in Figure 2.

The input of the model: Figure 2 highlights the structure of the input sequences. First, an end-of-sequence (EOS) tag is appended to the end of every sequence to indicate the end of the event series. The implemented EOS tag is added to the end of the sequences and handled similarly to all the other events in the subsequent steps. Moreover, the order of the events in the input sequence is reversed, since according to Sutskever et al. [16], the prediction accuracy significantly improves when the beginning of the input sequence is close to the beginning of the predicted sequence.

Embedding layer: The described sequence of input events needs to be transformed into a mathematically manageable vector of numerical values. Therefore, first, the symbols are encoded as one-hot encoded vectors, $\mathbf{oh}_t$ of binary values of length n_d, where n_d is the number of one-hot encoded symbols. In the one-hot encoded vectors, only one bit related to the encoded symbol is fired. A detailed explanation and visualization of one-hot encoding can be found in [34]. Then, the embedding layer transforms the one-hot coded vectors into a lower dimension (n_e) of continuous values using a $\mathbf{x}_t = \mathbf{W}_{\text{emb}}\mathbf{oh}_t$ linear transformation. Note that, in Figure 2, the embedded forms of the EOS symbol are denoted by the symbol EOS.

Encoder and decoder layers: The encoder LSTM layer processes the sequence of one-hot coded and then embedded symbols. Instead of calculating its output values, it maps the sequence into its internal states. These internal weights of the encoder layer represent the state of the process, which generated the events. These weights are used to condition the decoder layer, which means the transfer of information of that happened previously in the process and generally means copying the encoder layer's weights into the decoder layer, obtaining the same structure (of n_u LSTM units). These weights indicate the prediction required from the decoder layer. After the input of an (embedded) start-of-sequence symbol, the decoder layer predicts the next event of the predicted sequence iteratively, consistently applying the previously predicted event as the input for the prediction of the next event. This procedure is repeated until an end-of-sequence symbol is predicted or the maximum sequence length is reached.

Dense layer: After the decoder layer maps the input event $\mathbf{x}_{k''}$ into a vector of real values $\mathbf{h}''$ represented as $\mathbf{h}'' = [h''_1, \ldots, h''_{n_U}]$, these values are used to calculate the probabilities of occurrence of the events using the softmax activation function of the dense layer in Figure 2,

$$P\left(e_{(t+1)''} \mid \mathbf{x}_{t''}\right) = P\left(e_{(t+1)''} \mid \mathbf{h}_{t''}\right) = \frac{\exp\left(\left(\mathbf{h}_{t''}\right)^T \mathbf{w}_{s,j} + b_j\right)}{\sum_{j=1}^{n_d} \exp\left(\left(\mathbf{h}_{t''}\right)^T \mathbf{w}_{s,j} + b_j\right)}, \tag{4}$$

where $\mathbf{w}_{s,j}$ represents the j-th column vector of the weight matrix of the output dense layer of the network $\mathbf{W}_s$, and b_j represents the degree of bias. Once the probability of each state in our dictionary is determined, all the predictions above the defined threshold P_{thr} is accepted as the next event of the related future scenario,

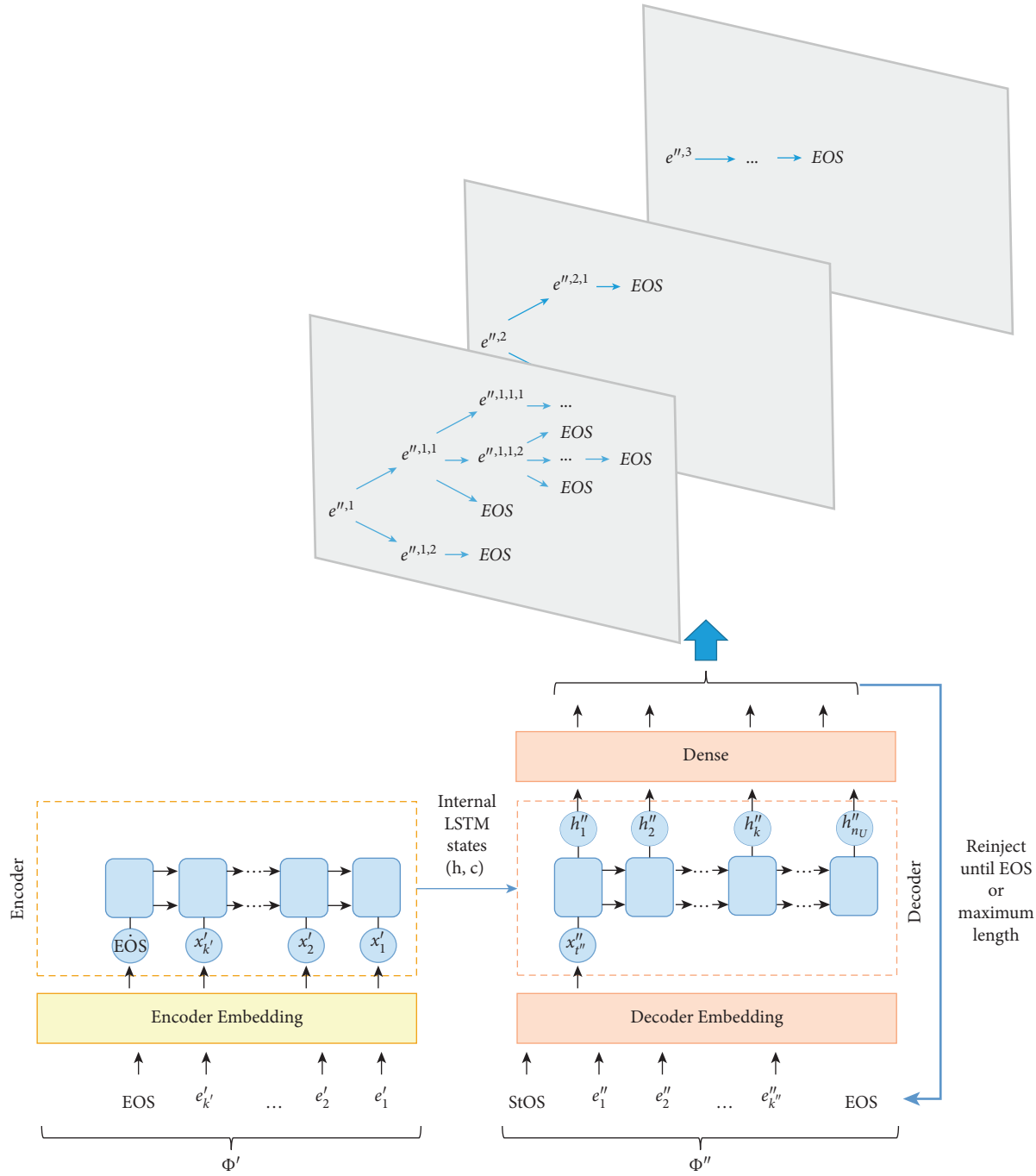


FIGURE 2: The illustration of the structure of the sequence-to-sequence event-scenario prediction. The encoder model maps the states of the input sequence into a fixed-length vector-based representation. Using these vector-based representations of input events as the initial state, the decoder model determines the next event. However, using the probabilities calculated by the dense layer, not just the event with the highest probability is recorded, but event scenarios are predicted using every prediction above a predefined threshold. The StOS and EOS tags mark the start-of-sequence and end-of-sequence tags, respectively.

$$\left\{e_{(t+1)''} \mid P\left(e_{(t+1)''} \mid \mathbf{h}_{t''}\right) > P_{\text{thr}}\right\}. \tag{5}$$

2.4. Creation and Traversal of the Probability Trees. Prior to prediction, the sequence of events that defines the state of the process is to be transformed to the internal state of the encoder layer. Then, these internal states of the encoder layer containing information on the history of the process are transferred to the decoder layer. The prediction starts with the input of a start-of-sequence symbol (marked as StOS in Figure 2). The decoder network generates the prediction of the next event, which is reintroduced into the input of the decoder network and applied as the input in the next time

step. By utilizing the original seq2seq learning method, the generated events are continuously appended to the predicted sequence of events. The feature added by the seq2probTree method is that after the first prediction step following the start-of-sequence symbol, we do not simply accept an event as the next with the highest probability. However, we take the entire output vector and apply equation (3), thus pruning the candidates for the next possible event. Then, we further explore the network of alternative events during which the probability of each upcoming event is determined (and stored if that probability is adequate), thereby realizing the beam search algorithm. The prediction process is continued until the layer generates the end-of-sequence symbol or reaches the previously set limit of the length of the predicted sequence in the case of every scenario.

The method results in a probability tree that is explored and recorded in a depth-first manner (Figure 3). The resource demand of this approach is significantly increased as it is necessary to store all the internal LSTM states and the previous prediction's output for each step—depending on the original number of the possible events—could be a memory hog. In addition, an increase in the inference runtime is expected as the time demand of the depth-first search algorithm is $O(|V| + |E|)$, where V and E stand for the number of vertices and edges in the tree, respectively. The pseudocode for the tree traversal and the recursive prediction step is given below.

2.5. Evaluation and Metrics. The evaluation of the model was carried out using metrics that measure the potential applicability of the method. Since the focus is on the development of a prediction system that draws attention to the most possible outcomes of the process, three performance metrics have been identified for characterizing the sequence containing the events that are found suitable in every step by using equation (3). Therefore, for easier notation, we introduce $\widehat{\Phi}$, a sequence containing the events with only adequate prediction probabilities in every step.

First, S_1 is the percentage of the $\widehat{\Phi}$ sequences that include at least one well-predicted event. For mathematical formulation, Φ is the sequence of events that we aim to predict, while Φ'' is our prediction. N is the number of sequences in the analyzed database, the cardinality of a set is marked with $|^*|$, while the common elements in two sequences are marked as their intersection. Mathematically, S_1 is expressed as follows:

$$S_1 = \frac{\sum_{n=1}^{N}\left(\left|\Phi_n \cap \widehat{\Phi}_n''\right| \geq 1\right)}{N}. \tag{6}$$

Second, $S_{\%}$, a set-based similarity measure that describes the well-predicted events as a percentage of the length of the target sequence has been defined. The events do not have to be in the order of occurrence, and $S_{\%}$ measures how accurately the type of events are predicted,

$$S_{\%} = \frac{\sum_{n=1}^{N}\left|\Phi_n \cap \widehat{\Phi}_n''\right|/\left|\Phi_n\right|}{N}. \tag{7}$$

Finally, S_{ED} was proposed, which is an edit distance-based similarity metric that provides the edit distance between the actual (target) and predicted sequence as a percentage of the length of the more extended sequence among them. The edit distance yields the minimum number of elements that must be inserted or skipped in the compared sequences in order to be identical. The edit distance of two sequences is marked with ED, and equation (8) mathematically describes the S_{ED} edit distance-based similarity metric,

$$S_{\mathrm{ED}} = \frac{\sum_{n=1}^{N}\mathrm{ED}\left(\Phi_n, \widehat{\Phi}_n''\right)}{N}. \tag{8}$$

These performance metrics are calculated for each sequence on the tree, whenever a leaf is found, that is, EOS is predicted, or the maximum sequence length is reached. However, in order to make the resulting sequences even more comparable, their confidence is also calculated. Confidence for each $\widehat{\Phi}$ is defined as a product of the supports of all the containing events in the sequence. The support of the event is the probability the LSTM calculated for that item, given the sequence of the previous events. For the events in the input sequence, the support is determined as a value of 1,

$$\text{confidence} = \prod_{i=1}^{k} P\left(e_i|\widehat{\Phi}_{i-1}\right). \tag{9}$$

3. Implementation and Results

In this section, a summary is provided on the implementation of the proposed method. Then, the used validation techniques are detailed, and the obtained results are evaluated. Since the implemented tool is used for diagnostic purposes, the results should be easily reproducible. Thus, the validation is performed by applying the proposed methods on examples with different complexities. First, the realized system is validated on a simple first-order Markov chain where the method's capability to reproduce the sequence tree is examined. Then, to demonstrate the proposed method's capability to understand higher-order relationships between events, a more complex benchmark dataset is generated using a tree-based system. Finally, the method is tested on a real-life production unit.

3.1. Realization of the seq2probTree Method. The described method was implemented in Python using the Spyder 4 Integrated Development Environment in the Anaconda open-source data science development platform. This platform was ideal for the task as most of the necessary libraries are included by default, thus minimizing the setup process for development. The LSTM RNN was implemented using Keras, a deep-learning application programming interface running on top of the TensorFlow end-to-end open-source machine learning platform. Keras API is well-known for its full-fledged documentation and high-quality example codes, which are usually very well commented for easy

Require: $\text{model}_{\text{LSTM}}$, $\text{eventseq}_{\text{input}}$
Create root node for $\text{Tree}_{\text{Event}}$
Append events in $\text{eventseq}_{\text{input}}$ to $\text{Tree}_{\text{Event}}$
$\text{input}_{\text{LSTM}} = \text{inputConversion}\ (\text{eventseq}_{\text{input}})$ \(⊳\) Conversion of input to match Encoder Embedding layer format
$\text{states}_{\text{LSTM}} = \text{encoder}_{\text{LSTM}}.\text{prediction}\,(\text{input}_{\text{LSTM}})$
$\text{RecursiveDecoding}(\text{input}_{\text{LSTM}}, \text{states}_{\text{LSTM}}, \text{Tree}_{\text{Event}})$

Algorithm 1: Preprocessing before prediction.

Require: $\text{input}_{\text{LSTM}}$, $\text{states}_{\text{LSTM}}$, $\text{Tree}_{\text{Event}}$
$\text{output}_{\text{LSTM}}, \text{states}_{\text{LSTM}} = \text{decoder}_{\text{LSTM}}.\text{prediction}\,(\text{input}_{\text{LSTM}}, \text{states}_{\text{LSTM}})$
$i = 0$
While There is $e'' | P(e'') > P_{\text{thr}}$ in $\text{output}_{\text{LSTM}}$ **AND** $i < N_{\text{Thr}}$ **do**
Add e'' to $\text{Tree}_{\text{event}}$
Delete e'' from $\text{output}_{\text{LSTM}}$
if not(e'' is EOS OR $\text{sequencelength}_{\text{MAX}}$ reached) **then**
$\text{RecursiveDecoding}(\text{input}_{\text{LSTM}} + e'', \text{states}_{\text{LSTM}}, \text{Tree}_{\text{Event}})$
$i = i + 1$

Algorithm 2: RecursiveDecoding function.

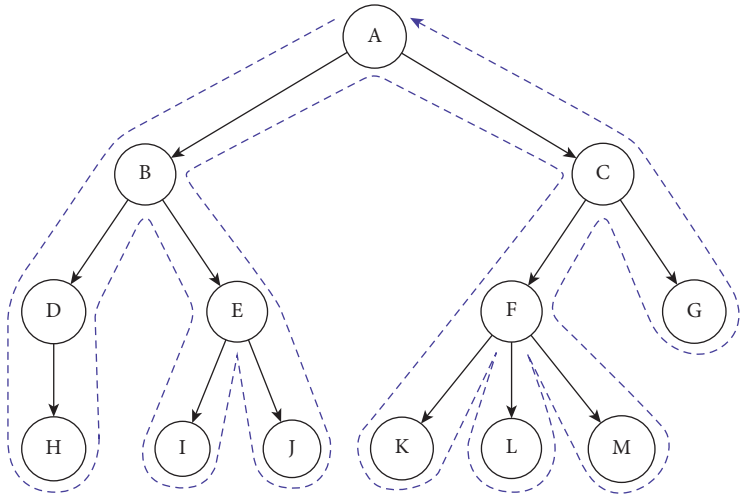


Figure 3: Depth-first traversal of a tree structure.

adaptation. In order to decrease the runtime of the training process of LSTM, the NVIDIA CUDA®Deep Neural Network (cuDNN) library was utilized. Since Keras is built on top of Tensorflow, which happened to be a cuDNN accelerated framework after the initial setup, the time required by the LSTM training was reduced tenfold. This speed increase was provided by an NVIDIA Geforce GTX 1080 Ti graphics processing unit.

The probability trees presented in this paper were generated using the ETE toolkit for Python, which provides a wide range of tree-handling options and node annotation features alongside a tree visualization system to output the resultant trees. The code of Markov chain models was created in MathWorks MATLAB environment for the ease of exporting the simulated data into *.xlsx* format and importing it into Python using the pandas library. However, due to the vast size of the training dataset for the third-order Markov model, MATLAB's *.m* format had to be utilized, which can be handled by SciPy (conveniently included in Anaconda).

The finished implementation consists of two routines. The first contains the selection of the desired dataset, the setup of the LSTM, the training procedure, and the creation of the training history plots. After the training process has been completed, the encoder and decoder models are saved, thus eliminating the necessity of running the model training with each subsequent session of the application of the tool. The second routine consists of loading the LSTM models, the recursive decoding, and all the functions necessary for the metric calculation and the tree generation and output.

3.2. Validation on First-Order Markov Model. In this section, a brief summary will be given on how the proposed method has been implemented. For the ease of validation, a simple Markov chain is used. The model consists of 12 states that follow each other in a row as a rule of thumb. The only 2 exceptions are state 4 and 7, which break this rule. While transitioning from state 4, there is a probability of 0.35 that the system will "reset," thus returning to state 1. If the system reaches state 7, there is a 30% chance that the system skips the following 2 states and goes right to 10. This behavior can be observed in Figure 4.

The dataset was established by creating 10000 sequences utilizing the described Markov chain. Each sequence starts from a randomly selected system state, and the length is also randomly determined between 9 and 12. After the generation of the dataset, the LSTM model was trained by using the following parameters:

(i) Embedding dimensions = 6
(ii) Latent dimensions = 15
(iii) Batch size = 256
(iv) Epochs = 70

The training's accuracy and loss can be observed in Figure 5. In order to validate the model's performance, a cross-check was made by feeding each state as an input to the encoder, thus initializing the internal LSTM states. It is important to note here that to initialize the encoder for the validation, not only the state from which the prediction starts needs to be used as the input but also the previous two states; as for the model training, each sequence in the database was separated after the third state as input and output. Then, one prediction step is completed, and the output of the LSTM is recorded. This is repeated for each state, creating the validation transition matrix, which is then compared to the transition matrix of the first-order Markov chain (part (a) in Figure 4). In Figure 6, each predicted value is illustrated in function of the original transition probability. The calculated coefficient of determination for this simple example is as high as 0.9994.

After the training was completed, the seq2probTree method was utilized with $P_{\text{thr}} = 0.2$ and by giving the input sequence of $[1, 2, 3]$ to the taught LSTM model. The maximum output sequence length was set to 12.

Figure 7 gives visual aid about the metrics placed at each node on the probability tree, while the acquired results can be observed in Figure 8. Each node on the tree has at least three properties: name, support, and confidence (top and bottom values, respectively). The EOS nodes also have the three performance metrics calculated for the given sequence: S_1, $S_{\%}$, and S_{ED}, values of which can be found in the right column in the specified order from top to bottom. For example, it can be observed from Figure 7 that the seq2probTree method predicted state 11 after the subsequence ending with state 10 with a probability of 0.49. In addition, the calculated probability of ending the sequence after state 11 is 0.5. We can also see that the confidence of Φ_k—thus, the whole sequence ending with EOS—is 0.04. The S_1 value also shows the highlighted Φ_k sequence that every entry (1.0) in the input database starting with the given $\Phi_{k'}$ subsequence—in this case [1 2 3]—has at least one state that has been predicted in $\Phi_{k''}$ by the method. $S_{\%}$ being 0.68 gives us the idea that the states predicted in $\Phi_{k''}$ occur in 68% of the database entries starting with [1 2 3]. The last metric of this EOS node on the probability tree—S_{ED}—shows that the average edit distance—thus the number of changes that need to be made to match the sequence—is 4.49, given the aforementioned $\Phi_{k'}$.

The properties of the first-order Markov chain are observable in the results. Both of the distinguished transitions are identifiable, and the predicted transition probabilities are within a margin of error of the Markov chains. The tree also reflects all the different length variants of each possible sequences.

3.3. Validation on Higher-Order Tree-Based System. As the LSTM-based deep-learning networks are explicitly developed to capture the long-term relationship in datasets, a higher-order system is used for further evaluation. The behavior of the system is based on a probability tree, which was pseudorandomly generated. Each node on the tree may have up to three children, with the system stating that it represents and the probability that state occurs also generated randomly. The sum of the probabilities of states originating from the same node is normalized to 1. The depth of the tree was determined randomly between 8 and 9—without considering the root (StOS) and leaf (EOS) nodes. The number of applied states is set to 4 to facilitate easier understanding and reconstruction of the results. However, at this complexity, it is already a difficult task. The states are represented by letters A, B, C, and D. The complexity of the system can be observed in Figure 9, while the inspected transition probabilities—thus the highlighted areas—are visible more transparently in Figures 10–12.

To utilize the seq2probTree method, a training dataset was created consisting of 10,000 simulations of the system starting from the root node and randomly determining the path—based on the transition probabilities—until a leaf node is reached. After the given amount of simulations were concluded, the resultant dataset was copied six times, as during the training, the sequences are split to input and target and this position, where the sequences are separated. as input and target is randomly selected. The reason for the six times multiplication is that the position of the cut is varied between the 1 st and the 6 th state in the sequence —separating the input and target after the selected state. The generated dataset was used for the training of the LSTM model by using the following parameters:

(i) Embedding dimensions = 2
(ii) Latent dimensions = 15
(iii) Batch size = 64
(iv) Epochs = 25

During the training on the dataset produced by the simulation of the proposed tree-based system, the accuracy and loss functions were also recorded. They can be

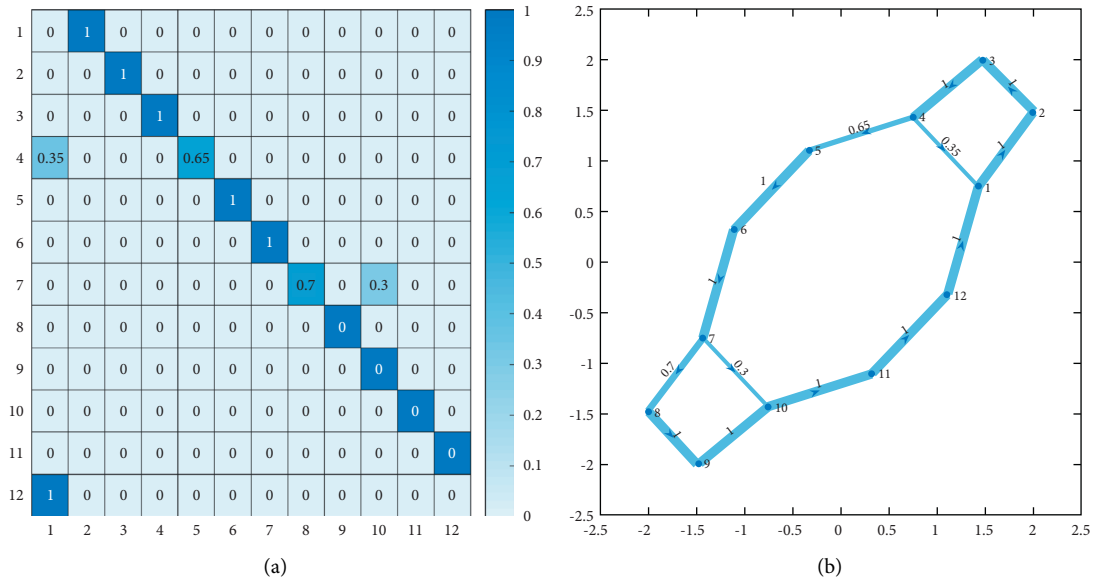


Figure 4: The transitional probabilities (a) and the directional graph (b) of the first-order Markov chain.

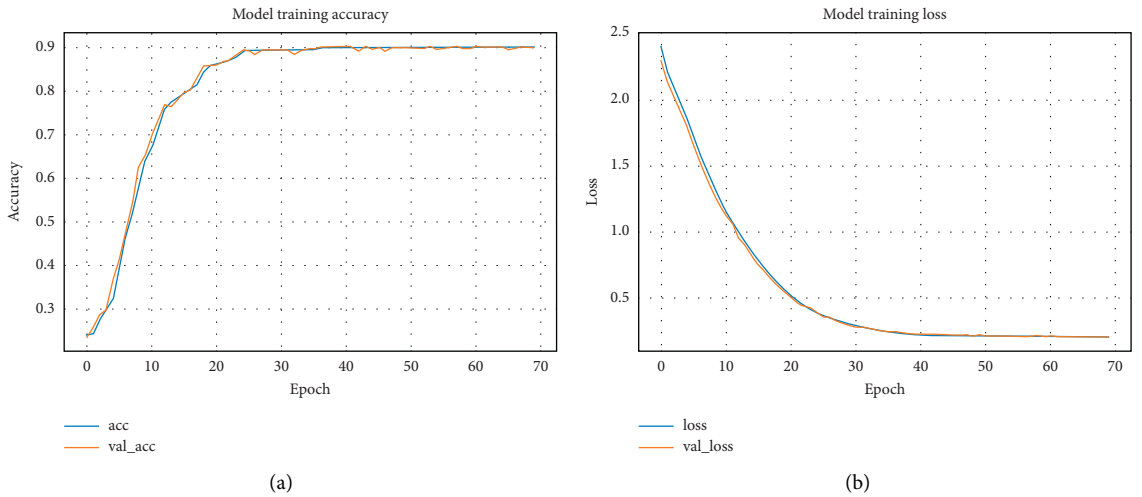


Figure 5: Training statistics on the first-order Markov chain.

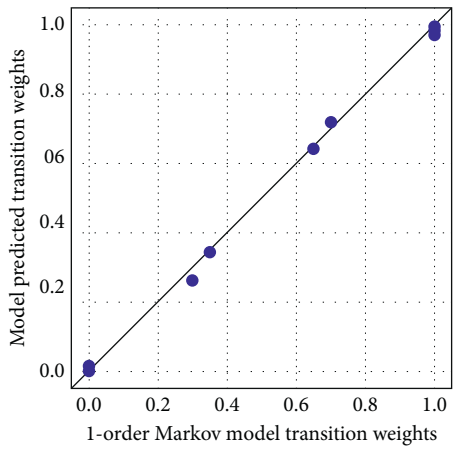


Figure 6: Crossvalidation of the transition probabilities of the first-order Markov chain.

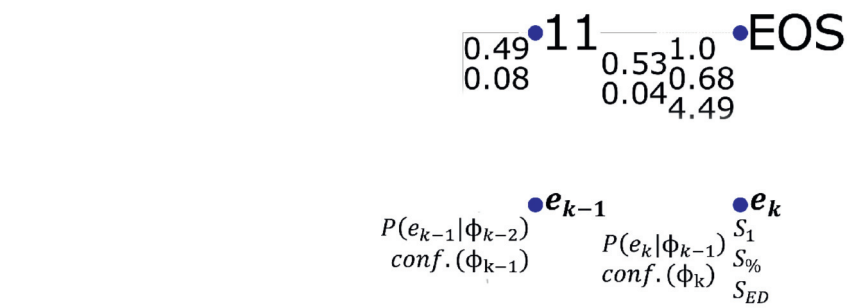


Figure 7: Explanation of the metrics located at the nodes of the probability tree.

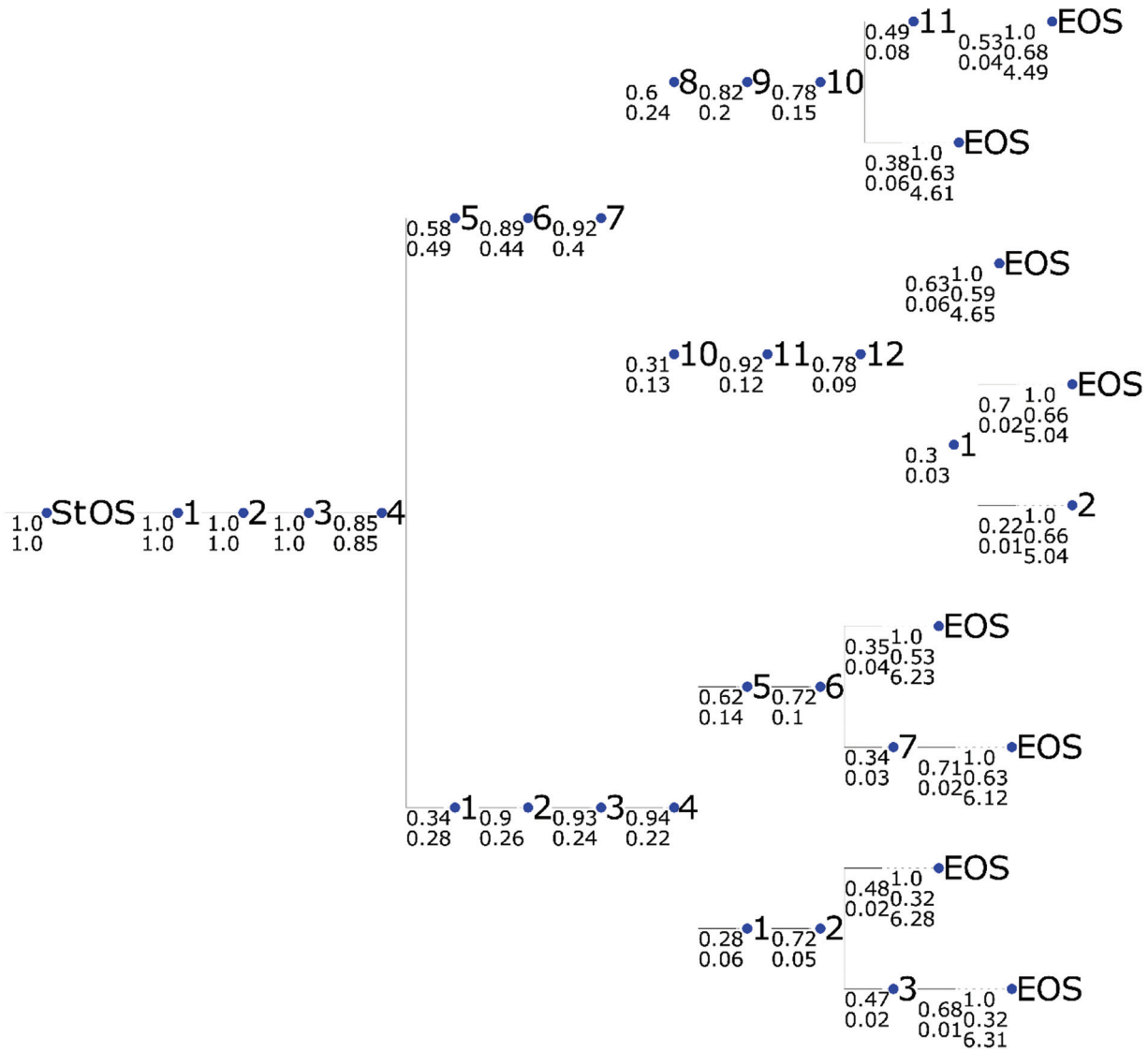


Figure 8: Output of the seq2probTree learning method for the first-order Markov model.

observed form Figure 13. It is important to note that during the training, 20% of the dataset was used as validation data, while dropout was not utilized in the LSTM layer.

After the training of the LSTM model on the aforementioned dataset, the seq2probTree method was applied with input sequences leading to the highlighted areas in Figures 14–16—[A D], [D D], and [B] respectively.

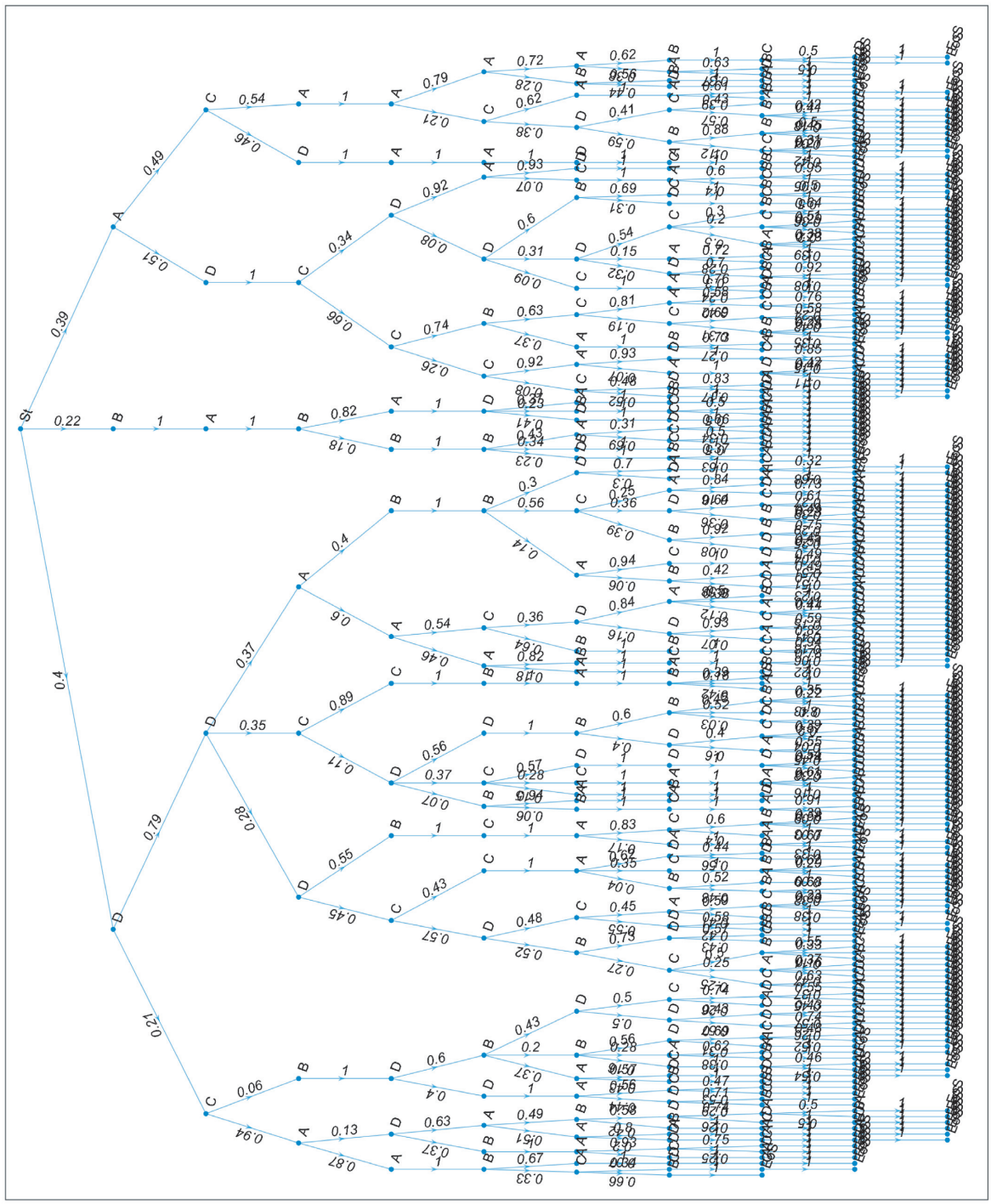


Figure 9: The state transition probabilities of the full tree-based system. The complexity of the system is easily observable.

In order to generate the smallest possible trees, consisting of only the states with the highest probability, the P_{thr} was set to a higher value of 0.25. In addition, TopN_{thr} parameter was introduced as the beam strength with a value of 2, which represents that only the two states with the highest probabilities are taken into consideration during the construction of the tree. With the maximal sequence length set to 9, these measures made sure that the size of the resultant tree is adequate and appropriate for evaluation.

By comparing the acquired probability trees to the tree that is defining the system's behaviour, it is observable that the seq2probTree method based on the LSTM model can capture the long-term relationship of the states of a system. Given the training accuracy as 0.86, the acquired results represent the original probability tree on which the system is based quite accurately. A few prediction errors are observable in the results. These discrepancies could be explained by pointing out that the input sequence is

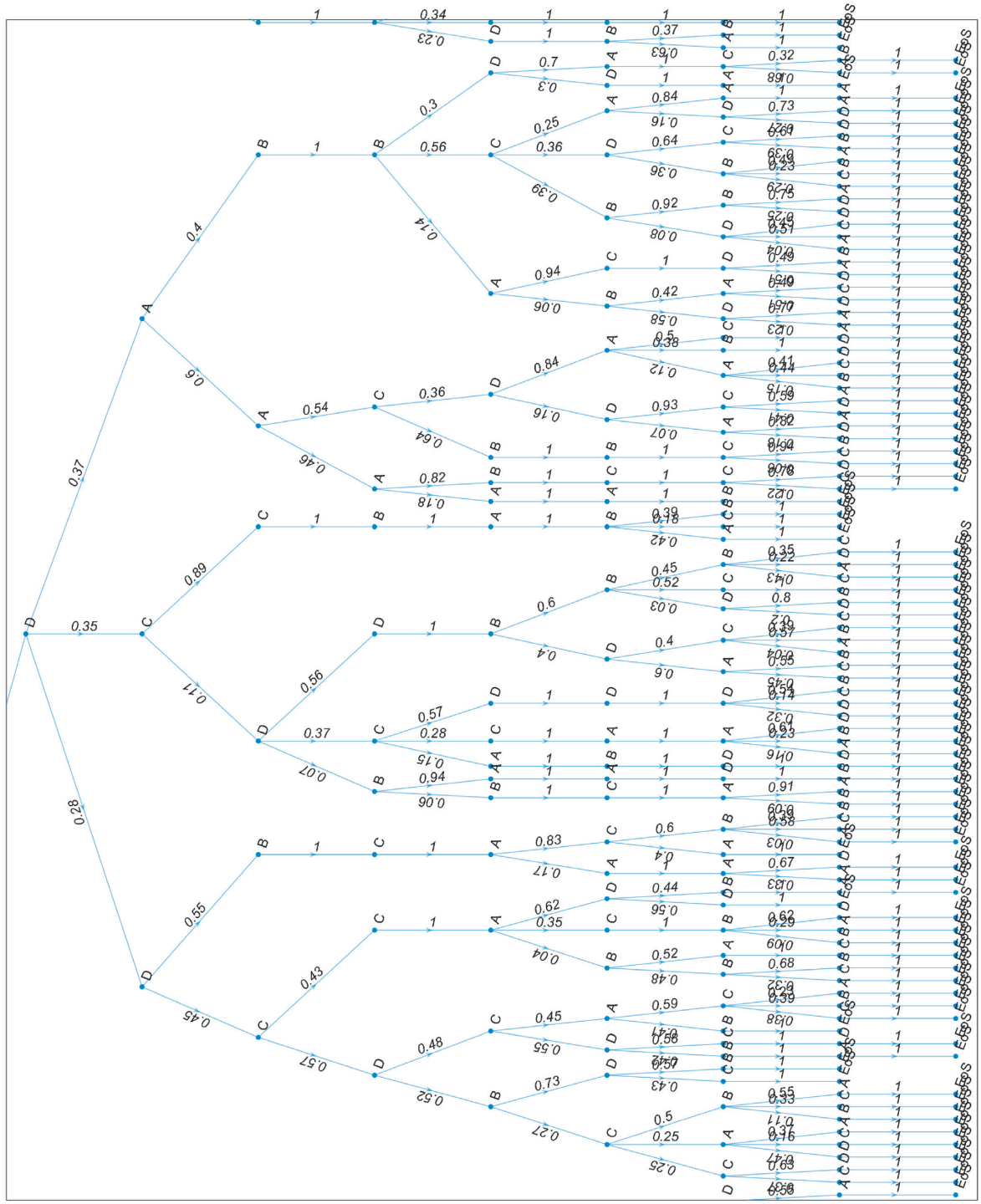


Figure 10: The state transition probabilities of the full tree-based system, assuming [D D] event history.

relatively scarce; thus, shorter patterns with high confidences can "mislead" the model.

As the seq2probTree method is proposed as a tool capable of online dynamic process supervision, the visual information it provides is crucial. While understanding the prediction tree with sparse input is an overwhelming task, as the input sequence expands with more system states, the less complex the probability tree structure becomes. Figures 16–21 represent the method's visual output, while step-by-step appending the input sequence starting from [B] to [B A B A D D] following the most probable path shown in Figure 16 (also the path of the sequence with the lowest S_{ED} metric). The results clearly show how the complexity of the acquired probability trees decreases by expanding the input sequences. The inferred state sequences are diversified by providing scarce input for the LSTM model, and a few erroneous conclusions are drawn. An excellent example for

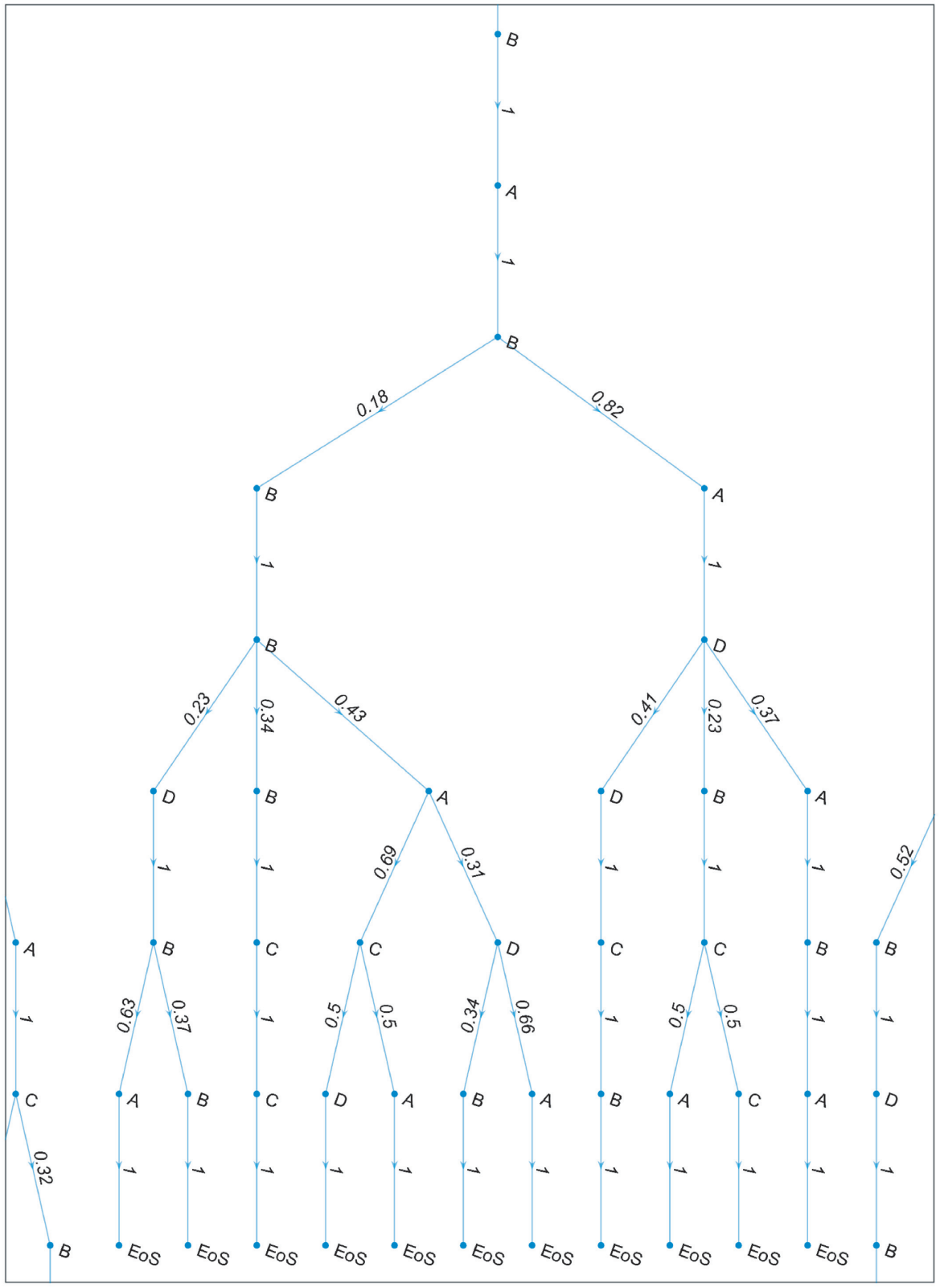


Figure 11: The state transition probabilities of the full tree-based system, assuming [B] event history.

this behavior is found in Figure 17, where after the [B A] input sequence, state A was predicted with a probability that fit P_{thr} along with state B, which should have been a sure transition.

To quantify the accuracy of the model for each aforementioned input sequence, an average error has been calculated and may be observable on Table 1. The error—just

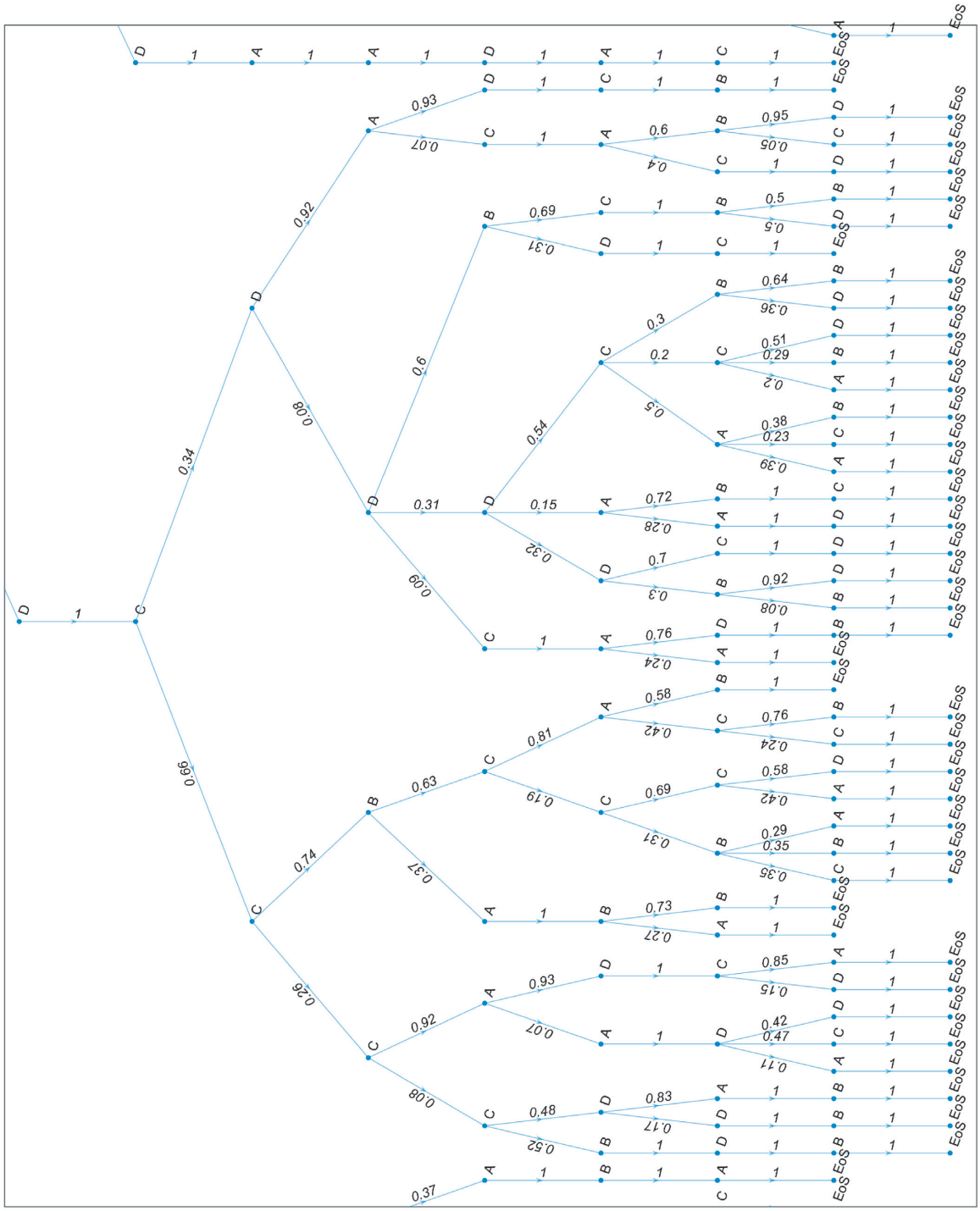


Figure 12: The state transition probabilities of the full tree-based system, assuming [A D] event history.

like the S metrics—is calculated for each predicted event sequence on the probability tree by simply counting how many elements from the end of the sequence are not found in the probability tree defining the system. The determined errors are then averaged out.

One question that arises during the utilization of the seq2probTree method is regarding the necessary length of the input sequence, after which the output is considered reasonably accurate. Table 1 gives us the idea that if the input sequence is at least four-element long, then the probability trees generated by the seq2probTree method will show no discrepancies when compared to the tree on which the behavior of the system is based. Thus, the probability trees for every possible 4-length input sequence were generated, and the aAverage errors were calculated. Since the same accuracy cannot be expected for all the input sequences—for

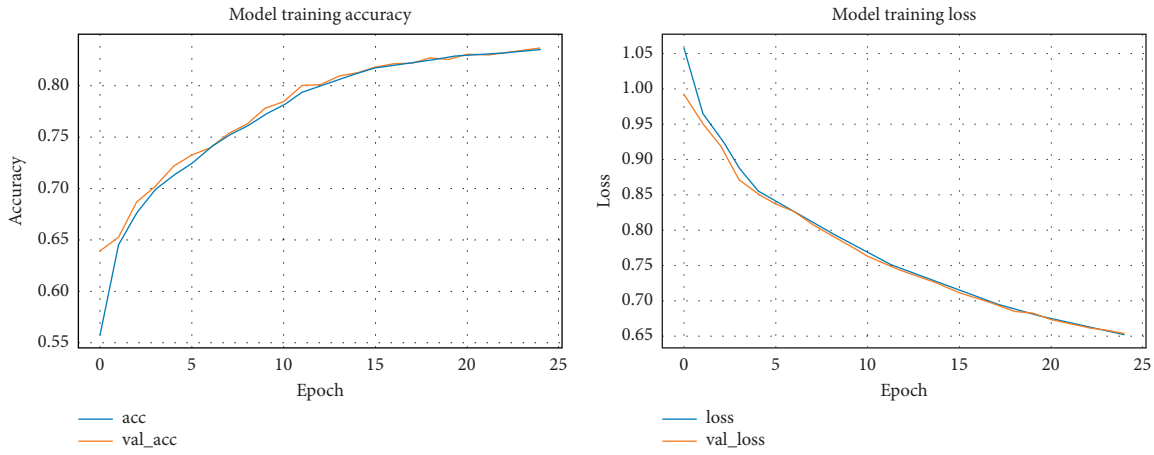


Figure 13: Training statistics on the tree-based system.

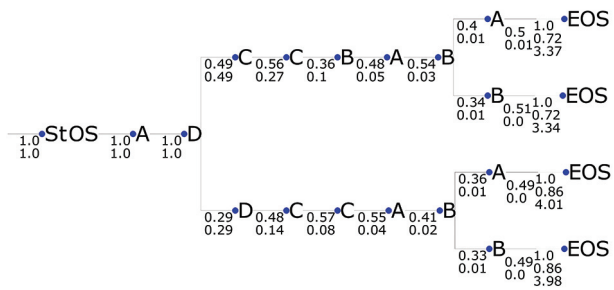


Figure 14: Output of the seq2probTree method for the tree-based system with input sequence [A D].

the sake of comparison—an additional weighting was applied to the calculated average error. The weight is calculated by determining the confidence for each input sequence and normalizing them based on the highest value. The resultant weighted average error values can be observed from Table 2. Based on the obtained results, it can be stated that after the 4 th input element, the prediction is quite precise for this example system. More significant discrepancies were observed in low confidence sequences, where even after the input sequence (several) diversions are possible.

By utilizing the seq2probTree method on this tree-based system, the capability of the algorithm for predicting higher-order event relationships has been verified with success. The average error value has been introduced to help the evaluation of the results when a direct comparison is possible with an original probability tree.

3.4. Case Study: Alarm Scenarios of a Hydrofluoric Acid Alkylation Production Unit. The proposed method has been applied to an alarm log of a hydrofluoric acid alkylation production unit to check the real-life performance. The process flow diagram of the technology can be observed in Figure 22.

The log used for this experiment was created by the operation of the production unit over a four-month-long (121 days) period, where all the incoming alarm and other events have been recorded. The unprocessed log contains precisely 200, 802 entries of which 30, 168 messages are unsuppressed alarm events. 8, 721 of these are alarms that were considered significant, thus were not shelved by the operators. The event sequences for the input of the tool were created by grouping them based on a time window while preserving their sequential temporal property. Thus, whenever a 600 sec gap is found after the last event, the two events are not considered related, and a new sequence is started. By using this strategy, the significant alarms were separated into 3, 330 sequences. Then, by considering only the event sequences with a minimum length of two, the number of valuable sequences got further reduced to 762. It is also important to note that this event database has a very high unique state count compared to the previous examples—the sequences are composed of 354 individual states. Due to confidentiality reasons, the name (the meaning) of the alarm tags has been removed.

Then, this sequence database was analyzed for frequent events that start sequences. To carry out the analysis for this case study, the four most frequent events were selected to be utilized using the seq2probTree method. The name of the selected events and their number of occurrence as the first in a sequence are highlighted in Table 3.

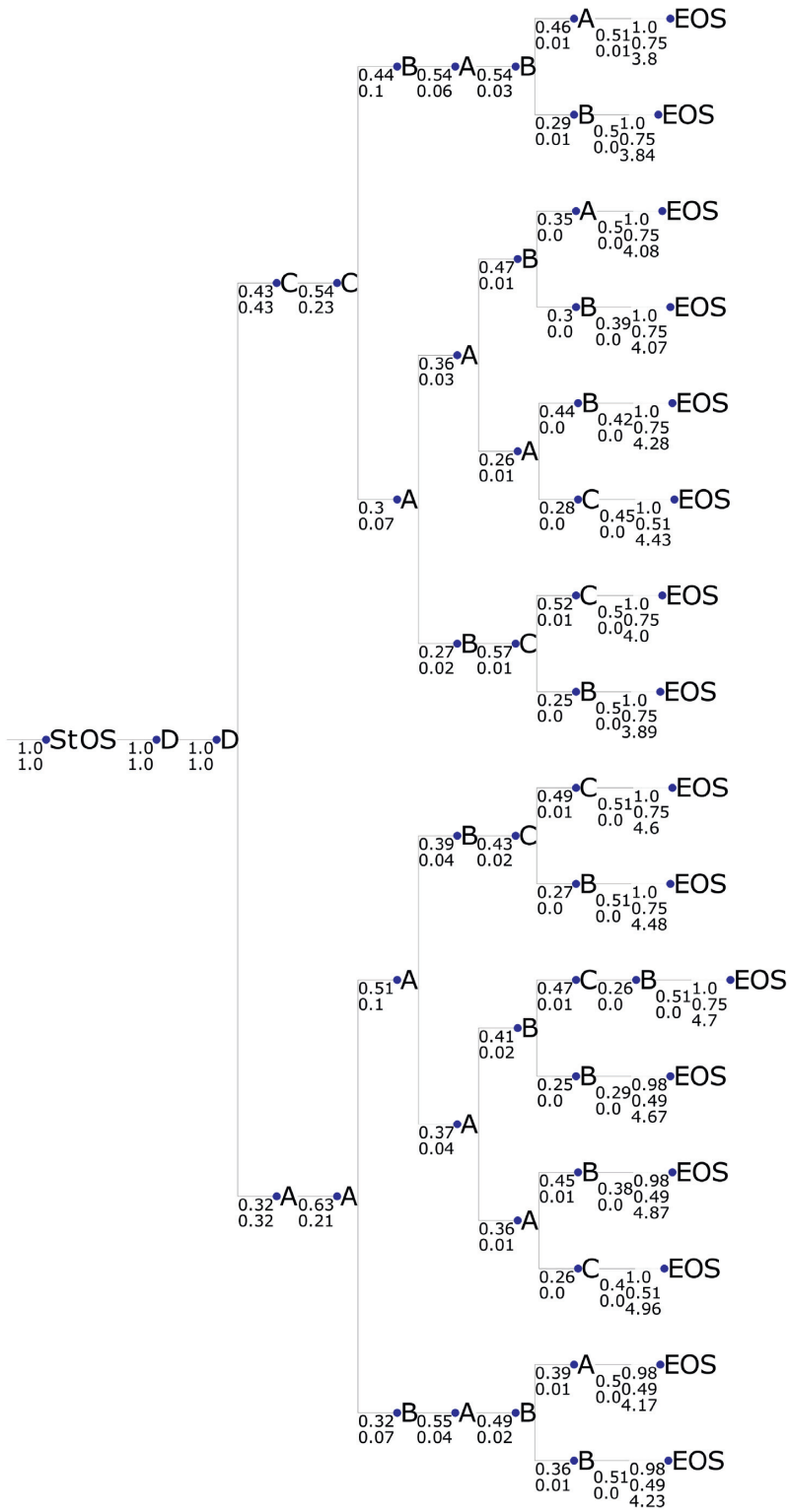


Figure 15: Output of the seq2probTree method for the tree-based system with input sequence [D D].

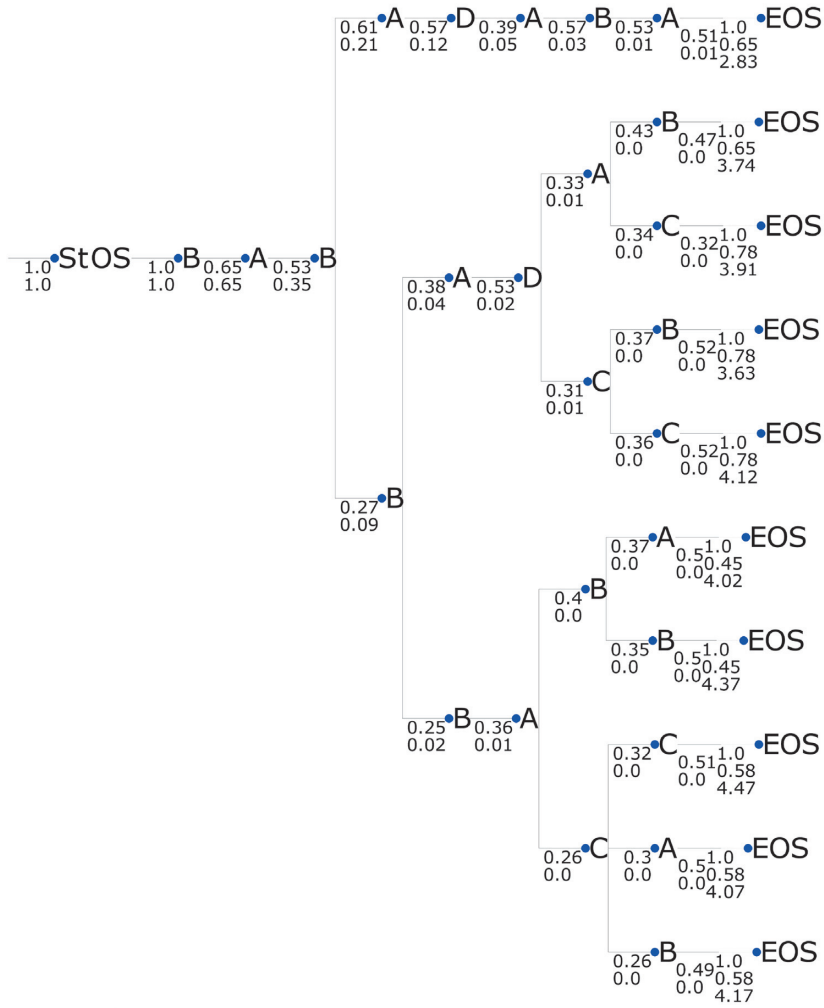


Figure 16: Output of the seq2probTree method for the tree-based system with input sequence [B].

After processing the event log, the seq2probTree method has been applied to the database. The training results from Figure 23 have been acquired by using the following LSTM and training parameters:

(i) Embedding dimensions = 5

(ii) Latent dimensions = 25

(iii) Batch size = 32

(iv) Epochs = 500

By using the aforementioned events as inputs for the seq2probTree method, the probability trees in Figures 24–27 have been created. The parameters of the beam search algorithm—P_{thr} and TopN_{thr}—were set as 0.065 and 3, respectively. Analyzing the trees, it is clear that the seq2probTree method is capable of learning and identifying the possible event scenarios. However, since the dataset is vastly diverse—especially since the seq2probTree method is also sequential position-sensitive—the probabilities of the individual transitions are pretty low; thus, the shallow P_{thr} value is justified. Moving lower with the probability threshold would have resulted in immense trees; thus, only the most frequent transitions are displayed in the figures. In Figure 27, one drawback of the method is also observed: in the longer sequences, which contain or start with [136711], often a recurring [361835] is present. This transition is so prominent that the LSTM model keeps on predicting it with a high probability. In these cases, only the defined maximal output sequence length parameter kept the seq2probTree method from creating an ever-growing branch on the tree.

Figure 24 illustrates well the different alarm sequences related to the depropanizer. The tree is initialized with the alarm message of the depropanizer pressure [136769], which can be followed by either the level alarm of one of the vessels of the depropanizer [137161] or an alarm of a pump [136711]. After the alarm on the depropanizer vessel, the alarm of the depropanizer pressure [136769] or the depropanizer feed can come in [353848].

The alarm sequences in Figure 25 are related to another scenario of the depropanizer. As can be seen, the alarm

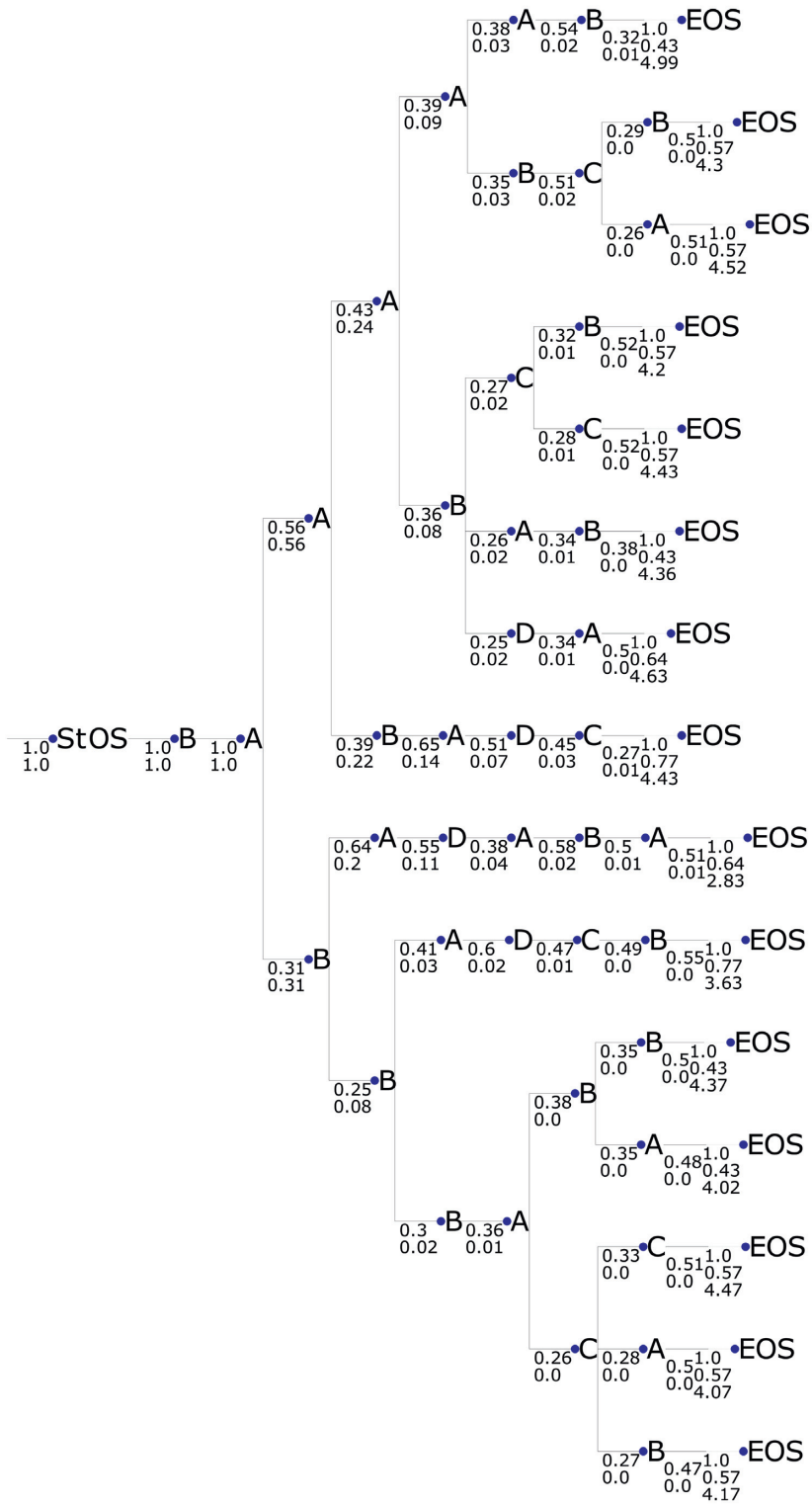


Figure 17: Output of the seq2probTree method for the tree-based system with input sequence [B A].

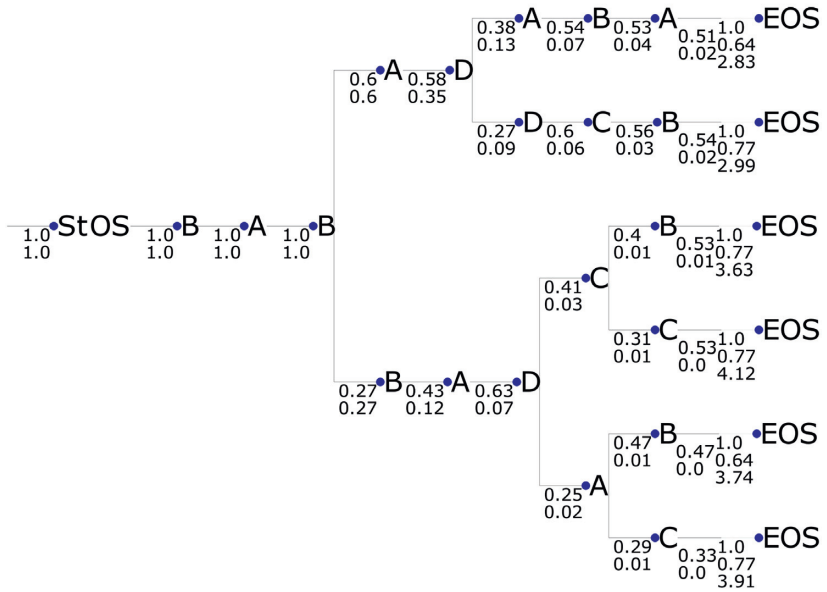


Figure 18: Output of the seq2probTree method for the tree-based system with input sequence [B A B].

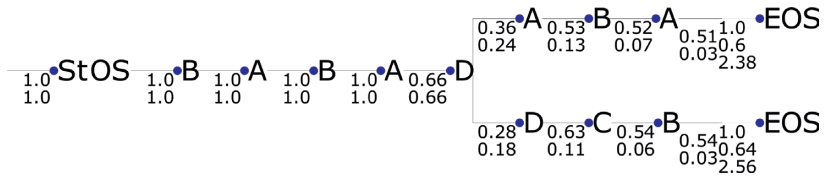


Figure 19: Output of the seq2probTree method for the tree-based system with input sequence [B A B A].

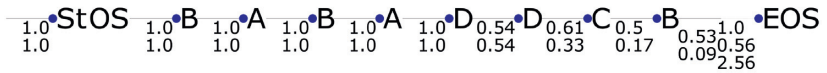


Figure 20: Output of the seq2probTree method for the tree-based system with input sequence [B A B A D].

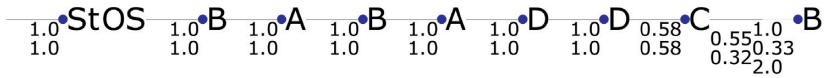


Figure 21: Output of the seq2probTree method for the tree-based system with input sequence [B A B A D D].

Table 1: The calculated average error values for different input sequences.

Input sequence	Average error
A D	3.5
D	2.5625
B	2.1
B A	3.4667
B A B	2.6667
B A B A	0
B A B A D	0
B A B A D D	0

Table 2: Prediction error statistics with every possible Φ_4' input.

SEQ	Conf.	Weight	Average error	Weighted A. E.
B A B A	0.1804	1	0	0
D C A A	0.0687	0.3808	2.8889	1.1001
D C A D	0.0013	0.0074	2.25	0.0166
D C B D	0.005	0.0279	1.8	0.0503
D D D C	0.0398	0.2207	4	0.8828
D D D B	0.0487	0.2698	0	0
D D C D	0.0122	0.0674	3.3333	0.2248
D D C C	0.0984	0.5456	2.5714	1.4031
D D A A	0.0702	0.3889	2.2222	0.8642
D D A B	0.0468	0.2592	4	1.0370
B A B B	0.0396	0.2195	1.8571	0.4077
A D C C	0.1313	0.7277	0	0
A D C D	0.0676	0.3749	0	0
A C D A	0.0879	0.4873	2.25	1.0964
A C A A	0.1032	0.572	2.4444	1.3983

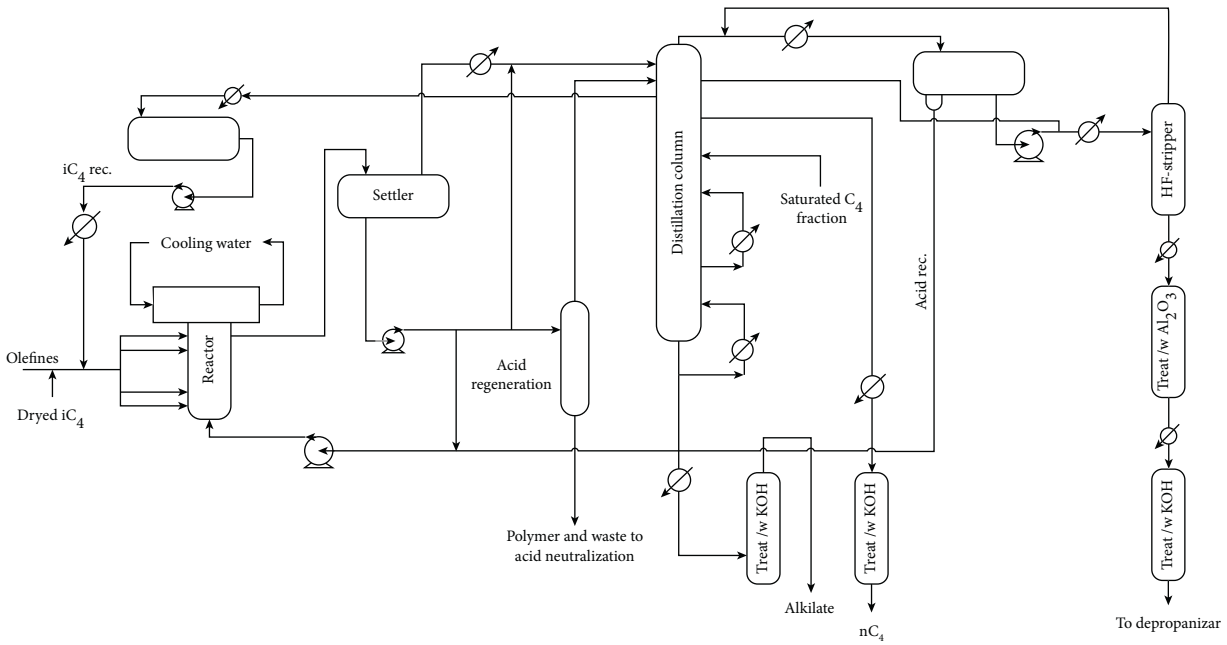


Figure 22: Process flow diagram of the hydrofluoric acid (HF) alkylation production unit.

Table 3: Occurrence statistics of the frequent sequence starting events in D_T.

Event ID	No. of occurrences
136711	127
136769	32
137438	31
137272	31

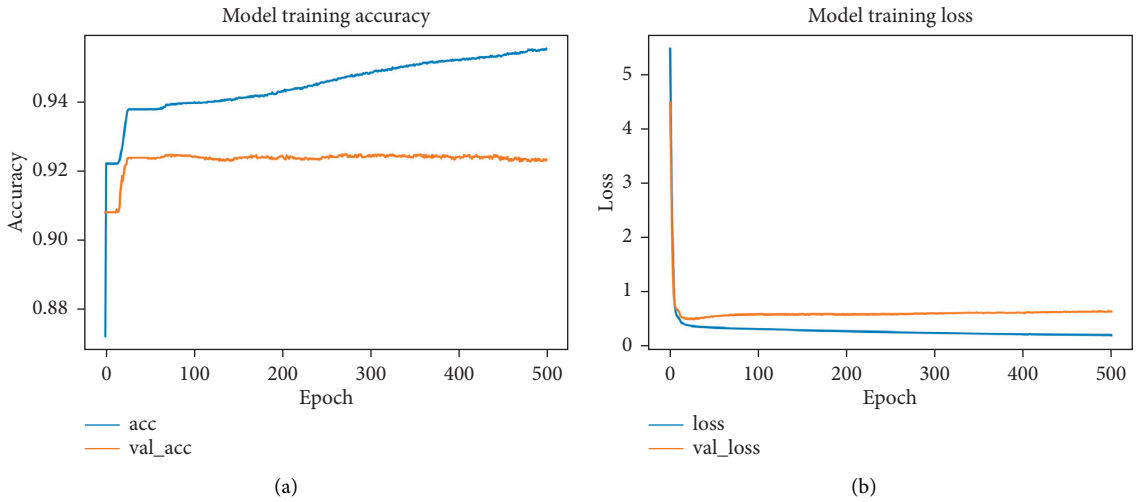


Figure 23: Training statistics on the log of the HF acid production unit.

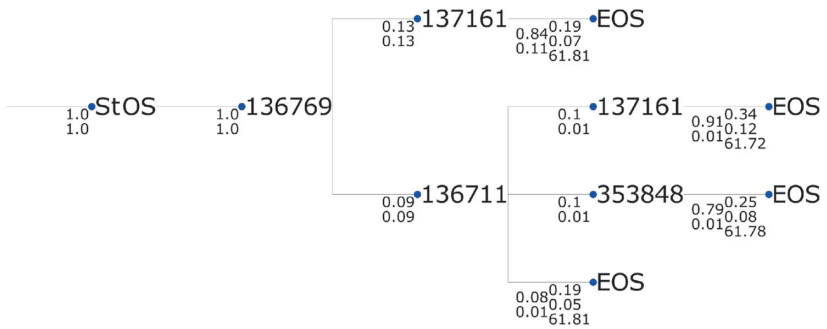


Figure 24: Output of the seq2probTree method for the HF production unit with input [136769].

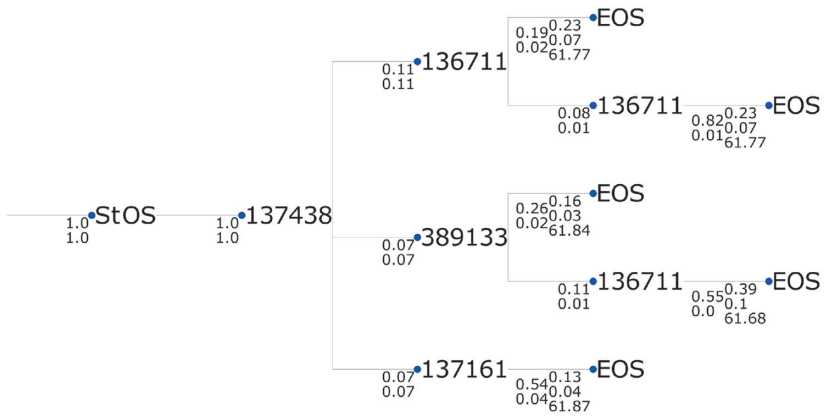


Figure 25: Output of the seq2probTree method for the HF production unit with input [137438].

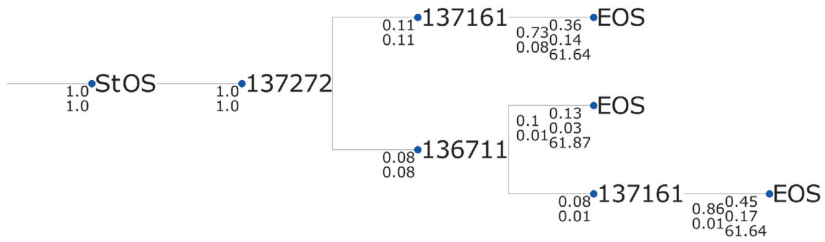


Figure 26: Output of the seq2probTree method for the HF production unit with input [137272].

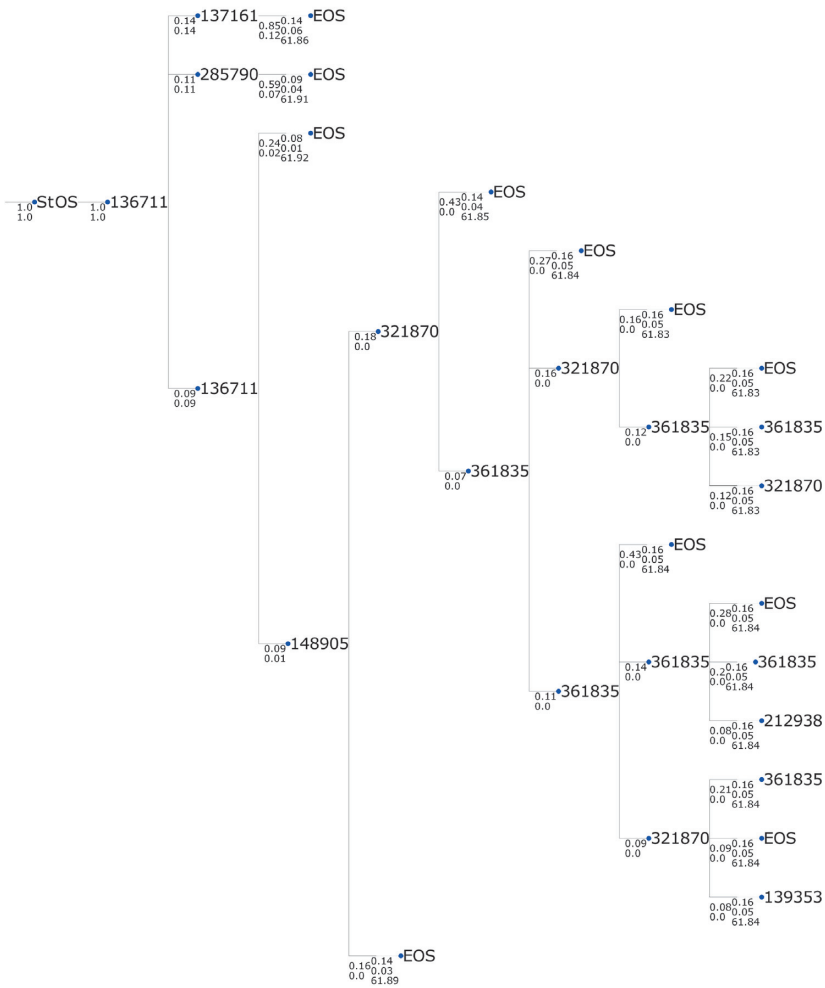


Figure 27: Output of the seq2probTree method for the HF production unit with input [136711].

message of the temperature alarm of the vessel [137438] can be followed by the pump alarm [136711] again, or the level alarm of the same vessel [137161], or another, rare alarm on a circulation pipeline.

Figure 26 is an excellent example that a problem in the bottom part of the stripper [137272] can generate a cascade of alarms on the connected units (pump and level of the vessel).

Similarly, very long alarm cascades of varying probabilities are generated in Figure 27. As we saw, the alarm of the pump [136711] reoccurs in many sequences, and, not surprisingly, it can induce the presence of several other alarms with different scenarios for the order of their occurrences.

4. Conclusion

By proposing the seq2probTree method, the application of the seq2seq learning algorithm is expanded by not only

considering the most probable item but also further exploring the alternative courses of an event sequence using the beam search algorithm during inference. This approach has been realized in Python environment by using state-of-the-art development tools.

The capability of the method has been demonstrated to reproduce the characteristics of a given system by applying it to a first-order Markov chain model. The provided transition probabilities were reasonably identified, but the approach was also capable of revealing the given unique attributes and quirks of the examined systems. The assumption that the seq2probTree method is capable of exploring higher-order relationships between events has been demonstrated and validated using a tree-based system as an example. In addition, the average error metric has been proposed to aid the user in determining the length of the input necessary for reliable prediction. Finally, the applicability of the proposed method was examined on a real-life practical example, where it produced valuable results even in the case of a highly diversified system. The proposed approach was able to map the typical alarm event scenarios and represent those in a visually interpretable manner in a hydrofluoric acid alkylation process.

Based on this evidence, it can be stated that the sequence trees created by the seq2probTree method properly represent the network of the possible alternate sequence of events. With this approach, the necessary visual output can be obtained for understanding and diagnostics of higher-order, complex systems.

Data Availability

The benchmark datasets and the code of the developed algorithms will be available on the GitHub profile and the website of the authors (https://www.abonyilab.com/) after the publication of the results.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the TKP2020-IKA-07 project financed under the 2020-4.1.1-TKP2020 Thematic Excellence Programme by the National Research, Development and Innovation Fund of Hungary. Gyula Dörgő was supported by the doctoral student scholarship program of the Co-operative Doctoral Program of the Ministry of Innovation and Technology financed from the National Research, Development, and Innovation Fund. The authors gratefully acknowledge the professional support of Ferenc Tandari who provided invaluable comments on the case study.

References

[1] D. Deitz, W. Irwin, G. Wilson et al., "Automatic linkage of process event data to a data historian," US Patent 7,275,062, 2007.

[2] I. W. Wilson and E. R. Heinzelmann, "Sequence of events recorder facility for an industrial process control environment," US Patent 7,840,285, 2010.

[3] M. Taub, R. Azevedo, A. E. Bradbury et al., "Using sequence mining to reveal the efficiency in scientific reasoning during STEM learning with a game-based learning environment," *Learning and Instruction*, vol. 54, pp. 93–103, 2018.

[4] J. S. Kinnebrew and G. Biswas, "Identifying learning behaviors by contextualizing differential sequence mining with action features and performance evolution," in *Proceedings of the International Conference on Educational Data Mining (EDM)*, Chania, Greece, June 2012.

[5] N. Béchet, P. Cellier, T. Charnois et al., "Discovering linguistic patterns using sequence mining," in *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 154–165, Springer, New Delhi, India, March 2012.

[6] R. Kant, S. H. Sengamedu, and K. S. Kumar, "Comment spam detection by sequence mining," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, pp. 183–192, Seattle, WA, USA, February 2012.

[7] Y. Fan, Y. Ye, and L. Chen, "Malicious sequential pattern mining for automatic malware detection," *Expert Systems with Applications*, vol. 52, pp. 16–25, 2016.

[8] G. Weiss, "Predicting telecommunication equipment failures from sequences of network alarms," 2001.

[9] S. Laxman, V. Tankasali, and R. W. White, "Stream prediction using a generative model based on frequent episodes in event sequences," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 453–461, Las Vegas, NV, USA, August 2008.

[10] R. Karoly and J. Abonyi, "Multi-temporal sequential pattern mining based improvement of alarm management systems," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 003870–003875, IEEE, Budapest, Hungary, October 2016.

[11] A. Belhadi, Y. Djenouri, J. C. W. Lin et al., "A general-purpose distributed pattern mining system," *Applied Intelligence*, pp. 1–16, 2020.

[12] M. d'Aquin and N. Jay, "Interpreting data mining results with linked data for learning analytics: motivation, case study and directions," in *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, pp. 155–164, New York, NY, USA, April 2013.

[13] M. El-Hajj and O. R. Zaïane, "Non-recursive generation of frequent k-itemsets from frequent pattern tree representations," in *Proceedings of the International Conference on Data Warehousing and Knowledge Discovery*, pp. 371–380, Springer, Prague, Czech Republic, September 2003.

[14] F. A. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[15] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," in *Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99*, vol. 2, Edinburgh, UK, September 1999.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 3104–3112, MIT Press, Montreal Canada, December 2014.

[17] A. Karatzoglou, A. Jablonski, and M. Beigl, "A Seq2Seq learning approach for modeling semantic trajectories and predicting the next location," in *Proceedings of the 26th ACM*

SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 528–531, Seattle, WA, USA, November 2018.

[18] J. Rebane, I. Karlsson, P. Papapetrou et al., "Seq2Seq RNNs and ARIMA models for cryptocurrency prediction: a comparative study," in *Proceedings of the SIGKDD Fintech'18*, London, UK, August 2018.

[19] T. Baumel, M. Eyal, and M. Elhadad, "Query focused abstractive summarization: incorporating query relevance, multi-document coverage, and summary length constraints into Seq2seq models," 2018, https://arxiv.org/abs/1801.07704.

[20] P. Wu, Z. Lu, Q. Zhou et al., "Bigdata logs analysis based on seq2seq networks for cognitive internet of things," *Future Generation Computer Systems*, vol. 90, pp. 477–488, 2019.

[21] S. Hwang, G. Jeon, J. Jeong et al., "A novel time series based Seq2Seq model for temperature prediction in firing furnace process," *Procedia Computer Science*, vol. 155, pp. 19–26, 2019.

[22] G. Dörgő and J. Abonyi, "Learning and predicting operation strategies by sequence mining and deep learning," *Computers & Chemical Engineering*, vol. 128, pp. 174–187, 2019.

[23] G. Dörgő, P. Pigler, M. Haragovics, and J. Abonyi, "Learning operation strategies from alarm management systems by temporal pattern mining and deep learning," in *Proceedings of the 28th European Symposium on Computer Aided Process Engineering*, A. Friedl, J. J. Klemeš, S. Radl et al., Eds., vol. 43, pp. 1003–1008, Elsevier, Amsterdam, Netherlands, 2018.

[24] B. Carter, J. Mueller, S. Jain et al., "What made you do this? understanding black-box decisions with sufficient input subsets," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics PMLR*, pp. 567–576, Naha, Okinawa, Japan, April 2019.

[25] E. Cohen and C. Beck, "Empirical analysis of beam search performance degradation in neural sequence models," in *Proceedings of the International Conference on Machine Learning PMLR*, vol. 97, pp. 1290–1299, Long Beach, CA, USA, June 2019.

[26] H. Scheidl, S. Fiel, and R. Sablatnig, "Word beam search: a connectionist temporal classification decoding algorithm," in *Proceedings of the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 253–258, IEEE, Niagara Falls, NY, USA, August 2018.

[27] Z. Li, J. Cai, S. He, and H. Zhao, "Seq2seq dependency parsing," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3203–3214, Santa Fem New Mexico, USA, August 2018.

[28] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach, and T. N. Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search," in *Proceedings of the Interspeech 2018*, pp. 2227–2231, Hyderabad, India, September 2018.

[29] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," 2017, https://arxiv.org/abs/1702.01806.

[30] D. Jahier Pagliari, F. Daghero, and M. Poncino, "Sequence-to-sequence neural networks inference on embedded processors using dynamic beam search," *Electronics*, vol. 9, no. 2, p. 337, 2020.

[31] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju et al., "Diverse beam search: decoding diverse solutions from neural sequence models," 2016, https://arxiv.org/abs/1610.02424.

[32] N. Humbatova, G. Jahangirova, G. Bavota et al., "Taxonomy of real faults in deep learning systems," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pp. 1110–1121, Seoul, South Korea, June 2020.

[33] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association*, Portland, OR, USA, September 2012.

[34] G. Dorgo, P. Pigler, and J. Abonyi, "Understanding the importance of process alarms based on the analysis of deep recurrent neural networks trained for fault isolation," *Journal of Chemometrics*, vol. 32, no. 4, Article ID e3006, 2018.

Hindawi
Complexity
Volume 2021, Article ID 9426547, 14 pages
https://doi.org/10.1155/2021/9426547

WILEY | Hindawi

Research Article

The Reciprocal Influence Criterion: An Upgrade of the Information Quality Ratio

Riccardo Rossi,[1] **Michela Gelfusa**,[1] **Filippo De Masi**,[1] **Matteo Ossidi**,[1] **and Andrea Murari**[2]

[1]*Department of Industrial Engineering, University of Rome Tor Vergata, Via del Politecnico 1, 00133 Rome, Italy*
[2]*Consorzio RFX (CNR, ENEA, INFN, Universita' di Padova, Acciaierie Venete SpA), Corso Stati Uniti 4, 35127 Padova, Italy*

Correspondence should be addressed to Riccardo Rossi; riccardo.rossi.en@gmail.com

Received 23 April 2021; Revised 12 July 2021; Accepted 3 August 2021; Published 12 August 2021

Academic Editor: M. De Aguiar

Understanding and quantifying the mutual influence between systems remain crucial but challenging tasks in any scientific enterprise. The Pearson correlation coefficient, the mutual information, and the information quality ratio are the most widely used indicators, only the last two being valid for nonlinear interactions. Given their limitations, a new criterion is proposed, the reciprocal influence criterion, which is very simple conceptually and does not make any assumption about the statistics of the stochastic variables involved. In addition to being normalised as the information quality ratio, it provides a much better resilience to noise and much higher stability to the issues related to the determination of the involved probability distribution functions. A conditional version, to counteract the effects of confounding variables, has also been developed, showing the same advantages compared to the more traditional indicators. A series of systematic tests with numerical examples is reported, to compare the properties of the new indicator with the more traditional ones, proving its clear superiority in practically all respects.

1. Determining Statistical Dependence between Quantities

The investigation of the reciprocal influence between systems is a major scientific objective in practically all disciplines. Both linear and nonlinear effects can be important, the latter becoming often dominant in complex phenomena [1]. The information about systems is typically obtained with measurements affected by various forms of uncertainties. The measurands can therefore be considered as random variables, and their mutual influence is partly deterministic and partly probabilistic in nature.

Quantifying the dependence between random variables is an essential statistical task for both bivariate and multivariate data. The most popular measure of dependence between two quantities is the Pearson product-moment correlation coefficient or "Pearson's correlation coefficient" (PCC) [2]. To alleviate its limitations (see Section 2), at the beginning of the last century, various alternative indicators based on ranked variables were developed, such as the Spearman ρ or Spearman rank correlation coefficient [3] and the rank correlation coefficient or Kendall τ [4]. After the Second World War information theoretic criteria, such as mutual information, became quite popular [5]. At the dawn of the new century, a lot of work was devoted to consolidating distance correlation, an indicator that is zero only when the two variables are not correlated [6]. In the last years, the advances in genetics have also motivated the development of various techniques for the analysis of the dependence between vectors in high dimensions. For this multivariate inference, very sophisticated approaches have been proposed, ranging from projection correlation [7] to Ball covariance [8] and Brownian distance [9]. All these techniques are based on specific assumptions about the statistical properties of the vectors of random variables to consider. They are also quite involved, both conceptually and numerically.

The indicators proposed in this paper are meant to deal only with bivariate cases, for which no prior information is known about the statistics of the data, so fully general

techniques are required. A conditional version to counteract the effects of confounding variables has also been developed. The indicators are also very simple from the point of view of the conceptual background, interpretation of the results, and implementation requirements.

Regarding the structure of the paper, Section 2 introduces the most consolidated indicators, typically used to assess the dependence between two variables. Section 3 discusses the main rationale behind the proposed alternative indicator: the reciprocal influence criterion. The systematic tests, comparing the properties of RIC with PCC and IQR, are reported in Sections 4 and 5, for functional and nonfunctional dependencies, respectively. The robustness against the noise of statistics, different from the Gaussian and against outliers, is discussed in Section 6. The conditional version of RIC is introduced in Section 7, before conclusions are drawn in Section 8.

2. Traditional and Alternative Criteria to Quantify the Correlations between Quantities

In order to fully appreciate the potential of the new indicators proposed in the paper, it is worth starting the discussion with a review of the traditional and alternative tools most used by practitioners for the analysis of bivariate dependence. The linear correlations between two quantities are typically calculated with the Pearson correlation coefficient (PCC). For a couple of random variables X and Y, indicating with cov as the covariance and with σ as the standard deviation, the PCC is calculated as [10]

$$\mathrm{PCC} = \frac{\mathrm{cov}(X, Y)}{\sigma_x \sigma_y}. \tag{1}$$

Even if it is very useful in practice, some limitations of the PCC, particularly, its vulnerability to noise, are very often overlooked. In any case, the detection of nonlinear correlations between two variables is a much more serious matter. To quantify nonlinear dependencies, historically, the first techniques developed were based on ranking variables (see Section 2.1). After the Second World War, indicators based on the probability distribution function (pdf) of the data have been developed; particularly, information theoretic criteria are popular (Section 2.2). Distance correlation is a much more recent development, first introduced in 2005, to remedy the problem that the Pearson correlation coefficient can be zero for dependent variables (Section 2.3).

2.1. Rank-Based Criteria. In this paper, the symbol ρ_S indicates the extension of the Pearson correlation coefficient introduced by Charles Spearman. Spearman's ρ_S is a nonparametric measure of the statistical dependence between the rankings of two variables; as such, it quantifies how well the relationship between two variables can be modelled by a monotonic function [11]. Spearman's ρ_S is calculated as the Pearson correlation between the rank values of the quantities involved and assesses how monotonic the relation between them is (whether linear or not). If there are no repeated data values, ρ_S assumes the values +1 or −1 when the two variables are perfect monotone functions of each other.

To calculate Spearman's ρ_S, the raw data X_i and Y_i are converted into ranks first: rank_X and rank_Y; the standard deviations of these two ranked variables are indicated with the symbols σ_{rank_X} and σ_{rank_Y}. Spearman's ρ_S is then defined as

$$\rho_S = \frac{\mathrm{cov}\left(\mathrm{rank}_x, \mathrm{rank}_y\right)}{\sigma_{\mathrm{rank}_x} \sigma_{\mathrm{rank}_y}}. \tag{2}$$

The Kendall rank correlation coefficient, named after Maurice Kendall, was developed at the end of the 30 s, and it is usually indicated with the Greek letter τ. It is meant to measure the rank correlation, i.e., the similarity of the ordering of the data when ranked. Intuitively, the higher the Kendall correlation between two variables, the more similar their rank; the Kendall τ is also normalised in the sense that it ranges between 1 and −1 [11].

Mathematically, the Kendall rank correlation coefficient is calculated as

$$\tau = \frac{(\text{Number of concordant pairs}) - (\text{Number of discordant pairs})}{\binom{n}{2}}, \tag{3}$$

where the denominator is the binomial coefficient $\binom{n}{2} = n(n-1)/2$.

2.2. Information Theoretic Criteria. A widely used indicator to investigate nonlinear interactions is the mutual information, which quantifies the information shared by two systems. The Shannon or discrete version is defined as [12, 13]

$$\mathrm{MI}_{\mathrm{Shan}} = \sum \sum \left(p_{xy} \log \left(\frac{p_{xy}}{p_x p_y} \right) \right), \tag{4}$$

where p_x and p_y are the discrete probabilities of two random variables X and Y and p_{xy} is their joint probability.

A differential version of the indicator can be formulated in terms of the probability densities f_x and f_y:

$$f_x = \frac{p_x}{\Delta x},$$
$$f_y = \frac{p_y}{\Delta y}, \quad (5)$$
$$f_{xy} = \frac{p_{xy}}{\Delta x\,\Delta y},$$

where Δx and Δy are the dimensions of the bins. The differential mutual information, $\mathrm{MI}_{\mathrm{diff}}$, is defined as

$$\mathrm{MI}_{\mathrm{Diff}} = \int\int\left(f_{xy}\log\left(\frac{f_{xy}}{f_x f_y}\right)\mathrm{d}x\,\mathrm{d}y\right). \quad (6)$$

It can be easily demonstrated that the two relations (4) and (6) are equivalent; the mutual information is, therefore, referred to as MI in the rest of the paper. MI has various positive properties but presents the main limitation of not being normalised. To obviate this drawback, the mutual information is typically divided by the joint entropy to obtain the so-called information quality ratio (IQR) [14]:

$$\mathrm{IQR} = \frac{\mathrm{MI}}{H(X,Y)}. \quad (7)$$

This quantity is normalised in the sense that it assumes only values between zero and one. It is important to remember that, in equation (7), the discrete or Shannon version of the joint entropy is to be used:

$$H_{S,XY} = -\sum\sum p_{xy}\log\left(p_{xy}\right). \quad (8)$$

The Shannon entropy is to be compared with the differential one, which can also assume negative values:

$$H_{D,XY} = -\int\int\left(f_{xy}\log\left(f_{xy}\right)\mathrm{d}x\mathrm{d}y\right). \quad (9)$$

Equation (7) is the commonly accepted version of the information quality ratio, normally adopted because the differential version of the joint entropy can be negative, with the obvious related problems and difficulties.

Unfortunately, mainly due to the denominator, the IQR_S has some limitations, particularly, a lack of robustness to noise and a strong dependence on the choice of the bins.

2.3. Distance Correlation. The objective of distance correlation (D_{corr}) consists of quantifying the dependence between two random vectors, which do not need to have necessarily equal dimension. D_{corr} has the clear advantage, compared to the PCC, that the population distance correlation coefficient assumes a zero value only if the two random vectors are independent. Therefore, distance correlation is meant to quantify both linear and nonlinear association between two random variables or random vectors [5].

The calculation of D_{corr} requires the definition of some other preliminary quantities. Indicating with (X_k, Y_k), $k = 1, 2, \ldots, n$, a sample from a pair of real-valued or vector-valued random variables (X, Y), the elements of the n by n distance matrices $(a_{j,\,k})$ and $(b_{j,\,k})$ are all pairwise distances:

$$a_{j,k} = \left\|X_j - X_k\right\|, \quad j,k = 1,2,\ldots,n,$$
$$b_{j,k} = \left\|Y_j - Y_k\right\|, \quad j,k = 1,2,\ldots,n, \quad (10)$$

where $\|\ \|$ denotes the Euclidean norm. Defining $\overline{a}_j$ and $\overline{a}_k$ as the jth row mean and the kth column mean, respectively, and with $\overline{a}$ the grand mean of the first vector mutual distance X matrix, one can then calculate all doubly centred distances (with the same notation for the Y vector matrix):

$$A_{j,k} = a_{j,k} - \overline{a}_j - \overline{a}_k + \overline{a},$$
$$B_{j,k} = b_{j,k} - \overline{b}_j - \overline{b}_k + \overline{b}. \quad (11)$$

The desired sample distance covariance is then simply the arithmetic average of the product $A_{j,k}B_{j,k}$:

$$d\mathrm{Cov}_n^2(X,Y) := \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n}A_{j,k}\,B_{j,k}. \quad (12)$$

Indicating with distance variance,

$$d\mathrm{Var}_n^2(X,X) := \frac{1}{n^2}\sum_{k,l=1}^{n}A_{k,l}^2, \quad (13)$$

finally, the distance correlation is

$$d\mathrm{Cor}(X,Y) := \frac{d\mathrm{Cov}(X,Y)}{\sqrt{d\mathrm{Var}(X)\,d\mathrm{Var}(Y)}}. \quad (14)$$

The main properties of dCor are that it assumes values between 0 and 1, and it is zero only if the two vectors are independent. The distance correlation software used in this work is the one published by Shen Liu [15].

3. The Reciprocal Influence Criterion: Rationale

As will be shown in the rest of the paper, all criteria summarised in the previous section have several drawbacks. In this work, a new indicator, able to quantify linear and nonlinear correlations between variables, is introduced. The new indicator, based on information theoretic quantities and named reciprocal influence criterion (RIC), has been designed to have the following properties.

3.1. Property 1. The indicator ranges from zero (no correlation) to one (perfectly correlated). The definition of correlation for this indicator is the following.

Two variables i and j are correlated when the knowledge of i helps to predict j and *vice versa*. The correlation indicator tends to one when the uncertainty of i (j) known j (i) goes to zero. Correlation must be equal to zero when the uncertainty of i (j) does not change when the j (i) is known. Note that, in this definition, symmetric equations cannot reach the value of one; as, for example, for the function $y = x^2$, since y is known, there are two valid values of x (excluded $x = 0$ and $y = 0$).

Table 1: The value of the various indicators for the cases of Figure 1.

	$y = x$	$y = x^2$	$y = e^x$	$y = \sin(3x)$
Pearson	0.99	0.03	0.93	0.01
Spearman	0.99	0.03	0.98	0.03
Kendall	0.93	0.02	0.90	0.01
Distance correlation	0.99	0.49	0.96	0.35
IQR	0.47	0.35	0.46	0.33
RIC	0.97	0.93	0.96	0.90

3.2. Property 2. The indicator does not vary as a function of the binning used for the probability density or distribution function calculations, i.e.,

$$\frac{\Delta N_{\text{bin}}}{N_{\text{bin}}} \gg \frac{\Delta \text{RIC}}{\text{RIC}}, \tag{15}$$

where ΔRIC denotes the difference between the indicator values calculated with and without the outliers.

3.3. Property 3. The indicator can be calculated indifferently for either discrete or continuous random variables.

3.4. Property 4. Small sensitivity to outliers is quantified as

$$\frac{N_{\text{outliers}}}{N} \gg \frac{\Delta \text{RIC}}{\text{RIC}}. \tag{16}$$

The formulation of the RIC that ensures the former properties is

$$\text{RIC} = 1 - \frac{A^{H_{xy}}}{A^{H_x+H_y}} = 1 - \frac{A^{H_{xy}}}{A^{H_x+H_y}} = 1 - \frac{1}{A^{\text{MI}}}. \tag{17}$$

The RIC indicator is based on the ratio between the amount of information (uncertainty) shared by the two variables with respect to the sum of their individual information (uncertainty). In some ways, its definition is very similar to the IQR indicator, with the main difference that the RIC is based on the exponential of the entropy A^H. This approach allows obtaining a much more reliable indicator, which satisfies the aforementioned four desirable properties, as demonstrated in the following, and presents some additional positive qualities, namely, symmetry, asymptotic consistency, and unbiasedness, which are discussed in Appendix A.

With regard to the free parameter A, the choice of its numerical value can be optimised, depending on the nature of the data and the objectives of the investigation. The results, reported in the rest of the paper, have been obtained by setting $A = 10$, to maximise the coherence of RIC with PCC in case of linear correlations; the equations in the following are also particularised for this numerical value. A detailed discussion of the RIC behaviour with the parameter A is provided in Appendix B.

Some of the aforementioned four properties can be analytically demonstrated. First, property 1 can be proven as follows. For not correlated variables, $H_x + H_y = H_{xy}$ and $\text{MI} \longrightarrow 0$; thus,

$$\lim_{\text{MI}\longrightarrow 0} \text{RIC} = \lim_{\text{MI}\longrightarrow 0} 1 - \frac{10^{H_{D,xy}}}{10^{H_{D,x}+H_{D,y}}} = \lim_{\text{MI}\longrightarrow 0} 1 - \frac{1}{10^{\text{MI}}} = 0. \tag{18}$$

In the case of partially correlated quantities, RIC is always positive, and for high mutual information, it tends to one. More specifically, it is known that, for high correlation levels, the joint entropy tends to the lowest entropy of the two variables ($H_{xy} \longrightarrow \min(H_x, H_y)$) [16], which means that $\text{MI} \longrightarrow \max(H_x, H_y)$. Assuming $H_y > H_x$, it follows

$$\lim_{H_{xy}\longrightarrow H_x} RIC = \lim_{H_{xy}\longrightarrow H_x} 1 - \frac{10^{H_{xy}}}{10^{H_x+H_y}} = \lim_{H_{xy}\longrightarrow H_x} 1 - \frac{10^{H_x}}{10^{H_x+H_y}} = \lim_{H_{xy}\longrightarrow H_x} 1 - \frac{1}{10^{H_y}} > 0. \tag{19}$$

Property 3 can also be proved. Indeed, writing the RIC using the differential information theoretic quantities and remembering that the sum of the entropies is equal to the sum of the mutual information and the joint entropy, it is possible to demonstrate that RIC_D is the same as RIC_S:

$$\text{RIC}_D = 1 - \frac{10^{H_{D,xy}}}{10^{H_{D,x}+H_{D,y}}} = 1 - \frac{1}{10^{MI}} = 1 - \frac{10^{H_{S,xy}}}{10^{H_{S,x}+H_{S,y}}} = \text{RIC}_S. \tag{20}$$

Moreover, RIC is directly correlated to the mutual information value; also, property 2 is expected to be satisfied (and it is validated in Section 4).

In terms of interpretation, RIC remains an information theoretic indicator, since it relies on the mutual information. On the contrary, the monotonic transformation of (15) provides a practically normalised indicator without having to make recourse to the joint entropy. The advantages of such a reformulation will be illustrated in detail, with the help of numerical tests, in the next sections.

4. The Reciprocal Influence Criterion: Numerical Tests for Functional Dependencies

In this section, the properties of RIC are investigated with the help of a series of numerical tests with synthetic data. The reference indicators to benchmark the performance of RIC are the ones reviewed in Section 2, the most used by the practitioners. Only bivariate dependences due to functional relations are considered; nonfunctional dependencies are the subject of Section 4.1. A general overview of the results is

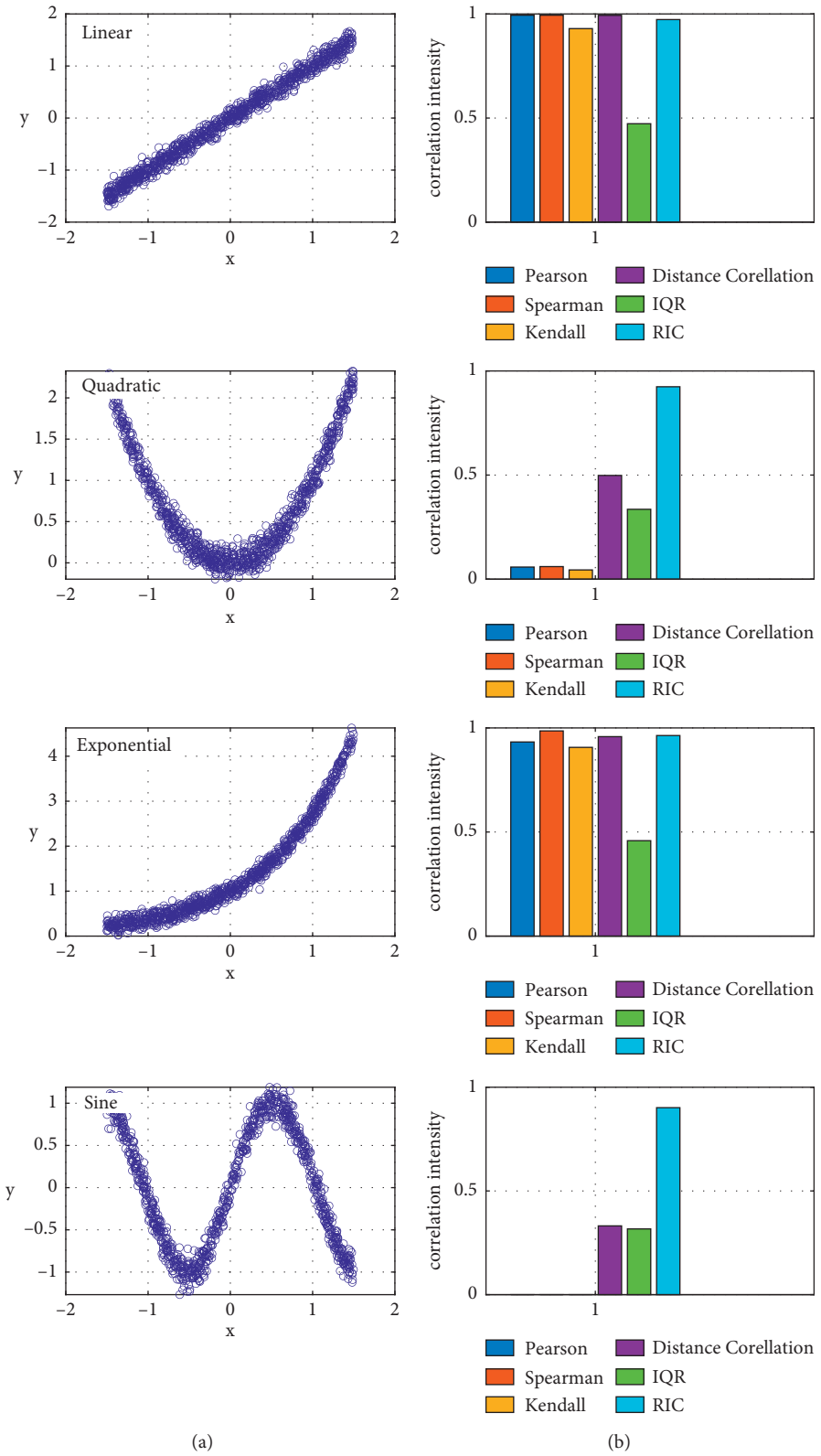


Figure 1: The main classes of functions tested. (a) Examples of synthetic data. (b) The numerical values of the indicators.

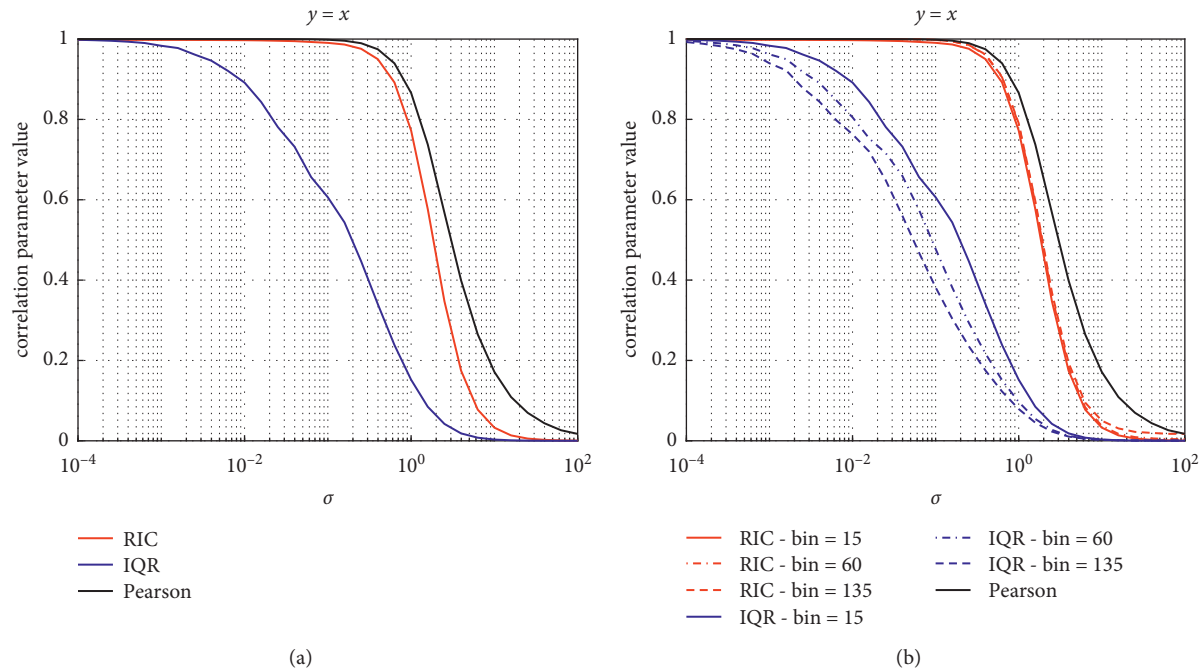


Figure 2: (a) Behavior of PCC, IQR, and RIC with the standard deviation of the additive noise for linear correlations between two variables: $y = x$. (b) Behavior of PCC, IQR, and RIC with the standard deviation of the additive noise when varying the binning.

provided first and some specific aspects are discussed in dedicated sections.

4.1. Comparison Overview. A first comparison between the RIC and the other criteria has been performed for the functional dependencies linear, quadratic, sinusoidal, and exponential. The results for a series of representative cases are reported in Table 1. A graphical overview is provided in Figure 1. Random noise of Gaussian distribution, with standard deviation equal to 10% of the quantity value, has been added to all variables.

The values of Table 1 and inspection of Figure 1 reveal that RIC never performs significantly worse than the other criteria for linear dependencies. RIC starts outperforming all other indicators in the case of nonlinear functions. Moreover, as expected, RIC provides much more reliable and reasonable results whenever the functional dependence is nonmonotonic and when even the ranked methods fail miserably. Also, the indicators based on the pdf, IQR, and distance correlation show significant difficulties to provide acceptable results for the nonmonotonic dependencies. All these are general properties not only true for the examples reported but also confirmed in all cases tested.

4.2. Linear Correlations: Effects of Gaussian Noise and Binning. This section, with the help of Figure 2, is simply aimed at supporting the statement that RIC can reproduce well the values of the PCC for linear correlations. The behaviour of the other indicators is also shown for completeness. The plots of Figure 2 refer to the case of perfect linear correlation between quantities: $y = x$. On the x-axis, the standard deviation of additive noise and zero mean and sampled randomly from a Gaussian distribution is reported. The results are fully general. The RIC reproduces quite well the values of the PCC, whereas the IQR is significantly more vulnerable to both noise and binning. A similar analysis indicates that RIC is also more robust against the presence of outliers; indeed, it can tolerate about even one order of magnitude more outliers than IQR, confirming that property 4 of Section 3 is satisfied (with 5% of outliers, the average Pearson coefficient variation is about 10% and the IQR variation is 12%, while the ΔRIC is 0.7%).

4.3. Nonlinear Correlations: Effects of Gaussian Noise and Binning. The competitive advantages of RIC become even more evident in the case of nonlinear correlations. Three exemplificative cases are reported in Figure 3, in which the proposed new criterion is compared with IQR_S. The functional dependencies reported are $y = x^2$, $y = \sin(x)$, and $y = \exp(x)$.

As expected, IQR is much more sensitive to the choice of the binning and the level of noise. RIC remains stable at a value very close to 1 for a much wider range of these factors. Moreover, IQR does not output a value of 1 even for perfect correlation between the two variables. This is a consequence of the denominator not being a normalised quantity. Again, also in the case of nonlinear correlations, RIC provides much more consistent results also in the presence of a significant number of outliers (the comparative resilience is similar to the case of linear correlations).

The positive qualities of RIC, compared to PCC and IQR, are not a negligible matter in practice because, in real-life applications, the effects of the noise and the uncertainties

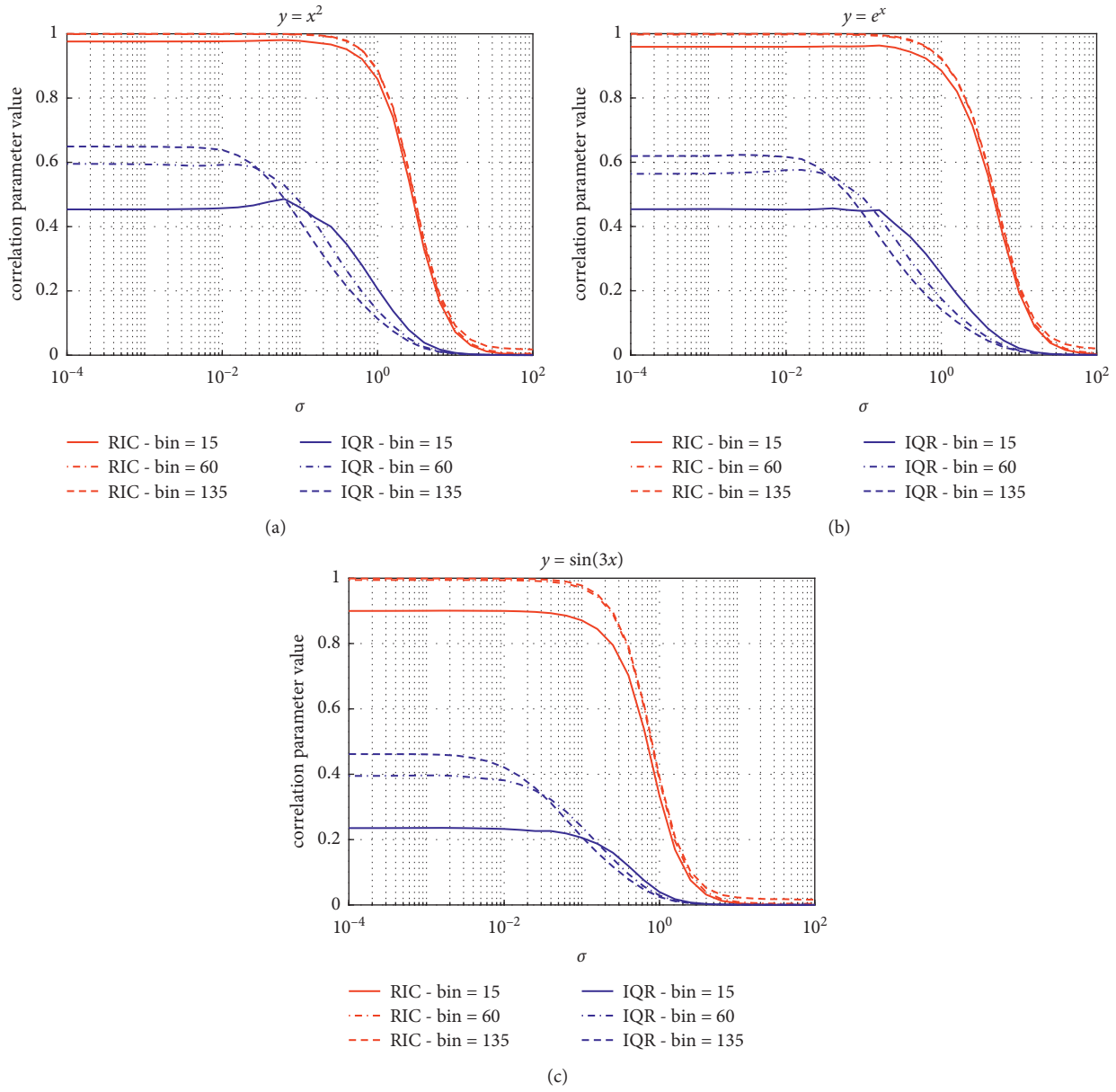


Figure 3: Behavior of PCC, IQR, and RIC with the standard deviation of the additive noise for various nonlinear dependencies. (a) $y = x2$. (b) $y = \sin(x)$. (c) $y = \exp(x)$.

about the details of the pdfs can have a strong effect on the conclusions.

It should be noted that, from the analysis performed as a function of the binning, it is clear that property 2 of Section 3 is satisfied, i.e., $\Delta N_{\text{bin}}/N_{\text{bin}} \gg \Delta\text{RIC}/\text{RIC}$.

5. The Reciprocal Influence Criterion: Numerical Tests for Nonfunctional Dependencies

The cases treated in this section, to exemplify the properties of RIC for nonfunctional dependencies (with additive Gaussian noise of mean equal to zero and standard deviation equal to 0.1), are shown in Figure 4. These types of dependencies are quite involved and difficult to resolve. They are fully nonlinear and they cannot even be represented by functions. For all these cases, RIC performs significantly better than all other indicators. A synthetic overview of the results is reported in Table 2.

Inspection of Table 2 and Figure 4 reveals that the RIC criterion is always higher than the others by a factor. A part of the case of the rhomboid dependence always provides a value of 0.8 or higher, whereas the other indicators are closer to zero. The RIC, therefore, provides a much more reliable indication that there is a strong correlation between the two variables involved. Even the two other most sophisticated criteria, the IQR and distance correlation, perform significantly worse for all the examples investigated.

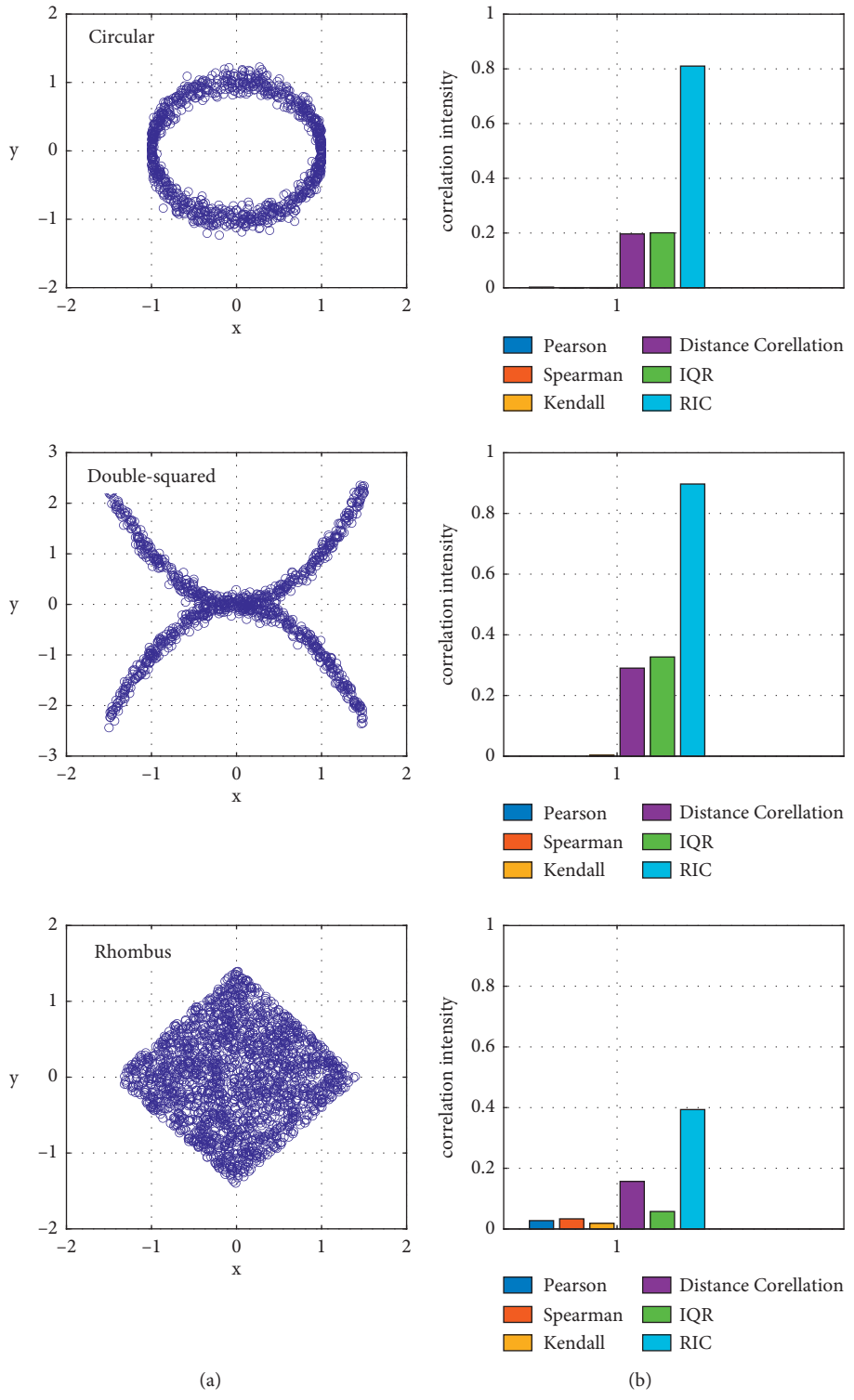


Figure 4: Main examples of the main functional dependencies investigated. (a) Examples of synthetic data. (b) The numerical values of the indicators.

6. Robustness to Noise of Different Statistics and Outliers

The signals and data acquired in many scientific disciplines are typically affected by noise. The assumption of Gaussian statistics is often justified, but there are also other important types of noise of great practical and theoretical importance. Two of the most relevant distributions are certainly the Poisson and gamma.

Poisson distribution:

Table 2: The value of the various indicators for the case of Figure 4.

	Circular	Double squared	Rhombus
Pearson	0.00	0.00	0.03
Spearman	0.00	0.00	0.04
Kendall	0.00	0.00	0.02
Distance correlation	0.19	0.31	0.15
IQR	0.21	0.31	0.05
RIC	0.80	0.89	0.38

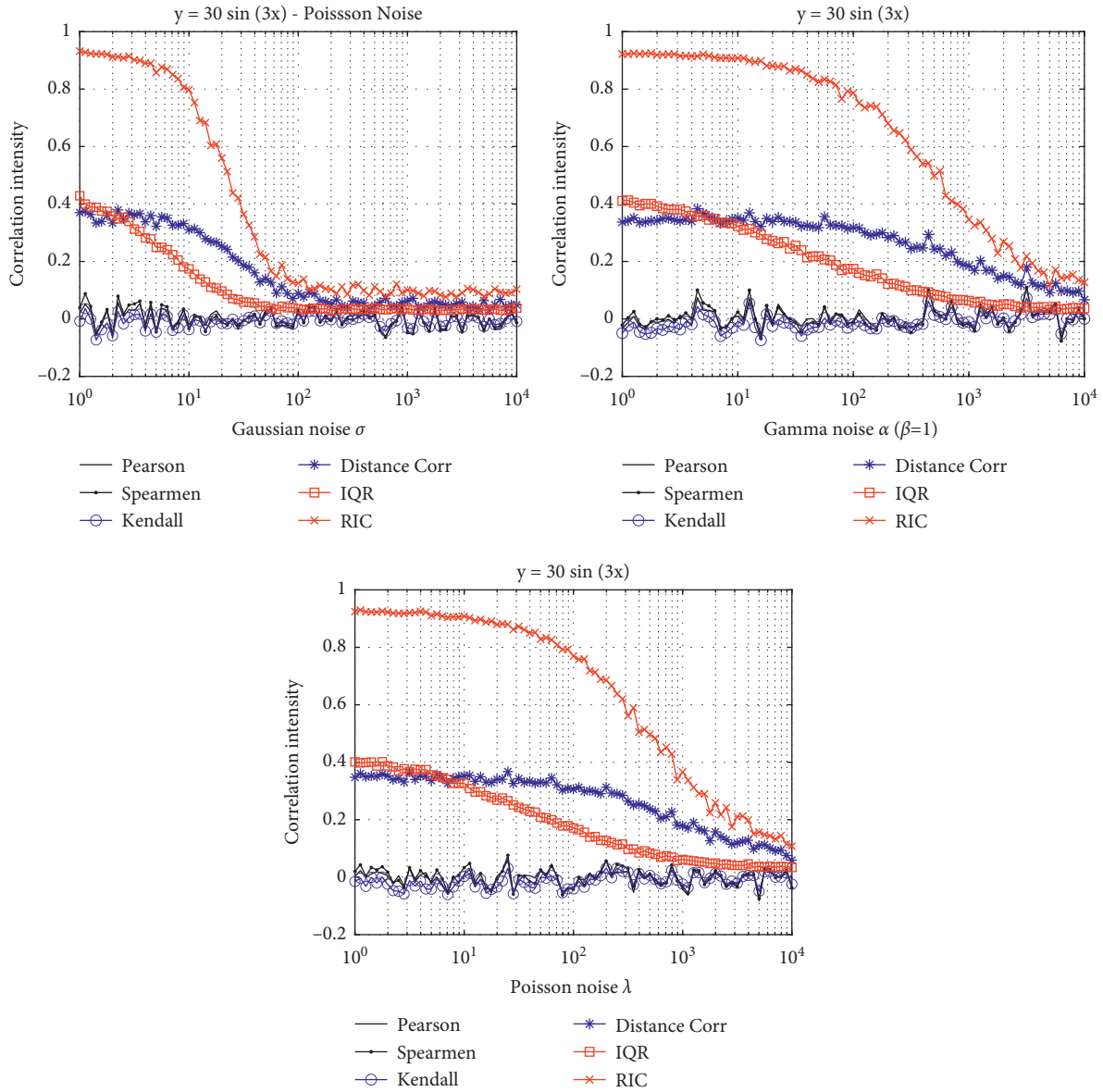


Figure 5: Comparison of RIC and the other criteria for nonlinear correlations, $y = 30 \sin(3x)$, in which the signals are affected by additive noise of different statistics.

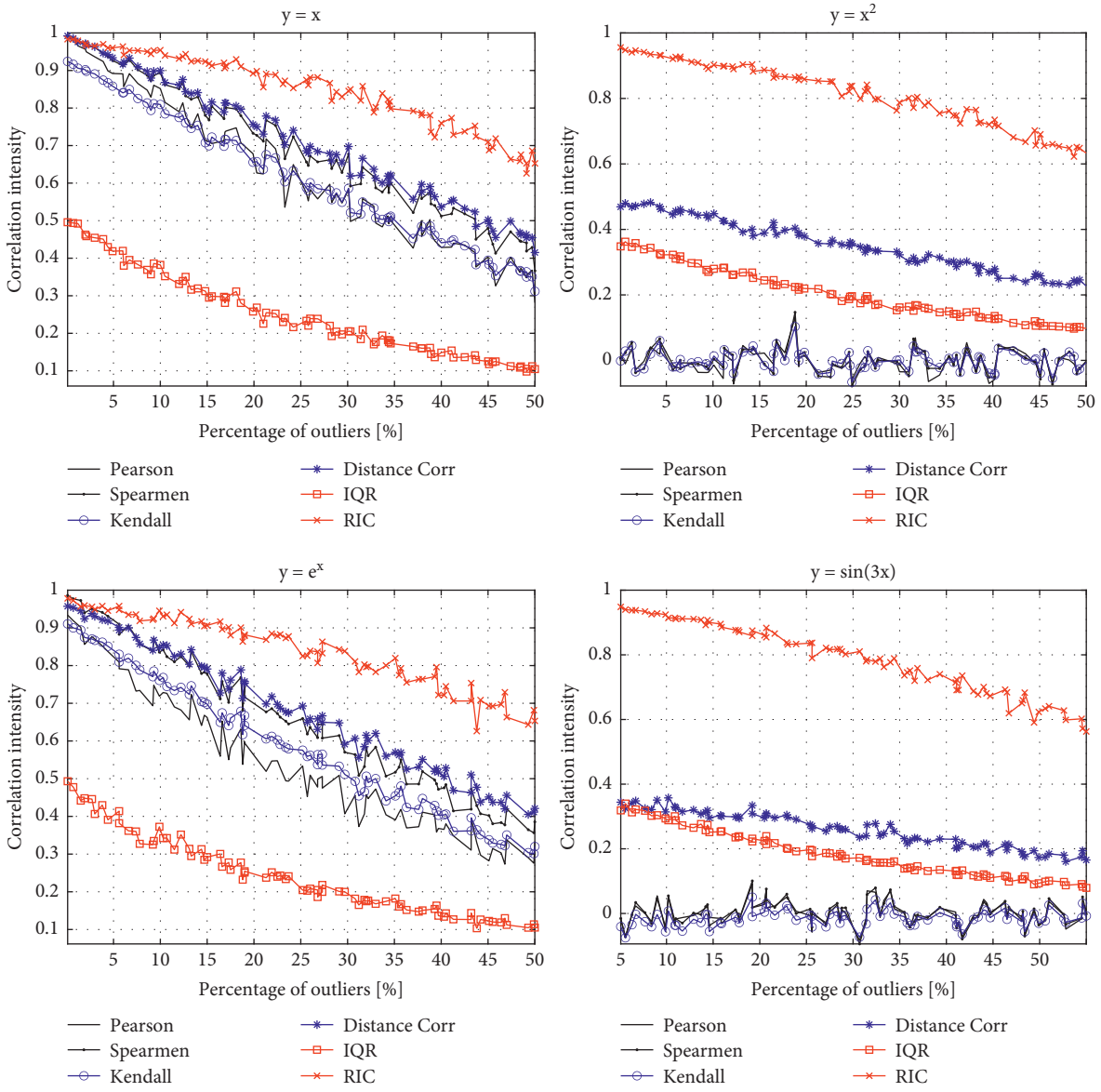


Figure 6: Robustness of RIC and other criteria against outliers. The x-axis reports the percentage of outliers generated randomly using a Gaussian distribution of large standard deviation.

$$f(x) = \frac{\lambda^n}{n!} e^{-\lambda}, \quad \forall n \in \mathbb{N}. \tag{21}$$

Gamma distribution:

$$f(x) = \frac{x^{k-1} e^{-x/\theta}}{\theta^k \Gamma(k)}, \quad \forall x > 0, k, \theta \in \mathbb{N}. \tag{22}$$

Figure 5 reports some comparative examples of the performance of the various indicators for these two distributions. The cases reported are full representatives of a series of systematic tests performed to investigate this point. In general, as for the Gaussian distribution reported in Figure 5 as a reference, RIC is also much less affected by additive noise of different distributions.

Another potential source of data contamination, of great practical relevance, is the presence of outliers. If the user is aware of the problem and has some prior information about the statistics of the outliers, some measures to remedy the situation can be taken before applying the dependence indicators [17, 18]. These measures belong to the family of robust statistics and can be quite effective. On the contrary, it is not always the case that the practitioner is aware of the issue and therefore investigating the robustness of the various dependence criteria to outliers remains a significant subject. Some representative results of a series of systematic tests, performed to assess this aspect, are shown in Figure 6 for the main classes of functional dependencies. In these cases, the outliers are generated as random Gaussian points with a mean equal to zero and standard deviation

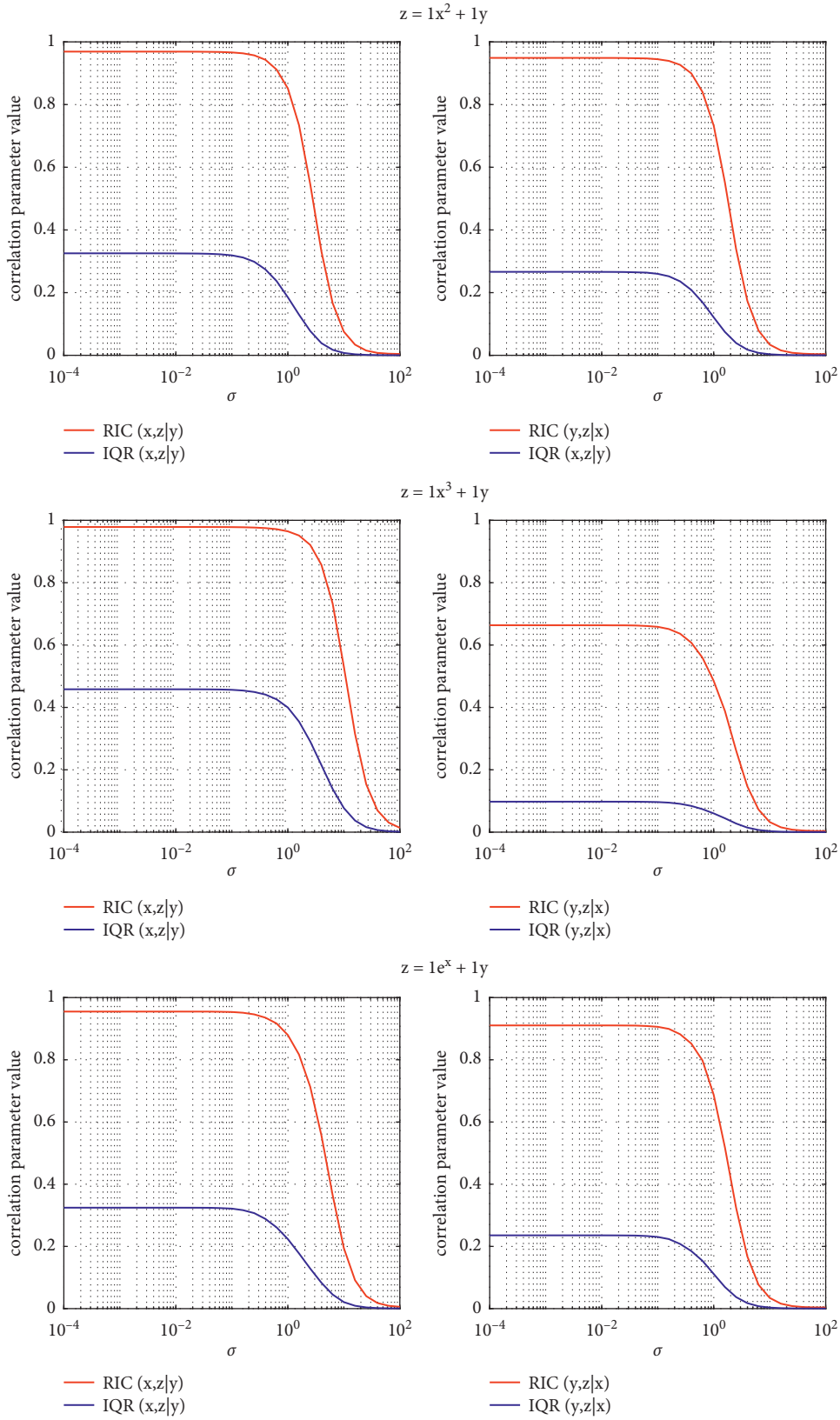


FIGURE 7: Comparison of $\mathrm{RIC}_{\mathrm{cond}}(X, Z|Y)$ and $\mathrm{IQR}_{\mathrm{cond}}(X, Z|Y)$ for various types of mutual influences in the presence of confounders.

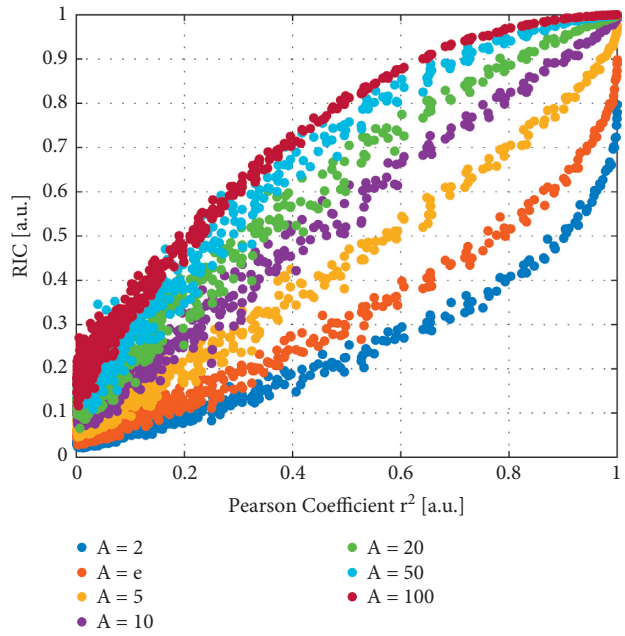


Figure 8: Trends of RIC versus the squared Pearson correlation coefficient for various values of A.

comparable to the range of the function. As expected, the most vulnerable indicator is the PCC. The rank-based criteria and the distance correlation are slightly more insensitive; the outliers must be relatively high both in number and amplitude to affect the ranking, before they have a detrimental effect on the values of these indicators. In any case, even in the presence of outliers, RIC remains the most robust criterion.

7. The Conditional Version of the Reciprocal Influence Criterion

In many applications of correlation analysis, one fundamental objective consists of determining the mutual influence between variables in the presence of confounding factors. It is therefore natural to investigate the potential of a conditional version of RIC:

$$\mathrm{RIC}_{\mathrm{cond}}(X, Z|Y) = 1 - \frac{1}{10^{\mathrm{MI}_{\mathrm{cond}}(X,Z|Y)}}. \quad (23)$$

In the following plots, the performances of $\mathrm{RIC}_{\mathrm{cond}}$ are compared with the ones of a conditional version of IQR:

$$\mathrm{IQR}_{\mathrm{cond}}(X, Z|Y) = \frac{\mathrm{MI}_{\mathrm{cond}}(X, Z|Y)}{H_{XZ}}. \quad (24)$$

The plots of Figure 7 report only the cases of nonlinear correlations ($z = x^2 + y$, $z = x^3 + y$, $z = e^x + y$), but the same conclusions apply also to linear effects.

In addition, in this application, RIC provides much better resilience to noise, and it is less dependent on the choice of the binning to determine the pdfs of the quantities involved. Moreover, even for very low levels of uncertainty, in the limit of no noise, it manages to identify more clearly the mutual correlations actually at play.

8. Discussion and Conclusions

To quantify the mutual influence between quantities, a new indicator has been introduced, the reciprocal influence criterion. A conditional version to separate the effects of confounding factors has also been devised. RIC reproduces the results of PCC in the case of linear correlations but is more robust against the influence of additive noise and outliers. In the case of nonlinear influences, RIC outperforms not only the ranked criteria and the distance covariance but also the information theoretic indicators such as the IQR in many respects; it provides more interpretable results and is more robust against noise and less sensitive to the choice of the binning of the pdfs involved. All these competitive advantages can be quite important in practice.

Other aspects, not to be neglected in the perspective of a wide application of the proposed indicator, are the fact that RIC is conceptually very simple, easy to implement, and fully general, in the sense that it does not rely on specific assumptions about the properties of the stochastic variables involved. In terms of requirements on the data, of course, enough examples must be available to properly calculate the pdfs, but again RIC is more parsimonious than the other indicators, which require estimating the probability distribution functions of the quantities involved.

Regarding future developments, it is planned to investigate whether alternative versions of the entropy and, therefore, of the derived quantities, can help improve the performance of RIC [19–22]. Furthermore, additional formulations, more suited to the investigation of actual causal relations than simple correlations, are also under consideration [23–26]. In terms of practical applications, some of the most immediate range from the investigation of synchronization experiments and disruptions in thermonuclear fusion [27–37] to the refinement of measurement techniques

and data analysis methods in earth sciences and plasma physics [38–40].

Appendix

A. Additional Useful Properties of the Reciprocal Influence Criterion

This appendix is devoted to showing how RIC satisfies some basic properties, which are desirable of any correlation criterion. They are symmetry, asymptotic consistency, and unbiasedness (independence from offset).

Symmetry: RIC (X, Y) is equal to RIC (Y, X). This is obviously true, being the mutual information symmetric. Indeed,

$$\mathrm{RIC}(X,Y) = 1 - \frac{1}{10^{\mathrm{MI}(X,Y)}} = 1 - \frac{1}{10^{\mathrm{MI}(Y,X)}} = \mathrm{RIC}(Y,X). \tag{A.1}$$

Asymptotical consistency: this property means basically that the indicator assumes appropriate values when the MI decreases toward zero or becomes very large. Also, in this respect, RIC behaves very satisfactorily. The indicator ranges from 0 to 1 and increases monotonically with MI, as can be derived directly from equations (19) and (20).

Independence to offset (unbiased): this additional very important property assures that a constant offset or bias in the data does not affect the results and can be written as RIC $(X + a,\ Y + b) = \mathrm{RIC}\ (X,\ Y)\ \forall a,\ b \epsilon P$. Also, this property is a direct consequence of the unbiased nature of the mutual information. Indeed,

$$\begin{aligned}\mathrm{RIC}(X+a,Y+b) &= 1 - \frac{1}{10^{\mathrm{MI}(X+a,Y+b)}} = 1 - \frac{1}{10^{\mathrm{MI}(X,Y)}} \\ &= \mathrm{RIC}(X,Y).\end{aligned} \tag{A.2}$$

B. The Choice of Parameter A

In general, the RIC indicator is defined as

$$\mathrm{RIC} = 1 - \frac{A^{H_{xy}}}{A^{H_x+H_y}} = 1 - \frac{1}{A^{\mathrm{MI}}}. \tag{B.1}$$

In this appendix, the behaviour of RIC with respect to A is discussed. The parameter A can indeed be optimised depending on the situation and the objectives of the analysis. This degree of freedom can become handy in various applications. The reason why A has been set equal to 10, to obtain the results presented in his work, is that, for this choice, RIC produces quite well the trends of the PCC in case the correlation between the two quantities analysed is linear. Figure 8 shows how the RIC values vary as a function of the linear correlations between x and y and A.

For $A = 10$, RIC varies almost linearly with squared PCC, and the consistency with this very popular indicator can be considered a positive quality in most applications.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Disclosure

Rossi and Gelfusa are the co-first authors. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data, in the writing of the manuscript, and in the decision to publish the results.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

All authors have contributed equally to this study.

References

[1] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, *Applied Multiple Regression/correlation Analysis for the Behavioural Sciences*, Psychology Press, Hove, East Sussex, UK, 3rd edition, 2002.

[2] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, no. 347, pp. 240–242, 1895.

[3] C. Spearman, "The proof and measurement of association between two things," *American Journal of Psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[4] M. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1-2, pp. 81–89, 1938.

[5] S. Guiasu, *Information Theory with Applications*, McGraw-Hill, New York, NY, USA, 1977.

[6] G. . J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing independence by correlation of distances," *Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[7] L. Zhu, K. Xu, R. Li, and W. Zhong, "Projection correlation between two random vectors," *Biometrika*, vol. 104, no. 4, pp. 829–843, 2017.

[8] W. Pan, "Ball covariance: a generic measure of dependence in banach space," *Journal of the American Statistical Association*, vol. 115, pp. 2020–529.

[9] G. J. Székely and M. L. Rizzo, "Brownian distance covariance," *Annals of Applied Statistics*, vol. 3, no. 4, pp. 1236–1265, 2009.

[10] SPSS Tutorials: Pearson Correlation, 2017, https://libguides.library.kent.edu/SPSS/PearsonCorr.

[11] G. W. Corder and D. I. Foreman, *Nonparametric Statistics: A Step-by-Step Approach*, Wiley, Hoboken, NJ, USA, 2014.

[12] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, Cambridge, UK, 2003.

[13] C. Arndt, *Information Measures, Information and its Description in Science and Engineering*, Springer Series: Signals and Communication Technology, 2004.

[14] T. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2006.

[15] R. Wilcox, "Introduction to robust estimation and hypothesis testing," *Statistical Modeling and Decision Science*, Elsevier/Academic Press, Amsterdam, Netherlands, 3rd edition, 2012.
[16] P. Baudot, M. Tapia, D. Bennequin, and J.-M. Goaillard, "Topological information data analysis," *Entropy*, vol. 21, no. 9, p. 869, 2019.
[17] S. Liu, *Distance Correlation*, 2020, https://www.mathworks.com/matlabcentral/fileexchange/39905-distance-correlation.
[18] T. P. Hettmansperger and J. W. McKean, *Robust Nonparametric Statistical Methods, Kendall's Library of Statistics*, John Wiley & Sons, Inc., New York, NY, USA, 1998.
[19] F. Nielsen and R. Nock, "A closed-form expression for the Sharma-Mittal entropy of exponential families," *Journal of Physics A: Mathematical and Theoretical*, vol. 45, no. 3, Article ID 032003, 2011.
[20] A. Rényi, "On measures of information and entropy," in *Proceedings of the Fourth Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 547–561, Berkeley, CA, USA, 1961.
[21] C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics," *Journal of Statistical Physics*, vol. 52, no. 1-2, pp. 479–487, 1988.
[22] C. Tsallis, *Introduction to Nonextensive Statistical Mechanics*, Springer, Berlin, Germany, 2009.
[23] M. C. Romano, M. Thiel, J. Kurths, and W. von Bloh, "Multivariate recurrence plots," *Physics Letters A*, vol. 330, no. 3-4, pp. 214–223, 2004.
[24] N. Marwan, M. Carmenromano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics Reports*, vol. 438, no. 5-6, pp. 237–329, 2007.
[25] T. Schreiber, "Measuring information transfer," *Physical Review Letters*, vol. 85, no. 2, pp. 461–464, 2000.
[26] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.
[27] A. Murari, M. Lungaroni, E. Peluso et al., "On the use of transfer entropy to investigate the time horizon of causal influences between signals," *Entropy*, vol. 20, no. 9, p. 627, 2018.
[28] A. Murari, E. Peluso, M. Gelfusa et al., "Application of transfer entropy to causality detection and synchronization experiments in tokamaks," *Nuclear Fusion*, vol. 56, no. 2, Article ID 026006, 2016.
[29] A. Murari, "How to assess the efficiency of synchronization experiments in tokamaks," *Nuclear Fusion*, vol. 56, Article ID 076008, 2016.
[30] A. Murari, T. Craciunescu, E. Peluso, E. Lerche, and M. Gelfusa, "On efficiency and interpretation of sawteeth pacing with on-axis ICRH modulation in JET," *Nuclear Fusion*, vol. 57, no. 12, Article ID 126057, 2017.
[31] E. Lerche, M. Lennholm, I. S. Carvalho et al., "Sawtooth pacing with on-axis ICRH modulation in JET-ILW," *Nuclear Fusion*, vol. 57, no. 3, Article ID 036027, 2017.
[32] L. Garzotti, "Investigating pellet ELM triggering physics using the new small size pellet launcher at JET," in *Proceedings of the 37th EPS Conference on Controlled Fusion and Plasma Physics*, European Physical Society, Dublin, Ireland, June 2010, https://ocs.ciemat.es/EPS2010PAP/pdf/P2.131.pdf.
[33] P. C. De Vries, M. Baruzzo, G. M. D. Hogeweij et al., "The influence of an ITER-like wall on disruptions at JET," *Physics of Plasmas*, vol. 21, no. 5, Article ID 056101, 2014.
[34] J. Paméla, F. Romanelli, M. L. Watkins et al., "The JET programme in support of ITER," *Fusion Engineering and Design*, vol. 82, no. 5-14, pp. 590–602, 2007.
[35] J. Pamela, "Overview of JET results," *Nuclear Fusion*, vol. 43, no. 12, 2003.
[36] A. Murari, P. Boutot, J. Vega et al., "Clustering based on the geodesic distance on Gaussian manifolds for the automatic classification of disruptions," *Nuclear Fusion*, vol. 53, Article ID 033006, 2013.
[37] A. Murari, F. Pisano, J. Vega et al., "Extensive statistical analysis of ELMs on JET with a carbon wall," *Plasma Physics and Controlled Fusion*, vol. 56, no. 11, Article ID 114007, 2014.
[38] F. P. Orsitto, A. Boboc, P. Gaudio et al., "Mutual interaction of Faraday rotation and Cotton-Mouton phase shift in JET polarimetric measurements," *Review of Scientific Instruments*, vol. 81, no. 10, Article ID 10D533, 2010.
[39] C. Bellecci, M. Francucci, P. Gaudio et al., "Application of a CO_2 dial system for infrared detection of forest fire and reduction of false alarm," *Applied Physics B*, vol. 87, no. 2, pp. 373–378, 2007.
[40] F. P. Orsitto, "Analysis of Faraday rotation in JET polarimetric measurements," *Plasma Physics and Controlled Fusion*, vol. 53, no. 3, 2011.

Hindawi
Complexity
Volume 2021, Article ID 1207167, 9 pages
https://doi.org/10.1155/2021/1207167

Research Article

Nuclear Fusion Pattern Recognition by Ensemble Learning

G. Farias,[1] **E. Fabregas**,[2] **I. Martínez**,[1] **J. Vega**,[3] **S. Dormido-Canto**,[2] **and H. Vargas**[1]

[1]*Pontificia Universidad Catolica de Valparaiso, Avenida Brasil 2147, Valparaiso 2362804, Chile*
[2]*Departamento de Informática y Automática, Universidad Nacional de Educación a Distancia, Juan Del Rosal 16, Madrid 28040, Spain*
[3]*Laboratorio Nacional de Fusión, CIEMAT, Avenida Complutense 40, Madrid 28040, Spain*

Correspondence should be addressed to G. Farias; gonzalo.farias@pucv.cl

Received 26 May 2021; Accepted 19 June 2021; Published 30 June 2021

Academic Editor: Atila Bueno

Nuclear fusion is the process by which two or more atomic nuclei join together to form a single heavier nucleus. This is usually accompanied by the release of large quantities of energy. This energy could be cheaper, cleaner, and safer than other technology currently in use. Experiments in nuclear fusion generate a large number of signals that are stored in huge databases. It is impossible to do a complete analysis of this data manually, and it is essential to automate this process. That is why machine learning models have been used to this end in previous years. In the literature, several popular algorithms can be found to carry out the automatic classification of signals. Among these, ensemble methods provide a good balance between success rate and internal information about models. Specifically, AdaBoost algorithm will allow obtaining an explicit set of rules that explains the class for each input data, adding interpretability to the models. In this paper, an innovative approach to perform an online classification, that is, to identify the discharge before it actually ends, using interpretable models is presented. In order to evaluate and reveal the benefits of rule-based models, an illustrative example has been implemented to perform an online classification of five different signals of the TJ-II stellarator fusion device located in Madrid, Spain.

1. Introduction

Energy is a crucial element for the subsistence of our modern civilization. Almost all human activities require energy to work. This requirement is increased year after year, especially due to the growing population, which is estimated by about 10 billion people in the middle of this century [1]. Nowadays, fossil fuels are the main source of energy because of their relatively low cost of production and high energetic capacity. However, they are not a long-term option. Alternatives like renewable energies are increasing their participation in modern life. However, the current technology of renewable sources is still not able to supply all energy needed [2, 3].

On the contrary nuclear sources can provide great quantities of energy. Although fusion energy is still developing, its potential is enormous, even compared with nuclear fission. Nuclear fusion is the process by which two or more atomic nuclei join together to form a single heavier nucleus. This is usually accompanied by the release of large quantities of energy. Fusion is the process that powers active starts, the hydrogen bomb, and some experimental devices. Nuclear fusion could be cheaper, cleaner, and safer. Fusion power would provide much more energy than any other technology currently in use, and the fuel required for fusion, mainly deuterium, exits abundantly in the oceans. Fusion could, in theory, supply all the energy needs of the world for millions of years [4, 5].

Achieving full control of the energy generated by nuclear fusion devices involves an analysis over huge databases with thousands of signals that is impossible to do it manually. This amount of data implies performing the analysis (e.g., finding significance or regular patterns) in high dimensional spaces and it is essential to automate the process using machine learning [6–9]. To this end, we can find several algorithms in the literature in order to perform pattern recognition in an automatic way. In the context of the pattern classification problem, the most popular algorithms

are Support Vector Machines and Neural Networks; both have shown high performance in previous applications in fusion [10–12], but with an important inconvenient: these algorithms produce black-box models, where it is not possible to obtain explicitly a simple mathematical relationship that outputs the classification.

The aim of this article is to present a new approach that combines the pattern (waveform) classification online with internal information about the decision model (i.e., interpretability). In the literature, there are several examples of using black-box models to automate the classification problem. The main reason for this is the high success rates reported in different topics such as nuclear fusion [12–14]. However, a black-box model does not give us any hint about the reason for the classification; for example, we are not able to know what the most important input variable is involved in the decision.

In this sense, ensemble methods provide a good balance between success rate and internal information about the model [15, 16]. Particularly, the AdaBoost algorithm allows obtaining an explicit set of simple rules that outputs the class of the signal from the input data [17]. Such property adds interpretability to the models, which could be useful to understand the reason for the classification process and, ultimately, for improving knowledge of the underlying physical phenomenon. This fact will allow a much more precise adjustment of the obtained models.

The main contributions of this article are as follows: (1) the waveform classification using ensemble methods generate rules-based models (if-then rules) that are not black boxes and could be useful to understand the entire process of the plasma discharges in nuclear fusion devices and (2) the classification system of waveform works online, which implies that we do not need to wait until the discharge finishes to obtain the class from the input data.

The rest of the paper is structured as follows: Section 2 introduces some basics aspects of the Nuclear Fusion Energy, the AdaBoost algorithm, and the signals used. Section 3 presents the offline and online approaches to classify the signals. A brief analysis of the models and features obtained is also presented. Finally, Section 4 summarizes the main conclusions.

2. Background

2.1. Nuclear Fusion Energy. In order to reproduce on the Earth the fusion power, some fusion reaction can be used. One of the most important is the deuterium-tritium cycle [18], which release 17.58 MeV as follows:

$$ {}_{1}^{2}\mathrm{D} + {}_{1}^{3}\mathrm{T} \longrightarrow {}_{2}^{4}\mathrm{He}\,(+3.52\,\mathrm{MeV}) + {}_{0}^{1}\mathrm{n}\,(+14.06\mathrm{MeV}). \quad (1) $$

In a fusion device, the reaction is produced at very high temperatures, about 150 million degrees Celsius. To this temperature, the matter inside of fusion devices is found like plasma, which is a state of matter similar to gas with a portion of its particles ionized. Magnetic fields are used to confine plasma in the shape of a torus. The most common configurations for magnetic confinement of plasma are stellarators and tokamaks. Figure 1 shows a simple and general scheme of the process of generating electrical energy from the nuclear fusion. The reactor uses deuterium (D) and tritium (T) to produce the reaction. The water is heated by the energy of the reaction and it feeds a turbine generator that produces the electricity.

The International Thermonuclear Experimental Reactor (ITER) is an international nuclear fusion research and engineering project, which is currently building the world's largest and most advanced experimental tokamak nuclear fusion reactor at the Cadarache (France) [18]. ITER is expected to demonstrate that more energy is obtained than is used to initiate the fusion process, something that has not been achieved by any experimental fusion reactor. After ITER, the first commercial demonstration fusion power plant, named DEMO [19], will be intended. Currently, there are many experimental fusion devices in operation. The Joint European Torus (JET) [20] is an experimental tokamak reactor located in Oxfordshire (UK). It is currently the largest facility of its kind in operation. TJ-II [21] is a medium size stellarator located at CIEMAT in Madrid (Spain). DIII-D is another tokamak machine developed by General Atomics in San Diego (USA) [22].

Experiments on fusion reactors are carried out by producing discharges or shot, in which plasma exists inside the torus. The duration of the shot is normally tens of seconds. ITER would keep the shot for about 30 minutes. During the discharges, many diagnostics around the reactor acquire data at high sampling frequencies. About 10 GBytes per discharge can be acquired in JET [20] (ITER could storage 1 TByte per shot). Bolometry, density, temperature, and soft X-rays are just some examples of the thousands of data sets acquired during a discharge. Huge databases, with an enormous amount of data, are a common situation in experimental fusion reactors.

However, nowadays only 10% of the generated data is processed, while the rest is not processed at all. Therefore, in order to achieve fusion energy as a clean, inexhaustible, safe, and cheap energy source, the current databases of experimental devices (tokamaks and stellarators) should be analyzed completely. Performing complete analysis will involve an optimal operation planning of ITER and, in turn, will be basic for a successful design of DEMO. For that reason, this project proposes the use of advanced pattern recognition and machine learning techniques in order to analyze in a faster and more efficient way massive fusion databases.

In this paper, the AdaBoost algorithm is used to build a rule-based model to classify five different waveform classes of the TJ-II stellarator. The plasmas in TJ-II are produced and heated with ECRH (2 gyrotrons, 300 kW each, 53.2 GHz, 2nd harmonic, X-mode polarization) and NBI (300 kW) [21]. Figure 2 shows a view of the TJ-II device.

2.2. AdaBoost Algorithm. The adaptive Boosting algorithm (AdaBoost) was proposed by Yoav Freund and Robert Shapire [17, 24]. AdaBoost is a general method to obtain a strong classifier (in our case a rule-based model) from a set of T weak classifiers (also called hypotheses, or rules in our

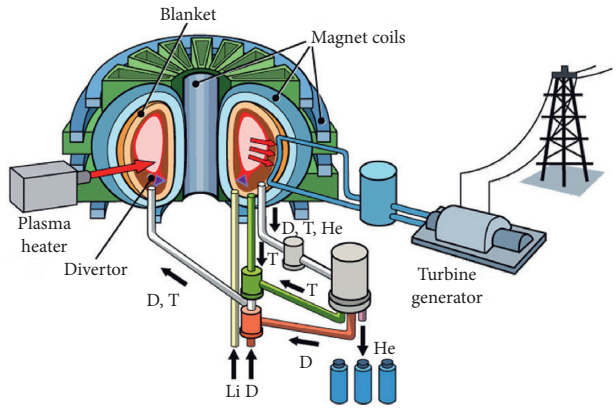


Figure 1: Electrical energy produced from fusion.

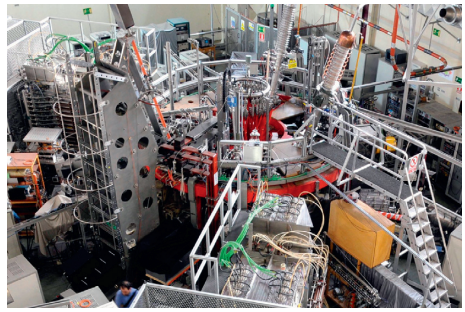


Figure 2: The TJ-II device [23].

case). This algorithm takes as input a set $S = \{F, y\}$, where F is a features vector of the signal that is going to be classified, y is the label of class for each signal (+1 or −1), and m is the total number of signals. Note that h_t is a rule, and T represents the number of rules that composes the rule-based model. Algorithm 1 is the pseudocode that shows the implementation of this algorithm.

The basic idea of boosting is to select the best weak (and simple) classifier after each iteration. The hypothesis selected is weighted according to its capacity to classify the training set correctly. Samples that were not correctly classified are also weighted in order to look for a suitable hypothesis for them in the next iteration. AdaBoost uses exponential error loss as an error criterion. The final model corresponds to a weighted sum of the selected weak hypotheses. The most important issue is that the resulting model is based on if-else rules, which means that the model is not based on a black-box type model. This represents an advantage compared to other classification algorithms [24].

In this way, AdaBoost can be used in a straight forward manner with signals. For example, a simple rule could be to predict a class if the average of the last 30 milliseconds is greater than a threshold. Thus, we can use if-then sentences such as if (avg (signal) > threshold) then +1, else −1 as a weak rule h_t (as in lines 9 and 10 of the pseudocode above). The output of the AdaBoost classifier will be finally the sign of the weighted sum of T rules (line 18 of the pseudocode) such as in equation (2). Note that α_t corresponds to the importance or weight of each class:

$$\text{Class}(F) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t \cdot (F)\right). \tag{2}$$

The algorithm can be easily extended for a multiclass problem (more than two classes) using the approach the one versus the rest, which implies building a model to classify the waveforms of a particular class (+1) versus the waveforms that belong to a different class (−1). This process is repeated in order to build one classifier for each class. For example, in [24], there is a detailed explanation about fundamental concepts of AdaBoost. In [11, 25], there are good descriptions about implementing classifiers for two or more classes in nuclear fusion databases combined with other algorithms (autoencoder and wavelet).

Figure 3 shows an illustrative example of the AdaBoost algorithm from [26], which is a previous work of the authors. There are seven samples of two classes (red circle and blue cross) in the upper image. Let us assume that a new sample located at (3.5; 3.0) has to be classified in one of the two classes. We can use the seven samples to build (train) a supervised data-driven model to predict the class of the new sample by using AdaBoost. The feature vectors are the respective Cartesian coordinates $x1$ and $x2$. After some iterations, the new sample is classified as a cross. The image

(1) **Input**: $S = \{(F_i, y_i), \quad \forall i = 1 \ldots m\}$
(2) #S: Training set example
(3) #D_1: Initial weight distribution
(4) #m: Size of the training set
(5) #$y_i \in \{-1; +1\}$
(6) $D_1(i) = 1/m, \quad \forall i = 1 \ldots m$
(7) #T rules that compose the rule-based model
(8) **for** $t := 1$ to T **do**
(9) #Get weak hypothesis h_t: $F \longrightarrow \{-1, +1\}$
(10) h_t = GetWeakHypothesis(F, D_t)
(11) $e_t = \sum_{i:h_t(F_i) \neq y_i} D_t(i)$
(12) $\alpha_t = (1/2)\ln((1 - e_t)/e_t)$
(13) #Update D_t distribution
(14) $D_{t+1}(i) = D_t(i)\exp(-\alpha_t y_i h_t(F_i))/Z_t, \quad \forall i = 1, \ldots, m$
(15) #Z_t: Normalization factor D_{t+1} is a distribution:
(16) $\#Z_t = \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(F_i))$
(17) **end for**
(18) **return** hypothesis combination: $C(F) = \sum_{t=1}^{T} \alpha_t h_t(F)$
(19) **end AdaBoost**

Algorithm 1: AdaBoost algorithm.

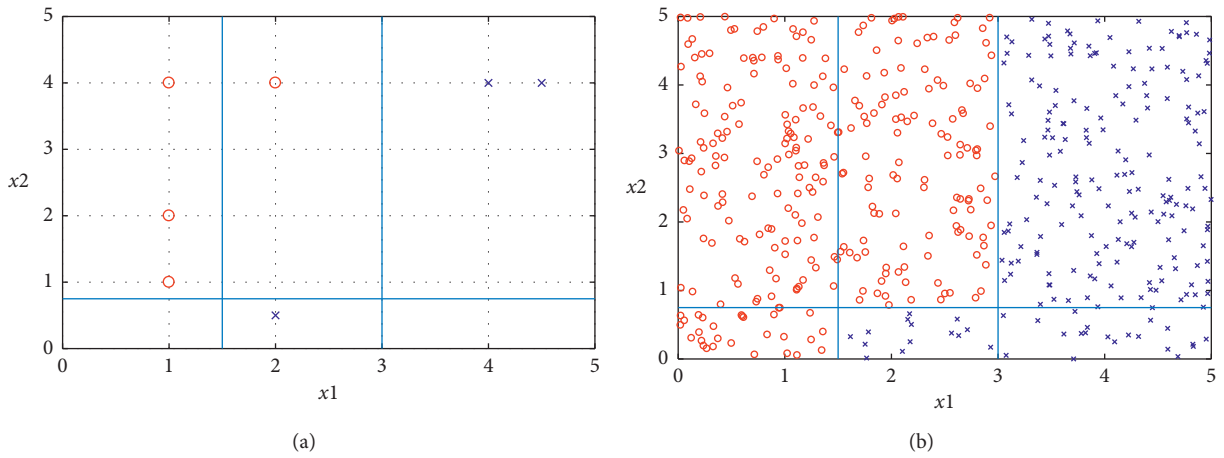


Figure 3: (a) Two classes (training samples). (b) Classification of 500 new samples.

below shows the result of testing the AdaBoost model for 500 new samples. Interested readers can find more technical details in the literature.

2.3. TJ-II Waveforms. The data generated by the experiments of the TJ-II device is stored in the relational database called TJ2RDB. This database allows searches to find shots with particular properties and to do scaling studies. More general information about this database can be found in [27, 28].

To illustrate the complexity of the data used for this work, we show the waveform of one of the signals involved in this research. Figure 4 shows the signal ECE7 for 200 shots. As it can be seen, for the same signal, the shape of the waveform is very different from a shot to another one. This implies that the classification of these kinds of signals can be a difficult task if it is carried out manually.

In order to evaluate and reveal the benefits of this approach, we have implemented a proof-of-concept using an illustrative example of the online classification for five different waveforms. This explanatory classification problem has been selected because the proposed approach can be easily compared with other previous works, where black-box algorithms have been implemented.

Figure 5 shows the temporal evolution of the 5 waveforms used to test the proposed approach in this work. From top to bottom and from left to the right, the waveforms (classes) are ECE7, GR, GR2, HALFAC3, and IACCEL1.

Table 1 presents a brief description of the selected TJ-II signals. Note that the selection of other signals might provide

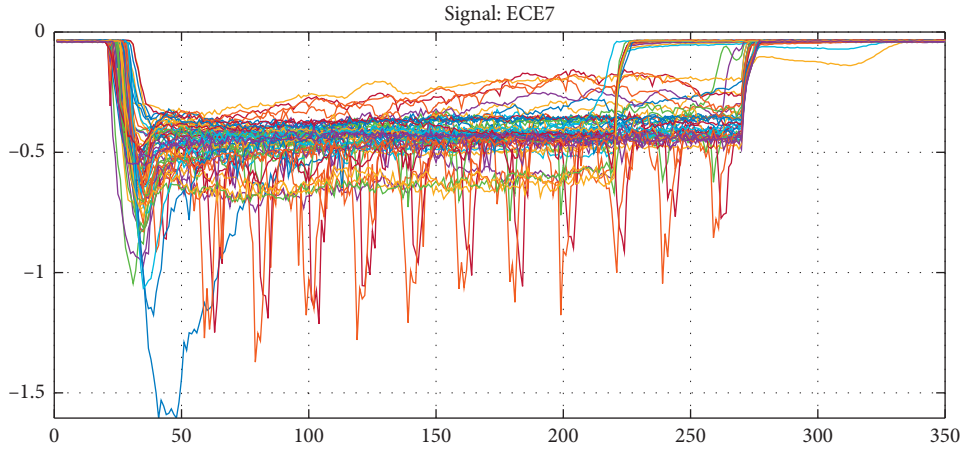


Figure 4: The plot depicts amplitude versus time (ms) of the ECE7 signal for 60 random shots.

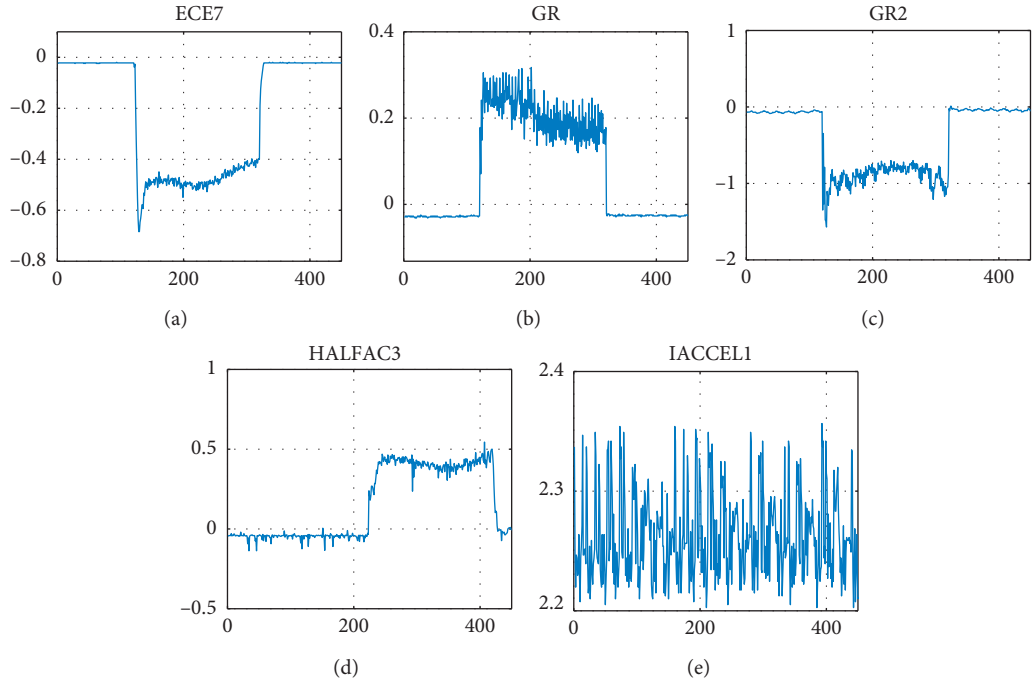


Figure 5: The plot depicts amplitude versus time (ms) of the five TJ-II waveforms classes: (a) ECE7, (b) GR, (c) GR2, (d) HALFAC3, and (e) IACCEL1 described in Table 1.

different results to those presented here, but the approach is enough general to obtain a classification with similar successful rates.

Finally, note that a supervised training scheme requires a previously labelled data set, and since in this context each signal is acquired by a separated sensor system, all the labels are known when data is stored. In a different context, the labelling process could imply the assistance of many specialists to obtain such data sets.

3. Waveforms Classification

The waveform classification is developed using AdaBoost with two approaches: (1) offline and (2) online. In the offline approach, the obtained model uses the entire signal to perform the classification, which involves the classification is done after the discharge has finished. On the other hand, a sensitivity analysis was also performed to select a reduced set of features in order to classify the waveform before it actually finishes, which could be very interesting for real-time applications. For comparison purposes, the rule-based model has been tested with signals used in previous works.

3.1. Offline Classification. For the offline approach, the AdaBoost algorithm has been implemented to classify the TJ-II waveforms using all the samples of the discharge. In this case, 340 waveforms have been used in total (68 waveforms for each class). Each entire waveform is resampled to 1024 samples to form the feature vector (F) in

Table 1: TJ-II waveforms.

Signal	Description	Class	Samples
ECE7	Electron cyclotron em.	1	500
GR	First gyrotron	2	500
GR2	Second gyrotron	3	500
HALFAC3	Hα line intensity	4	500
IACCEL1	Neutral beam injector	5	500

order to feed the AdaBoost algorithm. Finally, AdaBoost will output a rule-based model (AdaBoost model) that allows classifying a new waveform. Figure 6 shows the block diagram of the implemented stages.

Table 2 shows three rules (h_t) and their associated weights (α_t) of the obtained rule-based model to classify GR signals (class 2). Note that the features F_{578} (magnitude of GR signal at sample 578), F_{842} (sample 842), and F_{1024} (sample 1024) are used to perform the classification.

In the case of class 3 (GR2 signal), the classification can be performed by using only the following rule: if ($F_1 < -1.151$) then +1 else −1, which implies that using only the first feature of a discharge (F_1), the approach is able to classify GR2 signals successfully.

In order to evaluate the model, we split the data into two subsets (cross-validation). The training stage was carried out with 60% of the data set (205 waveforms for each of the 5 classes) while 40% of the data set was used for the test stage (135 waveforms for each class).

Table 3 shows the results of the offline classification of the 5 types of signals. As it can be seen, the results are encouraging. All the success rates are above 93%.

The average success rate of the ensemble model is up to 98%, improving the results of previous works. In [25], a Wavelet Transform with Support Vector Machines (WT + SVM) [25], the results were up to 92%. More recently, in [11], a Stacked Autoencoder (a type of Neural Network) in combination with Support Vector Machines (NN + SVM), the results were up to 94%.

One interesting thing about the proposed approach is the ability to see the importance of each feature to perform the classification. Figure 7 shows the features selected (samples) by the algorithm to classify the ECE7 signals. The blue line represents shot and red circles represent the features used to classify this signal.

The size of the circles is proportional to the importance of the rule (α_t) that uses the feature. As it can be seen, the most important values are located before sample 200, which implies that some signals could be classified at the beginning of the discharge; that is, an online classification could be performed.

Based on the previous results, the idea of online classification came up. In this way, it is not necessary to wait until the end of the discharge in order to perform the classification. The next section presents the online classification algorithm.

3.2. Online Classification. This approach starts the classification at the very beginning of the discharge. First, the signal is preprocessed in sliding windows obtained by grouping 10

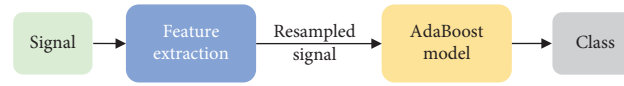


Figure 6: Block diagram of the offline application.

Table 2: Example of rules obtained to classify GR signals.

h_t	α_t
if ($F_{578} < 2.127$) then +1 else −1	0.3726
if ($F_{842} < 2.108$) then +1 else −1	0.3782
if ($F_{1024} < 0.131$) then +1 else −1	0.5675

Table 3: Results of the classification of the 5 signals.

Signal	Success rate (%)
ECE7	99.18
GR	98.35
GR2	100
HALFAC3	93.83
IACCEL1	100

consecutive samples and taking only one representative sample for each window. In this way, the signal is reduced by a factor of 10. Then, the feature extraction stage is applied to obtain some specific characteristics of the signal that helps in the classification. Figure 8 shows the block diagram of the online approach.

Figure 9 shows an explanatory diagram of the algorithm. The red solid line represents the signal that is being analyzed. The blue dashed rectangle represents the sliding window, which contains the segment of the signal analyzed at the current iteration.

Then, the four features of this window are obtained: average value (F_1), minimum value (F_2), maximum value (F_3), and, finally, slope value (F_4), which is calculated by performing a least squares adjustment. After that, the AdaBoost model classifies the signal into one of the five classes. When the result of the AdaBoost is three consecutive positive values (+1), the signal is classified as this class. In this example, the signal is classified as HALFAC3 (Class 4) as is represented by the red dashed rectangle.

Similar to the offline case, we can easily obtain the attributes used by the algorithm to perform the classification. Table 4 shows the features used for each class. Note that classes 1, 2, and 5 use only three features.

Table 5 presents the confusion matrix that shows the results of the online classification of the five classes. Rows represent the class that is being classified and the columns represent the predictions of the classification for the actual

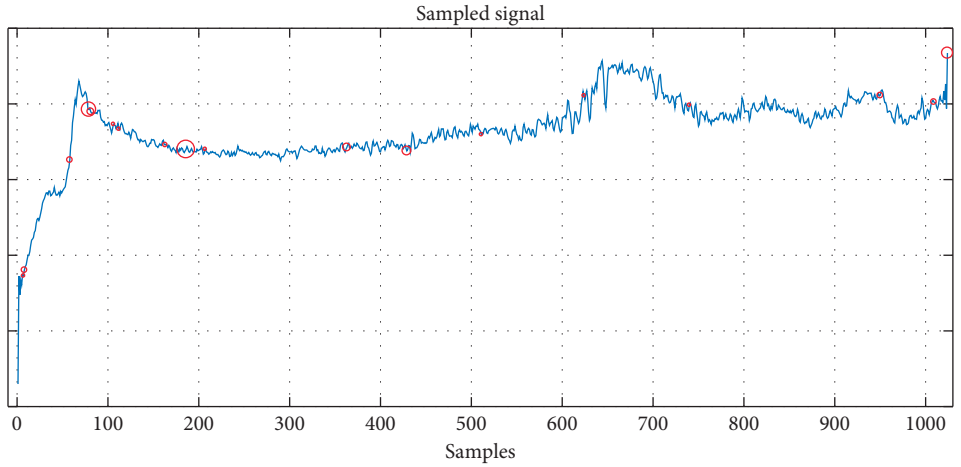


Figure 7: Features selected (samples marked as red circles) by the AdaBoost algorithm to perform the classification of ECE7 signals (Class 1).

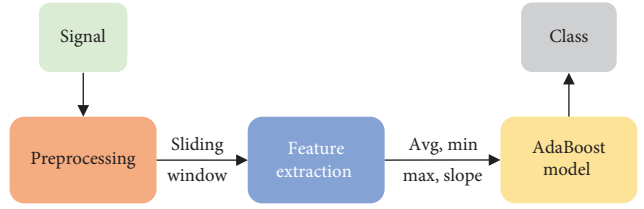


Figure 8: Block diagram of the online application of the approach.

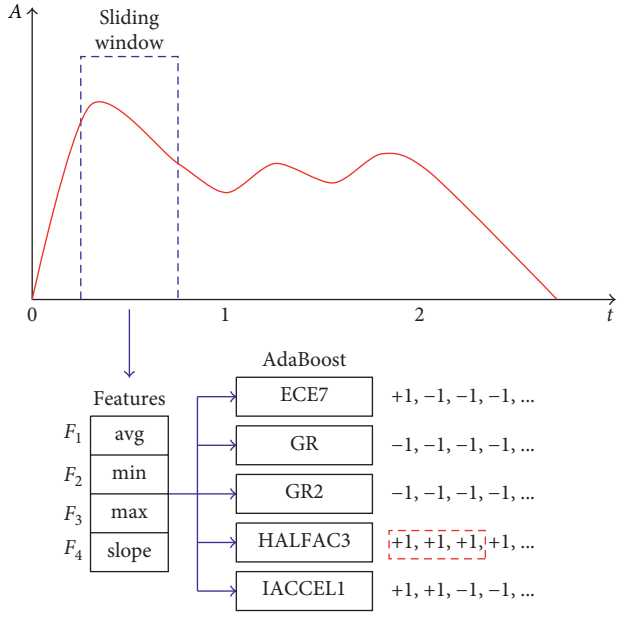


Figure 9: Diagram of the method.

signal. As it can be seen, almost all the 27 tested discharges for each class are correctly classified. This leads to the fact that the average success rate is over 99%.

Table 6 shows the time fraction of the discharge required to perform the online classification of the five classes for 27 randomly selected shots. The second column is the average (in percentage) of time fraction for all shots to carry out the classification. The third column is the standard deviation of the time faction (in percentage) needed to make the classification. The fourth and fifth are the minimum and maximum values. As it can be seen, the algorithm takes more time to classify the class HALFAC3 (0.23%) of the signal,

Table 4: Results of the classification of the 5 signals.

Class	Features
1	F_1, F_2, F_3
2	F_1, F_2, F_3
3	F_1, F_2, F_3, F_4
4	F_1, F_2, F_3, F_4
5	F_1, F_2, F_4

Table 5: Confusion matrix of the online classification.

		Predicted class				
		1	2	3	4	5
Actual class	1	**27**	0	0	0	0
	2	0	**27**	0	0	0
	3	0	0	**27**	0	0
	4	0	1	0	**26**	0
	5	0	0	0	0	**27**

Table 6: Results of the classification of the 5 signals.

Class	Avg (%)	**Std** (%)	Min (%)	Max (%)
ECE7	0.09	0.01	0.08	0.11
GR	0.14	0.22	0.08	1.21
GR2	0.09	0.01	0.08	0.11
HALFAC3	0.23	0.26	0.08	1.17
IACCEL1	0.09	0.01	0.08	0.11

which is a good result because this value is still short. The standard deviation is also short, which means that all the signals are classified with a small initial segment of each signal. The minimum value indicates that, for all classes, the algorithm never classifies a signal before 0.08% of the time. The maximum value indicates that the algorithm can classify all the signals before 1.21% fraction of the entire signals, which is a very good result.

The experiments were carried out on a PC with an Intel Core i7-8750H, 2.2 GHz, 16 GB of RAM, and Ubuntu 18.04.1 LTS operating system. For this configuration, the classification process time of each sliding window is less than 10 milliseconds (about 1 ms for feature extraction and less than 9 ms). Considering the nature of the rule-based model, this time could be clearly reduced when using embedded hardware such as field-programmable gate array (FPGA) or Application-Specific Integrated Circuit (ASIC).

4. Conclusions

This article proposes two approaches to perform a classification of five TJ-II waveforms using the ensemble algorithm AdaBoost. The first method is carried out in an offline manner and the signals are resampled to obtain distinctive attributes in the feature extraction stage. These features are translated into AdaBoost rules to classify the signals. With this method, the classifications can achieve high success rates and the classifiers are built with explicit relationships between features and rules of the AdaBoost algorithm, which allows designers to understand better the physical underlying phenomenon. In the second approach, the classification is made for performing online classification. Firstly, the signal is preprocessed in sliding consecutive windows. Then, the feature extraction stage is performed to obtain the average, the minimum, the maximum, and the slope of the signal. These features are translated into rules of the AdaBoost algorithm that is capable of classifying the signals. The main advantage of this approach is that we do not need to wait until the discharge has finished in order to classify, which means that the classification can be performed online. Almost all the 27 tested discharges for each class are correctly classified. The average success rate is over 99%. The results show that the online classification can be performed by using only a very small fraction of the discharge.

Data Availability

The data used to support the findings of this study have not been made available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Chilean Research and Development Agency (ANID) under the Project FONDECYT 1191188 and by the Spanish Ministry of Science and Innovation under Projects PID2019-108377RB-C31 and PID2019-108377RB-C32.

References

[1] J. P. DeLong, O. Burger, and M. J. Hamilton, "Current demographics suggest future energy supplies will be inadequate to slow human population growth," *PLoS One*, vol. 5, no. 10, Article ID e13206, 2010.

[2] A. Qazi, F. Hussain, N. A. Rahim et al., "Towards sustainable energy: a systematic review of renewable energy sources, technologies, and public opinions," *IEEE Access*, vol. 7, pp. 63837–63851, 2019.

[3] D. C. Momete, "Analysis of the potential of clean energy deployment in the European union," *IEEE Access*, vol. 6, pp. 54811–54822, 2018.

[4] J. K. Shultis and E. F. Richard, *Fundamentals of Nuclear Science and Engineering*, CRC Press, Boco Raton, FL, USA, 3rd edition, 2016.

[5] F. I. T. Petrescu, A. Apicella, R. V. V. Petrescu et al., "Environmental protection through nuclear energy," *American Journal of Applied Sciences*, vol. 13, no. 9, pp. 941–946, 2016.

[6] G. Farias, J. Vega, S. González et al., "Automatic determination of l/h transition times in DIII-D through a collaborative distributed environment," *Fusion Engineering and Design*, vol. 87, no. 12, pp. 2081–2083, 2012.

[7] T. Yokoyama, T. Sueyoshi, Y. Miyoshi et al., "Disruption prediction by support vector machine and neural network with exhaustive search," *Plasma and Fusion Research*, vol. 13, p. 3405021, 2018.

[8] G. W. Shin, J.-W. Juhn, G. I. Kwon, S. H. Son, and S. H. Hahn, "Automatic detection of L-H transition in Kstar by support

vector machine," *Fusion Engineering and Design*, vol. 129, pp. 341–344, 2018.

[9] A. Murari, M. Lungaroni, E. Peluso et al., "Adaptive predictors based on probabilistic SVM for real time disruption mitigation on JET," *Nuclear Fusion*, vol. 58, no. 5, 2018.

[10] G. Farias, R. Dormido, M. Santos, and N. Duro, "Image classifier for the TJ-II Thomson scattering diagnostic: evaluation with a feed forward neural network," in *Proceedings of the International Work-Conference on the Interplay between Natural and Artificial Computation*, pp. 604–612, Springer, Canary Islands, Spain, June 2005.

[11] G. Farias, S. Dormido-Canto, J. Vega et al., "Automatic feature extraction in large fusion databases by using deep learning approach," *Fusion Engineering and Design*, vol. 112, pp. 979–983, 2016.

[12] G. Farias, E. Fabregas, S. Dormido-Canto et al., "Applying deep learning for improving image classification in nuclear fusion devices," *IEEE Access*, vol. 6, pp. 72345–72356, 2018.

[13] L. Makili, J. Vega, S. Dormido-Canto et al., "Upgrade of the automatic analysis system in the TJ-II Thomson Scattering diagnostic: new image recognition classifier and fault condition detection," *Fusion Engineering and Design*, vol. 85, no. 3-4, pp. 415–418, 2010.

[14] B. Cannas, P. C. de Vries, A. Fanni et al., "Automatic disruption classification in JET with the ITER-like wall," *Plasma Physics and Controlled Fusion*, vol. 57, no. 12, Article ID 125003, 2015.

[15] L. Rokach, "Pattern classification using ensemble methods," *World Scientific*, vol. 75, Article ID 7238, 2010.

[16] J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren, "The online performance estimation framework: heterogeneous ensemble learning for data streams," *Machine Learning*, vol. 107, no. 1, pp. 149–176, 2018.

[17] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[18] M. Glugla, R. Lässer, L. Dörr, D. K. Murdoch, R. Haange, and H. Yoshida, "The inner deuterium/tritium fuel cycle of ITER," *Fusion Engineering and Design*, vol. 69, no. 1-4, pp. 39–43, 2003.

[19] D. Stork, *DEMO and the Route to Fusion Energy*, Karlsruhe Intl School on Fusion Technology, Karlsruhe, Germany, 2009.

[20] P. H. Rebut, R. J. Bickerton, and B. E. Keen, "The Joint European Torus: installation, first results and prospects," *Nuclear Fusion*, vol. 25, no. 9, pp. 1011–1022, 1985.

[21] C. Alejaldre, J. Alonso, L. Almoguera et al., "First plasmas in the TJ-II flexible heliac," *Plasma Physics and Controlled Fusion*, vol. 41, no. 3A, 1999.

[22] J. A. Boedo, D. Rudakov, R. Moyer et al., "Transport by intermittent convection in the boundary of the DIII-D tokamak," *Physics of Plasmas*, vol. 8, no. 11, pp. 4826–4833, 2001.

[23] National Fusion Laboratory of Spain (CIEMAT), *TJ-II Project*, CIEMAT, 2021, http://www.fusion.ciemat.es/tj-ii-2/, Madrid, Spain.

[24] R. Rojas, "AdaBoost and the super bowl of classifiers a tutorial introduction to adaptive boosting," Technical Report, Freie Universität, Berlin, Germany, 2009.

[25] S. Dormido-Canto, G. Farias, R. Dormido et al., "TJ-II wave forms analysis with wavelets and support vector machines," *Review of Scientific Instruments*, vol. 75, no. 10, pp. 4254–4257, 2004.

[26] G. Farias, S. Dormido-Canto, J. Vega, I. Martínez, L. Alfaro, and F. Martínez, "AdaBoost classification of TJ-II thomson scattering images," *Fusion Engineering and Design*, vol. 123, pp. 759–763, 2017.

[27] J. Vega, C. Crémy, E. Sánchez, A. Portas, J. A. Fabregas, and R. Herrera, "Data management in the TJ-II multi-layer database," *Fusion Engineering and Design*, vol. 48, no. 1–2, pp. 69–75, 2000.

[28] E. Sánchez, A. B. Portas, and J. Vega, "A relational database for physical data from TJ-II discharges," *Fusion Engineering and Design*, vol. 60, no. 3, pp. 341–346, 2002.

Hindawi
Complexity
Volume 2021, Article ID 9945218, 13 pages
https://doi.org/10.1155/2021/9945218

Research Article

A Comparative Analysis of Data-Driven Empirical and Artificial Intelligence Models for Estimating Infiltration Rates

Mohammad Zakwan [1] **and Majid Niazkar** [2]

[1]*Civil Engineering Department, Maulana Azad National Urdu University, Hyderabad, India*
[2]*Department of Civil and Environmental Engineering, Shiraz University, Shiraz, Iran*

Correspondence should be addressed to Majid Niazkar; mniazkar@shirazu.ac.ir

Received 16 March 2021; Revised 8 April 2021; Accepted 24 April 2021; Published 5 May 2021

Academic Editor: Teddy Craciunescu

Infiltration is a vital phenomenon in the water cycle, and consequently, estimation of infiltration rate is important for many hydrologic studies. In the present paper, different data-driven models including Multiple Linear Regression (MLR), Generalized Reduced Gradient (GRG), two Artificial Intelligence (AI) techniques (Artificial Neural Network (ANN) and Multigene Genetic Programming (MGGP)), and the hybrid MGGP-GRG have been applied to estimate the infiltration rates. The estimated infiltration rates were compared with those obtained by empirical infiltration models (Horton's model, Philip's model, and modified Kostiakov's model) for the published infiltration data. Among the conventional models considered, Philip's model provided the best estimates of infiltration rate. It was observed that the application of the hybrid MGGP-GRG model and MGGP improved the estimates of infiltration rates as compared to conventional infiltration model, while ANN provided the best prediction of infiltration rates. To be more specific, the application of ANN and the hybrid MGGP-GRG reduced the sum of square of errors by 97.86% and 81.53%, respectively. Finally, based on the comparative analysis, implementation of AI-based models, as a more accurate alternative, is suggested for estimating infiltration rates in hydrological models.

1. Introduction

Infiltration can be defined as the process by which water enters the surface of Earth [1]. It leads to the entrance of water into the soil, thereby catering to groundwater recharge and subsurface runoff. In essence, the infiltration phenomenon is among the most crucial processes of water cycle. Furthermore, estimates of infiltration capacity of soil is required in the design of efficient irrigation systems, estimation of evapotranspiration, groundwater recharge, surface runoff, effective rainfall, crop water requirement, and transport of chemicals in surface and subsurface water [2]. As a result, modelling and prediction of infiltration rates is an inevitable part of hydrological modelling. For instance, Morel-Seytoux [3] reviewed the importance of infiltration in large-scale hydrologic modelling. Furthermore, Šraj et al. [4] pointed towards the impact of the estimation of infiltration rates on the runoff hydrograph, which plays a vital role in watershed modelling and water management. Similarly, Wen et al. [5] demonstrated the implication of excessive infiltration on watershed models. Finally, these studies demonstrated why an accurate estimation of time-dependent infiltration is important in hydrological modelling.

Owing to the wide applications of the infiltration rate, its estimation has gained significant attention from researchers. Over the years, various infiltration models have been proposed by the researchers for the estimation of infiltration rates. They include models that have physical, semiempirical, and even empirical formulations. Despite the development of several models, no single model exists that outperforms other ones universally. The suitability of infiltration model for a particular site depends on the type of soil and field conditions [1]. In this regard, many comparative studies have been conducted to assess the suitability of various infiltration models for different soil types under varying field conditions. Mishra et al. [6] conducted one of the most

comprehensive analyses on suitability of infiltration models for different soils. Similarly, the methodology used to model infiltration rate has a significant impact on the estimation of infiltration. Deep and Das [7] compared various optimization algorithms to estimate the parameters of infiltration models. Nonetheless, the application of different optimization techniques can only move the solution from local optimum parameters towards global optimum parameters, while they cannot increase the flexibility of infiltration models to mimic actual infiltration rates. Haghiabi et al. [8] employed a dimensionless form of infiltration data to estimate infiltration parameters accurately. However, Zakwan [9] suggested that such transformation may not necessarily improve the accuracy of infiltration equations. Finally, Chen et al. [10] utilized genetic algorithm to improve the estimate of Green-Ampt infiltration model under a rainfall condition.

Recent applications of computational techniques in water resource engineering have widened the scope further [11–21]. With the advancement in the computational method and modelling approaches, the application of these approaches has provided a viable alternative towards the estimation of infiltration rates also. Kumar and Sihag [22] applied Gene Expression Programming (GEP) to model infiltration rates. Moreover, Dewidar et al. [23] proposed the application of fuzzy logic to estimate the infiltration rates. In addition, Patle et al. [24] employed a multiple linear regression model to predict time-dependent infiltration values based on several soil properties such as bulk density, silt, sand percentage, and moisture content. Furthermore, Sihag et al. [25] exploited the support vector machine (SVM) for modelling infiltration rates in sandy soil. Also, Pahlavan-Rad et al. [26] compared the performance of Multiple Linear Regression (MLR) and Random Forest Tree in depicting the spatial variation of infiltration rates and reported the superiority of Random Forest Tree over MLR. Recently, Sepahvand et al. [27] utilized several data-driven models to predict infiltration rates. Their investigation revealed the superiority of neural networks over other data-driven techniques such as model tree, Gaussian process, and regression analysis. According to the recent studies, considering time as the exclusive state variable in empirical models should be revisited in favour of better infiltration predictions, while the AI-based models were found to have a better performance in comparison with the conventional infiltration models. Therefore, despite previous efforts in improving the estimation of infiltration rates, further studies are needed to explore these issues.

The present study aims to compare the performances of different infiltration methods. Additionally, it attempts to assess the capability MGGP and of the novel hybrid MGGP-GRG to model the infiltration process. In a bid to seek for a better time-dependent infiltration model, the performances of the MGGP-based models were compared with those of the conventional models, regression techniques, and commonly used neural network.

2. Materials and Methods

2.1. Data. In the present study, the infiltration data reported by Sihag et al. [28] were utilized. The data were divided into training and testing data sets. To be more precise, 75% of the data were used for training, while the rest of the data were exploited to test the obtained results. Table 1 summarises the data sets used in the present paper.

Figure 1 shows the observed infiltration data at the same time duration. Also, it illustrates that the infiltration rate may be dependent on other factors (soil properties, such as bulk density and sand percentage) apart from time. The infiltration data set, which was obtained from the literature [28], belongs to the infiltration observations carried out at Davood Rashid and Honam regions in Lorestan Province and the Kelat region in Ilam Province in Western Iran.

2.2. Conventional Infiltration Models. There are a number of infiltration models available in the literature. Brief description of some of the commonly used infiltration models considered in the present study is as follows.

(1) *Horton's Model*

Horton [29] proposed an empirical equation, which is presented in the following equation, for exponential decay of the infiltration rate after analysing several infiltrometer data sets:

$$f = f_c + (f_0 - f_c)e^{-kt}, \tag{1}$$

where f is the infiltration capacity at any time t from the start; f_c is the final or ultimate infiltration capacity occurring at $t = t_c$; f_0 is the initial infiltration capacity at time $t = 0$; and K is Horton's decay coefficient.

(2) *Philip's Model*

Philip [30] proposed an infinite series solution of Richard's equation to drive a relationship between the cumulative infiltration (F) and soil properties. It is presented in the following equation:

$$F = st^{0.5} + Kt\,. \tag{2}$$

By differentiating the above equation, the infiltration rate may be represented as

$$f = \frac{1}{2}st^{-0.5} + K\,. \tag{3}$$

(3) *Modified Kostiakov's Model*

Kostiakov [31] observed the temporal variation of infiltration into soil and proposed a time-dependent infiltration model, invariantly known as Kostiakov's model. The major limitation of Kostiakov's model is that it approached to zero final infiltration rates rather than toward constant final infiltration rates and infinite infiltration rates at the start. Smith [32] modified Kostiakov's [31] equation to include the constant term f_i. The modified version is shown in the following equation:

$$f = (ab)t^{b-1} + f_i. \tag{4}$$

The parameters of different infiltration models were obtained by minimizing the sum of square of errors using a

Table 1: Training and testing data used.

Data	Count	Infiltration rate (cm/min)	Sand (%)	Density (g/cm^3)
Training	116	0.080–1.560	6.00–38	0.08–1.56
Testing	38	0.080–1.480	6.00–38.00	0.08–1.48

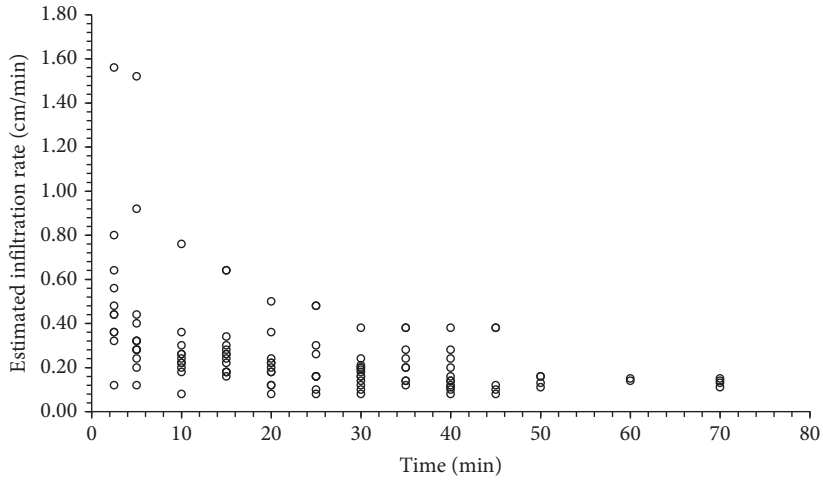


Figure 1: Observed infiltration data.

nonlinear optimization tool. Thus, the objective function becomes

$$\text{Min SSE} = \sum_{i=1}^{N} \left[f_{obs_i} - f_{est_i} \right]^2, \tag{5}$$

where f_{obs} is the observed infiltration rate and f_{est} is the estimated infiltration rate at any time t.

2.3. Multiple Linear Regression. MLR has been widely used in water resource engineering [17, 33]. It has also been applied to estimate the infiltration rate [25]. In accordance with MLR, infiltration rate can be expressed as

$$f = c_1 t + c_2 S + c_3 D + c_4, \tag{6}$$

where c_1, $c_2 c_3$,c_4, and c_5 are coefficients, f is the infiltration rate in cm/min; t is time in minutes; S is the percentage of sand; D is the density in g/cm^3.

2.4. Generalized Reduced Gradient (GRG). GRG is a gradient-based nonlinear optimization technique [34]. Earlier, Zakwan et al. [1] and Muzzammil et al. [35] suggested that GRG technique is superior to the conventional graphical method for estimating infiltration parameters and rating curve parameters. In accordance with GRG, the infiltration rate can be expressed as

$$i = c_5 t^{c_6} S^{c_7} D^{c_8}, \tag{7}$$

where c_5,c_6, c_7 , and c_8 are coefficients.

In the present study, GRG solver embedded in Microsoft Excel was used to estimate the infiltration rate based on minimizing the sum of square of errors. Detailed explanation on working of GRG technique is available in the literature [17, 36].

2.5. Artificial Neural Network. ANN is one well-documented AI model. It has been used for solving various problems in water resources and hydrological modelling [37, 38]. Generally, ANN has a few layers, whose neurons store data. The neurons in each layer (input, hidden, and output layers) are connected with neurons in the previous and next layers, whereas there is no connection between neurons in a typical layer [39]. The flexible architecture of ANN basically facilitates the estimation of a relationship between input and output data [40]. In this study, a feed-forward ANN was exploited to predict the rate of the infiltration. The controlling parameters of ANN were set as those used in the previous studies [41].

2.6. Multigene Genetic Programming. MGGP is a modified version of genetic programming (GP), which is classified as an AI technique [42]. Not only does it utilize genetic algorithm as its search engine but also it works as a flexible estimator without the need to know the shape of a prediction model under investigation [43]. In essence, MGGP follows a similar solving approach as GP using a tree-like structure, while it enables the use of more than one gene, i.e., tree, in each individual. This characteristic benefits MGGP in the light of developing estimation models when the relation among involved variables is complicated to study. As a result, a typical MGGP solution consists of a set of equations, each associated with one gene, which is algebraically summed up using weighting coefficients. These coefficients are calibrated in MGGP, while a term invariantly called as bias is also added to the final solution. The terms comprising the

final solution of MGGP help in improving its flexibility in capturing the relationship between input and output data.

In this study, an open-access code of MGGP was exploited. This code was adopted form the literature [44], while it was used in previous studies for other purposes [20]. It minimizes the root mean square of errors between the estimated and observed values of the normalized infiltration rates. Additionally, the MGGP parameters were selected from previous studies [20, 43]. Since each run of MGGP may result in a unique equation, more than 100 runs of MGGP were considered for developing the relation between the infiltration rates and other variables involved. The common number of MGGP runs in the literature is 50 [20, 45], while the double number of runs, i.e., 100, was taken into account to make sure that the best relation was achieved.

2.7. Hybrid MGGP-GRG Technique. The hybrid MGGP-GRG was first proposed for developing stage-discharge relationships in the literature [20]. In this technique, MGGP and GRG are used in two successive steps to find the best-fit model. Figure 2 depicts the flowchart of the hybrid MGGP-GRG for estimating infiltration rates. As shown, MGGP is initially operated to search for the best-fit form of equation to the data, while the GRG technique is utilized to optimize the coefficients of the equation obtained by MGGP. Hence, this hybrid technique not only benefits from the powerful capability of MGGP for seeking an accurate prediction model, but also uses GRG to enhance the performance of the estimation model.

2.8. Performance Evaluation Criteria. The performance of infiltration models and soft computing techniques was compared based on several criteria, which are presented in the following equations [28, 46]:

$$\begin{aligned}
\text{normalised root mean square (NRMSE)} &= \frac{\sqrt{\sum_{i=1}^{N}\left(f_{obs_i}-f_{est_i}\right)^2/N}}{f_{obs_{\max}}-f_{obs_{\min}}},\\
\text{Wilmott Index (WI)} &= \frac{\sum_{i=1}^{N}\left(f_{obs_i}-f_{est_i}\right)^2}{\sum_{i=1}^{N}\left(\left|f_{obs_i}-\overline{f}\right|+\left|f_{est_i}-\overline{f}\right|\right)^2},\\
\text{mean absolute square (MAE)} &= \frac{1}{N}\sum_{i=1}^{N}\left|f_{obs_i}-f_{est_i}\right|,\\
\text{mean absolute relative square (MARE)} &= \frac{1}{N}\sum_{i=1}^{N}\left|\frac{f_{obs_i}-f_{est_i}}{f_{obs_i}}\right|,\\
\text{maximum absolute relative square (MXARE)} &= \max\left(\left|\frac{f_{obs_i}-f_{est_i}}{f_{obs_i}}\right|\right), \quad for\ i=1,\ldots,N\\
\text{Nash Sutcliffe criterion (NSE)} &= \left[1-\frac{\sum_{i=1}^{N}\left(f_{obs_i}-f_{est_i}\right)^2}{\sum_{i=1}^{N}\left(f_{obs_i}-\overline{f}\right)^2}\right],
\end{aligned} \tag{8}$$

where f_{obs} is the observed infiltration rate, $f_{obs_{\max}}$ and $f_{obs_{\min}}$ are the maximum and minimum observed infiltration rates, f_{est} is the estimated infiltration rate at any time, and $\overline{f}$ is the mean of the observed infiltration capacity. Nash criterion has been widely used as an indicator for goodness of fit, while its value ranges from 0.0 to 1.0. The higher values of NSE indicate a better agreement between measured and estimated data. Similarly, WI values close to unity represent the best-fitted model. However, SSE, NRMSE, MAE, MARE, and MXARE should be as low as possible for the model with highest accuracy.

2.9. Sensitivity Analysis. In a bid to determine how much the results achieved by a typical model are sensitive to each input parameter, a sensitivity analysis can be conducted [47]. In this study, the parentage of the sensitivity analysis (SA) of the infiltration rate in respect of each input parameter (time, sand percentage, and density), which were selected based on Sihag et al.'s [28] study, is computed using [48]

$$SA_i = \frac{IR_{\max}(x_i)-IR_{\min}(x_i)}{\sum_{i=1}^{N}\left[IR_{\max}(x_i)-IR_{\min}(x_i)\right]}\times 100, \tag{9}$$

where $IR_{\max}(x_i)$ and $IR_{\min}(x_i)$ are the minimum and maximum infiltration rate determined by considering the variation of the input parameter (x_i) when each one of other input parameters are set as their average values. The more the SA percentage for a specific input variable, the higher the model is sensitive to that variable.

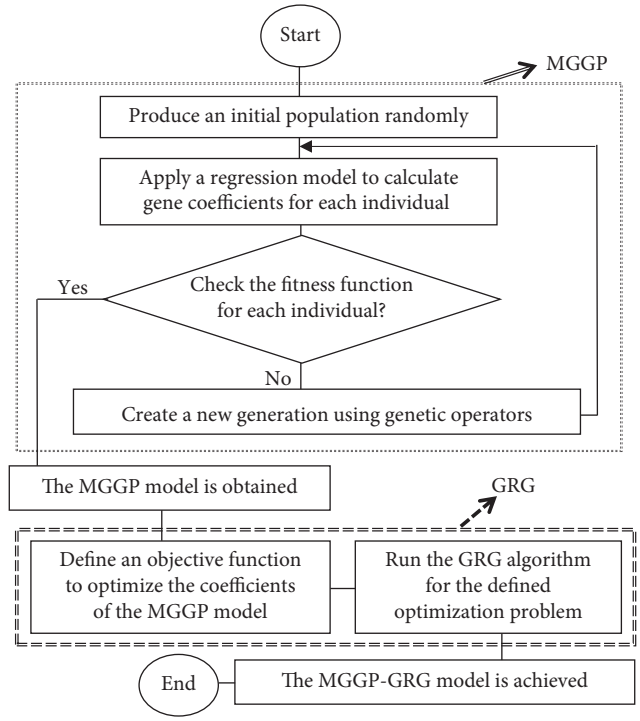


Figure 2: Flowchart of the hybrid MGGP-GRG for estimating infiltration rates.

2.10. Reliability Analysis. The reliability analysis is basically conducted to investigate the overall consistency of a prediction model. For this analysis, the relative error for each data point is achieved by the estimation model and compared with a threshold. Then, the number of cases, which have an equal or lower relative error than the threshold specified, is divided by the total number of points. Finally, the aforementioned ratio in the percentage would be the reliability metric, which demonstrates how reliable the prediction model performs in accordance with the desirable threshold. In this study, the reliability analysis was carried out for all methods used for predicting the infiltration rate, while the threshold was selected to be 20% based on the literature [49].

3. Results and Discussion

Accurate estimation of infiltration rate plays a vital role in various aspects of watershed hydrology. The present work focuses on improving the estimates of the infiltration rate through application of different soft computing approaches. The infiltration rates estimated by these techniques were compared with those approximated by the conventional infiltration models (Horton's model, modified Kostiakov's model, and Philip's model). In the conventional infiltration model, the observed infiltration rates and time were used as input data in accordance to model equations to obtain the estimated infiltration rate. On the other hand, in MLR, GRG, ANN, MGGP, and the hybrid MGGP-GRG models, the observed infiltration rates, time, sand percentage, and density were used as the input variables to obtain the estimated infiltration rates.

3.1. Comparison of the Conventional Infiltration Models. Table 2 presents the model parameters obtained in the training phase for the three conventional infiltration models. For the test phase, these parameters were used to estimate the infiltration rate based on equations (1), (3), and (4).

The results of different approaches considered in the present study were compared with respect to four criteria for both train and test data. This comparative analysis is shown in Table 3. In this comparison, the same data divisions were considered for all methods. The metrics used for comparing different infiltration models are given in Table 3. Based on Table 3, it may be observed that the performance of Horton's model was the worst for both training and testing parts of data. The modified Kostiakov's model improved the estimates of the infiltration rate by almost 4% and 10% as compared to those of Horton's model during training and testing, respectively. The performances of Philip's model and modified Kostiakov's model were almost comparable.

3.2. Comparison of the Conventional Models with Soft Computing Approaches. A perusal of Table 3 reveals that the technique used to model infiltration rates influences the estimates of the infiltration rate considerably. It can be observed that MLR provides the worst estimates of infiltration rates, which may depict the nonlinear nature of the infiltration process. The conventional models provide slightly better predictions of infiltration rates as compared to those obtained by MLR. Application of GRG solver further improves the estimate of infiltration as equation (7) involves a higher nonlinearity and more number of parameters as compared to equations (1)-(4). Before the application of

Table 2: Parameters obtained for the conventional infiltration models.

Models	Calibrated parameters		
Horton	Fc (cm/min) = 0.080	$f0$ (cm/min) = 0.535	k = 0.046
Modified Kostiakov	a = 1.699	b = 0.464	fi = 0.081
Philip	s (cm/min0.5) = 1.585	K (cm/min) = 0.061	—

Table 3: Comparative statistics for fit of model to the observed infiltration data.

Methods		Training phase					Testing phase				
		SSE (cm^2/min^2)	NRMSE	WI	MAE (cm/min)	NSE	SSE (cm^2/min^2)	NRMSE	WI	MAE (cm/min)	NSE
Conventional models	Horton	4.645	0.135	0.610	0.123	0.236	1.650	0.149	0.636	0.127	0.319
	Modified Kostiakov	4.485	0.133	0.613	0.118	0.263	1.474	0.141	0.686	0.119	0.392
	Philip	4.482	0.133	0.616	0.117	0.263	1.474	0.141	0.687	0.119	0.392
MLR		4.710	0.202	0.136	0.123	0.242	1.668	0.210	0.150	0.128	0.312
GRG		3.943	0.184	0.124	0.104	0.352	0.837	0.148	0.106	0.101	0.655
AI-based models	ANN	0.097	0.020	0.996	0.018	0.984	0.380	0.071	0.954	0.051	0.843
	MGGP	0.838	0.057	0.962	0.059	0.862	0.487	0.081	0.938	0.071	0.798
	MGGP-GRG	0.836	0.057	0.962	0.059	0.862	0.483	0.081	0.938	0.070	0.801

MGGP and the hybrid MGGP-GRG model, the observed infiltration rates were normalized as $i = (f_i - f_{\min})/(f_{\max} - f_{\min})$, where i, f_{min}, and f_{max} are the normalized, minimum, and maximum discharges of the ith observation. The normalized infiltration rate obtained from MGGP and the hybrid MGGP-GRG model are presented by the following equations, respectively:

$$i = \frac{0.01731\tanh(psqroot(t))}{\cos(\exp(s))} - 3.651\sin(\sin(s+d)) - 0.3395\,psqroot\left|\frac{psqroot(t)}{s+d}\right| - \frac{0.02371\tanh(\cos(d))}{\cos(\exp(s))} - 2.778\cos(\sin(s+d)) + 4.863, \tag{10}$$

$$i = \frac{0.01723\tanh\left(t^{0.4595}\right)}{\cos(\exp(s))} - 3.624\sin(\sin(s+d)) - 0.3395\,psqroot\left|\frac{psqroot(t)}{s+d}\right| - \frac{0.02399\tanh(\cos(d))}{\cos(\exp(s))} - 2.7507\cos(\sin(s+d)) + 4.8255, \tag{11}$$

where $psqroot(t) = \begin{cases} \sqrt{t} & t > 0 \\ 0 & t \le 0 \end{cases}$.

Figures 3–10 present the relative error plots obtained from different conventional infiltration models and computational techniques during training and testing. These figures also compare different methods based on MARE and MXARE for both train and test data. Although the relative error plots of the conventional infiltration models and other computational techniques (MLR, GRG, MGGP, and the hybrid MGGP-GRG) followed a similar sequence, the nature of relative error plots of ANN followed a different pattern during both training and testing. It may also be observed from Figures 3–10 that the relative errors achieved by ANN are the least as compared to others. On the other hand, relative errors obtained by Horton's infiltration model were the highest as compared to others. Furthermore, the AI-based models (ANN, MGGP, and the hybrid MGGP-GRG), which consider three independent variables (t, s, and d) instead of one variable (t), achieved much better MARE and MXARE in comparison with the empirical models during the training and testing phases. According to Figures 3–10, ANN and the hybrid MGGP-GRG resulted in the first and second best MARE and MXARE values, whereas MLR and Horton's model yielded to the first and second worst MARE and MXARE values for the train and test data.

Figures 11 and 12 depict the comparison between the observed and estimated infiltration rates obtained by the best-fit model (ANN and the hybrid MGGP-GRG) and the worst-fit model (Horton's model). It may be observed from Figure 11 that the infiltration rates estimated by ANN almost fit the observed data during training phase. On the other hand, the infiltration rates predicted by Horton's model deviated significantly from the observed data. The performance of the hybrid MGGP-GRG was better than that of Horton's model but poorer than that of ANN. During the testing phase, the estimates of the hybrid MGGP-GRG and ANN were almost identical as shown in Figure 12. The

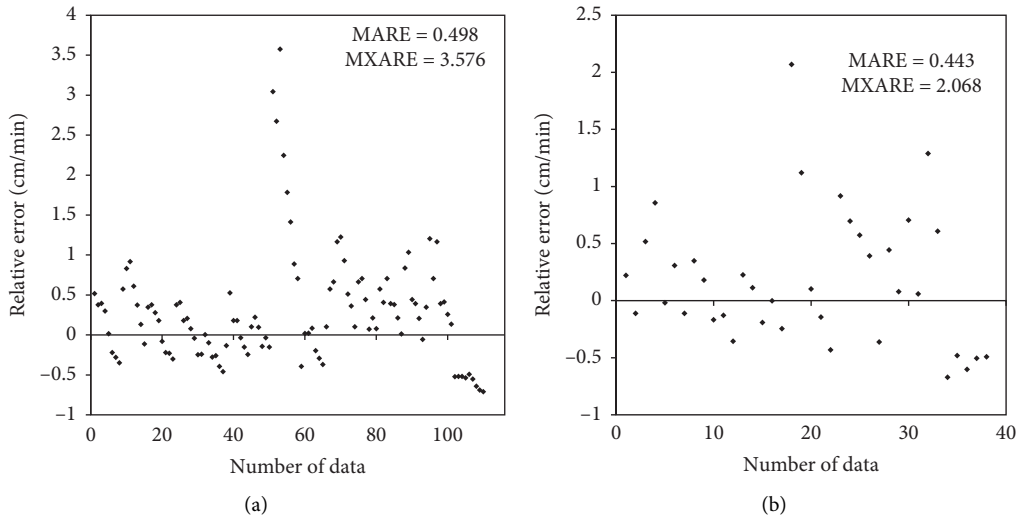


Figure 3: Relative error plots for Horton's model for the training and testing data.

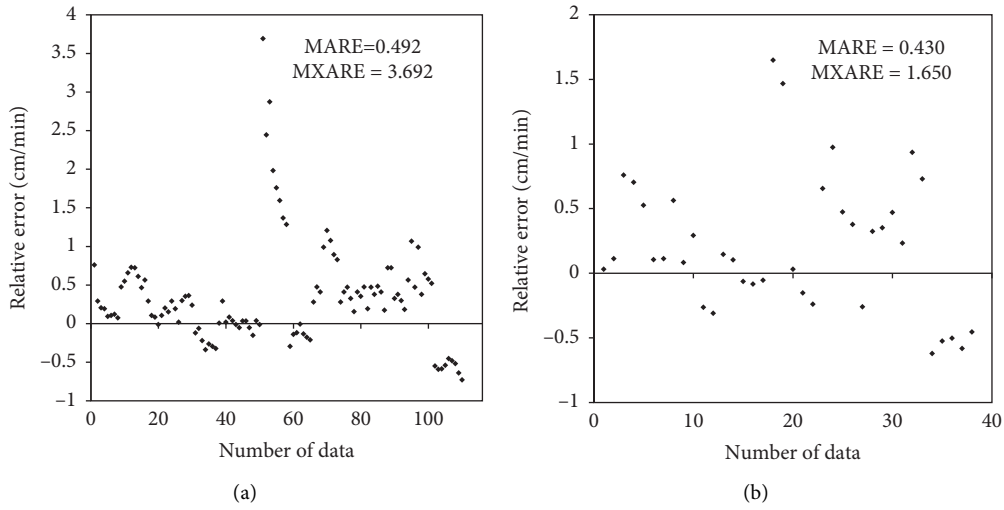


Figure 4: Relative error plots for modified Kostiakov's model for the training and testing data.

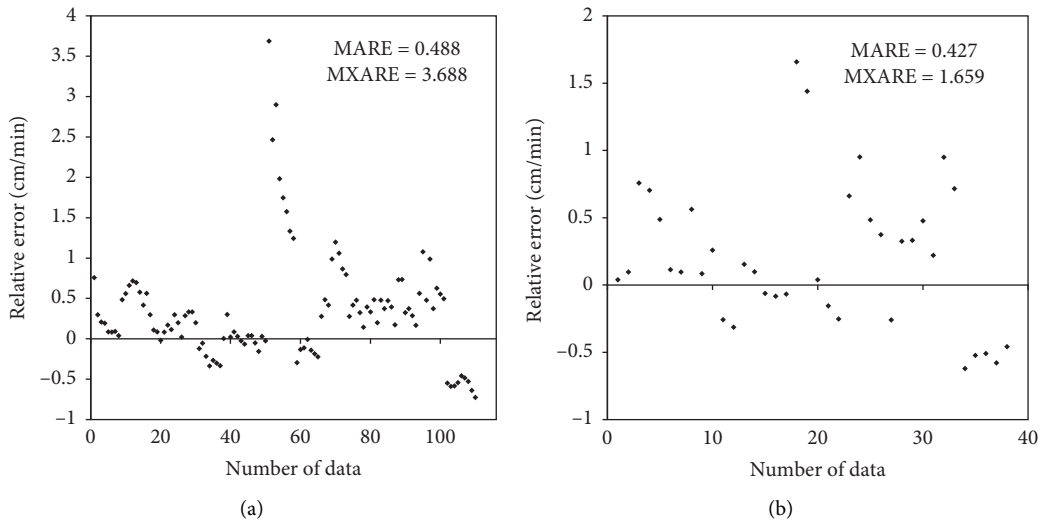


Figure 5: Relative error plots for Philip's model for the training and testing data.

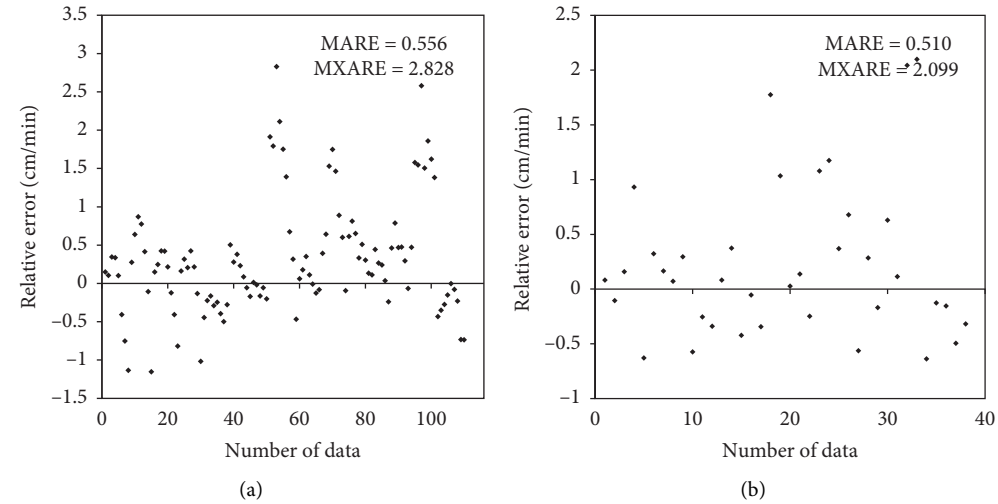


Figure 6: Relative error plots for the MLR model for the training and testing data.

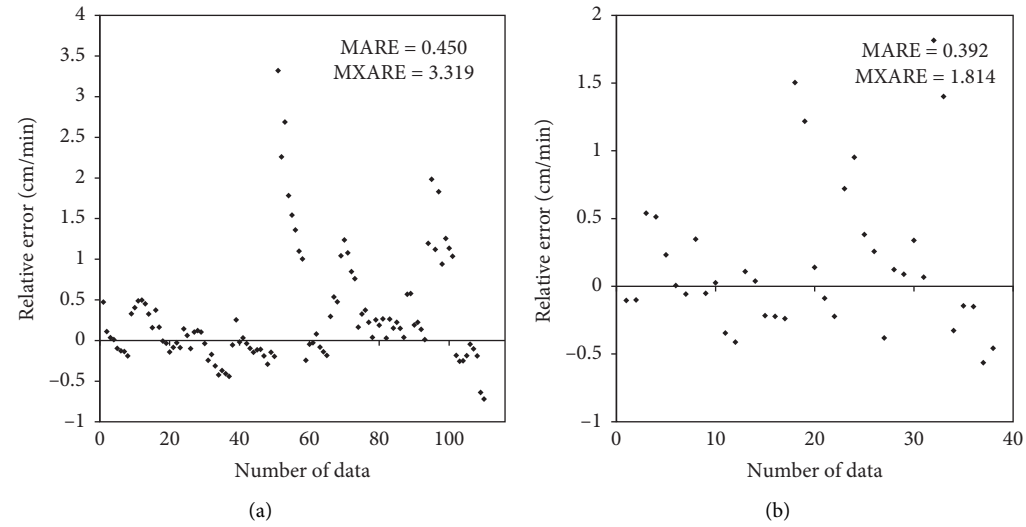


Figure 7: Relative Error plots for the GRG model for the training and testing data.

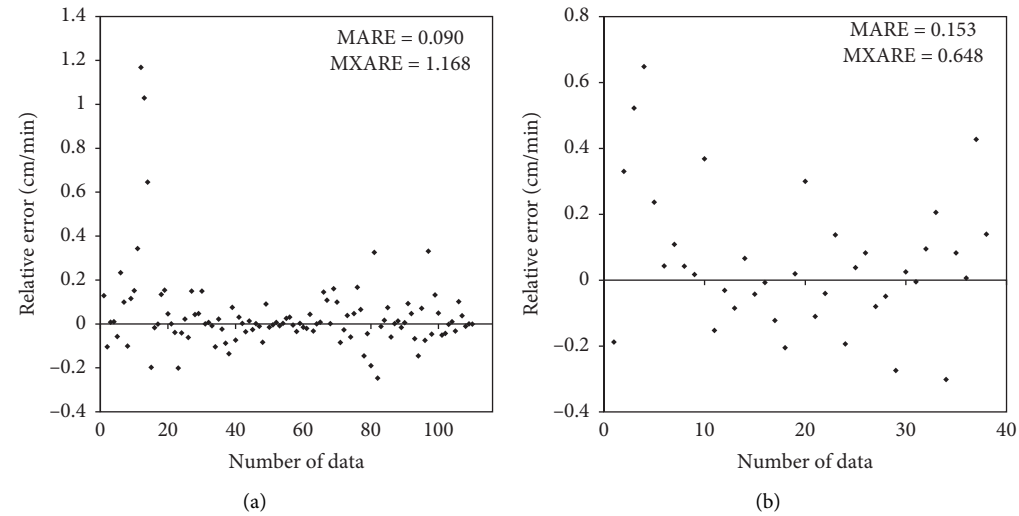


Figure 8: Relative error plots for the ANN model for the training and testing data.

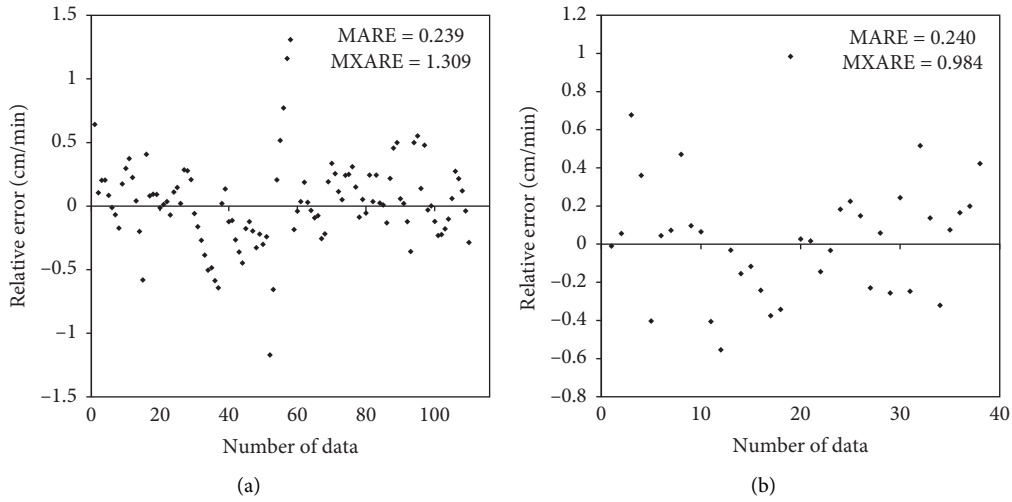


Figure 9: Relative error plots for the MGGP model for the training and testing data.

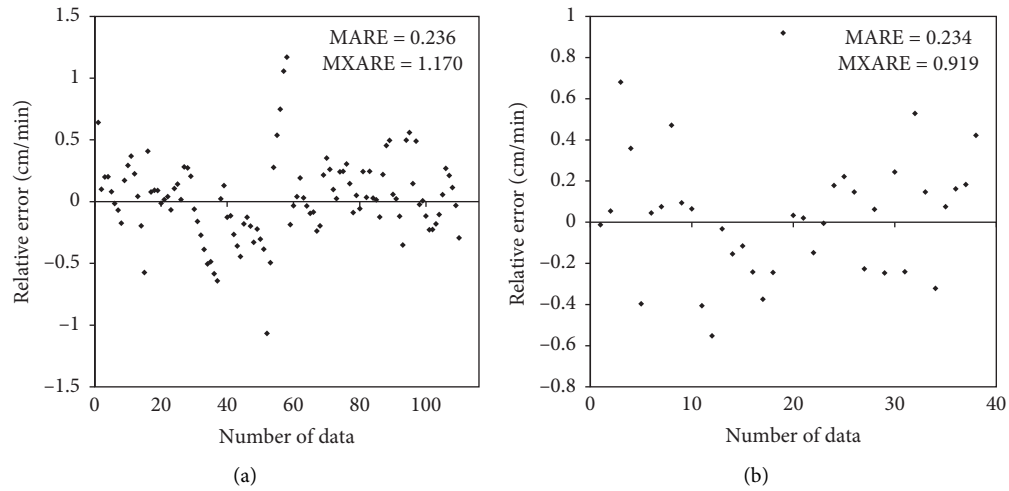


Figure 10: Relative error plots for the hybrid MGGP-GRG model for the training and testing data.

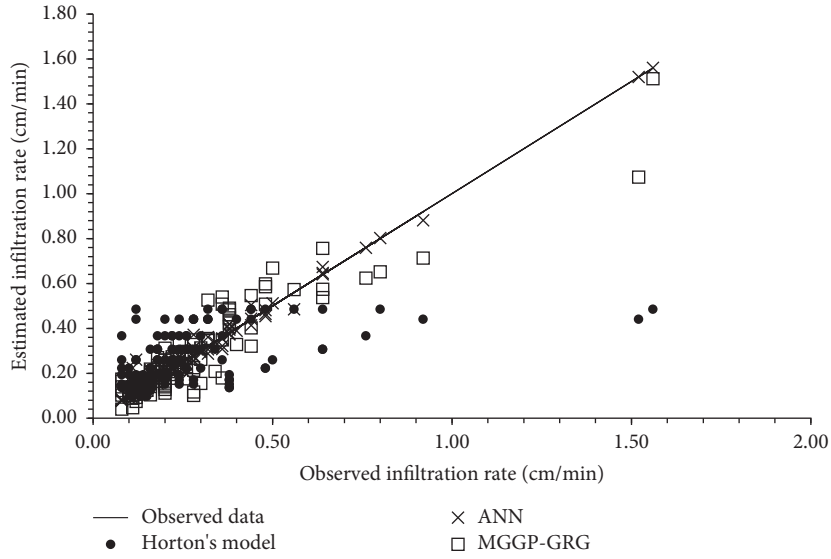


Figure 11: Estimated versus observed infiltration rates during the training phase.

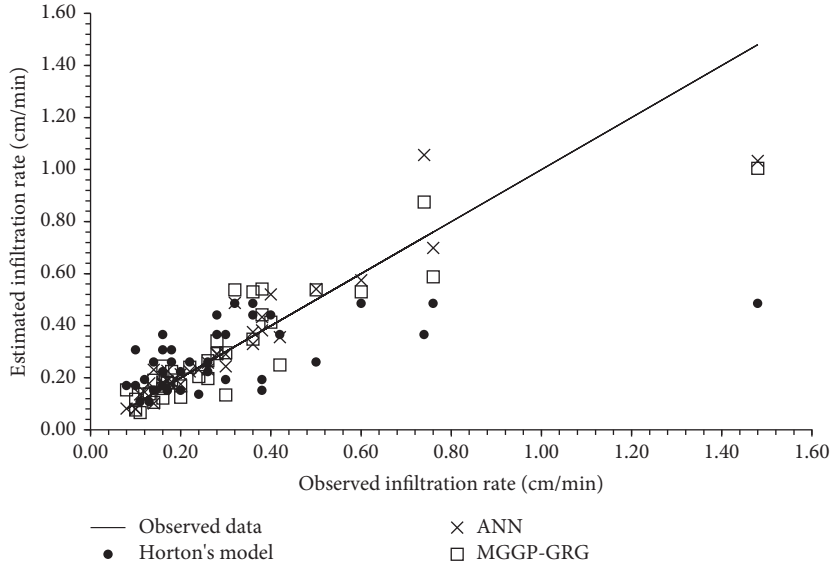


Figure 12: Estimated versus observed infiltration rates during the testing phase.

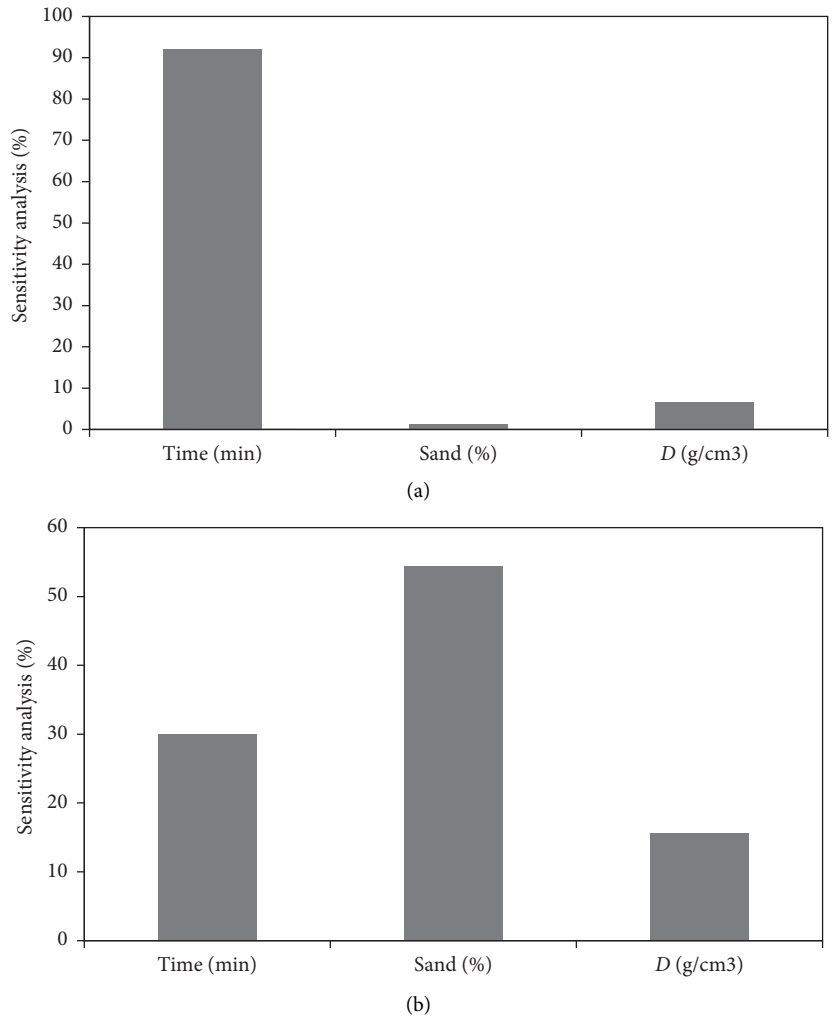


Figure 13: Results of the sensitivity analysis based on (a) ANN and (b) MGGP.

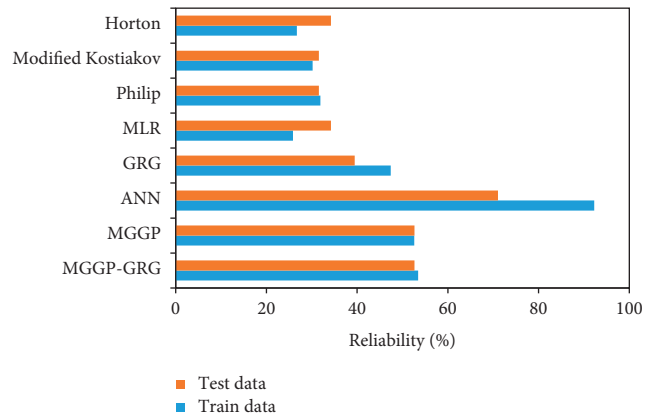


Figure 14: Results of the reliability analysis for train and test data.

estimates obtained by Horton's model during the testing phase were again significantly different from the corresponding observed values. Hence, Figures 11 and 12 obviously demonstrate how much the infiltration estimations can be enhanced by considering other variables involved in the process in addition to time, while they clearly indicate the better performances of AI-based models in comparison with those of the available empirical equations.

Figure 13 depicts the results of the sensitivity analysis, which was conducted for ANN and MGGP. As shown, time has the highest SA percentage (SA = 91.98%) for ANN, which implies that the infiltration rates predicted by ANN are mostly sensitive to time in comparison with other two input variables (sand percentage and density). This achievement is in agreement with the fact that the empirical models (such as Horton's and modified Kostiakov) used for estimating infiltration rates rely only on time. On the other hand, MGGP-based model, which yielded a lower accuracy for predicting infiltration rates than ANN, was found to be more sensitive to sand percentage than to time. Therefore, as infiltration rates may be affected by time based on the physical background of the problem statement, the results of the sensitivity analysis also indicate that ANN estimated infiltration rates better than the MGGP-based model.

The reliability analysis was carried out for the train and test data separately. The results of this analysis are presented in Figure 14. As shown, ANN achieved the highest percentages of reliability for both the train and test data. Furthermore, the reliability percentages obtained by MGGP and the hybrid MGGP-GRG were higher than those of empirical model, MLR, and GRG. Finally, the reliability analysis conducted in this study reveals the improvement made by the AI models over other data-driven methods available in the literature for predicting infiltration rates.

The structure of the equations developed by the conventional infiltration models, MLR and GRG, are known in advance of applying these methods. On the other hand, ANN, MGGP, and the hybrid MGGP-GRG are highly nonlinear techniques with greater degrees of freedom and complexity and, therefore, provide better estimates of the infiltration rate. However, more precise results are obtained by ANN, MGGP, and the hybrid MGGP-GRG at the expense of higher computational efforts. These machine learning tools require a considerable number of runs, unlike the conventional models and MLR in which a single attempt is sufficient for determining the model output. Based on the comparative analysis conducted in this study, ANN certainly yielded to the best estimates of infiltration rates. However, the estimates obtained from the hybrid MGGP-GRG were also comparable, especially, for the test data. Furthermore, unlike ANN, the hybrid MGGP-GRG model provided explicit equations for predicting infiltration rates, which can be implemented in a typical hydrological modelling or preferred in practice by engineers, which may be counted as an advantage of this AI-based technique.

4. Conclusions

In the present study, published infiltration data was used to assess the performances of MGGP and the hybrid MGGP-GRG technique in modelling the infiltration rates of soil. The estimated infiltration rates were compared with those obtained by the conventional models (Horton's model, Philip's model, and modified Kostiakov's model). It was observed that application of the hybrid MGGP-GRG and MGGP improved the estimates of infiltration rates as compared to the conventional infiltration model by over 80%. On the other hand, ANN provided the best estimates of infiltration rates. In addition to the accuracy improvement, the application of ANN, MGGP, and the hybrid MGGP-GRG increased the complexity of modelling equations. Future studies may focus on the comparison of the hybrid MGGP-based models with the other machine learning approaches, while applying the explicit infiltration models developed by either MGGP or the hybrid MGGP-GRG in hydrological models is anticipated in favor of assessing their performances in practice.

Data Availability

The data used in this study are available in the related literature.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

[1] M. Zakwan, M. Muzzammil, and J. Alam, "Application of spreadsheet to estimate infiltration parameters," *Perspectives in Science*, vol. 8, pp. 702–704, 2016.

[2] M. Zakwan, "Comparative analysis of the novel infiltration model with other infiltration models," *Water and Environment Journal*, vol. 33, no. 4, pp. 620–632, 2019.

[3] H. J. Morel-Seytoux, "Infiltration modeling component in a large-scale hydrologic model," *Journal of Hydraulic Research*, vol. 42, no. 6, pp. 666-667, 2004.

[4] M. Šraj, L. Dirnbek, and M. Brilly, "The influence of effective rainfall on modeled runoff hydrograph," *Journal of Hydrology and Hydromechanics*, vol. 58, no. 1, pp. 3–14, 2010.

[5] Y. Wen, C. Hu, G. Zhang, and S. Jian, "Response of the parameters of excess infiltration and excess storage model to land use cover change," *Journal of Hydrology and Hydromechanics*, vol. 68, no. 2, pp. 99–110, 2020.

[6] S. K. Mishra, J. V. Tyagi, and V. P. Singh, "Comparison of infiltration modelsfiltration models," *Hydrological Processes*, vol. 17, no. 13, pp. 2629–2652, 2003.

[7] K. Deep and K. N. Das, "Optimization of Infiltration parameters in hydrology," *World Journal of Modelling and Simulation*, vol. 4, pp. 120–130, 2008.

[8] A. Haghiabi, J. Abedi-koupai, M. Heidarpour, and J. Mohammadzadeh-habili, "A new method for estimating parameters of Kostiakov and Modified Kostiakov infiltration equations," *World Applied Science Journal*, vol. 15, pp. 129–135, 2011.

[9] M. Zakwan, "Assessment of dimensionless form of Kostiakov model," *Aquademia: Water, Environment and. Technology*, vol. 1, no. 1, p. 01, 2017.

[10] L. Chen, L. Xiang, M. H. Young, J. Yin, Z. Yu, and M. T. v. Genuchten, "Optimal parameters for the Green-Ampt infiltration model under rainfall conditions," *Journal of Hydrology and Hydromechanics*, vol. 63, no. 2, pp. 93–101, 2015.

[11] G. Tayfur, "Modern optimization methods in water resources planning, engineering and management," *Water Resources Management*, vol. 31, no. 10, pp. 3205–3233, 2017.

[12] A. D. Mehr, V. Nourani, E. Kahya, B. Hrnjica, A. M. Sattar, and Z. M. Yaseen, "Genetic programming in water resources engineering: a state-of-the-art review," *Journal of Hydrology*, vol. 566, pp. 643–667, 2018.

[13] C. Shen, "A transdisciplinary review of deep learning research and its relevance for water resources scientists," *Water Resources Research*, vol. 54, no. 11, pp. 8558–8593, 2018.

[14] H. Tyralis, G. Papacharalampous, and A. Langousis, "A brief review of random forests for water scientists and practitioners and their recent history in water resources," *Water*, vol. 11, no. 5, p. 910, 2019.

[15] S. Kambalimath and P. C. Deka, "A basic review of fuzzy logic applications in hydrology and water resources," *Applied Water Science*, vol. 10, no. 8, pp. 1–14, 2020.

[16] M. Sit, B. Z. Demiray, Z. Xiang, G. J. Ewing, Y. Sermet, and I. Demir, "A comprehensive review of deep learning applications in hydrology and water resources," *Water Science and Technology*, vol. 82, no. 12, pp. 2635–2670, 2020.

[17] M. Pandey, M. Zakwan, M. A. Khan, and S. Bhave, "Development of scour around a circular pier and its modelling using genetic algorithm," *Water Supply*, vol. 20, no. 8, pp. 3358–3367, 2020.

[18] M. Najafzadeh and G. Oliveto, "Riprap incipient motion for overtopping flows with machine learning models," *Journal of Hydroinformatics*, vol. 22, no. 4, pp. 749–767, 2020.

[19] M. Najafzadeh and G. Oliveto, "More reliable predictions of clear-water scour depth at pile groups by robust artificial intelligence techniques while preserving physical consistency," *Soft Computing*, vol. 25, pp. 5723–5746, 2021.

[20] M. Niazkar and M. Zakwan, "Assessment of artificial intelligence models for developing single-value and loop rating curves," *Complexity*, vol. 2021, Article ID 6627011, 21 pages, 2021.

[21] N. Niazkar, N. Talebbeydokhti, and S. H. Afzali, "One dimensional hydraulic flow routing incorporating a variable grain roughness coefficient," *Water Resources Management*, vol. 33, no. 13, pp. 4599–4620, 2019.

[22] M. Kumar and P. Sihag, "Assessment of infiltration rate of soil using empirical and machine learning-based models," *Irrigation and Drainage*, vol. 68, no. 3, pp. 588–601, 2019.

[23] A. Z. Dewidar, H. Al-Ghobari, and A. Alataway, "Developing a fuzzy logic model for predicting soil inltration rate based on soil texture properties," *Water SA*, vol. 45, no. 3, pp. 400–410, 2019.

[24] G. T. Patle, T. T. Sikar, K. S. Rawat, and S. K. Singh, "Estimation of infiltration rate from soil properties using regression model for cultivated land," *Geology, Ecology, and Landscapes*, vol. 3, no. 1, pp. 1–13, 2019.

[25] P. Sihag, N. K. Tiwari, and S. Ranjan, "Support vector regression-based modeling of cumulative infiltration of sandy soil," *ISH Journal of Hydraulic Engineering*, vol. 26, no. 1, pp. 44–50, 2020.

[26] M. R. Pahlavan-Rad, K. Dahmardeh, M. Hadizadeh et al., "Prediction of soil water infiltration using multiple linear regression and random forest in a dry flood plain, eastern Iran," *Catena*, vol. 194, Article ID 104715, 2020.

[27] A. Sepahvand, B. Singh, M. Ghobadi, and P. Sihag, "Estimation of infiltration rate using data-driven models," *Arabian Journal of Geosciences*, vol. 14, no. 1, pp. 1–11, 2021.

[28] P. Sihag, V. P. Singh, A. Angelaki, V. Kumar, A. Sepahvand, and E. Golia, "Modelling of infiltration using artificial intelligence techniques in semi-arid Iran," *Hydrological Sciences Journal*, vol. 64, no. 13, pp. 1647–1658, 2019.

[29] R. I. Horton, "The interpretation and application of runoff plot experiments with reference to soil erosion problems," *Soil Science Society of America Proceedings*, vol. 3, pp. 340–349, 1938.

[30] J. R. Philip, "Theory of infiltration," *Soil Science*, vol. 83, pp. 345–357, 1957.

[31] A. N. Kostiakov, "On the dynamics of the coefficients of water percolation in soils," in *Sixth Commission*, pp. 15–21, International Society of Soil Science, Part A, Vienna, AT, USA, 1932.

[32] R. E. Smith, "The infiltration envelope: results from a theoretical infiltrometer," *Journal of Hydrology*, vol. 17, pp. 1–21, 1972.

[33] M. Pandey, M. Zakwan, P. K. Sharma, and Z. Ahmad, "Multiple linear regression and genetic algorithm approaches to predict temporal scour depth near circular pier in non-cohesive sediment," *ISH Journal of Hydraulic Engineering*, pp. 1–8, 2018.

[34] M. Zakwan, M. Muzzammil, and J. Alam, "Estimation of soil properties using infiltration data," in *Proceedings of the*

National Conference of Advanced Geotechnological Engineering, pp. 198–201, Aligarh, India, December 2016.

[35] M. Muzzammil, J. Alam, and M. Zakwan, "A spreadsheet approach for prediction of rating curve parameters," in *Hydrologic Modeling. Water Science and Technology Library*, V. Singh, S. Yadav, and R. Yadava, Eds., Vol. 81, Springer, Singapore, 2018.

[36] A. R. Nawaz, M. Zakwan, I. Khan, and Z. A. Rahim, "Comparative analysis of variants of Muskingum model," *Water and Energy International*, vol. 63, no. 7, pp. 64–73, 2020.

[37] M. Niazkar and S. H. Afzali, "Developing a new accuracy-improved model for estimating scour depth around piers using a hybrid method," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 43, no. 2, pp. 179–189, 2018.

[38] M. Niazkar, N. Talebbeydokhti, and S.-H. Afzali, "Bridge backwater estimation: a Comparison between artificial intelligence models and explicit equations," *Scientia Iranica*, vol. 28, pp. 573–585, 2020.

[39] M. Niazkar, F. Hajizadeh mishi, and G. Eryılmaz Türkkan, "Assessment of artificial intelligence models for estimating lengths of gradually varied flow profiles," *Complexity*, vol. 2021, Article ID 5547889, 11 pages, 2021.

[40] M. Niazkar, "Revisiting the estimation of colebrook friction factor: a comparison between artificial intelligence models and C-W based explicit equations," *KSCE Journal of Civil Engineering*, vol. 23, no. 10, pp. 4311–4326, 2019.

[41] M. Niazkar, "Assessment of artificial intelligence models for calculating optimum properties of lined channels," *Journal of Hydroinformatics*, vol. 22, no. 5, pp. 1410–1423, 2020.

[42] M. Niazkar, N. Talebbeydokhti, and S. H. Afzali, "Novel grain and form roughness estimator scheme incorporating artificial intelligence models," *Water Resources Management*, vol. 33, no. 2, pp. 757–773, 2019.

[43] H. R. Niazkar and M. Niazkar, "COVID-19 outbreak: application of multi-gene genetic programming to country-based prediction models," *Electronic Journal of General Medicine*, vol. 17, no. 5, p. em247, 2020.

[44] D. Searson, *GPTIPS: Genetic Programming and Symbolic Regression for MATLAB*, https://www.researchgate.net/publication/277297934_GPTIPS_Genetic_Programming_Symbolic_Regression_for_MATLAB_User_Guide, 2009.

[45] J. S. Lee and K. D. Suh, "Development of stability formulas for rock armor and tetrapods using multigene genetic programming," *Journal of Waterway, Port, Coastal, and Ocean Engineering*, vol. 146, no. 1, Article ID 04019027, 2020.

[46] H. R. Niazkar and M. Niazkar, "Application of artificial neural networks to predict the COVID-19 outbreak," *Global Health Research and Policy*, vol. 5, no. 1, p. 50, 2020.

[47] M. Niazkar and S. H. Afzali, "Assessment of modified honey bee mating optimization for parameter estimation of nonlinear muskingum models," *Journal of Hydrologic Engineering*, vol. 20, no. 4, Article ID 04014055, 2015.

[48] A. Garg, K. Tai, V. Vijayaraghavan, and P. M. Singru, "Mathematical modelling of burr height of the drilling process using a statistical-based multi-gene genetic programming approach," *The International Journal of Advanced Manufacturing Technology*, vol. 73, no. 1-4, pp. 113–126, 2014.

[49] F. Saberi-Movahed, M. Najafzadeh, and A. Mehrpooya, "Receiving more accurate predictions for longitudinal dispersion coefficients in water pipelines: training group method of data handling using extreme learning machine conceptions," *Water Resources Management*, vol. 34, no. 2, pp. 529–561, 2020.

Hindawi
Complexity
Volume 2021, Article ID 5525271, 10 pages
https://doi.org/10.1155/2021/5525271

Research Article

An Improved Artificial Neural Network Model for Effective Diabetes Prediction

Muhammad Mazhar Bukhari,[1] Bader Fahad Alkhamees,[2] Saddam Hussain,[3] Abdu Gumaei,[4] Adel Assiri,[5] and Syed Sajid Ullah[3]

[1]*Department of Computer Science, National College of Business Administration & Economics, Lahore, Pakistan*
[2]*Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia*
[3]*IT Department, Hazara University, Mansehra 21120, KP, Pakistan*
[4]*Research Chair of Pervasive and Mobile Computing, Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia*
[5]*Management Information Systems Department, College of Business, King Khalid University, Abha 61421, Saudi Arabia*

Correspondence should be addressed to Saddam Hussain; saddamicup1993@gmail.com and Abdu Gumaei; abdugumaei@gmail.com

Received 5 February 2021; Revised 19 March 2021; Accepted 1 April 2021; Published 22 April 2021

Academic Editor: Michela Gelfusa

Data analytics, machine intelligence, and other cognitive algorithms have been employed in predicting various types of diseases in health care. The revolution of artificial neural networks (ANNs) in the medical discipline emerged for data-driven applications, particularly in the healthcare domain. It ranges from diagnosis of various diseases, medical image processing, decision support system (DSS), and disease prediction. The intention of conducting the research is to ascertain the impact of parameters on diabetes data to predict whether a particular patient has a disease or not. This paper develops an improved ANN model trained using an artificial backpropagation scaled conjugate gradient neural network (ABP-SCGNN) algorithm to predict diabetes effectively. For validating the performance of the proposed model, we conduct a large set of experiments on a Pima Indian Diabetes (PID) dataset using accuracy and mean squared error (MSE) as evaluation metrics. We use different number of neurons in the hidden layer, ranging from 5 to 50, to train the ANN models. The experimental results show that the ABP-SCGNN model, containing 20 neurons, attains 93% accuracy on the validation set, which is higher than using the other ANNs models. This result confirms the model's effectiveness and efficiency in predicting diabetes disease from the required data attributes.

1. Introduction

The revolution in artificial neural networks (ANNs) within the medical discipline research domain emerged in data-driven applications, particularly in the healthcare sector. It ranges from diagnosis of various kinds of diseases, image-processing in the medical field, decision support system (DSS), and disease prediction. Many contemporary ANN models such as deep learning models, recurrent neural networks, and genetic algorithms are significant in artificial intelligence, robotics, image processing, and several other cutting-edge technologies, especially in the health sector. ANN is an ideal tool that is used in the identification, analysis, and prediction of general and health sectors involving in Thrombo-embolic stroke disease, bone densitometry, hepatitis B, and breast cancer [1]. This paper presents the predictive framework for a successful diagnosis of diabetes using a dataset comprising female patients with corresponding attributes. It is quite pertinent to mention that typical regression models have been employed to solve the problem. Usually, these models' basis includes inference of statistical independence and interdependencies of their

input and output variables and uniformity of continuity and presence of external variables.

On the other hand, in most applications, these inferences are often offended or disregarded due to complicated physiological characteristics. To enhance patients' diabetes self-management, research in designing various models is in progress [2]. However, several models and programs have been developed to provide medical research benefits; rigorous testing of these models is partial and inadequate, resulting in very hard to perform and difficult to manage. Here, in this research, a well-organized and refined regression model, i.e., artificial backpropagation scaled conjugate gradient neural network (ABP-SCGNN), is proposed that predicts the underlying disease since ANN has been employed in medical research and analytical studies [3]. Backpropagation is supervised learning of ANN using gradient descent. The algorithm computes the gradient of the error function in connection with weights.

Diabetes is one of the diseases that affect problems associated with the hormone insulin. It may appear when the human body acts in response, particularly in a negative manner. Unfortunately, it has obtained the cure for nothing up till now, but people with diabetes can adopt certain precautions to manage their disease and enjoy a healthy life. This paper focuses on the successful prediction using ANN in the diagnostic approach by choosing some attributes carefully. Upon the positive predicted result, people may consult their doctors for prior precautions treatment. Typically, suffering from diabetes, patients become the victims of other serious diseases ranging from heart attack to kidney failure if not appropriately controlled [4]. In many countries, a significant cause of cardiovascular disease, blindness, kidney failure, and lower limb amputation is diabetes. The number of patients with diabetes has been increasing worldwide, nearly 200 million people are targeted, and more than half of the population was female. Females can quickly become victims of this disease during the age range from 25 to 44 and at high risk accordingly. Several research studies are in progress to predict diabetes for early diagnoses and cure hence [4, 5].

Such models can be designed and implemented using ANN that proved to be more helpful, efficient, and effective in several fields of medicine like analysis, diagnosis, and prediction and assist not only professionals but also ordinary people [5]. ANN is a representation of the human neural system mathematically, demonstrating the strength of training and generalization. Most of the ANN techniques are based on nonlinear functions in which the link or association of input features is either a bit complex or unknown. A series of nodes also called neurons that form an ANN is organized in different layers. In a typical statistical model of an ANN, each neuron is directly connected to the neurons of the other layers employing some weighted values that illustrate the strength or power of the connection between them [6, 7]. Each neuron input is affected with the weighted permutation of several input signals that may contain distinct computations and finally on the resultant output. These neurons apply transfer function to the weighted inputs to evaluate the threshold value. Using the activation function, the message is sent to the next concerned neuron if the threshold value exceeds.

Based on the functions of ANNs, it is significant to understand when it presents prediction, perception classification, and pattern recognition along with training accordingly [7]. Despite this, on the other hand, significant work is carried out towards ANN development for the applications in medical fields, for example, classifications, clustering, data optimization, and prediction based on a given set of inputs. Any ANN model consists of several components comprising layers, specifically an input layer, one or more hidden layers, an output layer, several neurons, and their corresponding interaction. Choosing these features is sensitive as few features result in slow training, and on the contrary, many features decrease the overall network processing power. There are different ways to determine these components, typically the pruning method and growing method. One can go with many features in the pruning method and reduce the network size with those not important components [8]. The growing process that was used in the proposed work adds desire components gradually. Several algorithms have been applied to perform various operations on the dataset used in this paper to select the attributes, processes the data, and predict diabetes accordingly. Diagnosis in the medical sector using the same diabetes dataset utilizing general regression neural networks (GRNNs), multilayer perceptron (MLP) neural networks, radial basis function (RBF), and feed-forward neural networks for comparing the performance of the MLP was tested for different types of backpropagation training algorithms [9]. The adaptive learning routine has been employed in [10].

The rest of the paper is organized as follows. Section 2 presents the previous related work of diabetes prediction. Section 3 explains the proposed methodology in which the ABP-SCGNN-based diabetes prediction framework is given in detail. Experiments and results are introduced in Section 4, and then a conclusion of the study with future work is shown in Section 5.

2. Literature Review

In 2011, Sapon et al. [1] took 250 diabetes patients who were both male and female ranging 25–78 years with 27 input variables to training the network to identify the disease pattern. Among the three algorithms, the Bayesian regulation algorithm presented the most excellent result in the prediction of diabetes to Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm and Quasi-Newton and Levenberg–Marquardt algorithms. BFGS Quasi-Newton possesses 0.86714 correlation coefficients with 578 epochs while Bayesian regulation acquires 0.99579 for 37 epochs only and Levenberg–Marquardt holds 0.6051 for only five epochs. In [1], the Bayesian regulation algorithm presented a good correlation between estimated targets and actual outputs (i.e., 0.99579) with 88.8% prediction accuracy that affirms the validation that exhibits the appropriateness of this algorithm to carry out the successful diabetes prediction.

For the sake of dataset classification, in 2012, Choubey et al. [8] applied a naïve Bayes (NBs) classifier, as well as a genetic algorithm (GA) with NBs methods to predict diabetes in females aged from 21–78 years. The total number of instances was 768. First, using naïve Bayes, the process of classification has been performed on PIDD, and a genetic algorithm was used to add and remove attributes from the dataset. It comparatively decreased the computational cost as well as time and increased ROC and classification accuracy. The result comparison on PIDD regarding accuracy with ROC, GA, and NB highlights the most accurate result and better ROC compared with other methods.

In 2003, Kayaer and Yıldırım [9] applied MLP neural networks, RBF, and GRNNs on the Pima Indians Diabetes data. The Levenberg–Marquardt training algorithm has demonstrated the best result against the training data. The accuracy of RBF was not better than that of MLP, even utilizing the all-underlying values. The GRNNs attained the best result using test data (i.e., 80.21%). This algorithm proves to be an excellent and pragmatic selection for the successful classification of diabetes data.

In 2016, Florez et al. [11] used 700 instances as training data that were selected randomly using software *R* to predict the intensity of diabetes. MSE was 2.952. Next, the higher risk subgroups of characteristics that lead to diabetes were obtained. The unmodified model is the number of times pregnant (PRG), plasma concentration in saliva (PLASMA), BP, body mass index (BODY), and diabetes pedigree function (PEDIGREE). PRG, PLASMA, BODY, and PEDIGREE have certain effects on predicting diabetes according to their coefficient. A large number of variables with a higher probability of containing diabetes possess the MSE to be 3.21068. Therefore, the concluding variables include PRG, PLASMA, BODY, and PEDIGREE.

Iqbal et al. [12] proposed an intelligent system for smart-cities using clusters and fuzzy inference systems to predict traffic congestion. The solution for traffic congestion is presented using smart mobile technologies. It motivates the idea for the prediction of diabetes using computational intelligence technologies.

Since the proposed framework is based on the ABP-SCGNN algorithm, the experiment might proceed with a small number of components, and a decision could be made on the performance of the learning curve. The features of the network can be concluded upon generating a small error rate when training complete.

3. Proposed Methodology

People of any age group may become the victim of diabetes. The reasons may vary among different age groups, gender, living style, glucose and insulin level, BP, and so on. At present, numerous algorithms, such as ANNs, SVM, and naïve Bayes (NBs) with fuzzy logic (FL), are incorporated for predicting the diagnosis of diabetes. These algorithms have their trade-offs regarding their processing time and accuracy rate, obtaining hidden information from the given data. Since in complex problem domains, finding an optimum solution may not require a trivial approach [13–16] and the proposed framework drawn in the following figure demonstrates the working process of ANN algorithms for predicting, validating, and systematically testing the network for the intended purpose to improve the self-reliance and significant certainty. Operations and parameters areyielded during ANN training, and a careful comparison is made on predicted and obtained values used for ANN optimization. The whole course of action is explained in Figure 1 schematically.

According to the framework, the network obtains dataset that consists of female diabetes patients' detail for predicting the disease diagnosis. Since the dataset contains different attributes/features (glucose, insulin, and BP) relevant to the patients' detail, a careful selection of these features must be performed as some features may contribute to misleading the result because of noise or null data. Sometimes, the value range among attributes is high. It is recommended to apply a formalized process for the sake of minimizing the erroneous result while finding a relatively improved result [9]. Data that need to train must be preprocessed before the evaluation process. Network architecture varies from the classifier to classifier, exhibiting the underlying algorithm parameters that are dependent on the classifier that is supposed to train the network. Other diabetes patients can use this network for predicting the diagnoses of their disease. To end with it, a general physician of diabetes or doctor of medicine performs an evaluation and appraisal process on this expected diagnosis. A detailed explanation of the proposed framework is provided as follows.

3.1. Data Collection and Representation. The dataset used in the study is obtained from the National Institute of Diabetes and Digestive and Kidney Diseases [17]. The purpose behind this is to predict the disease considering some selected diagnostic key attributes included in the dataset whether a person is a diabetes patient or not. The dataset contains the data for female patients only with minimum age 21 years for the resident of Arizona USA. The responder includes a binary value of either 0 (negative test representing diabetes) or 1 (positive test representing diabetes). Approximately 35% (268) patients are included in a class containing value 1. On the other hand, 65% (500) patients are included in another class containing a value of zero [9]. The researchers have mentioned many vital factors that may vary from patient to patient for diabetes currently or after a predefined period [18].

3.2. Feature Selection/Reduction. To diagnose diabetes correctly, key attributes are normally independent of each other. Medical experts examine the attributes carefully and decide if each is required in identifying the viable diagnosis. These attributes, called features in ANN, may be identified as diabetes symptoms and another relevant piece of information that assists in predicting diabetes. Choubey et al. [8] used GA for selecting attributes (features) and NBs for classification on PID dataset.

Since feature selection is a process of identifying the most relevant features extracted from the complete set, it

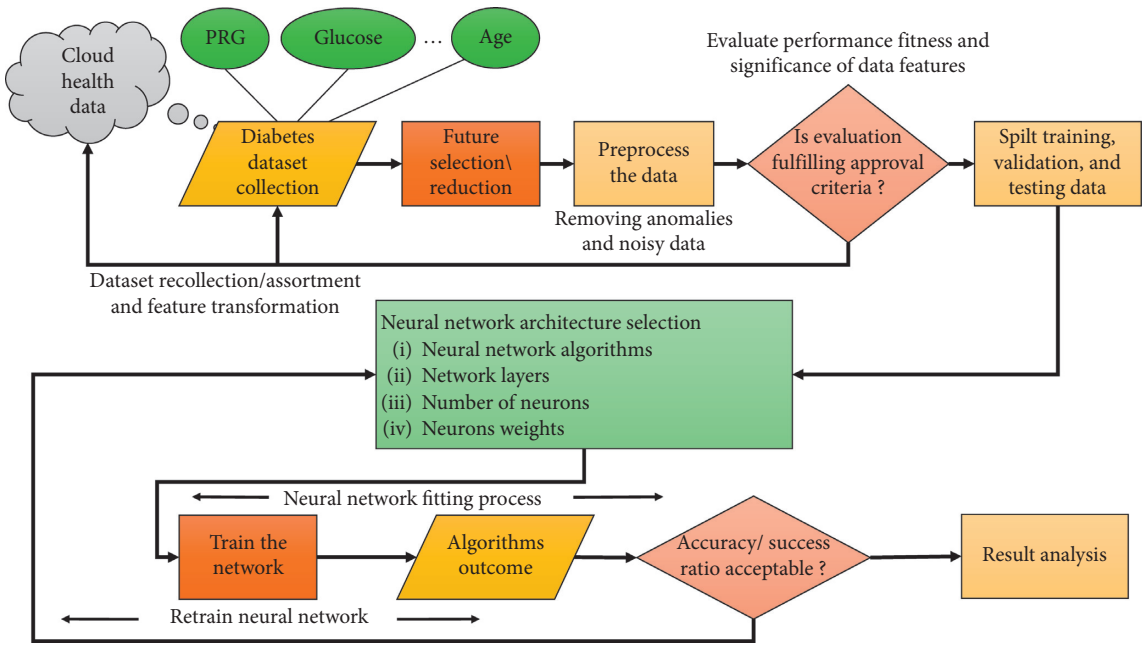


Figure 1: The proposed framework of diabetes prediction using neural network architecture selection.

increases the success ratio of predicting the diagnosis of diabetes patients [8]. Given the fact [18], these features must be robust and noise-free; hence, a careful selection is made in choosing these features with the entire dataset. Women may have diabetes from the sixth month of their pregnancy, and in addition to it, a certain level of glucose and insulin play a vital role in the diagnosis of diabetes. Eventually, there are eight explanatory input variables with one output responder variable that are included in the selected dataset. Each variable plays a vital role in diagnosing diabetes individually. Accumulating their values enables the network to be trained effectively and perform the diabetes prediction in return. The detail of both types of variables is explained in Table 1.

3.3. Preprocessing. Many methods are available to preprocess the data in the dataset before model evaluation [19, 20]. The preprocessed result must demonstrate the anticipated output. Learning rate, momentum, and time were taken; all are obtained in the response to the preprocessing process. Data must be transformed in the form of fulfilling the acceptance criterion. Accumulating the provided values in various features like blood pressure, skin thickness, insulin, and BMI must map with the outcome value. The diabetes dataset contains 768 instances with 8 input variables that are enough to predict the diabetes diagnosis using the simulation model. Noisy data have already been eliminated before receiving the dataset from the concerned authority [19].

3.4. Neural Network Fitting Process. The objective to accumulate the abovementioned information is to design such a neural network model that can predict precisely whether a particular patient possesses diabetes or not. Since ANN is the combination of many algorithms, some algorithms are used in the paper to train the network, obtain the prediction ratio, and given that compare the mutual results. After choosing the patients' dataset, the next step is to segregate the data according to the requirements. In our case, data are needed to be divided into three main segments as training, validation, and test data. Training data are presented to the network when the training session needs to commence while adjusting the network with an error reference. The diabetes dataset consists of 768 patients, and 70% (approximately 537) instances are used for training purposes. Multiple pieces of training may generate different results each time regarding various conditions as well as data. On the other hand, validation data need to measure the network generalization. When it reaches its optimum level, the validation process stops. Approximately 115 instances are selected to perform the validation process to evaluate the training impact pragmatically. In so far as test data are concerned, there is no effect or change on training data due to testing data and after training process. It enhances the independent measurement of the desired performance of the underlying network. A set of 115 instances are chosen for testing purposes, and each algorithm's overall performance is observed and recorded accordingly.

3.5. Proposed System Model. The ABP-SCGNN method architecture is segregated into four distinct parts, as depicted in Figure 2. These include initialization of weight, forward and backward propagation of error, updating of weight, and bias. The hidden layer consists of a number of neurons, and every neuron has an activation function as $f(x) = \text{Sigmoid}(x)$. Activation function gets the sum of weighted input $(w_{11} * x_1 + w_{21} * x_2 + w_{31} * x_3 + \cdots + w_{81} * x_8 + 1 * b)$ argument as follows:

Table 1: The input and output variables for the proposed system framework.

S/N	Input attributes	Description	Range values
1	Pregnancies	Number of times pregnant	0–17
2	Glucose	Plasma glucose concentration 2 hours in an oral glucose tolerance test	0–199
3	Blood pressure	Diastolic blood pressure (mm Hg)	0–122
4	Skin thickness	Triceps skinfold thickness (ram)	0–99
5	Insulin	2-hour serum insulin (mu U/m1)	0–846
6	BMI	Body mass index (weight in kg/(height in m)^2)	0–67.1
7	Diabetes pedigree function	Diabetes pedigree function	0.078–2.42
8	Age	Age (years)	21–81
		Output/responder variable	
Sr.	Input attributes	Description	Range values
1	Outcome	Diabetes, yes or no	01

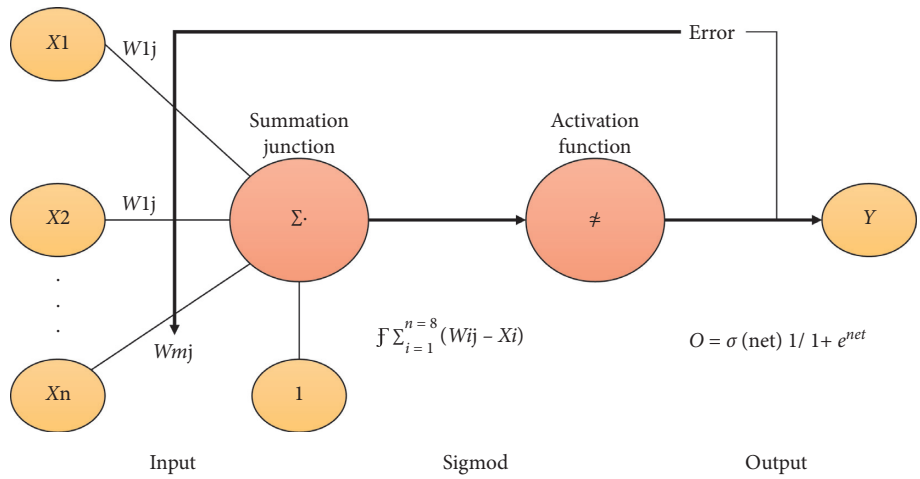


Figure 2: The architecture of the ABP-SCGNN model.

$$f\sum_{i=1}^{n=8} W_{ij}X_i + b. \tag{1}$$

Activation function was used to build a nonlinear transformation that is allocated to correspond nonlinear proposition or for assessing the complex functions. The inputs of the activation function were calculated by multiplying the weight by input and then added bias value. In the following section, the proposed neural network is explained how it performed its intended operations. To begin with, forward propagation is performed. First, backward propagation is made after comparing predicted Y output with actual output $\hat{Y}$ during the calculation of the gradient error at the output layer.

A neural network is trained with diabetes datasets by using the following algorithms:

(1) Multilayer perceptron (MLP)

(2) Bayesian regularized

(3) Scaled conjugate gradient

The proposed framework contains one input and one output layer with a single hidden layer containing eight input features (neurons) that produce one of the two outputs mentioned in the proposed research methodology. The dataset comprises input and output layers (X_i, Y_i), where X_i is the input layer and Y_i is the predicted output. If n is the total size of the entire set, then

$$X_n = \{(x_i, y_i), \ldots, (x_n, y_n)\}. \tag{2}$$

The parameters of feed-forward are denoted collectively as θ. Since training a neural network involves the gradient of the error function $E = (X, \theta)$ relevant to the weight w_{ij}^k (weight at node j in layer lk for node i), biases are b_i^k (bias at node i in layer lk), so as per learning rate, gradient descent updates the weights at each iteration as follows:

$$\theta_t = \theta_t - \alpha\frac{\partial E(X, \theta)}{\partial \theta_t}, \tag{3}$$

where θ_t represents ANN parameters at iteration t. The MSE in backpropagation is

$$E(X, \theta) = \frac{1}{2N}\sum_{i=1}^{N} (\hat{y}i - yi), \tag{4}$$

where $\hat{y}i$ is the predicted output and yi is the actual output of the input xi. The derivative of f (x) is represented as $f(x)$, and the derivative of the sigmoid function is $\sigma'(x)$. $b(k/i)$ (bias) in layer kth at node ith is included into the weight as w_{oi}^k with an output $o_0^{k-1} = 1$ at node 0 in layer $k - 1$; hence,

$$w_{0i}^{k} = b_{i}^{k}, \tag{5}$$

$$\vec{W} = \begin{pmatrix} w_{00} & \cdots \\ w_{0m} & \cdots \\ w_{10} & \cdots \\ w_{1m} & \cdots \\ & \cdot \\ & \cdot \\ & \cdot \\ w_{n0} & \cdots \\ w_{nm} & \cdots \\ w_{n+1},0 & \cdots \\ \cdot & \cdot \\ & \cdot \end{pmatrix}. \tag{6}$$

Weight and bias values are initialized with random values as one-time initiation; an updated weight value is used in subsequent iterations. Equation (6) can be written as follows:

$$\vec{W} = \begin{pmatrix} \vec{W}_{\mathrm{IH}} \\ \cdot \\ \vec{W}_{\mathrm{HO}} \end{pmatrix}. \tag{7}$$

Equation (7) demonstrates that $\vec{W}_{\mathrm{IH}}$ contains the weight from the input layer to hidden layer where as $\vec{W}_{\mathrm{HO}}$ is the weight representing from hidden layer to output layer.

$$\vec{W}_{\mathrm{IH}} = \begin{pmatrix} w_{00} & w_{01} & \cdots & w_{0m} \\ w_{00} & w_{11} & \cdots & w_{1m} \\ & \vdots & & \\ w_{n,0} & w_{n,1} & \cdots & w_{0m} \\ w_{00} & w_{11} & \cdots & \cdot \end{pmatrix}. \tag{8}$$

Equation (8) contains the weights between the input and hidden layer. It contains $n \times m$ elements. $W_{a,b}$ is the weight between element.

$$\vec{W}_{HO} = \begin{pmatrix} w_{n+1,0} & w_{n+1,1} & \cdots & w_{n+11,m} \\ w_{n+1,0} & w_{11} & \cdots & w_{n+2,m} \\ & \vdots & & \\ w_{n+k,0} & w_{n+k,1} & \cdots & w_{n+k,m} \\ w_{00} & w_{11} & \cdots & \cdot \end{pmatrix}. \tag{9}$$

In our case, only one output neuron is used. Equation (9) can also be written as

$$\vec{W}_{\mathrm{HO}} = \begin{pmatrix} w_{n+1,0} & w_{n+1,1} & \cdots & w_{n+1,m} \end{pmatrix}. \tag{10}$$

Corresponding to the original formulation $\mathcal{T}_{k-1}$,

$$a_{i}^{k} = b_{i}^{k} + \sum_{j=1}^{\mathcal{T}_{k-1}} w_{ji}^{k} o_{j}^{k-1} = \sum_{j=0}^{\mathcal{T}_{k-1}} w_{ji}^{k} o_{j}^{k-1}. \tag{11}$$

Then, the error can be calculated as

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}i - yi)^{2}. \tag{12}$$

After taking the derivative of Equation (12),

$$\frac{\partial E(X, \theta)}{\partial w_{ji}^{k}} = \frac{1}{N} \sum_{d=1}^{N} \frac{\partial}{\partial w_{ij}^{k}} \left(\frac{1}{2} (\hat{y}i - yi)^{2} \right) = \frac{1}{N} \sum_{d=1}^{N} \frac{\partial E_{d}}{\partial w_{ij}^{k}}. \tag{13}$$

The backpropagation algorithm is concerned with one I/O pair, and all I/O pairs in X may be produced in merging each gradient. For derivation, the error function is

$$E = \frac{1}{2} (\hat{y} - y)^{2}. \tag{14}$$

3.5.1. Error Function Derivatives. After applying the chain rule,

$$\frac{\partial E}{\partial w_{ji}^{k}} = \frac{\partial E}{\partial a_{j}^{k}} \frac{\partial w a_{j}^{k}}{\partial w_{ij}^{k}}, \tag{15}$$

where a_{j}^{k} treats as activation of node j in layer$k$.

Here,

$$\delta_{j}^{k} \equiv \frac{\partial E}{\partial a_{j}^{k}}. \tag{16}$$

The second term from the equation for a_{j}^{k} is as follows:

$$\frac{\partial a_{j}^{k}}{\partial w_{ji}^{k}} = \frac{\partial}{\partial w_{ij}^{k}} \left(\sum_{i=0}^{\mathcal{T}_{k-1}} w_{lj}^{k} o_{l}^{k-1} \right) = o_{l}^{k-1}. \tag{17}$$

Error function E partial derivative with w_{ij}^{k} (weight) is as follows:

$$\frac{\partial E}{\partial a_{j}^{k}} = \delta_{j}^{k} o_{i}^{k-1}. \tag{18}$$

So, the partial derivative of weight is a product of the error term φ_{j}^{k} at node j in layer k and the output o_{i}^{k-1} of node i in layer$k-1$.

3.5.2. The Output Layer. Backpropagation characterizes the value φ_{1}^{m}, where m is the final layer. Four-layer neural network possesses $m = 3$ for the ending layer and $m = 2$ for the second to the last layer. Expressing E (error function) in terms of the value a_{1}^{m} (since φ_{1}^{m} is a partial derivative for φ_{1}^{m}) gives

$$E = \frac{1}{2} (\hat{y} - y))^{2} = E \frac{1}{2} (g_{o}(a_{1}^{m}) - y))^{2}. \tag{19}$$

Then, applying the partial derivative,

$$\delta_i^m\left(g_o\left(a_1^m\right)-y\right)g_o'\left(a_1^m\right)=\left(\widehat{y}-ty\right)g_o'\left(a_1^m\right). \tag{20}$$

The partial derivative of$E$ and the error function ω_{i1}^m is

$$\frac{\partial E}{\partial w_{i1}^m}=\delta_1^m o_i^{m-1}=\left(\widehat{y}-y\right)g_o'\left(a_1^m\right)o_i^{m-1}. \tag{21}$$

3.5.3. The Hidden Layer. The hidden layers' error can be calculated as follows:

$$\delta_j^k=\frac{\partial E}{\partial a_j^k}=\sum_{l=1}^{N\mathcal{F}^{k+1}}\frac{\partial E}{\partial a_j^{k+1}}\frac{\partial a_j^{k+1}}{\partial a_j^k}, \tag{22}$$

where l ranges from rk+1 to the number of nodes.

Error term φ_l^{k+1} gives the following equation:

$$\begin{aligned}\delta_j^k&=\sum_{l=1}^{\mathcal{F}^{k+1}}\delta_l^{k+1}\frac{\partial a_l^{k+1}}{\partial a_j^k},\\ a_l^{k+1}&=\sum_{j=1}^{\mathcal{F}^k}w_{jl}^{k+1}g\left(a_j^k\right),\end{aligned} \tag{23}$$

remembering the definition of a_l^{k+1} where the $g(x)$ is the activation function.

$$\frac{\partial a_l^{k+1}}{\partial a_j^k}=w_{jl}^{k+1}g'\left(a_j^k\right). \tag{24}$$

As we know that

$$\delta_j^k=\sum_{l=1}^{\mathcal{F}^{k+1}}\delta_l^{k+1}w_{jl}^{k+1}g'\left(a_j^k\right)=g'\left(a_j^k\right)\sum_{l=1}^{\mathcal{F}^{k+1}}w_{jl}^{k+1}\delta_l^{k+1}, \tag{25}$$

the partial derivative of the Eerror function ω_{ij}^k for $1\le k<m$ is

$$\frac{\partial E}{\partial w_{ij}^k}=\delta_l^k o_i^{k-1}=g'\left(a_j^k\right)o_i^{k-1}\sum_{l=1}^{\mathcal{F}^{k+1}}w_{jl}^{k+1}\delta_l^{k+1}, \tag{26}$$

$$\frac{\partial E_d}{\partial w_{ij}^k}=\delta_l^k o_i^{k-1}. \tag{27}$$

For the partial derivatives and for the final layer's error term,

$$\begin{aligned}\delta_1^m&=g_o'\left(a_1^m\right)\left(\widehat{y}_d-y_d\right),\\ \delta_1^m&=g'\left(a_j^k\right)\sum_{l=1}^{\mathcal{F}^{k+1}}w_{jl}^{k+1}\delta_l^{k+1}.\end{aligned} \tag{28}$$

For the hidden layers' error term,

$$\frac{\partial E(X,\theta)}{\partial w_{ij}^k}=\frac{1}{N}\sum_{d=1}^{N}\cdot\frac{\partial}{\partial w_{ij}^k}\left(\frac{1}{N}\left(\widehat{y}_d-y_d\right)^2\right)=\frac{1}{N}\sum_{d=1}^{N}\cdot\frac{\partial E_d}{\partial w_{ij}^k}. \tag{29}$$

For combining the partial derivatives,

$$\Delta w_{ij}^k=a\frac{\partial E(X,\theta)}{\partial w_{ij}^k}. \tag{30}$$

For updating the weights,

$$w_{ij}^{k+1}=w_{ij}^k+\lambda\Delta w_{ij}^k. \tag{31}$$

3.6. Algorithm of ABP-SGCNN Model. Assume a as the learning rate and parameter initialization as w_{ij}^k, following steps are used to keep progress in the proposed algorithm.

3.6.1. Calculate the Forward Phase. For each pair of input and output $(\vec{x}_d, y_d)$, store the results $(\widehat{y}_d, a_j^k)$ and (o_j^k) for each node (j) in layer (k) by proceeding from layer zero, input layer, to layer (m), the output layer.

3.6.2. Calculate the Backward Phase. For each pair of input and output $(\vec{x}_d, y_d)$, store the results $(\partial E_d/\partial w_{ij}^k)$ for each weight (w_{ij}^k) connecting node (i) in layer $(k-1))$ to node (j) in layer (k) by proceeding from layer (m), the output layer, to layer (1), the input layer.

Evaluate the error term for the final layer using equation (2).

Backpropagate the error terms for the hidden layers δ_l^k, working backward from the$k=m-1$, by repeatedly using equation (3).

Evaluate the partial derivatives of the individual error E_d regarding w_{ij}^k by using equation (1).

3.6.3. Combine the Individual Gradients. Combine the individual gradients for each input-output pair $(\partial E_d/\partial w_{ij}^k)$ to get the total gradient $(\partial E(X,\theta)/\partial w_{ij}^k)$, for the entire set of input-output pairs $X=\left\{(\vec{x}_1,y_1),\ldots,(\vec{x}_N,y_N)\right\}$ by using equation (4) (a simple average of the individual gradients).

3.6.4. Update the Weights. According to the learning rate a and total gradient $(\partial E_d/\partial w_{ij}^k)$ and by using equation (5) (moving in the direction of the negative gradient), the weights are updated using equation (27).

Using equation (27), weights are updated accordingly.

4. Experimental and Analytical Verification

Probably, it is not likely to achieve the desired result during a single training iteration. However, sometimes, the model needs to train several times till it approaches near to the predicted outcome. After completing thousands of iterations, the measurement and evaluation results of predicting the diagnosis of diabetes are described in Table 2 for analysing the actual performance. Taking into account, optimizing the framework along with training the network for 5000 epochs and using 768 instances, the experiment demonstrates the potential benefit and worth of neural network algorithms. For all algorithms used in training, the learning rate was 0.25, and the momentum coefficient was 0.5. During the simulation, a number of neurons that were

Table 2: The performance analysis of the ABP-SCGNN model compared with standard ANNs models using an accuracy measure.

Algorithms	Training accuracy (%)			Validation accuracy (%)		
	5 neurons	20 neurons	40 neurons	5 neurons	20 neurons	40 neurons
Multilayer perceptron	91.6	93.6	94.9	90.8	91.9	93.2
Bayesian regularized	77.5	79.6	81	71	76.5	78.7
Proposed ABP-SCGNN	92.94	94.44	95.71	92.073	92.708	93.34

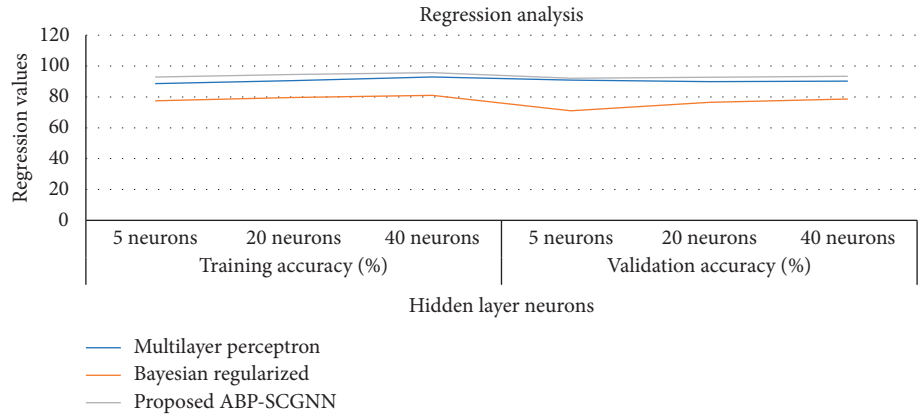


Figure 3: The regression accuracy results of the ABP-SCGNN model at different neurons of hidden layer.

taken in the hidden layer fall in the range 5, 10, 20, 30, 40, and 50 neurons. Results prove that the value of correlation coefficient particularly for proposed artificial back propagation scaled conjugate gradient neural network (ABP-SCGNN) is near to 1 that indicates pretty accuracy based on the regression accuracy plot among all other algorithms. Mean squared error (MSE) is calculated on all algorithms. The prediction accuracy is calculated with the ratio of the total number of predictions that are correct to the complete predictions. The proposed algorithm (ABP-SCGNN) produces more correct predictions with 93% prediction accuracy.

Figure 3 presents the MSE during the learning process for the hidden layer using different number of neurons in the range 5 to 50 which is computed against all algorithms. Table 3 reveals that the MLP algorithm contains 0.0026 using five neurons at the hidden layer; no doubt by increasing the number of neurons, i.e., 20 and 40, MSE was approaching low, but network complexity increased along with decreasing MSE rate. The Bayesian regularization algorithm has approximately high MSE, which is the same for a different number of neurons of the hidden layer. On the contrary, the proposed algorithm, ABP-SCGNN, attains relatively a lower MSE result when five neurons are used at the hidden layer, and this result is reduced more upon increasing the number of neurons, as shown in Figure 4. It confirms that using 20 neurons gets a lowest MSE and a high accuracy results. Moreover, the model achieves a regression accuracy value of 93%, as presented in Figure 3. Consequently, a trade-off between network complexity and performance of the model is approaching for 20 hidden neurons in both MSE and accuracy measures.

The regression accuracy plot in Figure 3 depicts a significant correlation of the proposed algorithm between targets and predicted values that are identified by the dashed line that exhibits the highest accuracy. Three different numbers of neurons were used in the hidden layer, as mentioned in Table 2. Prediction accuracy of MLP remains the same for all neurons used. In the same case with the Bayesian regularization algorithm, its regression accuracy result ranges from 70 to 78 using the same number of hidden neurons taken for the other two algorithms.

Since forward propagation concerns the inference phase of a feed-forward neural network and the learning phase deals with the backpropagation neural network, the learning phase is relatively slower than the inference phase because gradient descent needs to repeat several times during the whole process. Using parallel running, the work performance of underlying algorithms can be efficient. The computational complexity of the feed-forward neural network is calculated by splitting the computation in the training and inference phase. As shown in Figure 5, the overall time complexity will increase with the number of epochs. However, the efficiency of algorithms remains stable.

The following equation calculates the factor of complexity (Fc) according to Figure 5.

$$\frac{2 * it}{2 * (j * i + j * it)}, \tag{32}$$

where i is the input layer neurons, j is hidden layer neurons, and it is the number of iterations.

Likewise, The MSE time complexity can be calculated for the same algorithms as follows:

$$2 * ((j * i + j * k) * it), \tag{33}$$

where Tc is time complexity, i is the input layer neurons, j is hidden layer neurons, and k is the number of iterations. Figure 5 demonstrates the time complexity based on the

Table 3: The performance analysis of the ABP-SCGNN model compared with standard ANNs models using an MSE measure.

Algorithms	Training MSE rate			Validation MSE Rate		
	5 neurons	20 neurons	40 neurons	5 neurons	20 neurons	40 neurons
Multilayer perceptron	2.59E−03	1.99E−03	1.55E−03	2.61E−03	2.29E−03	2.13E−03
Bayesian regularized	4.17E−1	3.8E−1	3.54E−01	4.03E−01	3.96E−01	3.63E−01
Proposed ABP-SCGNN	2.6286−e3	2.3071E−3	1.6265E−3	2.381E−03	2.190E−03	2.054E−03

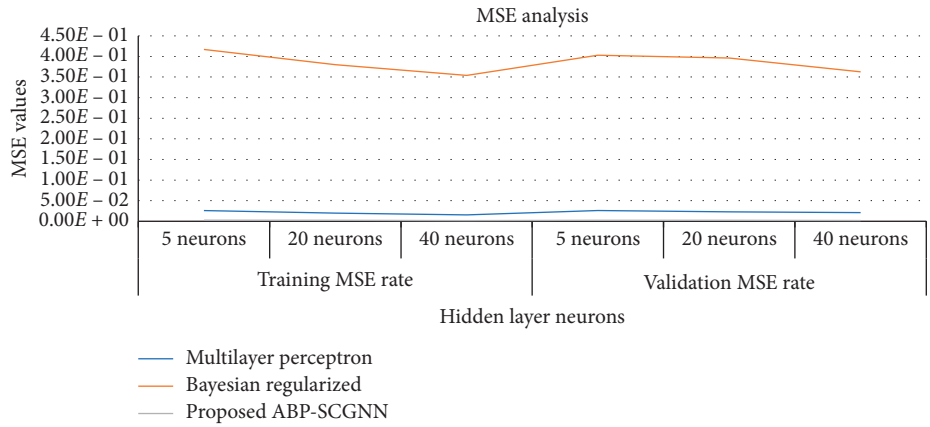


Figure 4: The MSE results of the ABP-SCGNN model at different neurons of hidden layer.

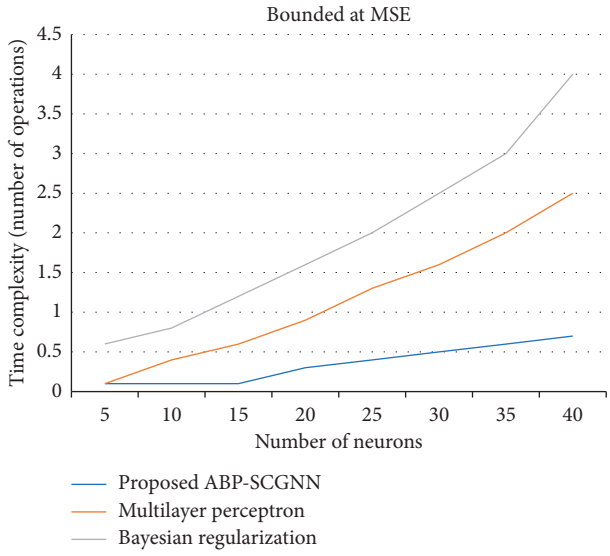


Figure 5: The time complexity of the ABP-SCGNN model at different neurons of hidden layer.

number of epochs by choosing three configurations of different neurons, such as 5, 20, and 40, as follows.

Indeed, with the increase in neurons, the success ratio increases to some extent. However, after examining 20 neurons, the success ratio remains the same in conjunction with increase in time complexity. Table 4 shows the comparison among various neural net algorithms with the proposed (ABP-SCGNN) algorithm that demonstrates its highest success ratio, such as 93%. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm contains an

Table 4: The accuracy results of the ABP-SCGNN model compared with some other models.

Algorithm	Accuracy (%)
BFGS [1]	88.8
Genetic algorithm [8]	87
GRNN [9]	80.21
ABP-SCGNN (proposed)	93

88.8% success ratio higher than ever used earlier in the underlying problem. On the other hand, the proposed algorithm shows the best result regarding success ratio and regression accuracy.

5. Conclusion

Using artificial backpropagation neural network (ABPNN), a pragmatic framework has been proposed to predict the diagnosis of diabetes. The presented results have been demonstrated in the paper that it can be viable to model an adaptive framework using APBNN. Since the applications of ANN have been organized differently from the classical approach, the features are provided as input variables rather than to provide them into the ANN.

The proposed ABP-SCGNN framework is effective and efficient, with a 93% success ratio when simulated with a test PIDD dataset [17]. To check the effective performance of ANNs and computational analysis, many performance indicators were calculated. Among these indicators, mean squared error (MSE) and regression accuracy analysis were examined anxiously. All neural networks were trained using 5 to 50 hidden layer neurons with the same

dataset. The best result was demonstrated at a hidden layer containing 20 neurons, specifically at the proposed ABP-SCGNN. The rest of the two algorithms remain with high MSE at the same 20 hidden layer neurons, whereas the presented ABP-GCGNN gives attractive results for the same dataset.

The experimental results demonstrated that ABP-SCGNN was the best algorithm that exhibited the highest prediction result among all algorithms. Since ANN possibly presents new approaches, strategies, and methodologies to remove uncertainties and reduce the potential instability of correlation, the intensity of network performance achieved using the proposed framework verified that ANN is functionally beneficial for successfully predicting the diagnose of diabetes with suitable disease input features and a corresponding set of instances.

Data Availability

The dataset used in this study is obtained from the National Institute of Diabetes and Digestive and Kidney Diseases, as cited in [17] (http://archive.ics.uci.edu/ml).

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

The authors are grateful to the Deanship of Scientific Research, King Saud University, for funding through Vice Deanship of Scientific Research Chairs.

References

[1] M. A. Sapon, K. Ismail, and S. Zainudin, "Prediction of diabetes by using artificial neural network," in *Proceedings of the 2011 International Conference on Circuits, System and Simulation*, vol. 2829, Singapore, Singapore, May 2011.

[2] A. Chavey, M.-D. Ah Kioon, D. Bailbé, J. Movassat, and B. Portha, "Maternal diabetes, programming of beta-cell disorders and intergenerational risk of type 2 diabetes," *Diabetes & Metabolism*, vol. 40, no. 5, pp. 323–330, 2014.

[3] D. Manzella, R. Grella, A. M. Abbatecola, and G. Paolisso, "Repaglinide administration improves brachial reactivity in type 2 diabetic patients," *Diabetes Care*, vol. 28, no. 2, pp. 366–371, 2005.

[4] E. O. Olaniyi and K. Adnan, "Onset diabetes diagnosis using artificial neural network," *International Journal of Scientific and Engineering*, vol. 5, no. 10, pp. 754–759, 2014.

[5] P. Rahimloo and A. Jafarian, "Prediction of diabetes by using artificial neural network, logistic regression statistical model and combination of them," *Bulletin de la Société Royale des Sciences de Liège*, vol. 85, pp. 1148–1164, 2016.

[6] S. S. Abbasi, B. Dashtbozorg, B. M. T. H. Romeny, and F. Fleuret, "Exploratory study on direct prediction of diabetes using deep residual networks," *European Congress on Computational Methods in Applied Sciences and Engineering*, vol. 1, pp. 797–802, 2017.

[7] O. Karan, C. Bayraktar, H. Gümüşkaya, and B. Karlık, "Diagnosing diabetes using neural networks on small mobile devices," *Expert Systems with Applications*, vol. 39, no. 1, pp. 54–60, 2012.

[8] D. K. Choubey, S. Paul, S. Kumar, and S. Kumar, "Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection," in *Proceedings of the Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing System (ICCCS 2016)*, pp. 451–455, Gurgaon, India, September 2016.

[9] K. Kayaer and T. Yıldırım, "Medical diagnosis on Pima Indian diabetes using general regression neural networks," in *Proceedings of the International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP)*, pp. 181–184, Stanbul, Turkey, June 2003.

[10] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, p. 261, Rockville Pike, MD, USA, April 1988.

[11] J. C. Florez, K. A. Jablonski, N. Bayley et al., "TCF7L2Polymorphisms and progression to diabetes in the diabetes prevention program," *New England Journal of Medicine*, vol. 355, no. 3, pp. 241–250, 2006.

[12] K. Iqbal, M. A. Khan, S. Abbas, Z. Hasan, and A. Fatima, "Intelligent Transportation System (ITS) for smart-cities using mamdani fuzzy inference system," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2, pp. 94–105, 2018.

[13] M. A. Khan, M. Umair, and M. A. S. Choudhry, "GA based adaptive receiver for MC-CDMA system," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, pp. 2267–2277, 2015.

[14] M. Umair, M. A. Khan, and M. A. Choudry, "GA backing to STBC based MC-CDMA systems," in *Proceedings of the Intelligent Systems Modelling & Simulation (ISMS), 2013 4th International Conference*, pp. 503–506, Bangkok, Thailand, January 2013.

[15] A. Atta, S. Abbas, M. A. Khan, G. Ahmed, and U. Farooq, "An adaptive approach: smart traffic congestion control system," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, pp. 1012–1019, 2018.

[16] M. A. Ullah, M. A. Khan, S. Abbas, A. Athar, S. S. Raza, and G. Ahmad, "Blind channel and data estimation using fuzzy logic-empowered opposite learning-based mutant particle swarm optimization," *Computational Intelligence and Neuroscience*, vol. 13, pp. 1–12, 2018.

[17] D. Dua and C. Graff, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, USA, 2019.

[18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[19] N. M. Nawi, W. H. Atomi, and M. Z. Rehman, "The effect of data pre-processing on optimized training of artificial neural networks," *Procedia Technology*, vol. 11, pp. 32–39, 2013.

[20] C. Boos, C. Scolaro, M. Pereira, and F. Azevedo, "Analysis of pre-processing methods for artificial neural network pattern recognition of EEG signals," in *Proceedings of the World Congress on Medical Physics and Biomedical Engineering*, pp. 558–561, Beijing, China, May 2012.