

Research Article

Performance Analysis of an Optimized ANN Model to Predict the Stability of Smart Grid

Ayushi Chahal ¹, Preeti Gulia ¹, Nasib Singh Gill ¹ and Jyotir Moy Chatterjee ²

¹Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak, Haryana, India

²Department of Information Technology, Lord Buddha Education Foundation, Kathmandu, Nepal

Correspondence should be addressed to Jyotir Moy Chatterjee; jyotir.moy@lbef.edu.np

Received 28 May 2022; Revised 30 June 2022; Accepted 2 July 2022; Published 3 August 2022

Academic Editor: Muhammad Ahmad

Copyright © 2022 Ayushi Chahal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The stability of the power grid is concernment due to the high demand and supply to smart cities, homes, factories, and so on. Different machine learning (ML) and deep learning (DL) models can be used to tackle the problem of stability prediction for the energy grid. This study elaborates on the necessity of IoT technology to make energy grid networks smart. Different prediction models, namely, logistic regression, naïve Bayes, decision tree, support vector machine, random forest, XGBoost, k-nearest neighbor, and optimized artificial neural network (ANN), have been applied on openly available smart energy grid datasets to predict their stability. The present article uses metrics such as accuracy, precision, recall, *f1*-score, and ROC curve to compare different predictive models. Data augmentation and feature scaling have been applied to the dataset to get better results. The augmented dataset provides better results as compared with the normal dataset. This study concludes that the deep learning predictive model ANN optimized with Adam optimizer provides better results than other predictive models. The ANN model provides 97.27% accuracy, 96.79% precision, 95.67% recall, and 96.22% *F1* score.

1. Introduction

An electric grid is said to be smart if it tends to replace traditional appliances with smart ones. A smart grid helps in the smart distribution of electric energy, provides smart meter infrastructure, etc. The smart grid also motivates the use of renewable energy resources rather than nonrenewable ones [1]. Smart grids can be used to enhance the smooth functioning of every domain like electricity generation either from renewable or nonrenewable energy resources, distribution of energy according to the demand and supply for different smart sectors like smart homes, smart offices, smart factories, electric vehicles, and so on as shown in Figure 1.

Nowadays, renewable energy resources dominate the market for electricity generation [2]. Smart grid faces multiple problems like power grid resilience, cyber security in a smart power grid system, smart energy management or distribution, etc. [3]. With the growing demand for renewable energy resources, grid topology used for electricity distribution has become more and more decentralized.

Consumer of the electricity can also behave like its producer. So unlike traditional grids, power generation and consumption can occur from any terminal point [4]; such terminals are called Prosumers. Generation and distribution of power are not limited to a central node anymore.

With this decentralized approach, it becomes very difficult to manage and keep track of the generation and consumption of energy. This creates chaos to maintain information on the demand and supply of electric power among the producers and consumers. This, in turn, makes it difficult to maintain the stability of the grid system. To remove this overhead, a decentralize smart grid control (DSGC) is proposed [5]. This control system keeps track of the power demand and supply frequency on every producer-consumer node of the grid. Different decentralized topologies can be used in smart energy grids for power consumption and production as shown in Figure 2.

DSGC increases the stability of the smart grid. DSGC model considers some parameters like price elasticity, balancing power flow, and reaction time of nodes. These

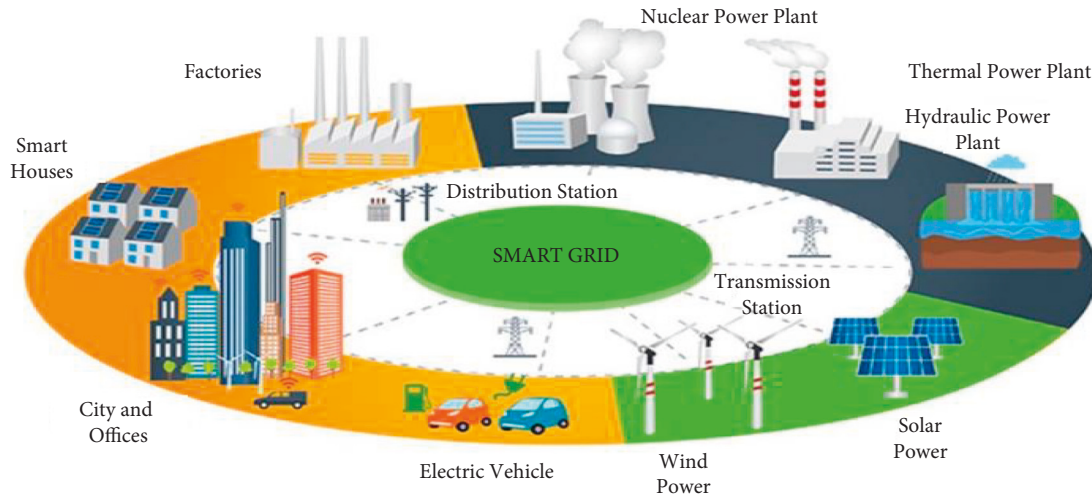


FIGURE 1: Smart grid. Source: Smart grid evolution, 2018. Eolas Magazine. URL: <https://www.eolasmagazine.ie/smart-grid-evolution/> (accessed 2.18.22).

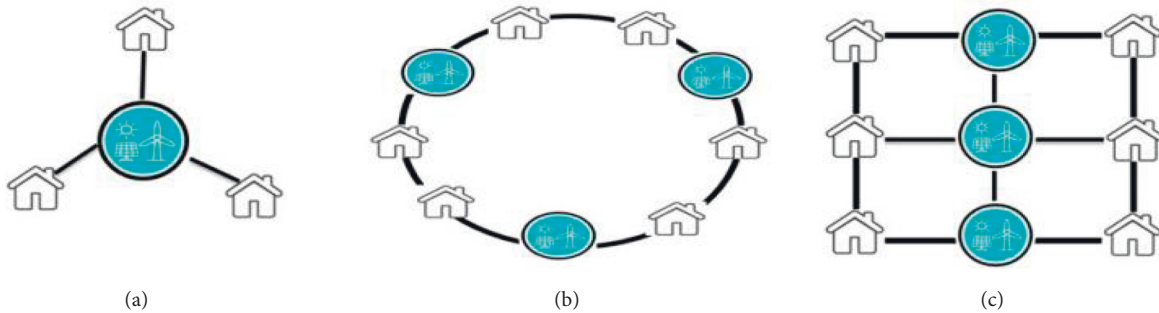


FIGURE 2: Decentralized smart grid topologies: (a) star with a central node as producer; (b) ring topology with decentralized 3 producer and 6 consumer nodes; (c) lattice topology with 3 producer and 6 consumer nodes.

parameters help to check the stability of the energy grid network. Balancing power flow deals with the units produced or consumed by producer or consumer, respectively. The reaction time of nodes deals with the change in the response according to the rise or fall in the price of power units. The most careful part of the decentralized smart energy grid is the information flow of electricity distribution. This information flow helps in deciding the stability of the decentralized smart grid. For predicting stability, different ML and DL models play an indispensable role [6].

The main objectives of this study are

- (i) This study provides a detailed overview of the integration of IoT with a smart energy grid with the help of three-layered architecture
- (ii) This article discusses the behavior of a smart grid dataset
- (iii) This study helps to understand the effect of data augmentation on prediction accuracy
- (iv) “ANN” model is proposed for the stability prediction of a decentralized grid system
- (v) This study outlines the comparative analysis of traditional machine learning algorithms with optimized ANN

The article is organized as follows: integration of IoT and the electric grid is very well explained in Section 2 with a layered architecture. The related literature review is given in Section 3. Section 4 describes a stepwise approach used in methodology for analysis purposes. It also provides a detailed overview of different prediction models used in this study. Section 5 discusses the experimental results and presents them in graphical form. Also, it provides a comparative analysis of all the models.

2. Layered Architecture for IoT and Smart Grid Integration

The Internet of Things (IoT) has provided innovative solutions to real-life problems [7]. IoT has upgraded different technologies like computational, sensing, communicational, etc. IoT can be integrated with grid systems which further helps to resolve problems of smart grids like balancing demand and supply of energy grid. [8].

IoT-integrated smart grid system is divided into three layers [9]. Figure 3 presents a layered structure of the IoT-integrated smart grid system. In bottom-up fashion,

- (i) *First layer* is the data collection layer. Different IoT devices like sensors and actuators are used over a

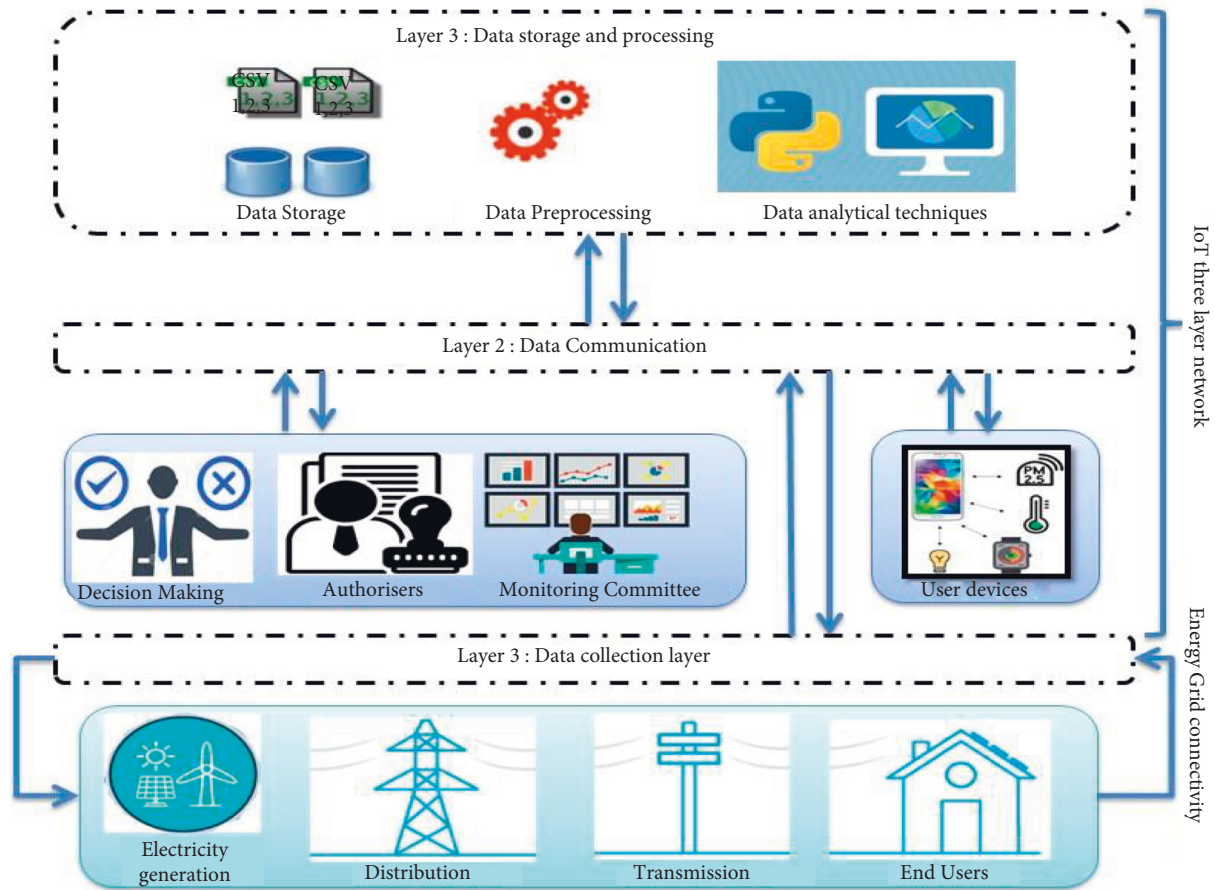


FIGURE 3: IoT-integrated smart energy grid three-layered network. This concept is partially adapted from Babar et al. [10].

wireless sensor network to collect data from smart homes, distribution centers, smart factories, or smart renewable energy generation systems. This data are fed into layer 2.

- (ii) *Second layer* is the data communication layer. In this layer, data are communicated to decision-makers, different monitoring centers, and authorities to determine priorities and parameters. User devices are also attached on the second layer only.
- (iii) *Third layer* is the data storage and processing. In this layer, data are processed and analyzed using intelligent techniques like artificial intelligence, machine learning, big data analytics, cloud computing, fog computing, edge computing, etc. These techniques process the data according to the guidelines and results required by the authorities and decision-makers presented in layer 2. This processed data are presented to the user in graphical, tabular, or textual form.

IoT can enhance the functioning of the smart grid by improving sensing and measurement [11]. It helps in automatic monitoring and remote control. IoT can help in forecasting the usage of renewable resources. The distribution of electricity is a very important part of the energy grid, which is automated by IoT. The smart meter is one of the most popular applications of IoT in the smart energy grid [12, 13].

3. Related Work

In the smart grid, work like assigning renewable energy resources, delivering short-term energy forecasting, and sensing the motion of the occupants are to be analyzed to enhance its efficiency. Yao et al. use machine learning models for the same and have given Machine Learning Energy-Efficient Framework (MLEEF) [14]. The authors used solar energy for this study. Occupant's profiles and energy profiles are studied in depth before applying machine learning frameworks to the dataset. Findings of the analysis are measured as accuracy for checking energy consumption and load forecasting. A hybrid machine learning model is proposed by Sharmila et al. for smart energy management by optimizing energy distribution from the energy sources to the recipient. In this model, SVM is used for regression and classification problems with the Big data five V's paradigm [15].

Energy Management Model (EMM) is presented by Ahmed et al. which uses ML models to manage energy flow for smart grid [16]. Both optimization techniques and machine learning models are used to manage the energy distribution. This study concluded that machine learning-based energy management models outperform the results of optimization-based EMM. Hourly data are collected from national renewable energy resources in the US for this study. Collected data statistics are used for its simulation. This

average dataset of 10 years is then distributed in three seasons: winter, summer, and spring. Season-wise energy management models are used for analysis to check which model performs better.

Arzamasov et al. focused on the DSGC system [17]. They focus on the frequency of the alternate current (AC) to check the stability of the system. It is discussed that frequency increases at the time of excess electricity generation and decreases at the time of reduced amount of electricity production. The authors focus to identify the instability of four node star network DSGC system after accessing the mathematical model discussed in the study of Schäfer et al.

Chen et al. used a deep residue learning model called Short-Term Long Forecast (SLTF) to forecast electric load over the network for a day ahead [18]. Authors have used ANN as a base. End-to-end complete new neural network is proposed in the study. Multiple neural networks ensemble in a single model. This ensemble is done in two phases. The first phase is to take snapshots of the training model. Authors have used Adam for optimization to enhance the learning rate. In the second phase, authors initialize different models independently. All models are trained by hyper-parameter tuning and then the average output of all the models is used for forecasting. A similar forecasting model using STLF is proposed by Y. Wang et al. in their study [47]. Ensemble Markov model is proposed to forecast the industrial electric load. In this model, authors have combined novel time series data mining techniques and then ensembled different prediction frameworks of the Hidden Markov Model (HMM). Min-Max normalization is used to normalize the data and this data are fed to the bagging algorithm for resampling. After sampling, hyper-parameters are set and HMM is applied to every sample. All results are ensemble based on log-likelihood. Different parameters like average absolute percentage error (MAPE), root mean square error (RMSE), and average absolute error (MAE) help to measure the efficiency of the model. Wang et al. use SVM and XGBoost machine learning models for forecasting industrial load [20]. Bayesian optimization algorithm is used to optimize the hyper-parameters of the XGBoost model. Different state-of-the-art methods are used to predict model accuracy.

An SVM model is proposed by Gupta et al. to predict blackout in smart grid [21]. This model is trained on the historic dataset and evaluated probabilistically. Ge et al. used a hybrid algorithm to forecast industrial power load [22]. At first, the k-means algorithm classifies the data in different clusters. After that, the reinforcement learning model is used with the SVM model. To optimize the reinforcement model, the author has used swarm optimization. This optimized hybrid algorithm has increased the accuracy of the model to predict the load over a real-time dataset.

Wei et al. have used Deep Belief Network (DBN) over the smart grid network to detect attacks for false data injection. Initially, researchers have used unsupervised machine learning algorithms to feed the Boltzmann machine with initial weights that will be used in DBN. Back-propagation technique of DBN helps to reduce errors in the top to bottom fashion and hence refine the model. This proposed approach of the researchers provides good results as compared with

the false data injection attacks detected through the SVM model [23].

Amarasinghe et al. used deep neural network models to forecast energy load balancing. Convolutional neural network (CNN) forecasts the load. Also, the results of CNN are compared with other predictive models like LSTM sequence-to-sequence, shallow ANN, restricted Boltzmann model, and SVM over the same kind of dataset. Root mean square error (RMSE) helps in the comparative analysis of different prediction models [24]. Muzumdar et al. presented the reasonableness of using ML predictive models to forecast the balance and imbalance of energy flow in the energy grid [25]. Authors have used historic communication and supplied data from the data source. After that, data preprocessing steps are applied. Refined data is split into training and testing datasets on which different machine learning models like multi-layer perceptron (MLP), bagging, AdaBoost, decision tree, random forest, naïve Bayes model, KNN, SVM, and gradient boosting. Accuracy, RMSE (for both 10 cross fold and random split), and MAE (for both random split and 10 cross fold) are tried to find the correct predictive model among all the ML models applied to the dataset. Z. Guan et al. use a nonparametric Bayesian clustering model in a smart grid system in order to preserve the privacy of the big data one gets from the smart grid. For this proposed model, the Infinite Gaussian mixture model (IGMM) is used in the implementation process. The Laplace mechanism is used to release data so that the privacy of data can be maintained [48].

Wang et al. proposed a data reduction technique for wireless sensor networks (WANs) to predict forthcoming data. This technique is divided into two phases: the first one is the data reduction phase (DRP). DRP helps in reducing the amount of data to be transmitted over WAN. The second phase is the data prediction phase (DPP), in which non-transmitted data are predicted at the base station itself. Data predictions are done on the basis of the Kalman filter [27].

Alsamhi et al. used the ANN model to predict signal strength emitted from drones. These signals are used to communicate with the IoT devices in a smart city environment. This signal strength is used to find the next location in the path of a flying drone [28]. Nyangaresi and Alsamhi presented a study for secure signal transfer in a smart grid. In smart metering, data are being transferred from consumers to power management centers. This study has proposed a traffic signaling protocol developed for preserving the privacy of the data. This protocol is found to be effective against Man-in-Middle attacks, desynchronization attacks, and impersonating attacks [29].

4. Methodology

4.1. Dataset Used and Its Preprocessing. This study has used a data set from the UCI repository [30]. This dataset has the readings of a renewable smart energy grid that is supposed to satisfy the needs of three consumers. This dataset contains 10,000 records originally. But this dataset is augmented for better results and the augmented version of the dataset contains 60,000 observations. This dataset contains 14

features, out of which 12 are independent primary features and 2 are dependent secondary features. Table 1 describes all the features of the dataset used for the study.

4.2. Stepwise Methodology Used. Figure 4 presents the stepwise approach used to predict the stability of the smart grid. A detailed explanation of the approach used in the research is as follows:

- (i) Firstly, a raw dataset is taken, and then it is augmented. Initially, the dataset has 6000 observations. Since the grid is considered to be symmetric and we are assuming three consumers in this study, this dataset is augmented by 3! ($3 * 2 * 1 = 6$ times). Data augmentation helps to get better accuracy in results.
- (ii) Next is the data preprocessing step. No feature engineering overhead is required because data are acquired from simulation exercises so there are no NaN or duplicate values.
- (iii) After that, some machine learning models like logistic regression, k-nearest neighbor (KNN), decision tree, support vector machine (SVM), random forest, naïve Bayes, and XGBoost are applied to the dataset to predict future values. Hyper-parameter tuning of these models is performed in order to get the best results. All the specifications of these ML models are explained in a subsection in Section 4.
- (iv) After hyper-parameter tuning, the performance of all the predictive models is measured based on the evaluation matrices used (explained in Section 5.1).
- (v) Deep learning model is applied to the scaled dataset. For applying ANN with the “Relu” activation function, the dataset is needed to be scaled. So, by importing the “StandardScalar” function of preprocessing package, the features of the dataset are scaled properly.
- (vi) After feature scaling, the proposed ANN model for smart grid is applied. Specification of the proposed ANN model is explained in detail in Section 4.2.8.

In this study, ML and DL models are deployed to predict the stability of the decentralized energy grid. Also, an artificial neural network (ANN), a deep neural network model, is used for the same. In the following subsection, these models are explained in detail.

4.2.1. Logistic Regression. Logistic regression works somewhat similar to linear regression. Logistic regression predicts the outcome on the basis of the individual characteristics of each feature [31]. The logistic regression model is very easy to regularize. It calibrates output based on predicted possibilities [32]. Suppose Y is a predicted output feature that depends on the predictor variable X and X can be given as $\{x_1, x_2, x_3, \dots, x_n\}$. y is regarded as the output variable given as follows:

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_nx_n + e, \quad (1)$$

$$y = \begin{cases} 1, & \text{if } a_0 + a_1x_1 > 0, \\ 0, & \text{if } a_0 + a_1x_1 \leq 0, \end{cases}$$

where $a_0, a_1, a_2, \dots, a_n$ are the regression coefficients. It uses logarithmic or logistic function for cost evaluation.

4.2.2. KNN. K-nearest neighbor (KNN) is a supervised machine learning model [33]. This supervised learning technique works on the principle of similarity among the neighbors. Classification of the data points is done on the basis of the distance between them. It is also called a lazy learner algorithm. KNN algorithm is used in many energy grid scenarios like to forecast low-voltage demand, and to measure current and its state [34, 35].

4.2.3. Naïve Bayes. Naïve Bayes network is also a prediction model that works on conditional probability [36]. For understanding purpose, let Z be a class label that will be classified on the basis of the conditional probability of independent features X and Y . Naïve Bayes [37] model will learn using the Bayes rule to calculate a conditional probability for class Z using instances $Y_0, Y_1, Y_2, \dots, Y_n$ of attribute Y . Posterior probability is used to predict the class. When attribute Y is independent of attribute X , then the posterior probability of Z can be calculated, when the following condition is hold [38]:

$$P(Y | X, Z) = P\left(\frac{Y}{Z}\right), \forall X, Y, Z, \text{ whenever } P(C) > 0. \quad (2)$$

4.2.4. Decision Tree. Decision tree is used to predict the categorical dataset. Nodes of the decision tree are applied with decision rules which help in categorizing the dataset in particular classes [39]. It helps to analyze the instance of a feature and predict the target value. Any kind of null value in data does not create any hurdle for the decision tree model. Data normalization or scaling is not necessary but any small change in data can make this model unstable [40].

4.2.5. SVM. It is a supervised machine learning model in which the type of kernel used plays an important part. Statistical learning is considered the base of this predictive model. Support vector machine (SVM) is used in classification as well as regression problems. Kernel used in SVM takes the input data and plots it to high-dimensional actual feature space. SVM model then creates a hyper-plane in feature space which is used to categorize the data points into different classes and to find out patterns for classification and regression problems. The cost function used in SVM to calculate prediction cost is given in equation (4).

$$C = \sum_{i=1}^l Lo + W(l, h), \quad (3)$$

where Lo = loss function and $W(l, h)$ = capacity of learning model.

TABLE 1: Performance parameters for prediction models.

S No.	Feature	Type	Significance
1	tau1, tau2, tau3, tau4	Primary, independent, and predictive	Represents reaction time of producer and three consumers (range: from 0.5 to 10)
2	p1, p2, p3, p4	Primary, independent, and predictive	Represents power produced (positive value) and consumed (negative value) by producers and consumers, respectively
3	g1, g2, g3, g4	Primary, independent, and predictive	Price elasticity coefficient of producer and consumer (range: from 0.05 to 1.00)
4	Stab	Secondary and dependent	Numerical real value (if positive, then linearly unstable, if negative, linearly stable)
5	Stabf	Secondary and dependent	Categorical feature, binary value labeled as stable/unstable

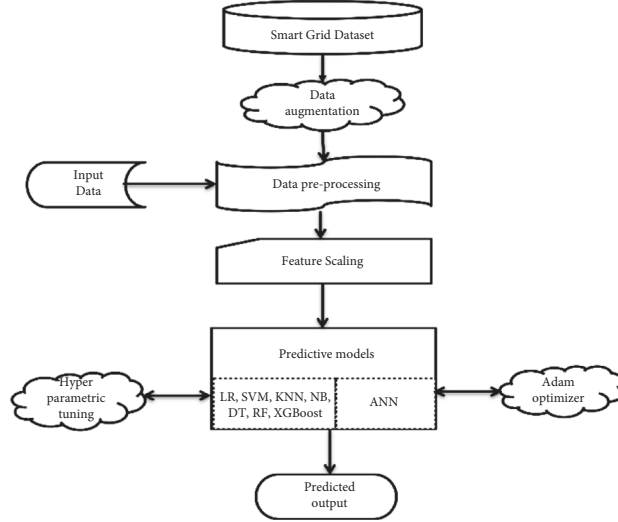


FIGURE 4: Flow chart of the methodology.

4.2.6. *Random Forest*. Random forest is made up of many decision trees. Random forest is one of the ensemble learning techniques which uses trees as a base to learn and predict [41]. Prediction is made by taking an average of the aggregate given by each tree individually. Random forest can also adapt sacristy very well [42]. Random forest supervised learning model cannot work efficiently on high-dimensional datasets. M number of multiple decision trees are combined to form a forest with a finite value.

$$m_{M,n}(x; \theta_1, \dots, \theta_m, D_n) = \frac{1}{M} \sum_{j=1}^M m_n(x; \theta_j, D_n), \quad (4)$$

where query decision query x is denoted as $m_n(x; \theta_j, D_n)$ and $\theta_1, \dots, \theta_m$ are random variables which are independent of each other and independent of D_n .

4.2.7. *XGBoost*. Exclusive gradient boosting technique has become very popular since it came into existence [43]. It is a tree boosting technique that is scalable in nature. It provides an easy way to prevent over-fitting problems [44]. It can be very helpful while using a sparse and low sample size dataset. When a dataset $D = \{(x_i, y_i)\}$ is given with n number of records and m number of features, then prediction of y_i represented as \hat{y}_i can be given with the mathematical principle of XGBoost as follows:

$$y_i^\wedge = \Phi(x_i) = \sum_{k=1}^k f_k(x_i), \quad (5)$$

where k is the adaptive functions used to predict the output and f_k is the regression tree decisive space [45].

4.2.8. *Proposed ANN*. Deep learning (DL) is a subset of machine learning. DL models like ANN (artificial neural network) [46], CNN (convolutional neural network) [47], RNN (recurrent neural network), and RBM (restricted Boltzmann machine) are one of the supervised deep learning techniques. Also, deep learning models like AE (AutoEncoders), RM, CNN, RNN, and RBM can be used as unsupervised model also [45]. For this study, we have used an optimized ANN as shown in Figure 5. ANN model has a single input layer, three hidden layers, and a single output layer. The input layer contains 12 nodes, the first and second hidden layers have 24 nodes, the third hidden layer has 24 nodes, and the output layer has a single node.

For the activation function, we choose rectified linear activation function which is popularly known as ReLU. ReLU function generally chooses the maximum value from the linear combination of inputs from the previous nodes for the output [48]. ReLU is chosen because this function gives output as either all zeros or ones. And with respect to our

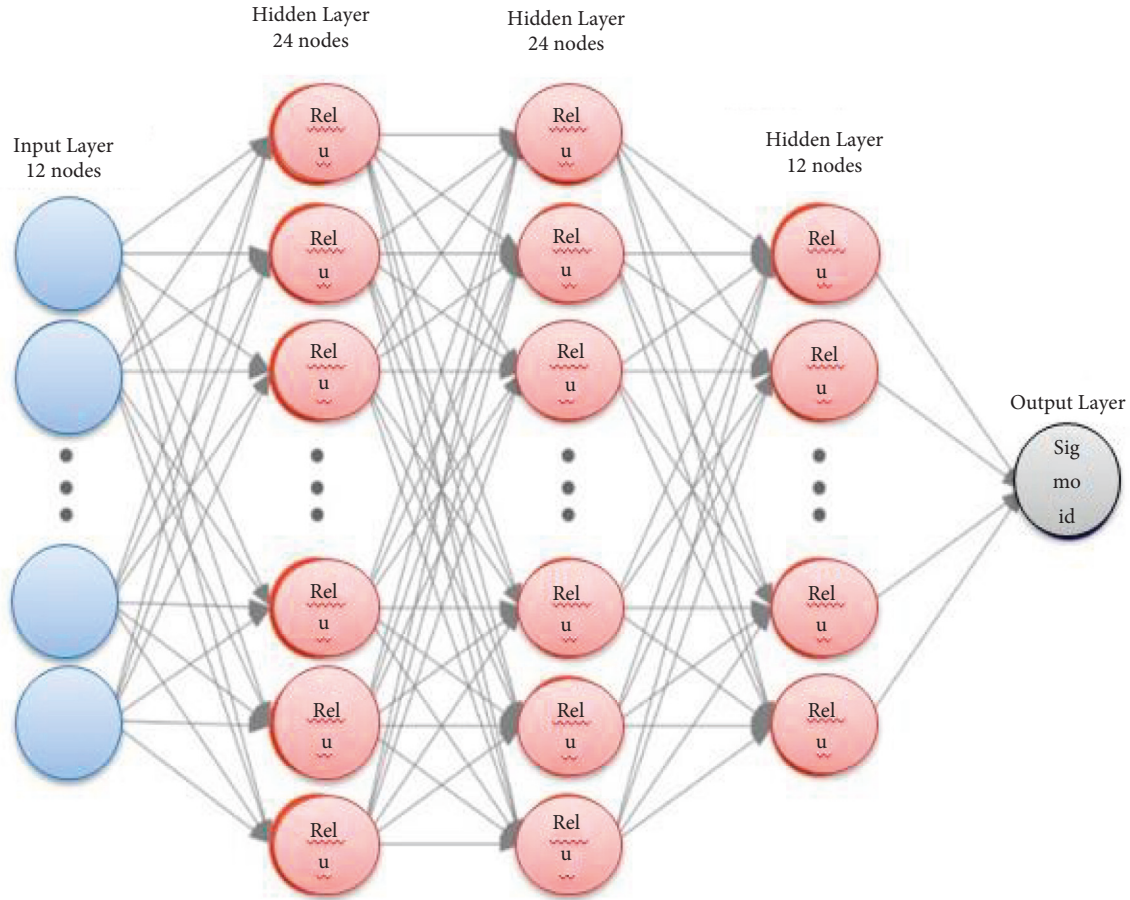


FIGURE 5: ANN model used for predictive analysis.

dataset, either the grid is stable represented as “1” or the grid is unstable represented as “0” and it has all numerical functions within a given range. “Sigmoid” function is used as an activation function for the output layer because the dataset has only two prediction classes, this implies the dataset will be classified logistically.

$$\text{ReLU}(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases}, \text{Sigmoid}(x) = \frac{1}{1 + e^{-z}}. \quad (6)$$

To enhance the functioning of ANN, an optimization technique named adaptive optimization algorithm also referred to as “Adam” is used to predict stability of the grid. Adam optimizer function helps to optimize the weights of ANN network. Adam optimizer helps to optimize the learning rate of ANN model [49].

Table 2 presents values of hyper-parameters of different machine learning and deep learning models used for this study.

5. Experimental Result and Discussion

In this section, details of the dataset used for this study are elaborated. All kinds of necessary evaluation matrices used for the analysis of the model are also elaborated in this section. Finally, performance comparison results are presented for all predictive models used.

5.1. Evaluation of Metrics. Repeated random test-train split validation method is used for consistency of machine learning models used for predictions. Repeated random test-train split is basically a hybrid technique of the Hold-out validation approach and K-Fold cross-validation method. Hold-out validation approach is basically known as train and test split.

In this method, the dataset is randomly split into training and testing datasets which may lead to under-fitting or over-fitting. K-Fold cross-validation is a technique in which the data set is divided into K folds and for all the splits, predictive models are applied. And for every split, accuracy is calculated. This helps to avoid over-fitting.

In the repeated random test-train split approach, one split the dataset in testing and training datasets, and then the splitting and evaluating processes are repeated for model prediction multiple times (as one does in K-cross-validation method). For repeated random test-train split, ShuffleSplit and cross_val_score libraries of sklearn are used in python. The number of splits used is 10 and the test dataset is 30% of the total dataset. Accuracy for each fold in repeated random test-train split approach is shown in Table 3. For every model, the highest accuracy achieved is highlighted.

In this analysis experiment, binary classification accuracy, precision, recall, F1-score, and ROC curve are considered for evaluation. This study has focused on the

TABLE 2: Hyper-parameters for predictive models.

Predictive models	Logistic regression	KNN	Naive Bayes	Decision tree	SVM	Random forest	XGBoost	Optimized ANN
Hyper-parameters used	<ul style="list-style-type: none"> ✓ Penalty = l2 ✓ Multi_class = auto ✓ c = 1.0 ✓ max_iter = 100 ✓ tol = 1e-4 	<ul style="list-style-type: none"> ✓ n_neighbour = 3 ✓ Weight = uniform ✓ algo = auto ✓ leafsize = 30 ✓ p = 2 ✓ Metric = minkowski 	<ul style="list-style-type: none"> ✓ n_class priors = none ✓ var_smoothing = 1e-09 	<ul style="list-style-type: none"> ✓ Splitter = best ✓ Criterion = entropy ✓ Max_depth = 90 ✓ Min_samples_split = 10 	<ul style="list-style-type: none"> ✓ Random_state = 1 ✓ Probability = true 	<ul style="list-style-type: none"> ✓ n_estimator = 100 ✓ criterion = gini ✓ min_sample_split = 10 ✓ max_feature = auto 	<ul style="list-style-type: none"> ✓ object = binary_logisticz ✓ n_estimator = 10 ✓ max_depth = 15 	<ul style="list-style-type: none"> ✓ Three hidden layers ✓ ReLU + Sigmoid activation function ✓ Adam optimizer

TABLE 3: Accuracy for every fold using repeated random test-train split.

S.No.	Model	Accuracy Fold1	Accuracy Fold2	Accuracy Fold3	Accuracy Fold4	Accuracy Fold5	Accuracy Fold6	Accuracy Fold7	Accuracy Fold8	Accuracy Fold9	Accuracy Fold10
1	Logistic regression	0.81405	0.81527	0.8145	0.81838*	0.8146	0.81561	0.8131	0.81116	0.81361	0.81272
2	KNN	0.813889	0.82283*	0.81522	0.82078	0.81656	0.81267	0.81639	0.81561	0.81628	0.81789
3	Naïve Bayes	0.832556	0.8315	0.82444	0.83039	0.83056	0.83256*	0.83239	0.82828	0.83311	0.83156
4	Decision tree	0.83305	0.83156	0.83183	0.837	0.83328*	0.83244	0.83178	0.8315	0.8345	0.83139
5	SVM	0.927667	0.92706	0.92622	0.92944*	0.92806	0.92628	0.92789	0.92767	0.92844	0.92911
6	Random forest	0.907056	0.90939	0.90572	0.90983	0.90928*	0.909	0.90772	0.91183	0.90817	0.91056
7	XGBoost	0.92983	0.92894	0.92806	0.9275	0.93406*	0.93206	0.92817	0.93133	0.93078	0.92794
8	Optimized ANN	0.9571	0.9705	0.9724	0.9771	0.9802	0.981	0.9781	0.9836*	0.9821	0.98

Bold face represents the highest accuracy achieved for every model among different folds.

accuracy of the model. Accuracy measures the correct values that are predicted as compared with the false value predicted and any prediction model is considered to be the best if it predicts maximum correct values. The more the accuracy is given by the model, the more it is considered to be suitable for prediction.

Accuracy (A) is calculated as a ratio of rightfully predicted and classified testing samples to the total testing sample size.

$$A = \frac{(\text{Correct predictions})}{(\text{total samples})}. \quad (7)$$

Precision (P) is also known as false alarm rate. It is calculated as the ratio of the number of correctly classified stable test samples to the total number of samples predicted as stable samples.

$$P = \frac{(\text{True Positives})}{(\text{True Positives} + \text{False Positives})}. \quad (8)$$

Recall (R) referred to as sensitivity or true positive rate, is the ratio of correctly classified stable test samples to total testing samples.

$$R = \frac{(\text{True Positives})}{(\text{True Positives} + \text{False Negative})}. \quad (9)$$

F1-score (F) is a weighted average of precision and recall.

ROC curve is basically a graph that helps in the visualization of binary classifier performance on all possible thresholds. ROC curve is plotted between true positive rate against false positive rate. In this analysis, all machine learning models are compared and visualized with the help of the ROC curve as shown in Figure 6.

As shown in Figure 6, SVM is providing the highest true positive rate with the lowest false positive value. Next, close to SVM is XGBoost classifier. So, among all machine learning models, SVM gave the best performance and KNN and logistic regression model gave the least performance.

5.2. Performance Comparison Analysis. Seven machine learning models and deep learning models are for the prediction of the stability of distributed smart energy grid.

Different parameters used for predictive analysis like accuracy, precision, recall, and $F1$ -score are already described in Section 4.2. Performance of different machine learning models is compared and shown in Table4. In this table, all parameters are taken as an average of repeated random test-train splits.

Table 4 shows the performance parameters of every model used for the study. Logistic regression model gave the least performance with an accuracy of 81.2%, precision as 75.8%, recall as 69.7%, and $F1$ -score as 72.6%. Now, as compared with the logistic regression model, KNN model gave a better performance. KNN prediction model gave an accuracy of 81.61%. Precision score for KNN is 76.3% and recall is 70.7%. $F1$ -score which is a harmonic mean of both precision and recall came out to be 73.4%.

After KNN next, Naïve Bayes is next in line. In terms of accuracy score, naïve Bayes and decision tree gave almost similar performance as accuracy for naïve Bayes model is 83.04%. But, true positive rate and correct prediction rate for decision tree are much better than that of naïve Bayes. Because precision for naïve Bayes came out to be 82.9% and recall is 66.2% for the same, $F1$ -score for naïve Bayes is 73.6%. Decision tree performs better than logistic regression, KNN model, and naïve Bayes. Accuracy for decision tree came out to be 83.28%. Precision is 86.5%, recall is 84.5%, and $F1$ -score is 85.5% for the same. According to the parameter results, random forest is next in line for better performance. Accuracy for random forest is 90.88%, precision value is 90.6%, recall is 82.4%, and $F1$ -score is 86.3% for random forest. SVM and XGBoost perform best among all machine learning models. Accuracy value for SVM and XGBoost came out to be 92.77% and 92.88%, respectively, Precision is 92.1% and 91.2%, respectively, recall is 87.1% and 88.5%, respectively, $F1$ -score is 89.5% and 89.8%, respectively.

Proposed deep learning model for smart grid, that is, optimized ANN model outperforms all other predictive models with accuracy as 97.27%, precision is 96.79%, recall is 95.67%, and $F1$ -score came out to be 96.22%. Following bar charts are used to visualize the comparative analysis of different predictive models on the basis of accuracy (Figure 7), precision (Figure 8), recall (Figure 9), and $F1$ -score

TABLE 4: Performance parameters for prediction models.

S. no.	Model	Accuracy	Precision	Recall	F1 score
1	Logistic regression	81.2	75.8	69.7	72.6
2	KNN	81.61	76.3	70.7	73.4
3	Naïve Bayes	83.04	82.9	66.2	73.6
4	SVM	92.77	92.1	87.1	89.5
5	Decision tree	83.28	86.5	84.5	85.5
6	Random forest	90.88	90.6	82.4	86.3
7	XGBoost	92.98	91.2	88.5	89.8
8	Proposed ANN	97.27	96.79	95.67	96.22

Bold face represents the best model among all other models.

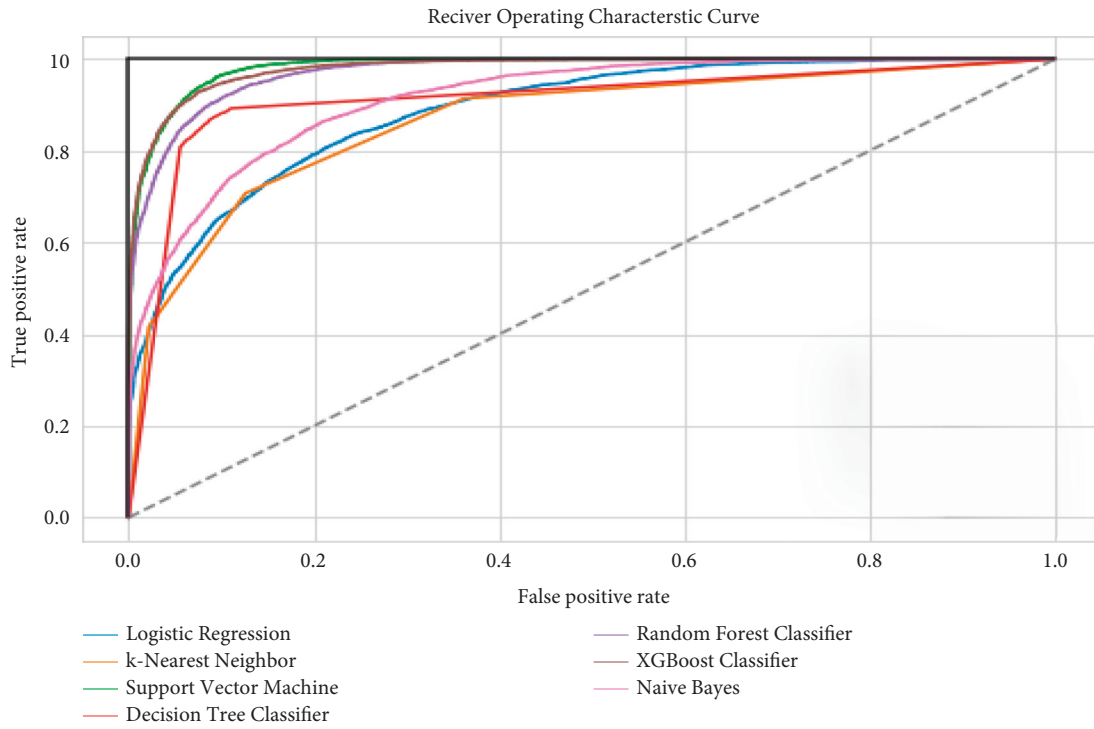


FIGURE 6: ROC curve of all machine learning models.

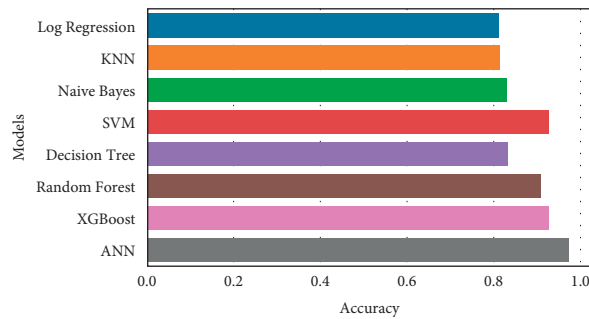


FIGURE 7: Comparative analysis of accuracy using different prediction models.

(Figure 10). On the basis of these charts and Table 4, this study concludes that ANN is the best model for stability prediction in smart grid.

Also, an augmented dataset helps to achieve better accuracy than that of the normal dataset. Dataset used in the study of Arzamasov et al. [17] used normal dataset which

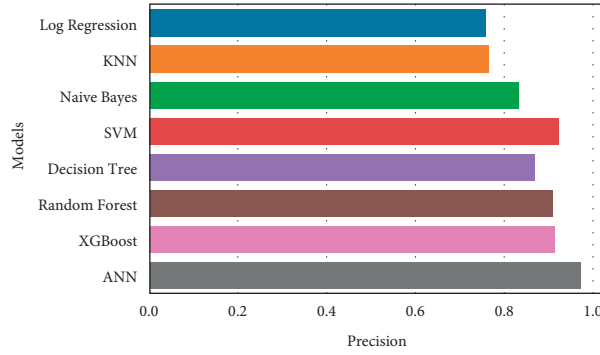


FIGURE 8: Comparative analysis of precision using different prediction models.

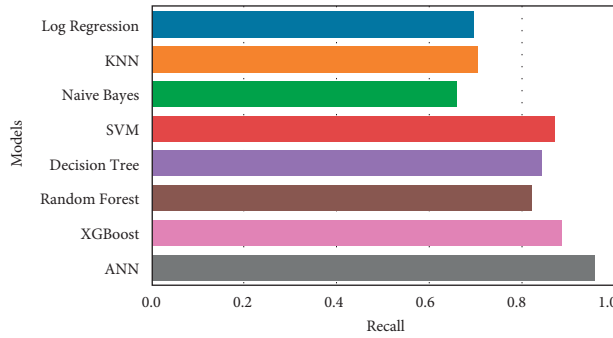


FIGURE 9: Comparative analysis of recall using different prediction models.

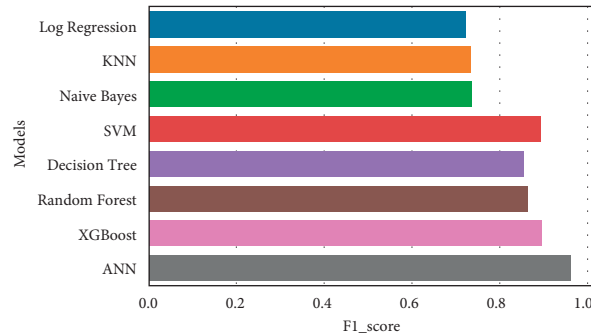


FIGURE 10: Comparative analysis of $F1$ -score using different prediction models.

TABLE 5: Comparison of the proposed model with the existing approach.

S. No.	Model	Accuracy (%)
1	Decision tree (Arzamasov et al. [17])	80
2	Proposed optimized ANN	97.27

gave an accuracy of around 80% for decision tree while augmented dataset with proposed model gave an accuracy of 97.27%. Table 5 shows the comparison of the proposed optimized ANN model with the model in this field with respect to accuracy.

6. Conclusion

Industry 4.0 has shown its impact by means of Industrial Internet viz. through IoT enabling technologies, taking smart decisions, automating industries deploying

machine learning, deep learning, data analytics, etc. Power grid is one of the domains where all these technologies are used to make it smart. In smart grid, producers and consumers are connected through communication lines or the Internet. IoT can help to convert conventional energy networks to smart energy grids. This article presents a three-layered integrated smart grid IoT network. This integration can play a very major role to predict the stability of smart power grid in near future using data collected from these communication lines or the Internet.

Different predictive models like logistic regression, naïve Bayes, KNN, decision tree, SVM, random forest, XGBoost, and optimized ANN are used to analyze openly available smart grid datasets. These predictive models are then compared based on accuracy, precision, recall, and F1-score. Also, ROC curve is used for the comparison of all the predictive models. Dataset is available on UCI machine learning repository and this dataset is firstly augmented and then scaled to find better results. As a result, we found that the decision tree predictive model on normal dataset gave an accuracy of 80% and the augmented dataset gave an accuracy of 83.28% for the same model.

Among all the predictive models, ANN optimized with Adam optimizer with ReLU and Sigmoid activation function gave an accuracy of 97.27% that wins the race. All comparative matrices are presented using bar charts. These bar charts help to conclude that optimized ANN which is a deep learning model outperformed the remaining machine learning predictive models. Further, other deep learning models will be proposed which may give much better results for prediction. A real-time dataset may be used in near future, to apply the same predictive models and analyze their efficiency for it.

Data Availability

Data will be available on request from the submitting author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] F.A. Almalki, S. H. Alsamhi, R. Sahal et al., "Green IoT for eco-friendly and sustainable smart cities: future directions and opportunities," *Mobile Networks and Applications*, vol. 19, p. 3, 2021.
- [2] V. V. S. Lanka, M. Roy, S. Suman, and S. Prajapati, "Renewable energy and demand forecasting in an integrated smart grid," in *2021 Innovations in Energy Management and Renewable Resources (Iemre 2021)*, A. K. Bar, M. Pal, S. Ghosh et al., Eds., IEEE, New York, 2021.
- [3] M. Jayachandran, C. R. Reddy, S. Padmanaban, and A. H. Milyani, "Operational planning steps in smart electric power delivery system," *Scientific Reports*, vol. 11, no. 1, p. 17250, 2021.
- [4] M. A. Mahmoud, A. Y. C. Tang, A. Maselena, F.-C. Lim, H. Kasim, and C. Yong, "Towards the development of a smart energy grid," in *Emerging Trends in Intelligent Computing and Informatics: Data Science, Intelligent Information Systems and Smart Computing*, F. Saeed, F. Mohammed, and N. Gazem, Eds., Springer International Publishing Ag, Cham, pp. 673–682, 2020.
- [5] B. Schäfer, M. Matthiae, M. Timme, and D. Witthaut, "Decentral smart grid control," *New Journal of Physics*, vol. 17, no. 1, p. 015002, 2015.
- [6] S. Azad, F. Sabrina, and S. Wasimi, "Transformation of smart grid using machine learning," in *2019 29th Australasian Universities Power Engineering Conference (Aupec)IEEE*, New York, 2019.
- [7] G. Kaur, P. Tomar, and P. Singh, "Design of cloud-based green IoT architecture for smart cities," in *Internet of Things and Big Data Analytics toward Next-Generation Intelligence, Studies in Big Data*, N. Dey, A. E. Hassanien, C. Bhatt, A. S. Ashour, and S. C. Satapathy, Eds., Springer International Publishing, Cham, pp. 315–333, 2018.
- [8] N. M. Kumar, A. A. Chand, M. Malvoni et al., "Distributed energy resources and the application of ai, iot, and blockchain in smart grids," *Energies*, vol. 13, no. 21, p. 5739, 2020.
- [9] S. M. A. A. Abir, A. Anwar, J. Choi, and A. S. M. Kayes, "IoT-Enabled smart energy grid: applications and challenges," *IEEE Access*, vol. 9, pp. 50961–50981, 2021.
- [10] M. Baba, M. U. Tariq, and M. A. Jan, "Secure and resilient demand side management engine using machine learning for IoT-enabled smart grid," *Sustainable Cities and Society*, vol. 62, 2020.
- [11] P. Tomar, G. Kaur, and P. Singh, "A prototype of IoT-based real time smart street parking system for smart cities," in *Internet of Things and Big Data Analytics toward Next-Generation Intelligence, Studies in Big Data*, N. Dey, A. E. Hassanien, C. Bhatt, A. S. Ashour, and S. C. Satapathy, Eds., Springer International Publishing, Cham, pp. 243–263, 2018.
- [12] M. Elsisy, K. Mahmoud, M. Lehtonen, and M. M. F. Darwish, "Reliable industry 4.0 based on machine learning and IoT for analyzing, monitoring, and securing smart meters," *Sensors*, vol. 21, no. 2, p. 487, 2021.
- [13] M. Babar, M. U. Tariq, and M. A. Jan, "Secure and resilient demand side management engine using machine learning for IoT-enabled smart grid," *Sustainable Cities and Society*, vol. 62, p. 102370, 2020.
- [14] R. Yao, J. Li, B. Zuo, and J. Hu, "Machine learning-based energy efficient technologies for smart grid," *International Transactions on Electrical Energy Systems*, vol. 31, no. 9, p. e12744, 2021.
- [15] P. Sharmila, J. Baskaran, C. Nayanatara, and R. Maheswari, "A hybrid technique of machine learning and data analytics for optimized distribution of renewable energy resources targeting smart energy management," in *2nd International Conference on Recent Trends in Advanced Computing Ictac-Disrup - TivInnovation, 2019*, P. Nithyanandam, R. Parvathi, R. J. Kannan, and A. N. Khan, Eds., Elsevier Science Bv, Amsterdam, pp. 278–284, 2019.
- [16] W. Ahmed, H. Ansari, B. Khan et al., "Machine learning based energy management model for smart grid and renewable energy districts," *IEEE Access*, vol. 8, pp. 185059–185078, 2020.
- [17] V. Arzamasov, K. Böhm, and P. Jochem, "Towards concise models of grid stability," in *Proceedings of the 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, IEEE, Aalborg, Denmark, 29-31 October 2018.
- [18] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2019.
- [19] Y. Wang, Y. Kong, X. Tang et al., "Short-term industrial load forecasting based on ensemble hidden Markov model," *IEEE Access*, vol. 8, pp. 160858–160870, 2020.
- [20] Y. Wang, S. Sun, X. Chen et al., "Short-term load forecasting of industrial customers based on SVM and XGBoost," *International Journal of Electrical Power & Energy Systems*, vol. 129, p. 106830, 2021.

- [21] S. Gupta, R. Kambli, S. Wagh, and F. Kazi, "Support-vector-machine-based proactive cascade prediction in smart grid using probabilistic framework," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2478–2486, 2015.
- [22] Q. Ge, C. Guo, H. Jiang et al., "Industrial power load forecasting method based on reinforcement learning and PSO-lssvm," *IEEE Transactions on Cybernetics*, vol. 52, no. 2, pp. 1112–1124, 2022.
- [23] L. Wei, D. Gao, and C. Luo, "False data injection attacks detection with deep Belief networks in smart grid," in *Proceedings of the 2018 Chinese Automation Congress, CAC*, pp. 2621–2625, Xi'an, China, 30 November 2018 - 02 December 2018.
- [24] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1483–1488, IEEE, Edinburgh, UK, 19–21 June 2017.
- [25] A. Muzumdar, C. N. Modi, G. M. Madhu, and C. Vyjayanthi, "Analyzing the feasibility of different machine learning techniques for energy imbalance classification in smart grid," in *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (Iccnt)*, 06–08 July 2019.
- [26] Z. Guan, Z. Lv, X. Sun et al., "A differentially private big data nonparametric bayesian clustering algorithm in smart grid," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2631–2641, 2020.
- [27] H. Wang, Z. Yemeni, W. M. Ismael, A. Hawbani, and S. H. Alsamhi, "A reliable and energy efficient dual prediction data reduction approach for WSNs based on Kalman filter," *IET Communications*, vol. 15, no. 18, pp. 2285–2299, 2021.
- [28] S. H. Alsamhi, F. Almalki, O. Ma, M. S. Ansari, and B. Lee, "Predictive estimation of optimal signal strength from drones over IoT frameworks in smart cities," *IEEE Transactions on Mobile Computing*, vol. 1, p. 1, 2021.
- [29] V. O. Nyangaresi and S. H. Alsamhi, "Towards secure traffic signaling in smart grids," in *Proceedings of the 2021 3rd Global Power, Energy and Communication Conference (GPECOM)*, pp. 196–201, IEEE, Antalya, Turkey, 05–08 October 2021.
- [30] *UCI Machine Learning Repository: Electrical Grid Stability Simulated Data Data Set*, Available at: <https://archive.ics.uci.edu/ml/datasets/Electrical+Grid+Stability+Simulated+Data+#> accessed, 2018.
- [31] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, vol. 24, no. 1, pp. 12–18, 2014.
- [32] R. H. Myers, *Classical and Modern Regression with Applications*, Duxbury press, vol. 2, p. 488, Belmont, CA, 1990.
- [33] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN Classification," *ACM Trans. Intell. Syst. Technol.* vol. 8, no. 3, pp. 1–43, 2017.
- [34] Y. Weng, R. Negi, C. Faloutsos, and M. D. Ilić, "Robust data-driven state estimation for smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1956–1967, 2017.
- [35] O. Valgaev, F. Kupzog, and H. Schmeck, "Low-voltage power demand forecasting using K-nearest neighbors approach," in *Proceedings of the 2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, pp. 1019–1024, IEEE, Melbourne, VIC, Australia, 28 November 2016 - 01 December 2016.
- [36] D. Vidaurre, C. Bielza, and P. Larrañaga, "Forward stagewise naïve Bayes," *Progress in Artificial Intelligence*, vol. 1, pp. 57–69, 2012.
- [37] N. F. Lepora, M. Evans, C. W. Fox, M. E. Diamond, K. Gurney, and T. J. Prescott, "Naive Bayes texture classification applied to whisker data from a moving robot," in *Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Barcelona, Spain, 18–23 July 2010.
- [38] I. H. Wijesinghe and S. D. Viswakula, "Machine learning for pre-auction sample selection," in *Proceedings of the 2018 National Information Technology Conference (NITC)*, pp. 1–8, IEEE, Colombo, Sri Lanka, 02–04 October 2018.
- [39] C. T. Yildiz and E. Alpaydin, "Omnivariate decision trees," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1539–1546, 2001.
- [40] P. H. Swain and H. Hauska, "The decision tree classifier: design and potential," *IEEE Transactions on Geoscience Electronics*, vol. 15, no. 3, pp. 142–147, 1977.
- [41] E. Scornet, "Random forests and kernel methods," *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1485–1500, 2016.
- [42] E. Scornet, G. Biau, and J.-P. Vert, "Consistency of random forests," *Annals of Statistics*, vol. 43, no. 4, pp. 1716–1741, 2015.
- [43] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pp. 785–794, Association for Computing Machinery, New York, NY, USA, 2016.
- [44] M. Ding, L. Zhao, H. Pei, and M. Song, "An XGBoost based evaluation methodology of product color emotion design," *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 15, vol. 15, p. 6, 2021 JAMDSM0075–JAMDSM0075.
- [45] A. Chahal and P. Gulia, "Deep learning: a predictive IoT data analytics method," *International Journal of Engineering Trends and Technology*, vol. 68, no. 7, pp. 25–33, 2022.
- [46] A. Sagu, N. S. Gill, and P. Gulia, "Artificial neural network for the internet of Things security," *International Journal of Engineering Trends and Technology*, vol. 68, no. 11, pp. 129–136, 2018.
- [47] G. Altan, "DeepOCT: an explainable deep learning architecture to analyze macular edema on OCT images," *Engineering Science and Technology, an International Journal*, vol. 34, p. 101091, 2022.
- [48] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: an efficient SMT solver for verifying deep neural networks," in *Computer Aided Verification, Lecture Notes in Computer Science*, R. Majumdar and V. Kunčák, Eds., Springer International Publishing, Cham, pp. 97–117, 2017.
- [49] Z. Zhang, "Improved Adam optimizer for deep neural networks," in *Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service, Banff, AB, Canada*, 04–06 June 2018.