Hindawi
Complexity
Volume 2022, Article ID 9928836, 19 pages
https://doi.org/10.1155/2022/9928836

Research Article

An Approach for Demand Forecasting in Steel Industries Using Ensemble Learning

S. M. Taslim Uddin Raju,[1] **Amlan Sarker**,[2] **Apurba Das**,[3] **Md. Milon Islam**,[1] **Mabrook S. Al-Rakhami**,[4] **Atif M. Al-Amri**,[4,5] **Tasniah Mohiuddin**,[6] **and Fahad R. Albogamy**[7]

[1]*Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh*
[2]*Department of Computer Science and Engineering, Daffodil International University, Dhaka 1207, Bangladesh*
[3]*Department of Industrial Engineering and Management, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh*
[4]*Research Chair of Pervasive and Mobile Computing, Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia*
[5]*Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia*
[6]*Department of Computer Science and Engineering, Military Institute of Science and Technology, Dhaka 1216, Bangladesh*
[7]*Computer Sciences Program, Turabah University College, Taif University, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Mabrook S. Al-Rakhami; malrakhami@ksu.edu.sa

Received 4 April 2021; Revised 11 November 2021; Accepted 8 December 2021; Published 25 February 2022

Academic Editor: Shahzad Sarfraz

This paper aims to introduce a robust framework for forecasting demand, including data preprocessing, data transformation and standardization, feature selection, cross-validation, and regression ensemble framework. Bagging (random forest regression (RFR)), boosting (gradient boosting regression (GBR) and extreme gradient boosting regression (XGBR)), and stacking (STACK) are employed as ensemble models. Different machine learning (ML) approaches, including support vector regression (SVR), extreme learning machine (ELM), and multilayer perceptron neural network (MLP), are adopted as reference models. In order to maximize the determination coefficient (R^2) value and reduce the root mean square error (RMSE), hyperparameters are set using the grid search method. Using a steel industry dataset, all tests are carried out under identical experimental conditions. In this context, STACK_1 (ELM + GBR + XGBR-SVR) and STACK_2 (ELM + GBR + XGBR-LASSO) models provided better performance than other models. The highest accuracies of R_2 of 0.97 and 0.97 are obtained using STACK_1 and STACK_2, respectively. Moreover, the rank according to performances is STACK_1, STACK_2, XGBR, GBR, RFR, MLP, ELM, and SVR. As it improves the performance of models and reduces the risk of decision-making, the ensemble method can be used to forecast the demand in a steel industry one month ahead.

1. Introduction

Demand forecasting indicates the prediction of the future needs of a product or service [1]. It is necessary to follow a procedure to attain a crystalline graph of the demand for identifying the pulse of the customer's need to hold their position in the market. From the last era, the steel industry in Bangladesh is a fast-growing industry in the local market. The industries managed to manufacture a large amount of steel to fulfill both local and international markets, but producing a large amount of steel without proper forecasting causes various problems. Demand prediction is used to support many fundamental business assumptions, including turnover, total revenues, income, capital consumption, chance evaluation and moderation plans, scope quantification, transportation and distribution plans, and more. Any type of misdeed assessment could cost decaying or scarcity of raw materials. It can also lead to overproduction or underproduction. All these cases erode the entire supply chain and total income, resulting in opportunity cost. Again, the entire industry setup depends on this demand, such as the amount of raw material, labor, and space. For these whole arrangements, time is also a crucial issue, as some processes have predefined deadlines that must be perfectly synchronized. For smart business strategy, the most important thing is to forecast the demand precisely but the industries do not have any intelligent method to measure the need perfectly. They follow the time series of their sales data and often skip factors, such as raw material supply, availability, and the number of workers at the factories, significantly influencing steel production.

Forecasting methods can be classified into three categories: (1) statistical methods, (2) artificial intelligence-based methods such as single machine learning (ML) methods, and (3) ensemble/hybrid methods. Most steel industries in Bangladesh use traditional statistical approaches. Statistical approaches, such as exponential smoothing [2], moving average [3], autoregressive moving average [4], and autoregressive integrated moving average [5], are most frequently used for time series prediction. The major drawbacks of these techniques are that the parameter values are fixed using statistical calculations. The error of estimation increases when the fluctuations in the entered data are high and do not yield convincing results for complicated time series patterns [6]. Thus, the companies need an intelligent decision support system that considers several factors.

Several researchers reveal that in the investigation of most cases, ML approaches have drawn much attention and could provide more accuracy than could traditional approaches [7]. Single artificial intelligence-based models, such as support vector machine (SVM), extreme machine learning, heuristic techniques, and multilayer perceptron (MLP), are widely used in various industrial aspects to predict demands because they demonstrate promising results in the areas of control, prediction, and pattern recognition [8–10]. Support vector regression (SVR) is popular for predicting future demand because of its outstanding generalization capability and no dependency over input space dimensionality [11]. It produces higher accuracy in agribusiness prediction [8] and supply chain demand forecasting [12]. Recently, MLP is used for monthly water demand prediction [10], wind speed prediction [13], and water demand prediction [14]. For improving MLP's prediction accuracy, different MLP architectures were used, and an optimization algorithm was used to tune its parameters [15]. The extreme learning machine (ELM) is another advanced model, which is a single hidden layer feed-forward neural network (SLFN) model with incremental learning speed and fast convergence, making it efficient and fast in learning [16]. It is widely used in applications, such as sales forecasting demand of fashion retailing [17] and sales prediction for the retail industry [18].

Since demand forecasting in steel industries is considerably challenging, it is impossible to solve this problem accurately using single ML models. No single model is ideally suited for various ML applications. Each method and application domain has some prerequisites, advantages, assumptions, and characteristics [19]. Generally, the performance of combined forecasting models is better than that of a single forecasting model [20]. The literature has described several strategies to enhance the predictive performance of regression models, and one of these is the regression ensemble [8]. The regression ensemble theory is built on ML, whose roots are related to the concept of divide-to-conquer, solving the constraints of ML models working in isolation [21]. An ensemble model is one in which numerous base models are constructed to address the same problem, with each model learning the dataset's feature attributes and making a prediction. As a result, the separate model's forecasts are integrated to generate the final projection. By combining the mean or weighted average, ensemble approaches for regression problems can be developed. The simple method of grouping regression ensembles by mean and weighted average is to use mean and weighted average. The regression ensemble models construct a collection of models in order to improve the predictive power of the selected models and the numerical goal variables [22, 23]. Ensemble methods are used in several studies, such as forecasting for energy consumption [24], agribusiness prediction [8], and wind power forecasting [25]. Although numerous frameworks have been established, there is always a need for improved forecasting accuracy and robustness, particularly in the steel industry.

This study proposes a new pipeline for demand forecasting in steel industries. From this aspect, this study explores the capacity of predictive regression ensemble models by comparing the ensembles among themselves and considering the single reference models to forecast the demand. The proposed pipeline includes data preprocessing, feature selection, hyperparameter tuning, cross-validation, and regression ensemble approaches to outperform the state-of-the-art results. Instead of using the median value of the attribute, the mean value of the attribute is utilized to fill in the empty area since it has a more central tendency to the mean of the attribute distribution than the median. The appropriate features are selected using feature selection algorithms (correlation-based, principal component analysis (PCA), and independent component analysis (ICA)) to avoid redundancy and model overfitting problems. Different single ML techniques, such as SVR, MLP, and ELM, are adopted as reference models. The ensemble bagging (RFR), boosting (GBR and XGBR), and stacking (STACK) models are used in our proposed framework to enhance demand forecasting robustness and efficiency. The grid search technique with cross-validation is used to select the optimal hyperparameters for each ML model. Comprehensive experiments are conducted on different data preprocessing and a combination of ML techniques to minimize the RMSE and maximize R^2 of demand forecasting models. All experiments are carried out under the same experimental settings and with the same data set as the previous experiment. Finally, we investigate the performance of regression ensemble approaches and verify that ensemble approaches outperform single reference models. The contributions of this paper are summarized as follows:

(i) Collect the dataset from a well-known steel industry in Bangladesh.

(ii) Present a modification of the theory underlying regression ensembles based on bagging (RFR), boosting (GBR and XGBR), and stacking (STACK) as well as single models (SVR, MLP, and ELM) (details in Supplementary Material Appendix A).

(iii) Find the best preprocessing pipeline using filling missing values, data transformation, standardization, and feature selection algorithms where the number of selected features is also varied.

(iv) Implement different ML regression models with its optimal hyperparameters, obtained using grid search algorithms with cross-validation. Investigate and analyze the performance of bagging, boosting, and stacking ensemble approaches and compare them with each other on the same dataset and preprocessing under the same experimental condition.

(v) Verify the superiority of the proposed ensemble approaches using Friedman test and Wilcoxon signed rank test.

The remainder of the paper is arranged in the following manner: Section 2 describes a collection of related studies for the purpose of forecasting. Section 3 illustrates the suggested approach, dataset, feature selection methods, and assessment measures. Various experimental findings are documented in Section 4 based on the interpretation of the data. Section 5 provides a conclusion as well as a scope for further development.

2. Related Works

Forecasting demand for industrial products is an urgent matter since a massive portion of a company's planning process is based on the amount of product to be produced. To meet the increasing demand, precise demand forecasting is required. In this section, we will discuss the work that has been done to anticipate demand in a variety of disciplines and will describe numerous exemplary studies.

Ribeiro and dos Santos Coelho [8] proposed a system for agribusiness prediction using ensemble methods. Bagging, boosting, and stacking ensembles along with single reference models named SVR, MLP, and KNN were used for their purposes. In this experiment, it was shown that ensemble methods performed better than single models. They obtained MAPE of 0.9787 and 0.7394 for both cases for best ensemble models. They did not apply any metaheuristics algorithm for optimizing hyperparameters. Yu et al. [9] developed an ensembling and decomposition algorithm with EEML for crude oil price forecasting. In Ref. [12], they introduced a system by ensembling regression algorithms and time series algorithms to forecast the supply chain demand. The system showed superior outcome because of the reality of invalidating the over-gauging and under-determining. Cankurt [26] employed a variety of regression models, including M5P and M5-Rule model trees, bagging, boosting, randomization, stacking, and voting, to anticipate tourism demand. In this case, they obtained R of 0.986 and a RAE of 14.96. The bagging and boosting methods have great significance for the improvement of performances in regression tree models.

Yang et al. [27] developed a system for forecasting agriculture commodities using the bagging and combining approaches with the Heterogeneous Autoregression (HAR) model. HAR model along with bagging and the principal component combination shows outstanding performance for agriculture commodities forecasting. In Ref. [28], they introduced a system by ensembling empirical mode decomposition (EEMD) to analyze global food price volatility. Tao et al. [29] proposed a method using a combination of ensemble empirical mode decomposition (EEMD), extreme learning machine (ELM), and ARIMA for forecasting hog price. They obtained the best-estimated accuracy of R = 0.848. Ribeiro et al. [30] designed nonlinear prediction models based on ensemble aggregation in order to improve the prediction accuracy of electricity load forecasting. In the proposed system, they used hourly load values from Italy in 2015 and Global Energy Forecasting Competition in 2012 to validate their proposed framework. Compared to the multilayer perceptron neural network (MPNN) and regression tree approach, their proposed forecasting framework based on wavelet ensemble provided a better performance.

da Silva et al. [31] introduced a decomposition-ensemble learning strategy for multi-step forward extremely short-term forecasting, which involved aggregating many regression models. They employed a range of preprocessing strategies to account for the system's high degree of input correlation. Across all time horizons, the proposed models outperform the CEEMD, STACK, and single models. In Ref. [32], they presented an excellent rolling decomposition-ensemble model for gasoline forecasting, which was both accurate and efficient. The researchers' experimental results demonstrate that the rolling decomposition-ensemble model is both accurate and resilient when it comes to projecting gasoline consumption levels and trends. A unique wind speed ensemble forecasting system (WSEFS) was developed by Liu et al. [33] in order to enhance point forecasting (PF) and interval forecasting (IF). They obtained MAPE of 1.9322%, 2.1579%, and 2.2808% for the 1st step, 2nd step, and 3rd step, respectively. The experimental results showed that the MOMA ensemble forecasting system is better than MOGWO and MODA. In order to estimate the sediment movement in open channels, Ebtehaj and Bonakdari [35] developed the ELM algorithm [35]. In all training and testing modes, the FFNN-ELM outperformed the FFNN-BP and GP methods, which were previously used. For the testing mode, they found RMSE = 0.121 and MARE = 0.023, respectively.

Considering the existing literature in Table 1, it is observed that ensemble models contribute significantly to determine predictions, more than traditional models in each

Table 1: Summary of most recent works for demand forecasting in various fields with their input factors and performances.

Publication	Objectives	Domain	Performance	Finding
Ribeiro and dos Santos Coelho [8]	Ensembling bagging (RFR), boosting (GBR and XGBR), and stacking (STACK), as well as adopting reference model SVR, MLP, and KNN	Agribusiness prediction	MAPE = 0.0093–1.6354, RMSE = 0.0013–0.0680	The ensemble approach outperforms the single models, especially the STACK model
Yu et al. [9]	Developing an ensembling and decomposition algorithm with ensemble empirical mode decomposition (EEMD) and extended extreme learning machine (EELM)	Forecasting crude oil price	MAPE = 0.0003, RMSE = 0.1431	This ensembling method shows better results than some existing popular model as well as single model in terms of accuracy, speed, and resilience
Adhikari et al. [12]	Ensembling time series methods and regression techniques in order to reduce forecast error from the actual value	Supply chain demand forecasting	TS FACC = 62% − 69% Reg FACC = 62% − 65%, En FACC = 65% − 71%	Showed superior outcome because of the reality of invalidating the overgauging and underdetermining and bringing the conjecture esteems closer to the genuine in the vast majority of the cases
Cankurt [26]	Developing M5P and M5-Rule model trees, randomization, boosting, bagging, voting, and stacking in order to anticipate the demand for tourism in Turkey	Tourism demand forecasting	$R = 0.9866$, $R^2 = 0.973$, RAE = 14.96, and RRSE = 16.77	The bagging and boosting methods have great significance for the improvement of performances in regression tree models
Yang et al. [27]	Developing the bagging and combining approaches with heterogeneous autoregression (HAR) model for the prediction of agriculture commodities' future	Forecasting agriculture commodities	$R^2 = 0.6263 - 0.3080$	HAR model with bagging shows outstanding performance comparing with AR benchmark
Wang et al. [28]	Ensembling empirical mode decomposition to analyze global food price volatility	Forecasting food price volatility	MSE = 74.29, MAE = 6.969, MAPE = 3.799	This model can successfully analyze the fluctuation of 3 types of agricultural commodities
Tao et al. [29]	Developing a combination of EEMD, ELM, and ARIMA	Forecasting hog price	$R = 0.281 - 0.848$	This model outperforms for the selected parts and claimed itself as an alternative for short-term forecasting for hog price
Ribeiro et al. [30]	Design of nonlinear prediction models for the ensemble aggregation of waveNet ensemble	Electricity load time series	–	All preprocessing stages and aggregation techniques contribute to overall performance, although perhaps not all to the same extent as a ceiling analysis would indicate
da Silva et al. [31]	For multi-step forward extremely short-term forecasting, decomposition-ensemble learning approaches are used. These methods include K-Nearest neighbors (KNN), partial least squares regression (PLSR), Ridge regression (RR), support vector regression (SVR), and Cubist regression (CR).	Wind energy forecasting	MAE = 101.32, MAPE = 8.63, RMSE = 138.97	CEEMD–BC–STACK a stacking-ensemble learning technique that significantly improved the accuracy of weak models CEEMD by merging and forecasting with a strong model.

Table 1: Continued.

Publication	Objectives	Domain	Performance	Finding
Yu et al. [32]	Proposing decomposition-ensemble learning model (ARIMA, SVR, ANN, RVFL, KRR, and ELM)	Gasoline forecasting	MAPE = 0.02 – 0.04	Decomposition-ensemble is better for prediction. Ensemble model or instantaneous frequency analysis is applicable for complex and irregular characteristics.
Liu et al. [33]	Developing a revolutionary wind speed ensemble forecasting system (WSEFS) to enhance point forecasting (PF) and interval forecasting (IF)	Wind speed forecasting	MAPE – 1st step, 2nd step, and 3rd step are 1.9322%, 2.1579%, and 2.2808%, respectively.	VMD technology is better than mayfly algorithm (MA) and ICEEMDAN. MOMA ensemble forecasting system is better than MOGWO and MODA.
Cook and Weisberg [34]	Developing imperialist competitive algorithms (ICA) and particle swarm optimization (PSO) algorithms were compared with the results of the MLP neural network trained with the back propagation algorithm	Nondeposition sediment transport prediction	MAPE = 2.7% – 6.52% and RMSE = 0.009 – 0.042	In comparison to the PSO and MLP algorithms, the ICA method is more accurate for computing the densimetric Froude number in pipe channels
Ebtehaj and Bonakdari [35]	Developing an extreme learning machine (ELM) and comparing with back propagation (BP), genetic programming (GP), and existing sediment transport equation	Sediment transport estimation	RMSE = .309 and MARE = .059	FFNN-ELM performs well and is also an alternative method in predicting the Fr

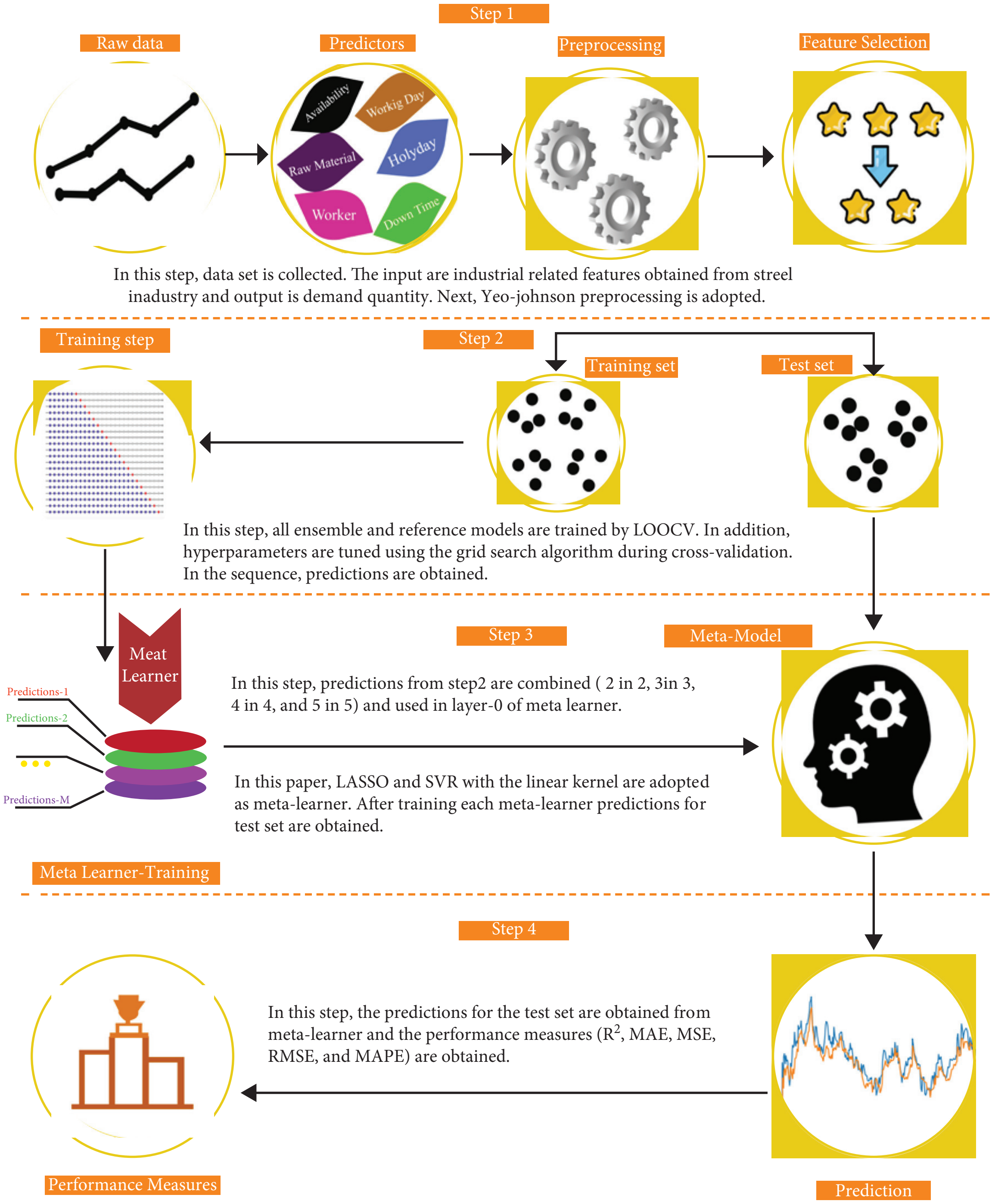
∗ MAPE = mean absolute percentage error, RMSE = root mean square error, MSE = mean square error, MAE = mean absolute error, R = Pearson correlation, R^2 = coefficient of determination, FACC = forecast accuracy check, TS FACC = time series FACC, Reg FACC = regression FACC, En FACC = ensemble FACC, RAE = relative absolute error, RRSE = root relative square error, CEEMD = complete ensemble empirical mode decomposition, VMD = variational mode decomposition, MOMA = multiobjective Mayfly algorithm, ICEEMDAN = improved complete ensemble empirical mode decomposition with adaptive noise, MOGWO = multiobjective grey wolf optimizer, MODA = multiobjective dragonfly algorithm, FFNN = feed-forward neural network, Fr = densimetric Froude number, MARE = mean absolute relative error.

case. Although several frameworks have already been developed, there is still a need for improvement in the accuracy and robustness of demand forecasting, especially in the steel industry. To sum up, there is up to now no proper pipeline for data preprocessing, features selection, hyperparameter tuning, and finally developed a regression ensemble method. This study uses bagging, boosting, and two-level stacking ensemble methods by analyzing the time series of historical data from the steel industry to achieve more propriety of forecasting results for demand. The steel industry follows the traditional time series trend to predict the demand, which fluctuates at a high quantity. To avoid this problem, this study combines multiple approaches instead of using a traditional single method to determine the precise result for the industry.

3. Materials and Methods

This section contains a concise description of the materials and method used. The suggested framework is depicted in Figure 1. The following are the primary phases in our suggested framework: (i) collection of industrial environmental data as the primary inputs of the framework; (ii) preprocessing the data including filling the missing values, Yeo–Johnson transformation, and standardization; (iii) discarding the irrelevant and redundant features to avoid overfitting of the models; (iv) applying the grid search algorithm with cross validation for hyperparameter tuning for each machine learning model; (v) development of two-level stacking ensemble method, where machine learning models with optimal hyperparameters are used as the baseline model; and (vi) evaluation metrics used to evaluate the proposed framework. These blocks are explained in the following sections.

3.1. Data Collection. The data were collected from a well-known prominent steel company named Bangladesh Steel Re-Rolling Mills Ltd., in Chittagong, Bangladesh. During the industrial attachment, some raw data were procured from sources, such as workers, production leaders, and human resources. Later, the data were closely knitted to build the dataset. The dataset comprises 132 cases and six input features from January 2009 to December 2019 (11 years). The key responsibility is to identify the demand of every month based on other factors. The dataset holds the amount of raw material used in a month, availability, the number of workers, working days, and other attributes. The data were gathered from their monthly and annual industrial reports from their official website, such as financial reports,



In this step, data set is collected. The input are industrial related features obtained from streel inadustry and output is demand quantity. Next, Yeo-johnson preprocessing is adopted.

In this step, all ensemble and reference models are trained by LOOCV. In addition, hyperparameters are tuned using the grid search algorithm during cross-validation. In the sequence, predictions are obtained.

In this step, predictions from step2 are combined (2 in 2, 3in 3, 4 in 4, and 5 in 5) and used in layer-0 of meta learner.

In this paper, LASSO and SVR with the linear kernel are adopted as meta-learner. After training each meta-learner predictions for test set are obtained.

In this step, the predictions for the test set are obtained from meta-learner and the performance measures (R^2, MAE, MSE, RMSE, and MAPE) are obtained.

Figure 1: Proposed architecture of automatic demand forecasting.

production reports, and some other necessary factors directly affecting their production achievements. Table 2 describes each feature and shows a statistical summary.

3.2. Data Preprocessing. The data preprocessing stage comprises missing value imputation and power transformation of data. Raw data inherit some missing attributes from various features that must be filled before applying any ML technique. Several imputation techniques can fill missing values. In our proposed method, the mean-based imputation technique is used, where the missing value is filled with the mean of the attributes of that specific feature.

After the imputation of missing or null values, the data power transformation is performed. In regression analysis, transformations are crucial [36]. Parametric, monotonic transformations are power transformations used to make data more Gaussian-like. This technique is useful in heteroscedasticity problems or other circumstances where data normality is required. Among the two most popular power

Table 2: Statistical description of influence features for forecasting the demand.

Indices	F_1	F_2	F_3	F_4	F_5	F_6	Outcome
Mean	8231.97	52444.08	167.42	28.93	1.46	60.09	51205.62
S.D.	2269.08	12385.54	43.10	1.28	0.90	9.68	10848.99
Q_{25}	6985.25	43731.75	135.00	28.00	1.00	52.00	45167.18
Q_{50}	7883.50	52941.50	150.00	29.00	1.00	62.00	52418.97
Q_{75}	9640.50	59861.00	200.00	30.00	2.00	67.25	61496.95
Range	(4010, 10178)	(30275, 60000)	(100, 200)	(26, 30)	(0, 4)	(38, 76)	(26072, 62275)
Skewness	0.297	0.123	0.454	0.322	1.403	−0.309	−0.624
Kurtosis	−0.520	−0.801	−0.740	2.569	1.180	−0.829	−0.553
Data type	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric

$*$ S.D. = standard deviation, F_1 = availability, F_2 = raw material, F_3 = worker, F_4 = working day, F_5 = holiday, F_6 = down time, outcome = demand level, Q_{25} = first quantile, Q_{50} = second quantile, Q_{75} = third quantile.

transformations methods are the Box–Cox and Yeo–Johnson transformations. Here, the Yeo–Johnson transformation is used because the Box–Cox transformation demands that input data are strictly positive, whereas both positive and negative data are endorsed by the Yeo–Johnson transformation [37]. The description of the Yeo–Johnson transformation can be given using

$$y^{*} = \begin{cases} \dfrac{\left((y+1)^{\lambda}-1\right)}{\lambda}, & \text{if } \lambda \neq 0, y \geq 0, \\[2ex] \log(y+1), & \text{if } \lambda = 0, y \geq 0, \\[2ex] -\dfrac{[(-y+1)\wedge\{2-\lambda\}-1]}{(2-\lambda)}, & \text{if } \lambda \neq 2, y < 0, \\[2ex] \log(-y+1), & \text{if } \lambda = 2,\ y < 0, \end{cases} \tag{1}$$

where y^{*} is the transformed value, y is a list of n strictly positive numbers, and λ is a hyperparameter used to control the transformation. Here, Scikit-learn implementation of PowerTransformer (method = “Yeo–Johnson,”, $*$, standardize = True) is used, performing the Yeo–Johnson power transformation operation with implicit data standardization with zero mean and unit variance to the transformed output.

3.3. Feature Selection.

3.3. Feature Selection. Feature selection or reduction reduces irrelevant, redundant, or partially important features that might mislead the model prediction, as the accuracy of an ML model depends on the features on which it has been trained. Feature reduction reduces the chances of overfitting because of the reduction of the redundant feature and lessens the model’s complexity. Several feature selection or reduction techniques exist. In our proposed method, PCA, ICA [36], and correlation-based feature selection algorithms were used to discard irrelevant features.

PCA is frequently employed in this capacity due to its adaptability and ease of implementation. PCA works on the premise of dividing data into an orthogonal space so that the eigenvectors corresponding to the greatest eigenvalues preserve the maximum data variance. PCA is a technique that focuses on the covariance matrix and second-order statistics. ICA decomposes observable data linearly into statistically independent components. For the correlation-based method, it classifies characteristics using a heuristic evaluation function that takes into account the correlation between the target outcome and their features. The design structure of both PCA and ICA follows the default implementation of Scikit-learn except the *n_components* parameter, resembling the number of features to be chosen by the respective algorithm, as the value of the parameter is driven from hyperparameter tuning. The design of PCA can be illustrated, respectively, such as (n_components, copy, whiten, svd_solver, tol, iterated_power) = ({4, 5, 6}, True, False, auto, 0.0, auto). Algorithms 1–3 summarize the procedures of PCA, ICA, and correlation-based feature selection algorithms, respectively.

3.4. Hyperparameters Determination. Hyperparameters define those values directly controlling the learning process of ML techniques and can be arbitrarily set by the user before starting the training phase. The correct combination of values is significant in achieving the best and quality model. Choosing the correct values for the optimal model is known as hyperparameter optimization or hyperparameter tuning [38]. Grid search and random search are both well-known techniques when tuning the hyperparameters of an estimator. This study used the grid search method based on cross-validation, resulting in the most precise predictions [39]. This algorithm splits the range of parameter values to be upgraded into the grid and across all points to obtain the optimal parameters. Different parameter combinations were evaluated for each model, which were divided into training and test sets using the cross-validation method [39]. Table 3 provides an overview of hyperparameters tuned using ML techniques and their range of tuning.

3.5. Cross-Validation in Time Series. Cross-validation is a widely used validation approach for tuning hyperparameters and assessing the effectiveness of machine learning techniques [40]. Different parameters must be stated for each case depending on the dataset. A grid search technique combined with cross-validation is effective at identifying the optimal hyperparameter combination for each model. As a consequence, forecasting errors associated with test samples may be decreased, allowing for the determination of the ideal

```
**Input:** $m-$ dimensional input data matrix $X \in \mathbb{R}^m$ with number of samples $N$, and variance threshold $T_{\text{var}}$
**Output:** reduced $L-$ dimensional data matrix $Y \in \mathbb{R}^L L < m$,
Load $X \in \mathbb{R}^m$, and calculate mean for each feature, $\mu_j = 1/N \sum_{i=1}^{N} X_{ij}$ for $j = 1, 2, \ldots, m$; subtract the mean from each corresponding dimension, $X_{ij} = X_{ij}' - \mu_j$ for $j = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, N$;
/ * Make each signal uncorrelated to each other * /
Calculate covariance matrix of $X', \sum_{m\times m} 1/N - 1\,[X']^T = X'$;
Solve the $\sum_{m\times m}$ as $\sum_{m\times m} = V^{-1}DV$, where $V \in \mathbb{R}^m$ is the matrix of eigenvector and $D_{m\times m}$ is the diagonal matrix containing eigenvalues on both sides of the diagonal matrix ;
Sort the eigenvector matrix $V$ in the descending order to the first $L-$ eigenvector that have variance $\geq T_{\text{val}}$ and form a projection matrix $P_{m\times L}$;
Finally, project on the PCA space, $Y = P^T X$;
```

Algorithm 1: Steps for the implementation of principal component analysis (PCA).

```
**Input:** $m-$ dimensional input data matrix $X \in \mathbb{R}^m$ with number of samples $N$, and variance threshold $T_{\text{var}}$
**Output:** reduced $L-$ dimensional data matrix $Y \in \mathbb{R}^L L < m$,
Select a nonquadratic nonlinear function $g$;
Initialize $W$ as $X = WH$, where $W\leftarrow$ ratio of source during mixing, $H\leftarrow$ matrix contains different components, and $X\leftarrow$ mixed output;
Perform PCA on $X$, as $X\leftarrow\text{PCA}(X)$ as in Algorithm 1;
**while** $W$ changes **do**
    Update $X\leftarrow E\{Xg(W^T X)\} - E\{g'(W^T X)\}$;
    Normalize $X\leftarrow W/\|W\|$;
Derive the new dataset by taking $Y = W^T X$, where $Y \in \mathbb{R}^L$;
```

Algorithm 2: Steps for the implementation of independent component analysis (ICA).

```
**Input:** $m-$ dimensional input data matrix $X \in \mathbb{R}^m$ with number of samples $N$, and expected outcome, $Y_O \in \mathbb{R}$
**Output:** reduced $L-$ dimensional data matrix $Y \in \mathbb{R}^L L < m$,
**for** $p\leftarrow 1$**to**$p \leq m$ **do**
$r_{pO} = \sum (X_p - \overline{X}_p)(Y_O - \overline{Y}_O)/\sqrt{\sum (X_p - \overline{X}_p)^2\sqrt{\sum (Y_O - \overline{Y}_O)^2}}$
Sort the correlation $r_{pO}$ in descending order to choose first $L$ features for $Y \in \mathbb{R}^L$;
```

Algorithm 3: Steps for the implementation of correlation-based feature selection (Corr).

collection of hyperparameters that enhance predictive performance while minimizing model overfitting [41]. The leave-one-out cross-validation procedure is acceptable in this scenario when dealing with time series data [42]. Alternatively, this method can be considered a sequential block cross-validation procedure and a subset of K-fold cross-validation.

Thus, the training set is iteratively constructed, with the training and validation sets being utilized concurrently, a process known as rolling cross-validation. This procedure is performed several times, with each iteration increasing the amount of observations in the training set and decreasing them in the validation set. The associated training set comprises only observations that happened before the observation in the test set. The dataset is partitioned into training and test sets, with 70% of the data used for training and verifying the models. The time series split notion is to divide the training set in half at each iteration, assuming that the validation set is still ahead of the training split. It is initially trained on a limited subset of data in order to forecast the next data point. Following that, the forecasted data points are incorporated into the succeeding training dataset, and subsequent data points are forecasted. This process is repeated until the complete training set has been utilized. Calculate the training outcome by estimating iteration performance assessments.

3.6. Structure of Stacked Ensemble Modeling. STACK modeling was conducted by considering two stages, level 0 and level 1, and the predictions of the base learner (level 0) are combined with the meta-learner (level 1). From the previous studies, it is shown that the support vector regression (SVR) and selection operator (LASSO) regression are used as the meta-learner [8, 25]. The key advantages of adopting SVR, and especially layer-1 in the STACK technique, are its ability to identify predictor nonlinearities and subsequently exploit

Table 3: Different machine learning techniques with hyperparameters to be tuned by the grid search algorithm during cross-validation.

Algorithms	Hyperparameters	Explanation	Grid
SVR	C	Regularization parameter of the error term	$1 - 1 \times 10^{-3}$
	Kernel	Kernel types applied in the algorithm	Linear, polynomial, RBF
	Epsilon	Border of tolerance	0.1 – 1
	Gamma	Kernel coefficient for rbf	$1 \times 10^{-3} - 0.1$
RFR	n_estimators	Number of trees in a forest	5 – 15
	Criterion	Measurement of the quality of a split	mae or mse
	max_depth	Highest depth of the tree	2 – 10
	min_samples_leaf	Least number of instances needed to split an internal node	2 – 10
	min_samples_split	Least number of instances needed to be at a leaf node	2 – 10
MLP	initial_learning_rate	Learning rate value at the starting point of training	$1 \times 10^{-3} - 0.1$
	Solver	Used for weight optimization	lbfgs, sgd, Adam
	learning_rate_adjustment	Learning the rate adjustment depending on the cost function's current value	Constant, adaptive
	hidden_layer_sizes	Layer: Number of layers between input and output layers	1 ~ 7
		Neurons: Number of hidden layer neurons	(4, 8, 12)
	activation_functions	Output of each neuron	Logistic, tanh, relu
	Alpha (L_2 penalty)	Reduces the influence of input parameters	$1 \times 10^{-3} - 0.1$
ELM	n_neurons	Number of hidden layer neurons	8
	activation_functions	Transformation function of hidden layer neurons	relu
	Alpha	Regularization strength	0.001
GBR	n_estimators	Number of boosting stages to carry out	200 – 500
	max_features	Number of features while considering the best split	Sqrt
	min_samples_leaf	Least number of instances needed to be at a leaf node	2 – 10
	max_depth	Utmost depth of individual regression estimators	2 – 10
	learning_rate	Shrinks the contribution of each tree	0.1 – 1
	Loss	Loss function based on order information of input variables	ls, lad, huber, quantile
XGBR	cosample_bytree	Subsample ratio of columns while building each tree	0.1 – 1
	Subsample	Subsample ratio of training samples	0.1 – 1
	reg_lamda	L_2 regularization On weight	0.1 – 1
	reg_alpha	L_1 regularization On weigh	0.1 – 1
	min_child_weight	Least sum of sample weight required in a child	1 – 10
	learning_rate	Step size reduction to prevent overfitting	$1 \times 10^{-3} - 0.1$
LASSO	Alpha (L_1 penalty)	A constant value that multiplies L_1	$1 \times 10^{-4} - 1 \times 10^{2}$

them to improve demand forecasts [8]. The SVR with linear kernel and selection operator (LASSO) regression model was utilized as a meta-learner in our experiment (level 1).

The following steps were adopted in this work.

(1) After doing the training session of the SVR/LASSO, RFR, MLP, ELM, GBR, and XBR models, the predicted results are combined (2 in 2, 3 in 3, 4 in 4, and 5 in 5) to build a STACK (SVR/LASSO) layer 0. Stack layer 0 does not use the model used in layer 1.

(2) For each STACK model, 56 models are analyzed, and best one is chosen for the study based on the test set results.

(3) The findings in Tables S1 and S2 indicate that models numbered 1–15 indicate a model combination of 2 in 2, models numbered 16–35 indicate a model combination of 3 in 3, models numbered 36–50 indicate a model combination of 4 in 4, and models numbered 51–56 indicate a model combination of 5 in 5 in the order specified in step 1.

(4) The performance evaluation measurements are achieved for the training and test sets after training each STACK model.

The working procedure of the stacking ensemble in this paper is described in Algorithm 4.

3.7. Performance Measures. Estimating the model's accuracy is crucial in designing ML models to define how well the model is predicting. It is used to determine the goodness of fit among models and data to compare various models for model selection. If $y_1, y_2, \ldots, y_t$ are T actual values and $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_t$ are corresponding predicted values, then the formulas are for evaluating the accuracy of the regression models as follows:

```
Input: Input dataset $D = \{X_i, y_i\}_{i=1}^{m}$ , where $(X \in \mathbb{R}, y \in Y), \Theta_{\mathrm{set}}$ is the set of optimal hyperparameter for each based regression model, $M$ is number of based model, $T$.
Output: final forecast demand level $\hat{Y}_f$ and performance indices.
Step 1: learn first-level base regression models;
/ * Loop for train and evaluate the first-level individual /regressor *
for $t \leftarrow 1$ to $T$ do
  Divide the dataset $D$ into $D^{\mathrm{train}}$ and $D^{\mathrm{test}}$;
  / * 70% data for training and validation, 30% for test set * /
  / * Leave-One-Out Cross-Validation * /
  for $i \leftarrow 1$ to $K$ ($K \leftarrow$ size of $D^{\mathrm{train}}$) do
    $D_i^{\mathrm{val}} = D^{\mathrm{train}}(i, :) \Rightarrow D_i^{\mathrm{train}} = D^{\mathrm{train}} / D_i^{\mathrm{val}}$;
    Train $M_t$ with optimal hyperparameter set $\Theta_{\mathrm{set}}$ on $D_i^{\mathrm{train}}$;
    Predict the demand level for $M_t$ with $D_i^{\mathrm{val}}$: $h_t \leftarrow M_t(D_i^{\mathrm{val}})$;
Step 2: create a new dataset from $D$;
for $t \leftarrow 1$ to $T$ do
  Create a new dataset $D'_{m \times l} = \{X'_i, y_i\}$ for meta-regressor,
  Where $X'_i = \{h_1, h_2, \ldots, h_t\}, h_t \leftarrow$ output of $i^{\mathrm{th}}$ model, $l \leftarrow$ number of based model;
Step 3: learn second-level regressor model;
/ * Loop for train and evaluate the final-level meta-regressor model
* /for $j \leftarrow 1$ to $K$ ($K \leftarrow$ size of $D^{\mathrm{train}}$) do
  $D_j'^{\mathrm{val}} = D'^{\mathrm{train}}(j, :) \Rightarrow D_j'^{\mathrm{train}} = D'^{\mathrm{train}} / D_j'^{\mathrm{val}}$;
  Train the meta-model $H_{\mathrm{meta}}$ with $D'^{\mathrm{train}}_j$ using $\Theta_{\mathrm{set}}$;
  Predict the demand level for $H_{\mathrm{meta}}$ with $D'^{\mathrm{val}}_j$;
Test set $D'^{\mathrm{test}}$ are used for the prediction and performance measure $(P_{H\mathrm{meta}})$ using $H_{\mathrm{meta}}$
return $P_{H\mathrm{meta}}$;
```

Algorithm 4: Demand forecasting using Stacking Ensemble techniques using cross-validation.

$$
\begin{aligned}
R^2 &= 1 - \frac{\sum_{t=1}^{t}\left(y_t - \hat{y}_t\right)}{\sum_{t=1}^{t}\left(y_t - \hat{y}_t\right)},\\
\mathrm{MAE} &= \frac{1}{T}\sum_{t=1}^{T}\left|y_t - \hat{y}_t\right|,\\
\mathrm{RMSE} &= \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)},\\
\mathrm{MAPE} &= \frac{1}{T}\sum_{t=1}^{T}\left|\frac{y_t - \hat{y}_t}{y_t}\right|
\end{aligned}
\tag{2}
$$

where $\overline{y} = 1/T \sum_{t=1}^{T} y_t$ and in this paper, training set $t = 1, \ldots, 90$ and test set $t = 91, \ldots, 132$ are adopted.

Along with the performance evaluation matrix mentioned above, several statistical tests [43, 44] are performed in this study to ensure the superiority of the proposed approach. The Friedman test is used to examine if the absolute percentage errors (APE) of the two models differ statistically significantly. Once statistical significance has been established, post hoc tests (nonparametric tests), such as the Wilcoxon signed-rank test, can be employed to assess if the APEs of the models change when compared to one another (lower tail) [44, 45]. Wilcoxon's null hypothesis indicates that there is no difference in APE between models 1 and 2, but the alternative hypothesis states that model 1 has a lower APE than model 2.

4. Experimental Results and Discussion

In this section, the preparatory analysis of steel industrial data used in this study is demonstrated in Section 4.2. The performance of the adopted models and statistical tests for test set errors are described in Section 4.3. Tables S1 and S2 represent the performance measurement indices of the 56 generated models.

4.1. Experimental Setup. A single computer (Asus X556U with an Intel® Core (TM) i5–72000U, central processor unit running at 2.50 GHz, 8.0 GB of random access memory, and an Nvidia GeForce 940MX graphics card) running the Windows 10 operating system was used to create the findings provided in Section 4. In order to implement the machine learning approaches and ensemble methods, we used the Python 3.6 programming language in conjunction with the Spyder computing environment, which is included in Anaconda.

4.2. Exploratory Analysis. Correlation analysis is a statistical approach used to determine the connection between two numerical variables. From an ML viewpoint, it indicates how the features correspond to the outcome. However, it is challenging to identify how features are interconnected. Data visualization can help determine how individual features might correlate with the outcome. Pearson's correlation coefficient is used to identify the relationship between two variables in a statistical analysis. In the range of +1 to −1, it means that there is no correlation at all, +1 indicates that there is a perfect positive correlation, and −1 indicates that there is a perfect negative correlation, according to the definition. After the Yeo–Johnson transformation has been performed to the training data set, the correlation matrix for the exploratory variables is shown in Figure 2. Figure 2 depicts the color scale of its association, which is represented on the right-hand side of the illustration. The light color indicates a close relation of 0, whereas the intense color indicates a close relation of +1 or −1. The indicators (F_1, F_2, and F_3) and the response variable (Demand) are highly positively correlated. Thus, the increment or decrement in the value of one tends to increment or decrement those that are highly correlated. However, indicator (F_5) is negatively correlated to the outcome (Demand), indicating that if the number of holidays in a month increases, the number of demands decreases and vice versa.

4.3. Evaluation of Proposed Models. In this study, the proposed models are trained using a set of optimal hyperparameters achieving the maximum predictive performance of each model achieved by grid search. The steel production data from January 2009 to December 2019, covering 132 months, are taken as the training and testing sets.

Table 3 presents an overview of hyperparameters tuned for each ML model, their explanation, and turning ranges. Table 4 represents the quantified results for selecting the best performing preprocessing and the number of selected features and ML models, where R^2 with standard deviation is stated for comparison. Table 5 summarizes each model's capacity to obtain the highest R^2 using the suggested pipeline, along with the optimal preprocessing and feature selection algorithms and the number of selected features. In addition, Table 5 illustrates the best-tuned hyperparameters using the grid search. The analysis of Table 4 reveals that when suitable preprocessing is used, various models produce superior outcomes. The different architectures of the MLP model are shown in Table 6. Table 7 summarizes the performance metrics used to evaluate each model, which include R^2, MAE, RMSE, and MAPE. When either correlation-based or PCA-based feature selection is applied, each model achieves the best results for filling missing values, Yeo–Johnson transformation, and data normalization (Tables 4 and 5). For SVR, the estimated accuracy of $R^2 = 0.931$ is obtained from preprocessed data and correlation-based feature selection.

The comprehensive experiments were performed on the same dataset to get the best architecture for the MLP model. Eight separate MLP models (Table 6) were implemented and evaluated, with 1–7 hidden layers, where the number of neurons served as a hyperparameter for selecting the best numbers. The experimental results in Figure 3 indicate that the optimal architecture is the MLP layout with $M = 4$ hidden layers (H_1, H_2, H_3, and H_4) and $N_1 = 12$, $N_2 = 12$, $N_3 = 12$, and $N_4 = 8$ neurons. In addition, the presence of additional hidden layers with fewer samples, like in the steel dataset, limits the MLP model's capability (Figure 3). Because of the limited data, such as in the steel dataset, the wide depth of the MLP model could be overfitted and cause gradient fading problems. Table 3 lists the optimal hyperparameters of the best MLP model. The models have used the ReLU activation function and Adam solver. It was trained on 200 epochs with a constant learning rate, batch size, and a regularization parameter of 0.01, 32, and 0.1, respectively. To reduce overfitting, the dropout layer was used, randomly dropping 60% of neurons. The highest accuracy R^2 from the MLP model is 0.961 when we perform data preprocessing and PCA-based feature selection. Similarly, the ELM model with eight neurons in the hidden layer obtained the best result. Table 3 lists the optimal hyperparameters of the best ELM model. The model used the ReLU as the transformation function of hidden layer neurons, and the optimal regularization parameter was 0.001. The best-estimated accuracy (R^2) of the ELM model with preprocessed data and correlation-based feature selection is 0.942.

Feature selection methods are used to improve the overall performance of each model (correlation-based, PCA, and ICA). It is possible to reduce the dimensions of a higher-dimensional space to a lower-dimensional space using PCA by selecting the orthogonal projections with the highest variance. The ICA theory implies that data are only partly independent if their variances across characteristics are larger than their co-variance. The number of computers being used has a significant impact on PCA performance. Because the ICA-based feature selection technique is used to find newly specified mutually independent components, it is possible that correlation with the desired output will be lost when the procedure is used to discover new predefined mutually independent components. Due to the fact that both PCA and ICA create new components in an unsupervised manner, it is not possible to guarantee greater performance on the steel dataset. Correlation-based feature selection, on the other hand, takes into consideration the relationship between quality and outcomes in order to discover the most closely related features. As shown in Table 4, the majority of models perform better when four features, F1, F2, F3, and F6, are used. These four features were chosen using a correlation-based feature selection technique.

Further improvement of demand forecasting was obtained using regression ensemble models. Bagging (RFR), Boosting (GBR and XGBR), and stacking (STACK) regression ensembles were adopted to improve the performance of demand forecasting. Table 5 presents the performance evaluation of the adopted models. Furthermore, the results are sorted regarding R^2 in the ascending order for the test set results. Finally, the best models present the lower RMSE and higher R^2 in the test set. RFR is the ensemble learner built-in unpruned decision tree, and it reduced the effects of overfitting by combining multiple trees. Table 5 shows the optimal hyperparameters for the RFR model. The best-estimated accuracy (R^2) of the RFR model is 0.966 obtained from preprocessed data and PCA-based feature selection. The RFR performance of the models is better for SVR, MLP, and ELM in terms of the RMSE, that is, it has lower RMSE values. GBR and XGBR are also used to increase the accuracy of forecasts. Extreme gradient boosting is a specific variant of the gradient boosting strategy that discovers the ideal tree model by employing a more exact approximation than the conventional gradient boosting method. The best-estimated accuracy (R^2) of the GBR model is 0.969, obtained from preprocessed data and correlation-based feature selection. The XGBR can reduce the loss by showing an extreme gradient capability. The highest accuracy (R^2) of XGBR is 0.974, and the lowest RMSE is 0.151. The RMSE of XGBR is significantly lower

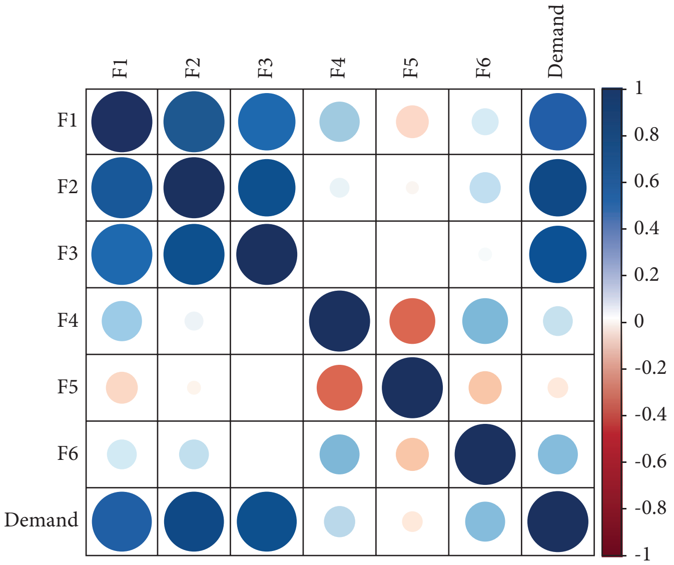


Figure 2: Correlation matrix for all influence features corresponding to the demand.

Table 4: Summary of all extensive experiments to select the best performing preprocessing feature selection methods with the number of features and regression models.

ML method (s)	Preprocessing	Algorithm	n_features	Performance (R^2)
SVR	Raw data	N/A	6	0.898
	Processed data	N/A	4	0.919
		Corr	**4**	**0.931**
		PCA	4	0.923
		ICA	4	0.923
RFR	Raw data	N/A	6	0.918
	Processed data	N/A	4	0.929
		Corr	4	0.966
		PCA	**4**	**0.967**
		ICA	4	0.952
MLP	Raw data	N/A	6	0.845
	Processed data	N/A	4	0.911
		Corr	4	0.957
		PCA	**4**	**0.961**
		ICA	4	0.950
ELM	Raw data	N/A	6	0.841
	Processed data	N/A	4	0.849
		Corr	**4**	**0.942**
		PCA	4	0.887
		ICA	4	0.909
GBR	Raw data	N/A	6	0.934
	Processed data	N/A	4	0.944
		Corr	**4**	**0.969**
		PCA	4	0.949
		ICA	4	0.934
XGBR	Raw data	N/A	6	0.949
	Processed data	N/A	4	0.953
		Corr	**4**	**0.974**
		PCA	4	0.956
		ICA	4	0.952

∗ N/A = none. Note: the best approaches were shown in bold type.

than the reference models and RFR and GBR. The best result of XGBR is obtained when a child's minimum amount of weight is less than 4, and a subsample ratio to construct a tree is 0.7.

Finally, the stacking ensemble method is used for integrating multiple-base models in order to reduce prediction errors to the smallest possible amount. According to the results from the test set, level 0 of the STACK$_1$ method is

Table 5: Best–-performing ML model and preprocessing and tuned hyperparameters with the highest possible accuracy (R^2).

ML techniques	Best preprocessing	Best hyperparameters	Performance
SVR	Processed data Corr (n_attributes: 4)	C: 100 Kernel: RBF Epsilon: 0.1 Gamma: 0.001	R^2: 0.931
RFR	Processed data PCA (n_attributes: 4)	n_estimators: 10 Criterion: mse max_depth: 8 min_samples_leaf: 2 min_samples_split: 2	R^2: 0.966
MLP	Processed data PCA (n_attributes: 4)	initial_learning_rate: 0.01 Solver: Adam learning_rate_adjustment: Constant hidden_layer_sizes: (12, 12, 12, 8) activation_functions: relu Alpha (L_2 penalty): 0.01	R^2: 0.961
ELM	Processed data Corr (n_attributes: 4)	n_neurons: (8) activation_functions: relu Alpha: 100	R^2: 0.942
GBR	Processed data Corr (n_attributes: 4)	n_estimators: 250 max_features: Sqrt min_samples_leaf: 2 max_depth: 2 learning_rate: 0.2 Loss: lad	R^2: 0.969
XGBR	Processed data Corr (n_attributes: 4)	cosample_bytree Subsample reg_lamda reg_alpha min_child_weight learning_rate	R^2: 0.974
STACK (SVR)$_1$	Processed data Corr (n_attributes: 4)	C: 100 kernel: RBF	R^2: 0.977
STACK (LASSO)$_2$	Processed data Corr (n_attributes: 4)	Alpha ($L1$ penalty): 0.001	R^2: 0.977

Table 6: The different architectures of MLP with the corresponding number of hidden layers and the number of neurons in each layer.

Numerous architectures	The number of hidden layers and the number of neurons in each layer
Architecture$_1$	$H_1 \in R^8$
Architecture$_2$	$H_1 \in R^8, H_2 \in R^{12}$
Architecture$_3$	$H_1 \in R^4, H_2 \in R^8, H_3 \in R^{12}$
Architecture$_4$	$H_1 \in R^{12}, H_2 \in R^{12}, H_3 \in R^{12}, H_4 \in R^8$
Architecture$_5$	$H_1 \in R^{12}, H_2 \in R^{12}, H_3 \in R^8, H_4 \in R^{12} H_5 \in R^8$
Architecture$_6$	$H_1 \in R^{12}, H_2 \in R^{12}, H_3 \in R^8, H_4 \in R^{12} H_5 \in R^4, H_6 \in R^{12}$
Architecture$_7$	$H_1 \in R^8, H_2 \in R^8, H_3 \in R^{12}, H_4 \in R^{12} H_5 \in R^{12}, H_6 \in R^{12}, H_7 \in R^{12}$

formed of the models ELM, GBR, and XGBR, with SVR as the first model in level 1. For STACK$_2$, the levels 0 and 1 are made of ELM, GBR, and XGBR, with LASSO as the level 1 component. All of the models in Table A1 have the same performance (R^2) as the models numbered 14, 24, 33, 35, 45, 50, and 55. Model 35, on the other hand, is selected for the STACK$_1$ technique because its complexity is smaller than that of other configurations, and it has the lowest MAPE. In a similar process, the models numbered 33, 35, 50, and 56 in Table A2 exhibit the same level of performance (R^2). For the STACK$_2$ technique, model 35 is also picked because its complexity is lower than that of other configurations, and it has the lowest MAPE of any of the models tested. The best-estimated accuracy of STACK$_1$ is 0.977, whereas the best-estimated accuracy of MAPE is 0.445. In a similar vein, the best-estimated accuracy of STACK$_2$ is 0.977, and the best-estimated accuracy of MAPE is 0.463. According to Table 7, based on the findings of the test phase, the approaches based on ensemble learning produced results that were compatible with the objective of minimizing error.

Figure 4 illustrates the violin graph for the APE distribution of each model that was utilized to produce predictions for the test set, as shown by the APE distribution of each model. The mean APE is shown by the white dot in the center of the chart. Ensemble-based techniques, as compared to other models, significantly lower the APE to the absolute bare

Table 7: Comparing stacking ensemble model with the best performing ML models.

Models	R^2	MAE	RMSE	MAPE
SVR	0.931	0.202	0.246	0.902
ELM	0.942	0.183	0.226	0.880
MLP	0.961	0.149	0.186	0.569
RFR	0.966	0.133	0.172	0.517
GBR	0.969	0.125	0.164	0.401
XGBR	0.974	0.120	0.151	0.619
STACK$_2$	**0.977**	**0.112**	**0.143**	**0.463**
STACK$_1$	**0.977**	**0.112**	**0.144**	**0.445**

The best approaches were shown in bold type.

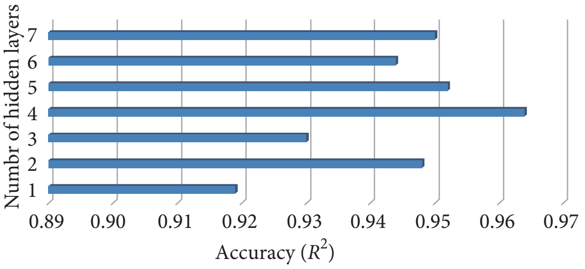


Figure 3: Performance of several MLP architectures with the purpose of picking the optimal one with the maximum accuracy ((R)^2), where the best corresponding models are presented in Table 6.

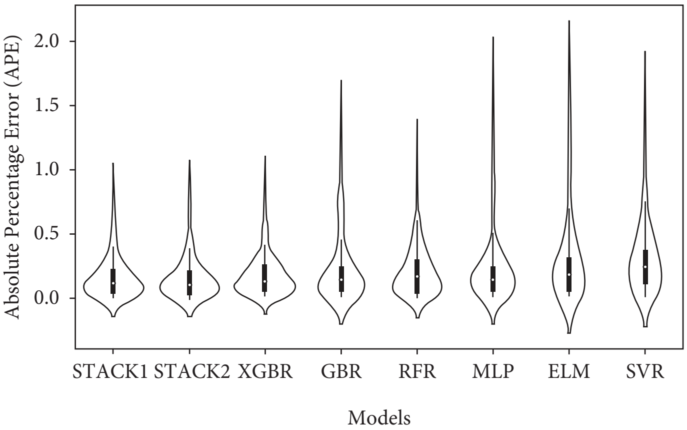


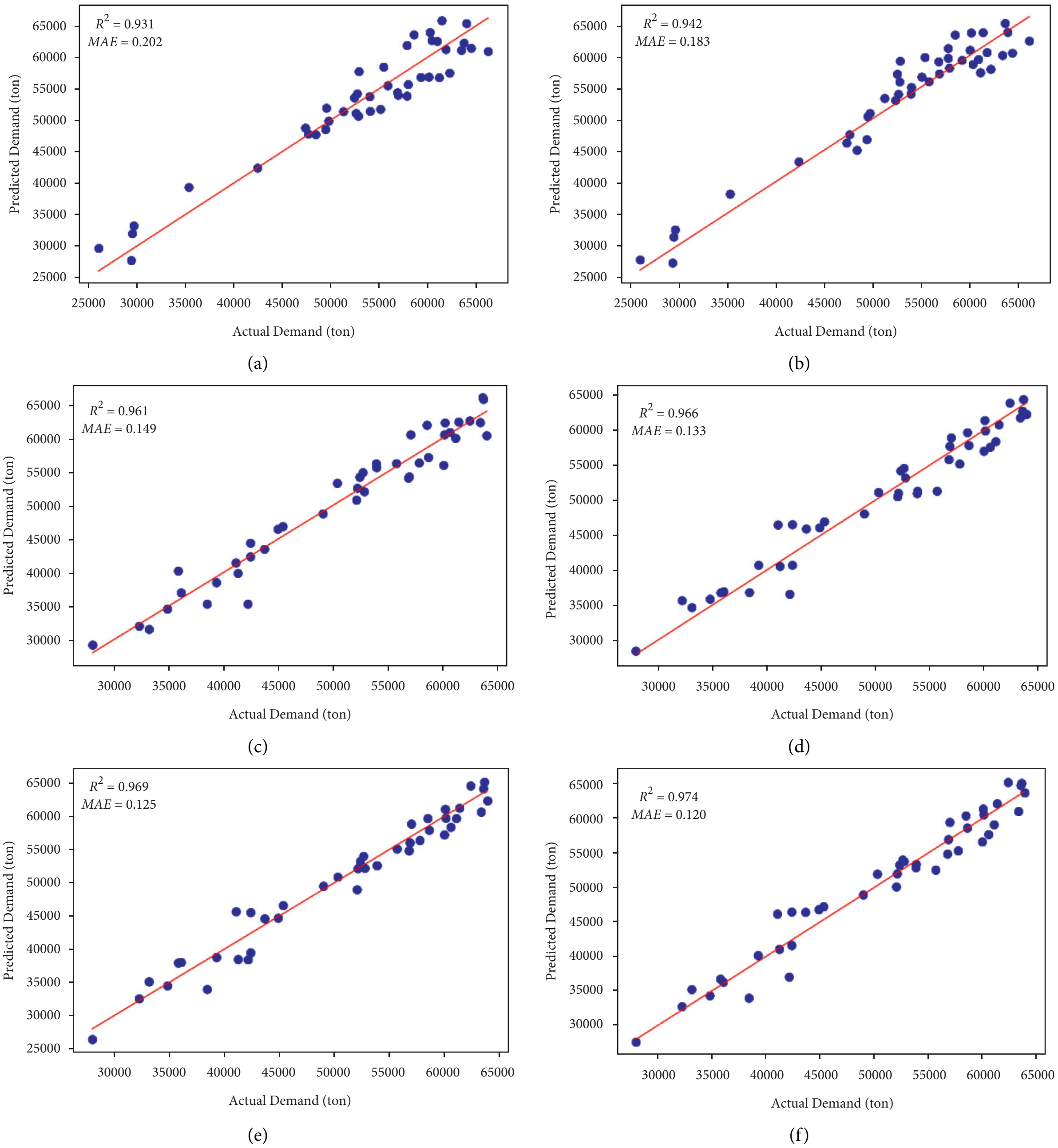
Figure 4: Violin plot to represent the APE of the models.

minimum. In this way, we can show that a model (for the test set) with lower metric values in Table 7 has a more stable APE and less volatility than a model with higher metric values. The Friedman test established that the APEs for the accepted models varied in the test set ($\chi_7^2 = 72.1875$, $p-$value< 0.05). This implies that there exist models with observed APE values that are equal to or less than those of the others. In addition, Table 8 depicts the results of the Wilcoxon signed rank test (lower tail) for measuring the APE reduction of the assessed models in the test set, in the presence of a statistically significant difference as revealed by the Friedman test ($\chi_7^2 = 72.1875$, $p-$value< 0.05).

At the 5% level of significance, the APE of the STACK$_1$ model is fewer than the APEs of the RFR, MLP, ELM, and SVR models, as shown in Table 8. It is statistically equivalent when the STACK$_1$ model is compared to other models with error rates at the 5% threshold of statistical significance. In addition, when the 5% threshold of significance is utilized to compare the models, Table 8 reveals that the APE of the STACK$_2$ model is lower than the APEs of the RFR, MLP, ELM, and SVR models. Using the % level of statistical significance, the STACK$_2$ model is compared to other models, and the errors are statistically equivalent. This highlights the advantages of the stacking ensemble models that we provide. Ensemble-based models, on average, have a lower APE than ELM and SVR. As a result, the ability of this approach to learn the data could be described using smaller estimation errors and variance between the ensemble methods than with the others, confirming the validity of this methodology. At the 5% level of significance, the APE of the STACK$_1$ model is fewer than the APEs of the RFR, MLP, ELM, and SVR models, as shown in Table 8. When the STACK1 model is compared to other models, the errors are statistically equal at the 5% level. Similarly, Table 8 reveals that the APE of the STACK$_2$ model

Table 8: Wilcoxon signed rank test statistic (W) (lower tail) and p value for APE comparisons.

Model_1 vs Model_2	W	p – value	Model_1 vs. Model_2	W	p value	Model_1 vs. Model_2	W	p value
STACK_1 vs STACK_2	3794	>0.05	STACK_2 vs RFR	2947	<0.05	GBR vs RFR	3740	>0.05
STACK_1 vs XGBR	4085	>0.05	STACK_2 vs MLP	2882	<0.05	GBR vs MLP	3374	>0.05
STACK_1 vs GBR	3383	>0.05	STACK_2 vs ELM	2270	<0.05	GBR vs ELM	2897	<0.05
STACK_1 vs RFR	3057	<0.05	STACK_2 vs SVR	1901	<0.05	GBR vs SVR	2392	<0.05
STACK_1 vs MLP	2856	<0.05	XGBR vs GBR	3814	>0.05	RFR vs MLP	3857	>0.05
STACK_1 vs ELM	2213	<0.05	XGBR vs RFR	3462	>0.05	RFR vs ELM	3048	<0.05
STACK_1 vs SVR	1843	<0.05	XGBR vs MLP	3144	<0.05	RFR vs SVR	2372	<0.05
STACK_2 vs XGBR	4023	>0.05	XGBR vs ELM	2720	<0.05	MLP vs ELM	3222	<0.05
STACK_2 vs GBR	3333	>0.05	XGBR vs SVR	1957	<0.05	MLP vs SVR	2593	<0.05
						ELM vs SVR	3657	>0.05



(a)

(b)

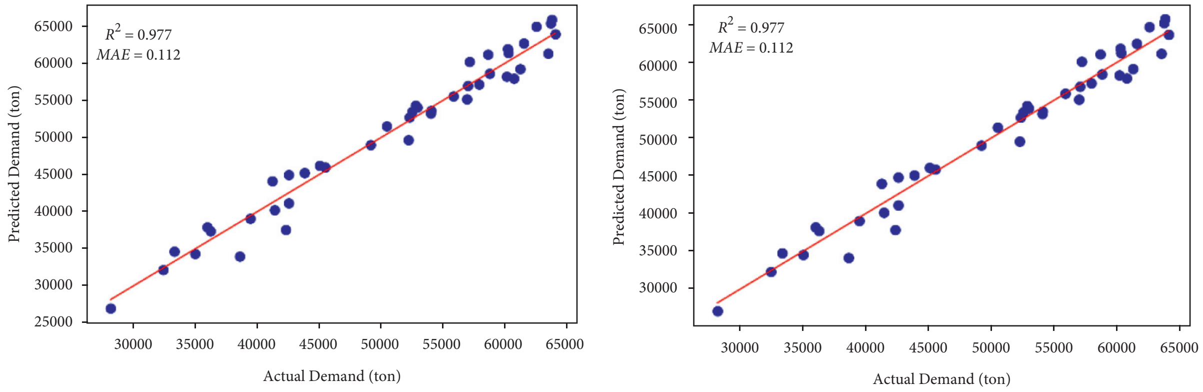
(c)

(d)

(e)

(f)

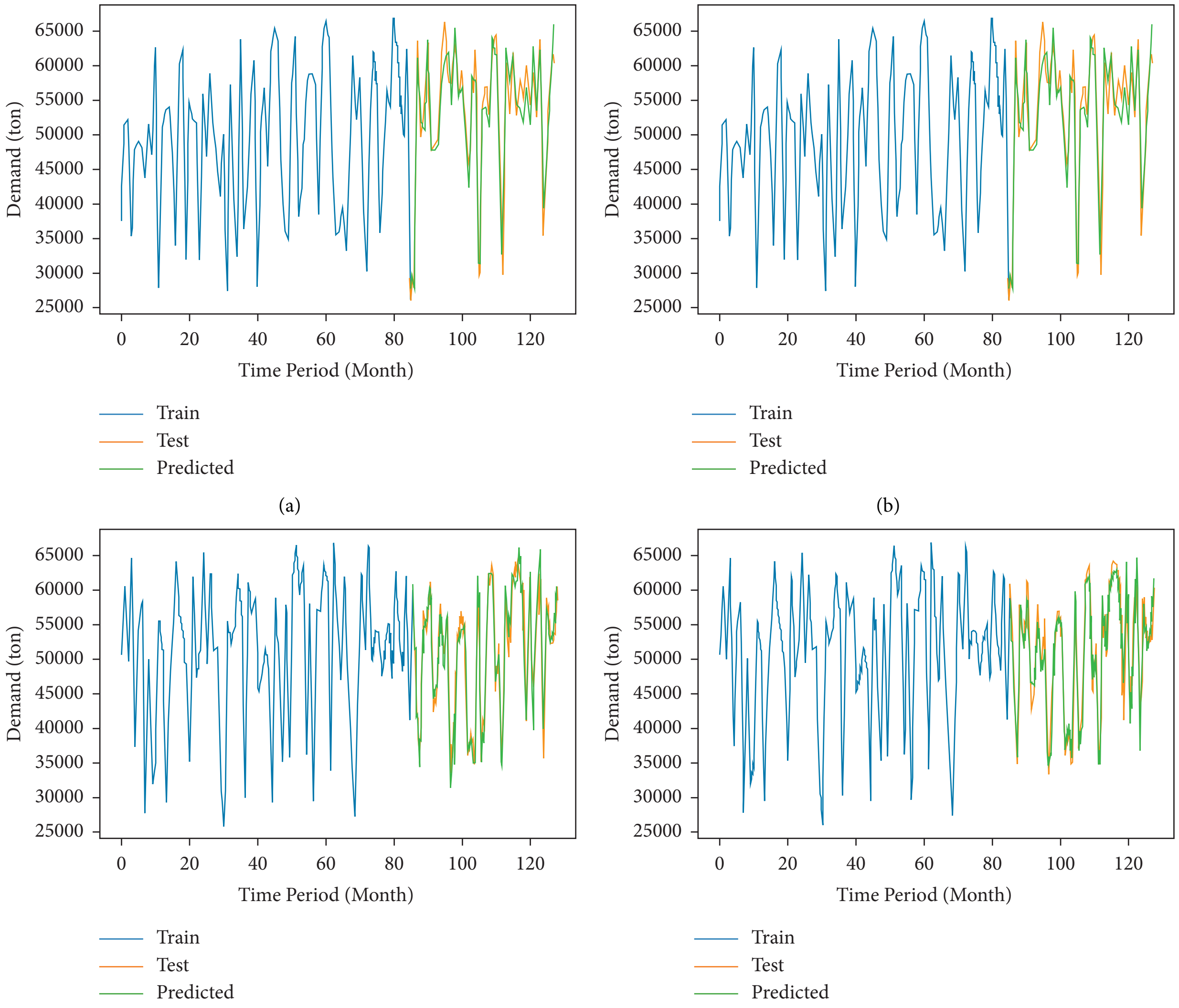
Figure 5: Continued.



(g)

(h)

FIGURE 5: Correlation-based comparison between predicted demand and actual demand during the testing phase (a) SVR, (b) ELM, (c) MLP, (d) RFR, (e) GBR, (f) XGBR, (g) STACK_2, and (h) STACK_1.



(a)

(b)

(c)

(d)

FIGURE 6: Continued.

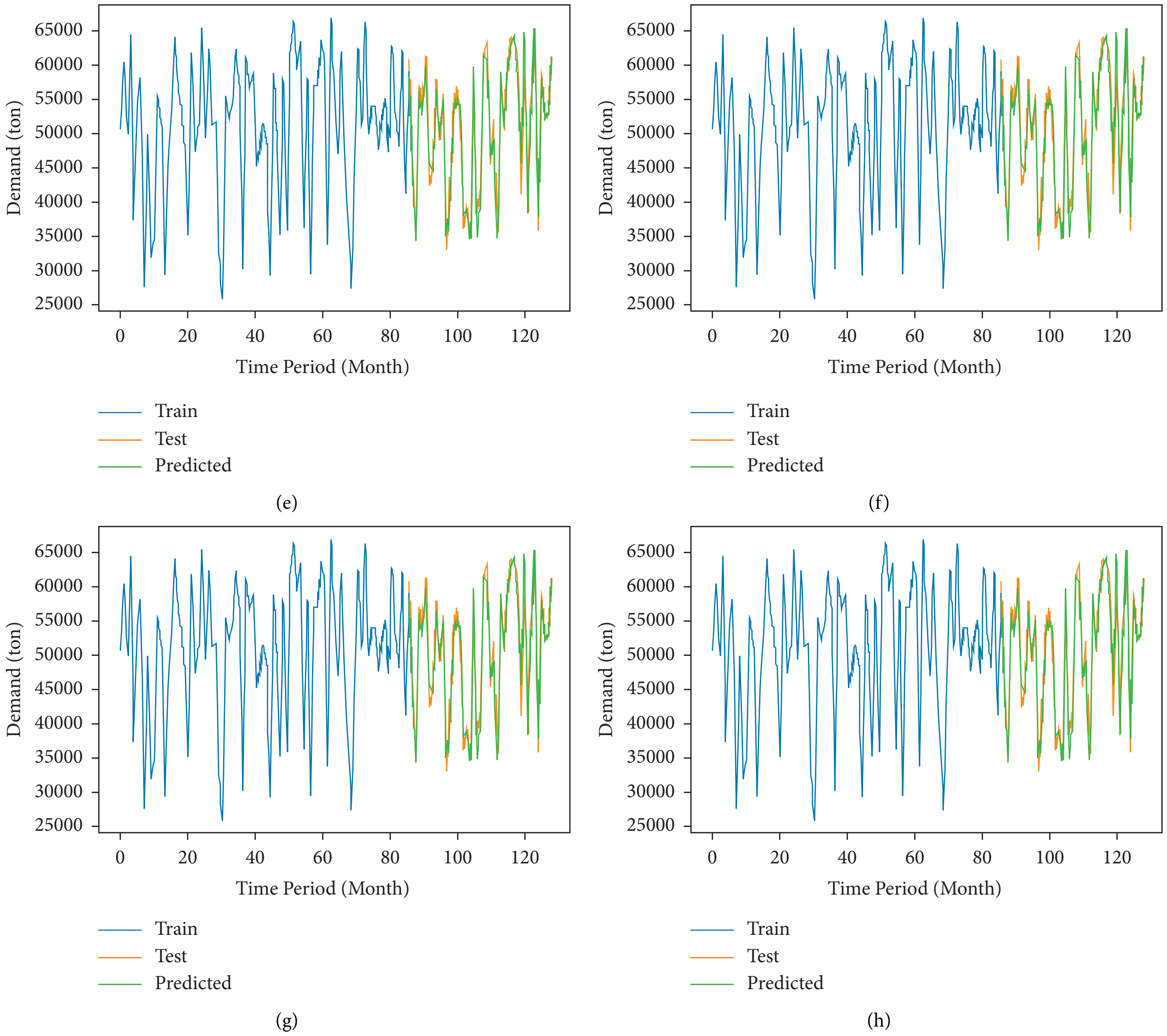


Figure 6: Graphical representation of actual vs predicted demand obtained by (a) SVR, (b) ELM, (c) MLP, (d) RFR, (e) GBR, (f) XGBR, (g) STACK_2, and (h) STACK_1 models, respectively.

is less than that of the RFR, MLP, ELM, and SVR models at the 5% level of significance. When the STACK_2 model is compared to other models, the errors are statistically equal at the 5% level. This demonstrates the advantages of the stacking ensemble models we proposed. Ensemble-based models, on average, have a lower APE than ELM and SVR. Thus, the capacity of this method to learn the data could be expressed using lower estimation errors and variance between the ensemble methods than with the others, demonstrating the correctness of this approach.

Furthermore, a relationship between the actual and predicted demand was established. Figures 5 and 6 show these techniques for better understanding. Figure 5 shows the correlation-based comparison between actual and predicted demand levels for both reference models and regression ensemble models. Figure 6 provides a pictorial view of actual vs predicted demand. Figure 6 shows that the demand pattern arbitrarily fluctuates because of the impact of the variables affecting it.

As shown in Figures 5 and 6, models that are capable of providing predictions that are consistent with the observed values are able to learn from data behavior. The improved performance attained during the training phases is maintained during the test phases, suggesting that the regression ensemble methodology is reliable in terms of achieving established predictions. This is supported by the capability of machine learning models to manage nonlinearities and model the complicated interaction between response variables and input features.

5. Conclusions

Precise demand forecasting significantly influences improving the performance and durability of the steel industry. This study compares the predictive performance of STACK, GBR, XGBR, and RFR regression ensembles and the MLP, ELM, and SVR reference models. In order to improve the prediction performance of regression ensemble models, data preparation and feature selection procedures are critical. The proposed preprocessing scheme improves the raw dataset quality, where filling the missing values and data standardization are the main concerns. The Yeo–Johnson transformation is used to influence the features and response variables. While PCA and ICA solely focus on interfeature redundancy, correlation-based feature selection might improve interfeature correlation. Hyperparameters are tuned to find the optimal hyperparameter set for each ML technique using a grid search algorithm. The best-performing models are combined in STACK_1 to form level 0. SVR with linear kernels and LASSO regressions are adopted as meta-learners in level 1. The Friedman and Wilcoxon signed-rank tests (lower tail) are used to validate the models' APE differences. Regarding the findings, two models may be used to forecast one month as follows: STACK_1 (ELM + GBR + XGBR-SVR) and STACK_2 (ELM + GBR + XGBR-LASSO). The test set results demonstrate that ensemble approaches outperform single models, notably the STACK model, in forecasting demand in the steel industry.

Future research will (i) develop other ensemble techniques and integrate other ML regression techniques into the ensemble; (ii) include other influence variables such as occasion and political factors; (iii) collect more information, in this case only 132 months of production data are used; and (iv) extend to other industrial fields to evaluate their generality and flexibility to predict several types of demand. [36, 46, 47].

Data Availability

The data used in this study are available from the corresponding author upon request.

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. The authors also thank the Deanship of Scientific Research and RSSU at King Saud University for their technical support. The authors are also grateful for the support from Taif University, Taif, Saudi Arabia. The APC is funded by Taif University Researchers Supporting Project Number (TURSP-2020/331), Taif University, Taif, Saudi Arabia.

Supplementary Materials

Figure S1: general architecture of stacking ensemble; Figure S2: architecture of MLP, with M hidden layer $(\mathbf{H})$ and $\mathbf{N_M}$ neurons in $\mathbf{H_M}$ layer, for forecasting the demand in the proposed framework; Figure S3: Single hidden layer architecture of ELM is used in this study; Table S1: performance metrics for evaluating STACK models that are used to anticipate steel demand one month ahead when the metalearner is SVR with a linear kernel; Table S2: performance metrics for evaluating STACK models that are used to anticipate steel demand one month ahead when the metalearner is LASSO. (*Supplementary Materials*)

References

[1] W. J. Stevenson, M. Hojati, and J. Cao, *Operations Management*, McGraw-Hill Irwin, New York, NY, USA, 2007.

[2] V. Bianco, O. Manca, S. Nardini, and A. A. Minea, "Analysis and forecasting of nonresidential electricity consumption in Romania," *Applied Energy*, vol. 87, no. 11, pp. 3584–3590, 2010.

[3] A. Azadeh, S. F. Ghaderi, and S. Sohrabkhani, "Forecasting electrical consumption by integration of neural network, time series and anova," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1753–1761, 2007.

[4] M. Manera and A. Marzullo, "Modelling the load curve of aggregate electricity consumption using principal components," *Environmental Modelling & Software*, vol. 20, no. 11, pp. 1389–1400, 2005.

[5] V. Ş. Ediger and S. Akar, "Arima forecasting of primary energy demand by fuel in Turkey," *Energy Policy*, vol. 35, no. 3, pp. 1701–1708, 2007.

[6] W.-Y. Chang, "A literature review of wind forecasting methods," *Journal of Power and Energy Engineering*, vol. 2, no. 4, pp. 161–168, 2014.

[7] R. Carbonneau, K. Laframboise, and R. Vahidov, "Application of machine learning techniques for supply chain demand forecasting," *European Journal of Operational Research*, vol. 184, no. 3, pp. 1140–1154, 2008.

[8] M. H. D. M. Ribeiro and L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Applied Soft Computing*, vol. 86, Article ID 105837, 2020.

[9] L. Yu, W. Dai, and L. Tang, "A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting," *Engineering Applications of Artificial Intelligence*, vol. 47, pp. 110–121, 2016.

[10] A. Altunkaynak and T. A. Nigussie, "Monthly water demand prediction using wavelet transform, first-order differencing and linear detrending techniques based on multilayer perceptron models," *Urban Water Journal*, vol. 15, no. 2, pp. 177–181, 2018.

[11] M. Awad and R. Khanna, "Support Vector Regression," in *Efficient Learning Machines*, pp. 67–80, Apress, Berkeley, CA, USA, 2015.

[12] N. C. D. Adhikari, R. Garg, S. Datt, L. Das, S. Deshpande, and A. Misra, "Ensemble Methodology for Demand Forecasting," in *Proceedings of the 2017 International Conference On Intelligent Sustainable Systems (ICISS)*, pp. 846–851, Palladam, India, December 2017.

[13] R. C. Deo, M. A. Ghorbani, S. Samadianfard, T. Maraseni, M. Bilgili, and M. Biazar, "Multi-layer perceptron hybrid model integrated with the firefly optimizer algorithm for windspeed prediction of target site using a limited set of neighboring reference station data," *Renewable Energy*, vol. 116, pp. 309–323, 2018.

[14] I. S. Msiza, F. V. Nelwamondo, and T. Marwala, "Water demand forecasting using multi-layer perceptron and radial basis functions," in *Proceedings of the 2007 International Joint Conference On Neural Networks*, pp. 13–18, Orlando, FL, USA, August 2007.

[15] M. K. Hasan, M. A. Alam, D. Das, E. Hossain, and M. Hasan, "Diabetes prediction using ensembling of different machine learning classifiers," *IEEE Access*, vol. 8, Article ID 76516, 2020.

[16] C. Chen, K. Li, M. Duan, and K. Li, "Extreme Learning Machine and its Applications in Big Data Processing," in *Big Data Analytics For Sensor Network Collected Intelligence*, pp. 117–150, Elsevier, 2017.

[17] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decision Support Systems*, vol. 46, no. 1, pp. 411–419, 2008.

[18] F. L. Chen and T. Y. Ou, "Sales forecasting system based on gray extreme learning machine with taguchi method in retail industry," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1336–1345, 2011.

[19] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[20] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[21] F. Divina, A. Gilson, F. Goméz-Vela, M. García Torres, and J. Torres, "Stacking ensemble learning for short-term electricity consumption forecasting," *Energies*, vol. 11, no. 4, p. 949, 2018.

[22] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression," *ACM Computing Surveys*, vol. 45, no. 1, pp. 1–40, 2012.

[23] E. Soares, P. Costa Jr, B. Costa, and D. Leite, "Ensemble of evolving data clouds and fuzzy models for weather time series prediction," *Applied Soft Computing*, vol. 64, pp. 445–453, 2018.

[24] M. A. Khairalla, X. Ning, N. T. Al-Jallad, and M. O. El-Faroug, "Short-term forecasting for energy consumption through stacking heterogeneous ensemble learning model," *Energies*, vol. 11, no. 6, p. 1605, 2018.

[25] M. Tahir, R. El-Shatshat, and M. Salama, "Improved stacked ensemble based model for very short-term wind power forecasting," in *Proceedings of the 2018 Fifty Third International Universities Power Engineering Conference (UPEC)*, pp. 1–6, Glasgow, UK, September 2018.

[26] S. Cankurt, "Tourism Demand Forecasting Using Ensembles of Regression Trees," in *Proceedings of the 2016 IEEE Eighth International Conference on Intelligent Systems (IS)*, pp. 702–708, Sofia, Bulgaria, September 2016.

[27] K. Yang, F. Tian, L. Chen, and S. Li, "Realized volatility forecast of agricultural futures using the har models with bagging and combination approaches," *International Review of Economics & Finance*, vol. 49, pp. 276–291, 2017.

[28] L. Wang, W. Duan, D. Qu, and S. Wang, "What matters for global food price volatility?" *Empirical Economics*, vol. 54, no. 4, pp. 1549–1572, 2018.

[29] X. Tao, L. Chongguang, and B. Yukun, "An improved eemd-based hybrid approach for the short-term forecasting of hog price in China," *Agricultural Economics*, vol. 63, no. 3, pp. 136–148, 2017.

[30] G. T. Ribeiro, V. C. Mariani, and L. D. S. Coelho, "Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting," *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 272–281, 2019.

[31] R. G. da Silva, M. H. D. M. Ribeiro, S. R. Moreno, V. C. Mariani, and L. d. S. Coelho, "A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting," *Energy*, vol. 216, Article ID 119174, 2021.

[32] L. Yu, Y. Ma, and M. Ma, "An effective rolling decomposition-ensemble model for gasoline consumption forecasting," *Energy*, vol. 222, Article ID 119869, 2021.

[33] Z. Liu, P. Jiang, J. Wang, and L. Zhang, "Ensemble forecasting system for short-term wind speed forecasting based on optimal sub-model selection and multi-objective version of mayfly optimization algorithm," *Expert Systems with Applications*, vol. 177, Article ID 114974, 2021.

[34] R. D. Cook and S. Weisberg, "Graphs in statistical analysis: is the medium the message?" *The American Statistician*, vol. 53, no. 1, pp. 29–37, 1999.

[35] I. Ebtehaj and H. Bonakdari, "Assessment of evolutionary algorithms in predicting non-deposition sediment transport," *Urban Water Journal*, vol. 13, no. 5, pp. 499–510, 2016.

[36] I. Ebtehaj, H. Bonakdari, and S. Shamshirband, "Extreme learning machine assessment for estimating sediment transport in open channels," *Engineering with Computers*, vol. 32, no. 4, pp. 691–704, 2016.

[37] I.-K. Yeo and R. A. Johnson, "A new family of power transformations to improve I. Ebtehaj and H. Bonakdari, "Assessment of evolutionary algorithms in predicting non-deposition sediment transport," *Urban Water Journal*, vol. 13, no. 5, pp. 499–510, 2016.

[38] I.-K. Yeo and R. A. Johnson, "A New Family of Power Transformations to Improve Normality or symmetry," *Biometrika*, vol. 87, no. 4, pp. 954–959, 2000.

[39] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks : The Official Journal of the International Neural Network Society*, vol. 13, no. 4-5, pp. 411–430, 2000.

[40] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012.

[41] H. Yasin, R. E. Caraka, A. Hoyyi, and Tarno, "Prediction of crude oil prices using support vector regression (svr) with grid search-cross validation algorithm," *Global Journal of Pure and Applied Mathematics*, vol. 12, no. 4, pp. 3009–3020, 2016.

[42] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.

[43] P. Shine, M. D. Murphy, J. Upton, and T. Scully, "Machine-learning algorithms for predicting on-farm direct water and electricity consumption on pasture based dairy farms," *Computers and Electronics in Agriculture*, vol. 150, pp. 74–87, 2018.

[44] S. P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis, "Forecasting stock market crisis events using deep and statistical machine learning techniques," *Expert Systems with Applications*, vol. 112, pp. 353–371, 2018.

[45] M.-W. Li, Y.-T. Wang, J. Geng, and W.-C. Hong, "Chaos cloud quantum bat hybrid optimization algorithm," *Nonlinear Dynamics*, vol. 103, no. 1, pp. 1167–1193, 2021.

[46] B. E. Flores, "The utilization of the wilcoxon test to compare forecasting methods: a note," *International Journal of Forecasting*, vol. 5, no. 4, pp. 529–535, 1989.

[47] G.-F. Fan, S. Qing, H. Wang, W.-C. Hong, and H.-J. Li, "Support vector regression model based on empirical mode decomposition and auto regression for electric load forecasting," *Energies*, vol. 6, no. 4, pp. 1887–1901, 2013.