IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 1 March 2024, accepted 9 April 2024, date of publication 15 April 2024, date of current version 23 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3389039

RESEARCH ARTICLE

Self-Supervised Adaptive Learning Algorithm for Multi-Horizon Electricity Price Forecasting

MUHAMMAD AHSAN ZAMEE[1], (Member, IEEE), **YEONGSANG LEE**[2], AND **DONGJUN WON**[2], (Member, IEEE)

[1]School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
[2]Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea

Corresponding author: Dongjun Won (djwon@inha.ac.kr)

This work was supported in part by the Energy Cloud Research and Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, Information and Communication Technology (MSIT) under Grant 2019M3F2A1073401; and in part by Korean Government (MSIT) under Grant 2023R1A2C2005090.

ABSTRACT Forecasting accuracy of electricity prices is crucial to the optimal operation of the electricity market, as improper forecasting can lead to inefficiencies, increased costs, and market instability. Thus, it is highly desired to develop a robust electricity price forecasting framework. The development of an optimal forecasting model depends on the proper choice of exogenous variables, and as the impact/characteristics of the input variables may change over time, thus the choice of appropriate external variables should be a dynamic task. Therefore, it is necessary to develop an online adaptive forecasting model, which will not only continuously forecast but also learn automatically by sensing the changes in the relationship of the variables. To sense the changes and to develop a parsimonious model proper feature engineering is required. Multi-level correlation with multicollinearity has been considered as the feature engineering tool for online training to create an accurate forecasting model. After analyzing existing studies and analyzing the gaps, an approach is proposed, utilizing a General Regression Neural Network (GRNN) with advanced feature engineering and simultaneous adaptive learning, that can outperform traditional models like ANN, RNN, and LSTM in terms of forecasting accuracy.

INDEX TERMS Electricity price forecasting, online adaptive learning, maximal information coefficient, general regression neural network, long short-term memory, recurrent neural network, artificial neural networks.

I. INTRODUCTION

From the producer to the consumer electricity price (EP) plays an important role in the electricity market industry. Depending on the market price, not only do the producers decide when to participate but the consumers can also decide when more electricity can be consumed. Thus, EP can be considered a market-controlling parameter and helps understand the market participant's behavior (producer and consumers). Often large-scale consumers schedule their energy consumption based on the EP. Thus, if the producers can forecast the EP accurately, optimal energy consumption scheduling can be performed without much trouble. Also, in modern-day power systems, multi-time scaled markets such as forward, spot, and balancing markets are found in many countries [1]. A proper forecasting model also helps the producers to determine when to participate and gain profit from the market. Thus, without a proper forecasting model, optimal operation/scheduling becomes a daunting task from both ends. Thus, the development of an optimal EP forecasting model will always be an important task for optimal market operation.

But the conventional forecasting approach has an inherent problem of developing a forecasting model based on fixed-sized data. The issue can be explained in figure 1 and discussed in the next subsection.

A. RESEARCH GAP OF THE CONVENTIONAL METHODS AND POSSIBLE SOLUTION APPROACH

Figure 1 consists of two significant portions to understand the research gap, namely Fig 1(a) describes the error performance

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan-Li Sun.

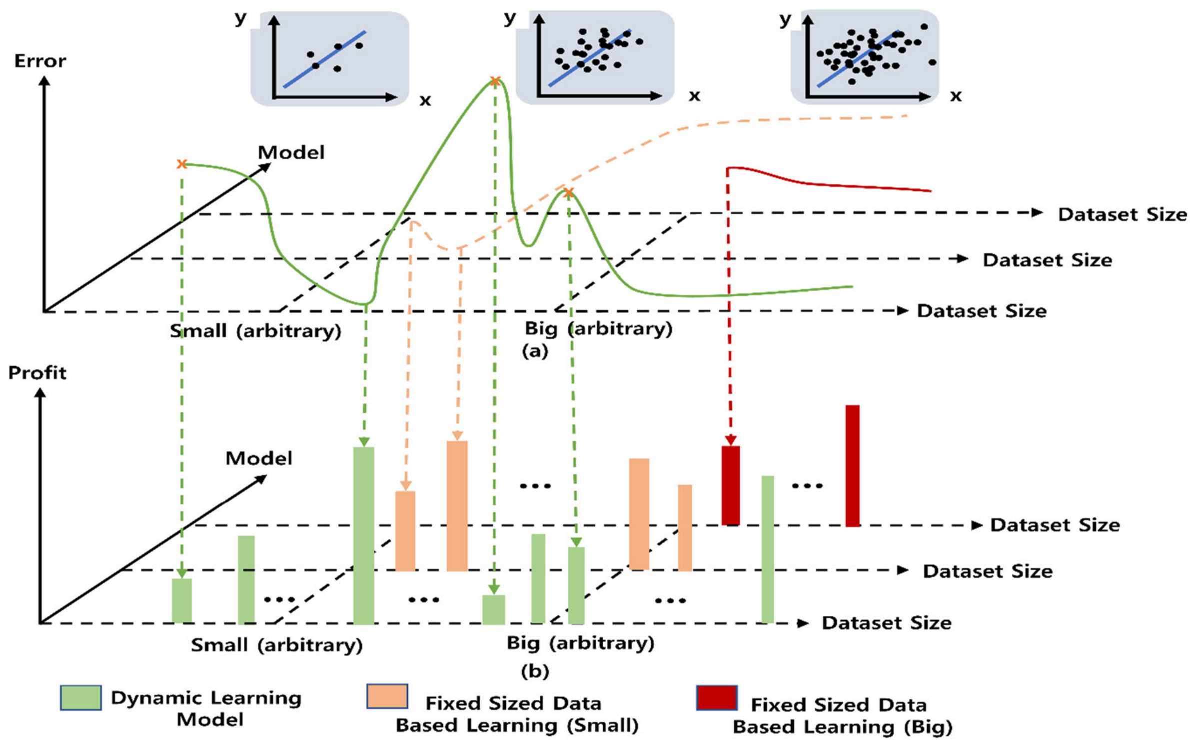

FIGURE 1. Impact of different model development approach from (a) error and (b) profit perspective.

of the forecasting models with the increase in dataset size, and Fig 1(b) shows possible profit achieved by the stakeholders (producer/consumer) based on the error performance. The small subplots at the top of Fig 1(a) show the accumulation of data over time and the corresponding change in the relationship between the possible inputs and inputs and the output parameter. It should be mentioned that error and profit have a reciprocal relationship, which means more profit can be achieved when smaller errors are found and vice versa.

These significant portions can be explained from three different perspectives, where two of them are based on fixed-sized data namely small (orange line) and big (Brown line), and the remaining is based on the adaptive/online/dynamic learning approach. Let's consider the learning scenario with a small fixed-sized dataset-based approach. In such an approach no error is obtained until the model is trained and deployed until a certain amount of data (relatively small) is accumulated. Also, once deployed the performance is satisfactory for a brief period of time. This is because as more data is gathered in the system (subplots), the model developed based on the limited data fails to describe the dynamics of the system and error keeps on piling as time goes by.

This on the other hand reduces the profit or in the worst case may cause loss of the entities. The case of big-sized fixed database learning can be explained by a similar approach. Here, the model is trained after a sufficiently large amount of data is gathered. Sufficiently large data defines enough data to represent the system dynamics properly. Thus, it is expected the error performance will continue to be satisfactory and remain under some tolerance limit. But deployment of the model after the large data accumulation may cause opportunity loss for the entities as the data explaining model dynamics may be accumulated long before the model is actually deployed. Therefore, in brief, the issue with fixed-sized data-based forecasting model development and deployment can be summarized as Error accumulation or loss in future profit (for the small-sized databases) and opportunity loss (for the big-sized databases).

Therefore, adaptive/online/dynamic learning approaches can be considered to overcome the inherent problem of fixed-sized database model development and uncertainty in the system. The concept of dynamic learning can be explained by the green line shown in the figure. The orange crosses on the green line of fig 1(b) denote model training points. The adaptive learning-based forecasting models are supposed to work in a closed-loop architecture with the environment. The models should be trained only when it is necessary. The necessary condition can be defined by the developer of the forecasting model. As a rule of thumb, two possible conditions can be considered for the training of a model (i) if the error of the model surpasses some predefined limit and (ii) if the system encounters a new relation with the possible input and output variables due to uncertainty. The advent of a new relationship can be reported through appropriate correlation analysis between the variables and is discussed in detail in section III. According to figure 1(a), the adaptive learning-based model goes under the model development process when the first orange cross appears. The error curve shows that after the first training and deployment of the forecasting model, the error starts decreasing and the profit starts increasing (figure 1(b)). But once a change in the relation between input-output variables occurred a sharp change in error can be seen, which initiates the retraining of the model (second orange cross). And the retraining continues to repeat every time a rise in error is found (third orange cross). This adaptive learning approach is practically viable and profitable as it overcomes the error accumulation problem of small fixed-sized database learning methods by performing necessary retraining cycles when the error increases or surpasses some tolerance limit. Also, it overcomes the problem of opportunity loss of the big fixed-sized database by deploying the forecasting model from the very beginning of the operation.

Therefore, in this research, an adaptive learning-based model has been proposed utilizing the potential of correlation of different external variables with the output variable. The proposed method has been verified through multi-horizon forecasting performance using a real dataset.

To address the issue and to develop a self-supervised adaptive forecasting model a detailed literature review has been performed to find the status of the considered research and find out the research gap and provide an optimal solution under realistic conditions. A well-disciplined literature review helped to find out the research gaps, the necessity of research, and possible solution approaches to solve the issue. The status of the corresponding research is discussed in the following literature review section.

For the convenience of the reader, the paper has been organized as follows. The related works and contributions of the work have been discussed in detail in Section II. The proposed methodology is discussed in detail in Section III. In Section IV, experiment, and result analysis for three different multi-horizon electricity forecasting has been conducted. This section also discusses the robustness of the proposed algorithm by applying it to another dataset, and the impact of not considering feature engineering and sensitivity analysis under varying hyperparameter choices. A summary of the work's advantages and future directions is given in Section V.

II. REALATED WORKS

Electricity price forecasting is a primitive study under the power system domain. Thus, a considerable amount of work is found (Approximately 357) using keywords such as electricity price forecasting/prediction. The found publications are arranged in fig 2 in order according to their year of publication. The increasing trend in publications shows the concurrent necessity of such research. The found publications are categorized into several categories based on the type of model (Machine learning/Deep Learning/Hybrid), learning method(online/offline), feature engineering, etc. Machine learning (ML) models are those that use diverse statistical models and general single-hidden-layer neural networks. A deep learning model is a model that uses advanced techniques such as Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Long Short Term Memory (LSTMs), and others. The term hybrid model refers to models that combine multiple methodologies together.

From the reviewed literature, it has been found that simple machine learning and simple hybrid models are deployed the maximum number of times (96 and 85 respectively). It should be mentioned that models without feature engineering and online training characteristics are defined as simple models. Along with them, a good number of publications are found using machine learning with feature engineering (63) and hybrid models with feature engineering (61). Compared to the machine and hybrid models applications DL models are less found in the literature (simple deep learning models (25) and deep learning with feature engineering (17)). Whereas the application of adaptive learning is rarely found, and among all the publications number of publications for adaptive/online learning are found is 10. More importantly, the simultaneous application of feature engineering and adaptive/online learning has been found only once. Thus, it can be understood that the proposed method discussed in the work is overlooked in the existing literature and thus should be considered for further progress in the field.

In the following subsections among the found 357 papers, 51 papers are discussed according to different types of models (machine learning/deep learning/hybrid model). As feature engineering and adaptive learning are key tools for the proposed methodology thus the discussion has been made in terms of the application of feature engineering and adaptive learning.

A. MACHINE LEARGING-BASED APPLICATIONS

In [2] and [3] along with Pearson correlation analysis, a random forest (RF) has been used as a forecasting tool for EP forecasting. But in that work, Pearson correlation has been used, which suffers from application for only variables with a linear relationship. In [4] multiple conventional statistical models such as naive, Autoregression (AR), Vector Autoregression (VAR), Autoregressive with Extra Input (ARX), etc. have been used as forecasting tools. The proposed framework also suggested the use of linear and logistic models for feature engineering. Regularized quantile regression averaging has been proposed for probabilistic EP forecasting in [5]. The reported work includes the Least Absolute Shrinkage and Selection Operator (LASSO) for feature selection. Feature engineering using binary genetic algorithm and Principal Component Analysis (PCA) with an Adaptive neuro-fuzzy inference system (ANFIS) -based forecasting model was developed in [6]. In [7] fractional Brownian motion has been used for the development of a discrete increment model for EP forecasting. As feature engineering, by evaluating long-range dependent characteristics and Hurst exponent [7]

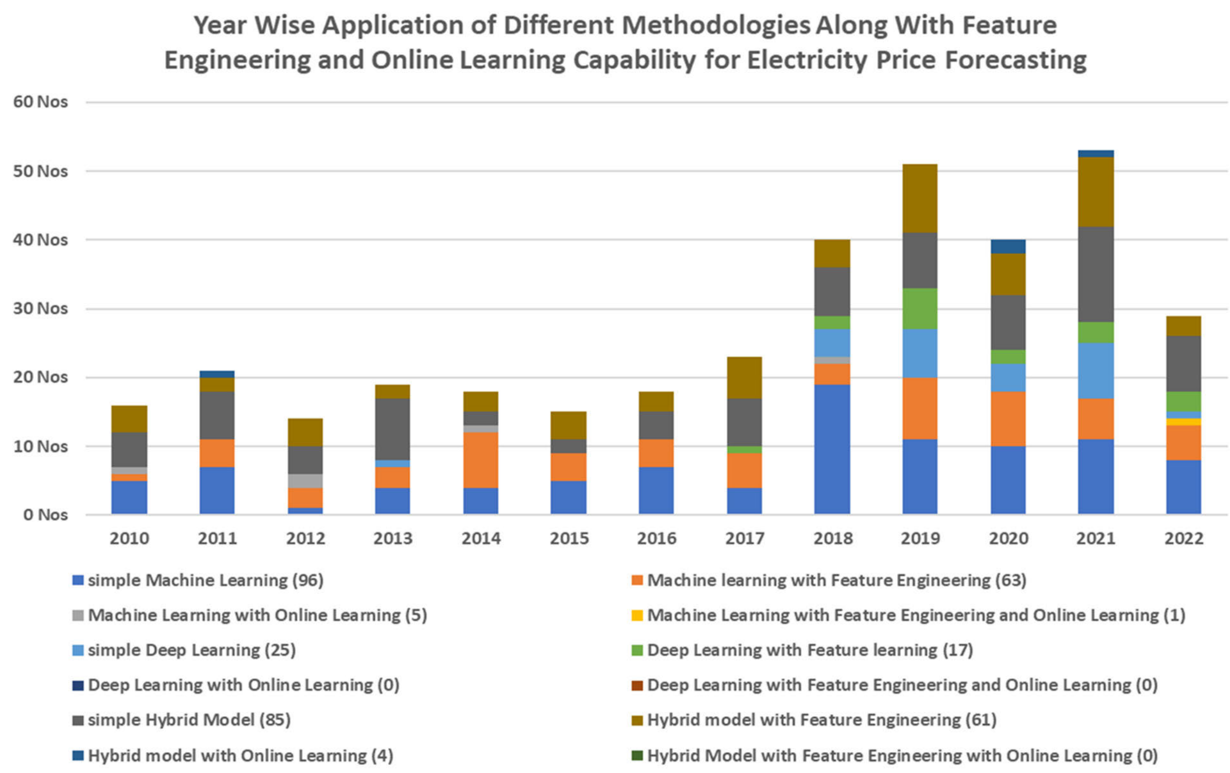

FIGURE 2. Year wise trend in related publications for electricity price forecasting.

necessary features were extracted. To forecast the electricity clearing price Bayesian Extreme Learning Machine (BELM) has been proposed in [8]. The algorithm also uses a maximum relevance algorithm for input variable selection. In [9] Enhanced Radial Basis Neural Network has been used. In the same literature, features have been selected using a Decision Tree (DT), and a Recursive feature elimination technique. And for feature extraction autocorrelation has been used. Enhanced k nearest neighbor (kNN) along with similar feature engineering methods mentioned in [9] is found in [10] for short-term EP forecasting. In [11] Dynamic Trees have been used and compared against the random forest approach. The use of one of the widely used neural network architectures Multi-Layer Perceptron (MLP)/Artificial Neural Network (ANN) has been found in [12], and [13]. Among them in [12] for feature selection information-theoretic criteria with a hybrid filter wrapper approach implemented via real coded genetic algorithm have been applied. In [14] Support Vector Machine (SVM) has been used. The work also discussed different feature engineering techniques such as principal component analysis-dynamic programming for time series segmentation, recursive feature elimination, and minimum redundancy maximum relevance feature selection. Along with the pinball loss function as a feature engineering tool, the application of the rolling window forecast model was found in [15] for forecasting the intraday spread densities of electricity prices. In [16] SVM, RBFNN (Radial Basis Function Neural Networks, WNN(Weighted Nearest Neighbor)), etc. different types of machine learning models were applied for medium term EP forecasting. Among different types of electricity prices spot price is an important category that's often required to be forecasted. One such forecasting model using the Holt-Winters model, the Recursive Least Squares-Auto Regressive (RLSAR) model has been reported in [17]. In the same literature the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) have been reported as a tool for feature engineering. Autoregressive–moving-average model (ARMA) a well-known forecasting model has been deployed and discussed in [18] along with time embedded algorithm. Point and interval forecasting is an important task for zonal EP have been performed using a semi-parametric heteroscedastic additive regression model as in [19]. In the mentioned article nonlinear backfitting algorithm was reported for feature engineering. Among a few other ML-based methods Enhanced Probability Neural Networks are found in [20].

Application of adaptive learning or online training was found in [2], [3], [11], [13], and [20], but none were applied along with feature engineering except [2]. But in that literature, for feature engineering, the Pearson correlation has been used, which is as mentioned, suitable only for variables with linear relations. But, in real-world problems, relations between variables are restricted not only to linear relationships. Nonlinear (monotonic and non-monotonic), discrete relations between the variables are found often in the systems. Thus, adaptive learning should be employed with appropriate type-dependent correlation analysis for feature engineering.

B. DEEP LEARNING-BASED APPLICATION

The use of Deep Neural Networks (DNN) has been reported on many occasions for day-ahead EP forecasting [21], [22]. Along with the application of DNN, bayesian optimization and functional analysis of variance have been used for input variable selections in [22]. The long Short-Term Memory (LSTM), a special variant of the RNN model which already gained popularity for its forecasting applications has been reported in [23] and [24]. Also, some articles alongside [21], [25] DNN, SVR, and LEAR have been used for creating a DL-based model. In the mentioned works, the use of shap value, autoregressive and random forest algorithms have been reported. For real-time Locational Marginal Price (LMP) forecasting in [26], Generative Adversarial Network (GAN) has been used. In [27] Deep Convolutional Neural Network (CNN) has been used for short-term electricity load and price forecasting. similar to [21], the use of a random forest algorithm for feature engineering was also found in [27]. In [28] Bayesian Recurrent Neural Network (RNN) has been used for day-ahead EP forecasting in Europe. The authors proposed the persistence approach also known as Davies Bouldin method [28] for feature engineering. The application of the probability density function for the day ahead EP forecasting has been found in [29]. In the reported work, a Deep Gabor convolution mixture network has been used for forecasting. Whereas Gabor convolutional and pooling layers have been used for feature variable selection.

In the discussed articles among the other feature engineering techniques, entropy, and mutual information, [23], GAN learn Spatio-temporal correlation [26], recursive feature elimination technique [27], CNN [24] was found. But interestingly, the application of adaptive/online learning has not been found.

C. HYBRID MODELS BASED APPLICATIONS

For real-time EP forecasting, quadruple branch CNN autoencoder has been used in [30]. In this work, features have been chosen through the pretraining of an autoencoder network. In [31] Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors (SARIMAX) was used along with Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) to reveal the risk factors. The dominant features were found using the ACF and PACF analysis. In [32] Marine Predators Algorithm has been used to optimize the regularized limit learning machine for the development of a forecasting model. The use of the spearman correlation coefficient and grey correlation was found in this work for the feature selection algorithm. In [33] Online Sequential Extreme Learning Machine has been used for online learning and forecasting purposes. But along with online learning use of feature engineering was not found for this work. In [34] bidirectional LSTM and a multi-head self-attention mechanism have been used. The algorithm was developed and applied for multi-horizon EP forecasting. An algorithm for gradient boosting on DTs namely catboost is used for feature finding. In [35] along with multiple feature engineering techniques such as extreme gradient boosting, elastic net and random forest Simulated Annealing optimized self-attention LSTM has been used. For forecasting of LMP in [36] Convolutional LSTM (CLSTM)-Based Generative Adversarial Network has been used. Where CLSTM has been used to find the spatio-temporal correlations among historical LMPs. In [37] Extreme Gradient Boosting (Xgboost) and random forest with bayesian linear regression have been used. Where Extra trees feature importance and univariate feature selection have been used for selecting the influencing input variables. In [38] DNN with Stacked Pruning Sparse Denoising Auto Encoder (SPSDAE) has been used. In this research, SPSDAE was used to individually decrease the noise of data sets with different sources. For the choice of input variables, Tensor Canonical Correlation Analysis (TCCA) has been performed. For day-ahead electricity price forecasting, in [39] Sequential Minimal Optimization (SMO) based Regression has been used along with genetic algorithm and tree-based method for feature engineering. To develop an optimal BP Neural Network for short-term EP forecasting, the use of another metaheuristic algorithm, Simulated Annealing Particle Swarm Optimization (SAPSO) is found in [40]. In the mentioned work for feature engineering, Maximal Information Coefficient (MIC) and Pearson correlation analysis have been used. In [41] Multi Branch Gated Recurrent Unit (GRU) has been used, where branches of GRU were used to do the feature engineering. To develop a probabilistic day ahead EP forecasting Mixture Density Recurrent Neural Network with L1 norm-based feature selection has been used in [42]. In [43] for adaptive learning, an adaptive hybrid model using SARIMA and Self-Adaptive Particle Swarm Optimization (SAPSO)-optimized Deep Belief Network (DBN) has been used. Where SAPSO was used to optimize the Variational Mode Decomposer (VMD). In [44] improved multi-objective sine cosine algorithm-based regularized Extreme Learning Machine has been used for multistep EP forecasting. In [45] wavelet transform-based Stacked Autoencoder and LSTM have been used to forecast the EP. In [46] Differential Evolution-based SVM has been used. Multiple types of feature engineering methods such as Random Forest, relief F algorithm, and grey correlation analysis, followed by kernel function and PCA are reported in the mentioned work. In [47] MLR with Autoregressive integrated moving average (ARIMA) and Hot Winters models have been used for the day ahead EP forecasting. The use of regression, p-value, and R square has been found too for feature engineering. An ensemble model of relevance vector machine, Xgboost, and elastic net regression have been used for EP prediction in [48]. The method consists of and uses mutual information and elastic net regression coefficient for selecting the input variables. In [49] Trigonometric Seasonal Box-Cox

TABLE 1. Comparison of the proposed and other methodologies.

Number	Category	Feature Engineering	Online Training	Combined Feature engineering with online training	Remarks
1	Machine Learning Based Applications [2-20]	[2-10, 12, 14-17, 19]	[2, 3, 11, 13, 20]	[2]	but limited due to the linear nature of chosen feature engineering method (Pearson correlation)
2	Deep Learning Based Applications [21-29]	[21-29]	No	No	-
3	Hybrid Application [30-51]	[30-44, 46-51]	[33, 43]	[43]	not practically feasible due to time complexity as optimizing VMD using SAPSO is itself a time-consuming task
4	Proposed Method	Yes	Yes	Yes	Practically feasible

Transformation with ARMA residuals Trend and Seasonal Components (TBATS) have been used and perform better over conventional methods such as ARIMA and ANN. In [50] grey wolf optimized Enhanced Recurrent Extreme Learning Machine and Enhanced logistic regression have been used. In the mentioned work Classification and Regression Tree (CART), Relief-F, and Recursive Feature Elimination (RFE) techniques have been used for feature selection. In [51] Grid search tuned Enhanced CNN and Enhanced SVR have been used. XG-Boost (XGB), DT, Recursive Feature Elimination (RFE), and Random Forest (RF) have been used for feature selection. In [50] authors have proposed a short-term electricity price forecasting method using an extreme learning machine, where classification and regression tree, relief-F, and recursive feature elimination were used for feature engineering.

The stated works are typically applied for day-ahead forecasting tasks. The most common input factors for forecasting include lagged load data, lagged electricity price, temporal data (day/hour/month/holiday, etc.), sale and purchase bidding data, natural gas price, LMP, crude oil price, and different meteorological data (temperature/dew point/humidity, etc.). Most research has focused on developing accurate models (reduced error) for the available data. However, except [2], [43] no instance of taking the data pattern changes during the online learning-based method has been taken into account. Also, as mentioned, that work suffers from an inappropriate choice of correlation method. In this work, the research gap has been identified and a method proposed to solve it.

From the above-detailed discussion, the summary has been organized and represented in Table 1. Table 1 has been summarized in terms of the presence of feature engineering, online learning, and their combined approach. Altogether only in 2 instances, their combined application has been found but again they suffer from inappropriate choice of feature engineering and complex time-consuming optimization of feature engineering module. Thus, to overcome the problems discussed in section I-A an effective feasible solution has been proposed in this study which considers feature engineering in online data and performs feature engineering without any additional optimization task.

D. CONTRIBUTION OF THIS RESEARCH

According to the literature review in section II, self-supervised adaptive learning methods considering appropriate feature engineering and online learning are only found on rare occasions. Inappropriate modeling can result in losses for stakeholders due to the neglect of such a phenomenon. As a result, in this work, a novel adaptive learning algorithm is proposed for electricity price forecasting. This algorithm performs feature engineering by analyzing multilevel correlation and multicollinearity, developing the model, training, forecasting, and retraining if necessary. The research contribution can be summarized as follows:

1. A multilevel MIC-based correlation-multicollinearity analysis was conducted based on the relationship between variables for dynamically sampled datasets.
2. A closed-loop platform has been developed, with a pool of different machine-learning models to choose the best model based on multi-index performance evaluation.
3. Among the pool of machine learning models, a GRNN-based electricity price forecasting model was found to perform satisfactorily by selecting dominant lagged exogenous variables through correlation-multicollinearity analysis. As compared to fully connected shallow neural networks, Recurrent Neural
4. Networks (RNN), and Long Short Term Memory (LSTM) models, the proposed model shows better performance.
5. The strength of the proposed algorithm has been justified by applying for multi-horizon forecasting ranging from an hour ahead, intraday ahead (12 hours), and day ahead (24 hours) forecasting. The robustness of the algorithm has been justified by applying it to multiple datasets.

III. PROPOSED METHODOLOGY

The proposed algorithm for adaptive learning and forecasting is shown using a flowchart in Figure 3. The measurable sensor data are collected from all over the system and stored in a database. The database could be a cloud server, which

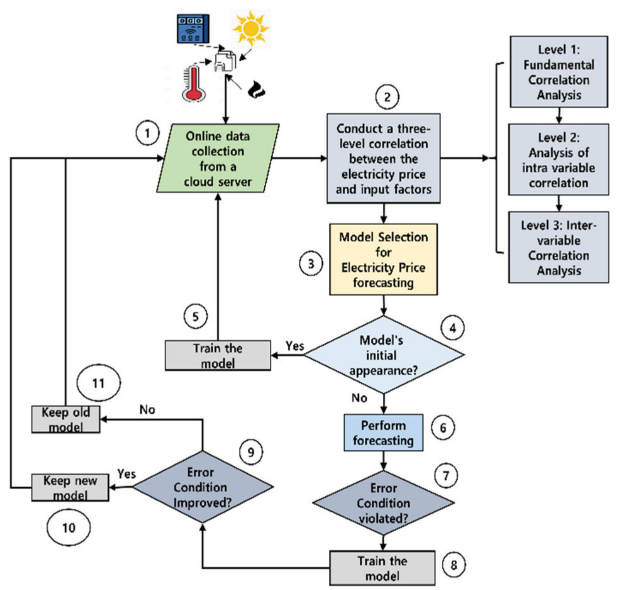

FIGURE 3. Flowchart of proposed framework.

continuously stores and feeds the data into the training and forecasting module. Upon receiving multidimensional data which includes historical EP, meteorological, gas price data, etc., the training and forecasting module conducts a three-level correlation analysis using the MIC proposed in this work. The benefits of using MIC, multilevel correlation, and related equations are described in the following subsections for detail and clear understanding. Multilevel correlations provide a unique set of combinations of input variables which also includes the lagged variables. Due to the online and adaptive structure of the proposed algorithm, the data will be continuously entered and go through this multilevel correlation analysis in every cycle. Thus, every time, either a new unique set of input variables or a previously encountered set of input variables will appear. If a completely new set of variables advents the model will go under the training cycle without performing any forecasting task. Else, the known model will perform the forecasting task and the performance will be evaluated. Performance is validated by the error of the model. If the error condition is violated, meaning if the error exceeds some specified condition, the model will due a training cycle. If the performance of the retrained model is better than the existing model, the existing model will be replaced with the newer one. Otherwise, the existing model will prevail. This complete cycle online train/test will repeat in every cycle and work in a closed-loop manner. Upon the arrival of each data multilevel correlation is evaluated, thus it is expected to have a stable correlated model after a while. The stable correlated model indicates the best model due to its explainability, error performance, and the number of training cycles. The detail of the model is also explained through the pseudocode of the algorithm (Algorithm 1). Each portion of algorithm 1 is discussed in detail below:

Algorithm 1 Pseudocode for Adaptive Learning Algorithm

1: **while** True **do**
2: Obtain data, set ML model parameters, prediction horizon, data step
3: Multilevel Correlation Analysis (Algorithm 2)
4: From the set $S_M = Model_{set}$ choose the forecasting model
5: **if** Model Appearance Count=1 **then**
6: Train the model: *Train* $(S_M)\rightarrow$ *Trained Model*
7: Test, and store the *Trained Model*, related errors, and fitting results (MAPE, APE, R^2)
8: **else**
9: forecast using the model and calculate Mean MAPE
10: **if** $MAPE_t > MAPE_{t-1} + w_1 MAPE_{t-1}$ **then**
11: set $MAPE_{beforeTrain} = MAPE_t$
12: Train the forecasting model and calculate $MAPE_{afterTrain}$
13: **if** $MAPE_{afterTrain} < MAPE_{beforeTrain}$ **then**
14: Discard the old model and Keep the new model
15: **else**
16: Keep the model before retraining
17: **end if**
18: **else**
19: Maintain the old model
20: **end if**
21: **end if**
22: **end while**

A. ANALYZING THE ALGORITHM IN DETAIL (PSEUDOCODE)

At the beginning (line 2), data are collected online, and the type of machine learning model (e.g. ANN/RNN/LSTM/GRNN, etc.) related training parameters, prediction horizon (an hour ahead/intraday (12 hours) ahead/day ahead (24 hours), etc.), data step. The data step determines the interval between two consecutive possible training/testing/retraining (if necessary) cycles. In general, the data step should be one, thus upon arrival of each new data the model be evaluated based on its performance.

But, if necessary, users can change this parameter and perform the necessary tasks at a longer horizon. Upon setting the necessary initial parameters for the algorithm, a multilevel correlation will be performed (line 3). The detail of the multilevel correlation is discussed in the next subsection. Upon the choice of input variables from multilevel correlation analysis, the model S_M will be selected from a set of models, namely $Model_{set}$ (line 4). Each of the models in the model set consists of different combinations of input variables found from the multi-level correlation analysis.

$$Model_{set} = \{M_1, M_2, \ldots, M_n\}$$

where, M_n represents n^{th} model set. As mentioned, different types of machine learning models have been used for this work for the purpose of comparison. From the experimental analysis section, it can be found that GRNN has outperformed all the other considered models. The details of the GRNN model and related equations are discussed in subsection III-C. It should be mentioned that at the very beginning, the model set will be empty. Every time a new model advent they will be added to the set. Once added to the set, the models will not be removed from the set ever. As the future behavior of the systems is uncertain, thus the stored models can be used in the future if a similar relationship between the external/input and the regressand is found.

Considering the multilevel correlation if a completely new combination of new input variables is found (line 5), then the model will directly go for the training cycle, also the related information of the model will be stored (lines 5-7). The related information includes different error/performance metrics. In this algorithm, as performance matrices Mean Absolute Percentage Error (MAPE) of the fitting, Mean Absolute Percentage Error (MAPE) of the forecast horizon, and fitting of the model through R^2 have been considered. In most literature along with MAPE, MSE or RMSE is another popular performance index. There are certain advantages when MAPE is considered. As MAPE interprets results in percentages thus for industry or for boarder class of readers/audience the results become easier to comprehend compared to other performance indexes (for example MSE or RMSE). Also, as in MAPE error is normalized by the actual values, thus MAPE is less sensitive to the scale of data. These reasons make MAPE a choice for performance index selection. MAPE considers the error of each point of the curve and due to the online nature of the algorithm their means are evaluated, thus overall fitting performance can be understood. Similar to MAPE, R^2 is also used as an indicator of fitting evaluation of the curves. The more the value of R^2 (near 1), the better the fitting is. On the other hand, it often may happen that, a model may have a lower MAPE, but its forecasting error (MAPE for forecasting horizon) may not be satisfactory. Thus, as a performance metric, the mean of MAPE for forecasting horizon is also considered so that the performance of the proposed algorithm can be evaluated correctly.

On the contrary, if a model reappears, initially it will perform forecasting (testing). If the MAPE of the model surpasses a certain percentage of the MAPE calculated at the previous entry of the data (line 10) then the MAPE of the current model will be stored in a temporary variable named as $MAPE_{beforeTrain}$ and the retraining cycle will be initiated. This condition can be mathematically expressed as

$$MAPE_t > MAPE_{t-1} + wMAPE_{t-1}$$

where, $MAPE_t$ and $MAPE_{t-1}$ are the MAPE calculated when data received at t^{th} and $(t-1)^{th}$ time respectively. w is the weightage parameter set as the maximum permissible limit to re-initiate the training and can be varied between 0 to arbitrarily any large number. 0 indicates any error greater than the previous MAPE will re-initiate the retraining cycle, and thus can be considered as the strictest condition. This condition will try to achieve the best possible model with minimum error and thus more often will perform the training cycle. But frequent training cycle of models reduces the generalization capability of the model and may restrict them from participating in the market due to poor performance. Thus, it is recommended to choose a value more than 0. In this work, we have chosen 0.1 (10%) for all other algorithms, except GRNN. From the experimental analysis section, it can be found that the learning capability of all the models except GRNN is not good enough. It can be judged based on the error performance of the trained models. The error of the GRNN model (MAPE) is minute compared to the other models. Thus to make the trained model using GRNN a large value of weightage parameter has been chosen (1000). Meaning that if the error of the GRNN model exceeds 1001 times of the previous error, only thus retraining will be performed. Even with this large number from the experimental analysis section GRNN has performed extremely well compared to other renowned algorithms (ANN, RNN, and LSTM).

The MAPE of the retrained model $MAPE_{afterTrain}$ should be evaluated and compared against $MAPE_{beforeTrain}$ (lines 12-13). Because the training cycle may improve the model forecasting accuracy but is not guaranteed due to system characteristics, the setting of the optimization and model parameters, and training time. As suggested the proposed will be completely self-supervised, therefore, to omit the human intervention, optimization parameters, for example, optimization algorithms, learning rate, the maximum number of iterations, etc. will be fixed. Thus, as the retraining of the model does not guarantee an improvement in model accuracy, the retrained model should be reevaluated. The reevaluation determines either to keep the retrained model or to discard and continue with the previous model. By performing this comparison user can decide to either keep the new model (line 14) or go with the previous model (line 16). In the following subsection, details of the performed feature engineering have been discussed.

B. FEATURE ENGINEERING/MULTILEVEL CORRELATION ANALYSIS

1) CHOICE OF FEATURE ENGINEERING TECHNIQUE

Feature engineering is one of the major tasks associated with any machine learning model. Proper feature engineering can help detect the changes in the system due to uncertainties of nature. Negligence or improper choice of feature engineering leads to the development of a poor model that affects forecasting accuracy, generalization, and explainability. Therefore, in this work, a careful approach has been considered using a robust feature engineering method, which can find out the meaningful input variables so that the model credibility becomes strong, and generalization can be achieved.

As found in the literature review section, it can be understood that many types of feature engineering methods from statistical and machine learning-based approaches can be adopted. Machine learning-based approaches are suitable and can have great outcomes in terms of offline model developments, as they separately need time for training and testing, and validation. But in real-time applications, due to time constraints, this approach would not be sufficient and may hamper the performance. Therefore, statistical or data analysis-based approaches are more suitable for real-time-based feature engineering. Among many other statistical approaches, one of the popular methods is correlation analysis. Correlation analysis provides useful insights into the relationship between the variables. Using this potential of correlation analysis, the selection of input variables for an explainable machine learning model would be appropriate and easier. From the literature, many methods for correlation analysis can be found. But they have specific applications and limitations too. Cherry picking without justification of application violates the generalization of the model. For example among the frequently used correlation analysis, Pearson [40], spearman [32], and point biserial [52] methods are often found in the literature. But as mentioned, Pearson correlation analysis is only applicable to variables with a linear relationship. Thus with a nonlinear relationship, the proper characteristics would not be found using the Pearson method. on the other hand, the spearman correlation method can be used for variables with nonlinear relationships, but they can capture only monotonic relations. Also, both methods are applicable to continuous variables. For discrete or categorical variables point biserial methods can be used.

Thus, in a system with different variable types (linear/nonlinear/continuous/discrete/categorical), different types of correlation methods should be chosen for efficient model development. If the nature of the system variables is hard to identify in advance, then it would be difficult to choose the appropriate correlation methods for the system. Also, multiple uses of different types of correlation analysis will make the algorithm complex, also the future addition of new variables in the system will also require modification in the existing algorithm for proper operation. Such human interaction would make algorithm development an inefficient task. Thus, to avoid such complex procedures and multiple issues, the Maximal Information Coefficient (MIC) [53] has been selected in this work. MIC can be applicable for any type of variable irrespective of type, which means it can be equally applicable and can correctly capture the relations of continuous/discrete/linear/nonlinear variables. In the next subsection, a detailed analysis of multilevel correlation analysis utilizing the potential of MIC has been presented through the in-detail discussion of the pseudocode of algorithm 2.

2) DETAIL ANALYSIS OF MULTILEVEL CORRELATION

Multilevel correlation is one of the major tasks performed in this work and needs separate algorithm representation to understand its workflow and impact on the system design. Thus, a separate pseudocode describing the working structure of the multilevel correlation is shown in algorithm 2. Multilevel correlation consists of three levels of analysis, namely primary, secondary, and tertiary correlation analysis. Apart from the basic correlation analysis from the primary level, secondary and tertiary analyses have been performed to identify the existence of the multicollinearity between the selected variables and omit them. The omission of variables due to multicollinearity essentially creates the forecasting model with the only minimum required input variables.

Algorithm 2 Pseudocode for Multilevel Correlation Analysis

Initialize th_1, th_2, th_3, and o_{dmax}
while $X_n != X_{ALL}$ **do**
  While $MIC_{n_{dm}n_{dm}} > th_1 \& \; del_{jd1} < o_{dmax}$ **do**
Create lagged input and output variables using equation (2)
    calculate $MIC_{n_{dm}n_{dm}}$ using equation (1)
  end while
end while
while $X_n != X_{ALL}$ **do**
If $X_n > 1$ **then**
  while $MIC_{n_{dm}n_{dm}} < th_2 \parallel del_{jl2} > o_{dmax}$ **do**
Create lagged input and output variables using equations (3)
    Calculate $MIC_{n_{dm}n_{dm}}$ using equation (1)
    end while
end if
end while
Create lagged input variables for tertiary correlation analysis (4)
Calculate $MIC_{n_{dm}n_{dm}}$ using equation (1) and stored in H
Identify matrix positions with $MIC_{n_{dm}n_{dm}} > th_3$ and corner items
Revise the delay order of each input variable and finalize the input matrix

Thus, training in the forecasting models will be more efficient and generalization conditions could be satisfied. This should be mentioned that Multicollinearity is a phenomenon in which the system can be explained by only one of two highly correlated explanatory factors. Multicollinearity analysis should be taken into account in addition to correlation analysis in order to construct a parsimonious model. Now each level of the multilevel correlation analysis is discussed below using the pseudocode shown in algorithm 2 and related equations.

3) PRE CORRELATION PARAMETER SETUP

It should be mentioned that prior to the operation of multilevel correlation 4 parameters namely th_1, th_2, th_3, and o_{dmax}, need to define for the smooth operation of the correlation analysis. Where, th_1, th_2, and th_3 are the threshold parameters of primary, secondary, and tertiary correlation analysis respectively. These parameters are used as a base parameter for comparison with the calculated correlation of each possible input variable and their delays with the output variable. Such comparison helps decide the variables to be possible contenders for the input variable. The details of the conditions for these comparisons are given in the pseudocode and also discussed below while discussing the corresponding operations. o_{dmax} is the maximum allowable delay for individual variables. This variable is important to limit the total number of input variables and is a user-choice parameter. If a large value of o_{dmax} is considered, due to the consideration of more delays longer correlation calculation time will be required. On the contrary, smaller values hinder the forecasting model development process. Thus this parameter is of great importance and should be chosen with great care. The details of the choice of this parameter can be found in the experimental analysis section.

4) PRIMARY CORREALATION ANALYSIS

Primary correlation is the correlation analysis that is required to perform between each possible input/external variable and their lagged variables to the predicting variable. To perform the correlation, MIC has been chosen and can be calculated using the following equation:

$$MIC\,(A,B) = max\frac{I\,(A,B)}{log_2 min\,\{n_A, n_B\}} \tag{1}$$

From equation (1) we can see that MIC(A, B) is the mutual information between two random variables A and B normalized by their minimum joint entropy. I interpret the MIC as the percent of variable B that can be explained by variable A. More on MIC can be found from [53]. A grid should be created to calculate the MIC for each pair of data. The number of cells in these grids can vary, and the primary purpose of creating a grid is to explore and capture the relationship between two variables. Following that for each grid mutual information needs to be calculated. Mutual information identifies the mutual dependence between the two variables. After that, mutual information should be normalized by the logarithm of the minimum of the numbers of bins in A and B. This process makes the score independent of the grid size, as a result, comparisons across different grid sizes can be performed. Once this is done, the maximum normalized information score obtained across all the grid sizes is determined as the MIC.

Soon as the required parameters are initialized (line 1), the primary correlation should be performed for all the variables (line 2) by rearranging/resizing the input and output variables until the condition of line 3 is violated. Resizing is necessary because each variable should have the same size since the delays of each individual variable are taken into account when performing the correlation between the variables. The equal-sized input and the output variable can be created from the data set by using the following equation:

$$\begin{cases} X_{del_{jd1}} = IO\left[o_{dmax} - \left(del_{jd1} - 1\right)\right. \\ \left. : T_d - 2f_h - \left(del_{jd1} - 1\right)\right] \\ Y = O\left[o_{dmax} + f_h : T_d - f_h\right] \end{cases} \tag{2}$$

In the above two equations, $X_{del_{jd1}}$ is the delayed input variable of the j^{th} variable, whereas the subscript $d1$ denotes the delayed input variable from level 1. Y is the output variable. IO, $O$is the original input and output dataset from where the resizing is performed. del_{jd1} can vary from 1 to maximum o_{dmax}. T_d is the total data length after the arrival of each new data, and f_h is the forecasting horizon. It is important to mention that, during the correlation calculation, the actual value of output ($f_h - step\ future\ value$) is not known, rather only the current value is known. Thus, the most recent output should be correlated with the delayed input variables, and for the generalization, it is assumed that the same relation will prevail within the forecasting horizon. Thus to create the input variable, the actual data length (T_d)-forecasting horizon (f_h) amount of data should be taken into account. Or in other words, the input variables should have at least f_h step delays than the output variable.

Thus the condition for performing primary correlation (line 3) can be summarized as follows. The variable del_{jd1} will start from 1 and will continue to increase until the calculated correlation exceeds the set threshold value th_1 and the number of delays del_{jd1} is do not exceed the maximum number of allowed delays o_{dmax} line 3-6)

As mentioned, the iterative operation of this task should be performed for all the possible input variables. The primary correlation analysis ensures that all the possible input variables are selected along with their delayed copies based on the set threshold value th_1. Unless the nature of the variables is known in advance, it is recommended that a lower value of correlation should be considered (0.10-0.50). The choice of a higher value (> 0.50) removes the variable with a lower correlation. If higher correlated variables do not exist, with a higher choice of th_1 the algorithm will fail to progress.

5) SECONDARY CORRELATION ANALYSIS

After selecting the variables from the primary correlation analysis, the secondary correlation calculation will be performed. secondary correlation finds multicollinearity between intra-variable delays. For example, if 3 delayed copies of any arbitrary variable x are found from primary correlation analysis, in secondary correlation analysis, multicollinearity will be analyzed between these 3 delayed copies. Upon the analysis, if any multicollinearity is found, one or more delayed copies will be removed. Again to perform secondary correlation, similar to primary correlation analysis two variables should have the same length and can be created (line 11) using the following equations:

$$\begin{aligned} X_{del_{jd2}} = IO[&O_{dmax} - (del_{jd1} - (del_{jd2} - 1)) : \\ &T_d - 2f_h - (del_{jd1} - (del_{jd2} - 1))] \end{aligned} \tag{3}$$

In equation 3, del_{jd2} can vary between 1 to del_{jd1}. Although each input may have a different number of delays, due to the (del_{jd2}- 1) term on both sides of the colon symbol of equation (3) each delayed input can maintain the same length. The value of (del_{jd1} will continue to increase if the condition of line 10 of the pseudocode continues to satisfy. According to line 10 to perform the secondary correlation, the correlation value $MIC_{n_{dm}n_{dm}}$ should be less than th_2. The choice of the value of th_2 is interesting and different from the choice of a threshold quantity of level 1 th_1. Unlike primary correlation, the threshold quantity for the secondary correlation should be higher (> 0.80). This is because in primary correlation the objective was to collect the maximum number of possible input variables with all their possible delayed copies. But in the secondary, the objective is to reduce the number of delayed copies by finding the existence of multicollinearity within the selected variables (intra-variable collinearity). The second condition in line 10 defines how long the multicollinearity checking should be performed. According to the second condition the multicollinearity should be performed until the del_{jd2} exceeds the maximum allowable delayed length defined by the user n_{dmax}.

It is expected that, similar to the primary correlation the secondary correlation should be performed for all the selected input variables from the primary correlation (line 9). But for all the selected variables secondary correlation is not possible to perform. If more than one delayed copy of any variable exits thus only the secondary correlation for that individual variable will be performed, as it represents the existence of multiple delays of the variable.

6) TERTIARY CORRELATION ANALYSIS

The tertiary correlation is performed to identify the existence of the multicollinearity between the remaining delayed variables (after the secondary correlation) of each variable. Thus this tertiary correlation can be defined as the intervariable multicollinearity analysis. Again similar to the previous two correlation analyses, to perform the tertiary correlation, the input variables can be resized using the following equation (line 16):

$$X_{del_{jd3}} = IO[o_{dmax} - \left(del_{jd3} - 1\right) : T_d - 2f_h - M_d - (del_{jd3} - 1)] \tag{4}$$

In equation 4, del_{jd3} can be varied between 1 to as high as del_{jd2}. In equation 4, an additional variable namely the highest delay H_d can be found, so that the size of both of the variables becomes equal. To be more specific, all the input variables should have the same size. But each input variable may have a different number of delayed variables, thus to match the length of the input set, M_d has been considered.

An example explaining the impact of M_d can be discussed to understand its operation. As the name suggests, M_d represents the highest delay, meaning that it is the maximum value of delay that is found among all the variables. A system with 3 different variables can be considered to elaborate the example. And after primary and secondary correlation analysis, the number of delays for each variable has been found as 15, 20, and 17. As 20 is the highest delay among the 3 variables, thus 20 should be set as M_d. To explain the impact of M_d, along with M_d we need to also choose some numbers for the variables T_d, f_h, and o_{dmax} arbitrarily. Let us choose T_d as 250, f_h as 1, and o_{dmax} as 25. Thus the size of a delayed variable of variable 1 would be, IO[25-(15-1):250-2*1-20-(15-1)]=IO[11:214], and for variable 2 would be IO[25-(20-1)):250-2*1-20-(20-1)]=IO[6:209]. So it can be seen that, although they are different variables with different numbers of delays, the data length would be the same.

After finalizing the input variables for tertiary correlation analysis, an input matrix having the same data length for each variable including the delayed variables will be created. Afterward, the correlation between each variable should be performed and stored in a Square matrix (K) (line 17).

$$K = \begin{bmatrix} MIC_{1_{d1}1_{d1}} & MIC_{1_{d1}1_{d2}} & \cdots & MIC_{1_{d1}n_{dm}} \\ \vdots & \vdots & \ddots & \vdots \\ MIC_{n_{d1}1_{d1}} & MIC_{n_{d1}1_{d2}} & \cdots & MIC_{n_{dm}n_{dm}} \end{bmatrix}$$

In the K matrix, each row represents the correlation of each delayed variable with the other delayed variables. For example, $MIC_{n_{d1}1_{d1}}$ is the correlation coefficient of the first lagged variable of the n^{th} variable with the first delayed variable of the 1^{st} variable. The diagonal elements correspond to self-correlation thus it will always be 1, thus should not be considered to avoid the self-correlation. For the other matrix positions where the threshold condition is not satisfied, those positions will be replaced with an arbitrarily large value. Now neglecting the diagonal items and position with higher threshold values the final delay order of each input variable and final input matrix for the training model can be created (lines 18-19). Once the multilevel correlation is done, model evaluation and the remaining part of algorithm 1 can be performed as discussed initially.

C. CHOICE OF FORECASTING MODEL

In this work, four different renowned machine learning models have been used for the proposed algorithm namely Artificial Neural Network, Recurrent Neural Network, General Regression Neural Network, and Long Short Term Memory. All these models have a proven track record of successful application for time series forecasting tasks. By comparing the performance of these four models under the proposed algorithm the best suitable model for such an application can be decided. From the experimental analysis section, it can be found that by considering multi-index comparative analysis GRNN was found successfully dominant over the other three models. Thus GRNN has been selected as the model to describe in the paper. The details of the other three respective models can be found in many works of literature.

GRNN was proposed by D.F. Specht [54]. According to [55] GRNN can be used as a good solution for an online dynamical system. Unlike generic neural network models, GRNN does not require any backpropagation, and also demonstrates high accuracy due to the use of gaussian functions. The general structure of a GRNN can be represented using figure 4.

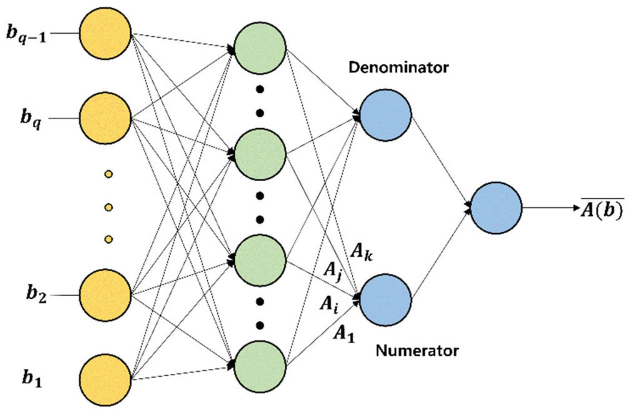

FIGURE 4. General structure of GRNN.

The following equation can be used for approximating the output of GRNN.

$$A(b) = \frac{\sum A_i - \frac{z_i^2}{2\sigma^2}}{\sum e^{-\frac{z_i^2}{2\sigma^2}}} \tag{5}$$

where, $z_i^2 = (b - b_i)^T\,(b - b_i)$. b and bi are the input and training samples respectively. Ai is the output of sample i. In the output approximating equation, only the unknown variable is (σ), hence the optimal value of the training process depends only on this parameter. Thus training is much simpler and faster than any conventional neural network. Faster training time makes it a good choice for adaptive online/dynamic learning. The justification for this statement can be found in the experimental analysis of this work also.

Also, it is worth mentioning that, in this work, all the inputs to forecasting models are normalized using the following equation [56]:

$$\bar{x}_t = \frac{x_t - x_{tmin}}{x_{tmax} - x_{tmin}} \tag{6}$$

IV. EXPERIMENTAL ANALYSIS OF THE PROPOSED ALGORITHM

A. DATA IMFORMATION AND EXPERIMANT SETUP

The data for this work has been taken from a public dataset available in [57]. The data consists of hourly data from the NEPOOL region (courtesy ISO New England) from 2004 to 2008. The dataset contains 14 variables in addition to the electricity price. The introduction of the variables can be found in Table 2. In this work, we have used 1-year full data and considered it as streaming data, meaning that instead of using 1-year data altogether the data has been inserted in the system sequentially. The proposed algorithm has been applied for different multi-horizon (1-hour, 12-hour, and 24-hour) forecasting to exploit the potential of the algorithm. Forecasting horizons have been chosen based on their importance in terms of application. For example, 1-hour ahead forec asting can be used for real-time pricing tasks, 12 hours can be used for intra-day, and 24 hours ahead forecasting can be used for the most commonly used day-ahead forecasting task. The algorithm has been implemented using MATLAB 2022a, in a PC with 12th generation Intel(R) Core(TM) i9-12900K 3.20 GHz and 32GB of RAM configuration.

TABLE 2. Data characteristic.

Number	Parameters
Var1	Dry bulb Temperature
Var2	Dew Point Temperature
Var3	Hour of day
Var4	Day of the Week
Var5	Holiday/Weekend Indicator (0 or 1)
Var6	System load
Var7	Previous day's average load
Var8	Load from the same hour the previous day
Var9	Load from the same hour and same day from the previous week
Var10	Previous day's average price
Var11	Price from the same hour the previous day
Var12	Price from the same hour and same day from the previous week
Var13	Previous day's natural gas price
Var14	Previous week's average natural gas price
Var15(output)	Electricity Price

TABLE 3. Training parameters for different machine learning models.

Model	Parameter	value
ANN	optimization algorithm	levenberg Marquardt
	Transfer function	Positive linear transfer function
	Max Epoch	1000
RNN	optimization algorithm	levenberg Marquardt
	Max Epoch	20
LSTM	optimization algorithm	ADAM
	Max Epoch	1000
	Mini Batch Size	128
	Learning rate	0.005
GRNN	sigma	0.009

For comparison with the GRNN, 3 different other renowned models have been chosen for performance comparison, namely ANN, RNN, and LSTM. The parameter setting for all four models can be found in Table 3.

The number of hidden layers for ANN, RNN, and LSTM is highly important and can affect the model's performance. Using the recommendations from [58], the number of hidden layers has been determined as follows.

$$Hidden_{neuron} = \frac{2}{3} \times No\ of\ input\ variables$$

Along with the mentioned model parameters, threshold values for three-level multicollinearity have been selected also. The threshold value of primary correlation analysis th1 is selected as 0.5 for 1 hour and 0.2 for 12 and 24 hours ahead forecasting horizon respectively. While performing the forecasting task, as future data is unknown, thus current data correlation is assumed to prevail in the forecasting horizon. As 1 hour is just the immediate data to forecast, the probability of prevailing correlation is high, thus a higher correlation threshold value is selected for 1 hour ahead forecasting task. Whereas due to the longer range of forecasting horizon for 12 hours and 24 hours ahead task, the correlation prevailing probability is supposed to be weaker, thus a lower correlation threshold value has been chosen, such that more initial variables can be selected and forecasting performance does not hamper. As the secondary and tertiary correlation omits the initially selected variables from the primary correlation analysis thus, a higher correlation threshold has been selected for secondary and tertiary correlation multicollinearity analysis (0.9) irrespective of the forecasting horizon. As mentioned in the precorrelation parameter setup subsection, the maximum allowable delay *odmax* also requires discussion on the choice of this variable. For the shortest forecasting horizon (1 hour ahead) this value is chosen as 20. That means for each variable maximum of 20 lagged copies can be considered. Even for a shorter range, this value is considerably large, because this problem has 15 variables (including the output itself), thus considering 20 lagged copies for each variable a maximum of 300 inputs can be found. Although the maximum allowable number of inputs is high, but due to multilevel correlation analysis only impactful variables will be selected. Thus the total number of variables will vary under 20 for 1 hour ahead of forecasting as can be found in figure 8 as discussed in section IV-C. For 12 hours and 24 hours ahead forecasting tasks *odmax* has been selected as 10 only. This is because of the lower threshold value for the longer forecasting horizon. As for the longer forecasting horizon, a lower correlation threshold has been chosen, thus it can be assumed that more lagged copies of variables could be selected. Thus to create a computationally efficient model a smaller value of *odmax* has been chosen.

Under the mentioned data and experimental setup condition, the multi-horizon forecasting task is performed and the corresponding results are discussed in the following subsection.

B. RESULT ANALYSIS

After applying the proposed adaptive learning algorithm using four different machine learning models for three different multi-horizon tasks results have been found and listed in Table 4. Also, the complete forecasting figure for all three forecasting tasks is shown in Figure 5.

Figure 5 also shows the minute details by displaying three different time zone figures as an inset of each figure. The figure illustrates how each model performs after the completion of training/retraining/testing cycles after 1 year of streaming data. It should be mentioned that training/retraining/testing occurs based on the conditions derived from multilevel correlation analysis (input combination) and forecasting accuracy. The result shows that in all cases GRNN performs the best fit compared to other algorithms (0.9614 for hour ahead, 0.9094 for intraday, 0.9851 for day ahead), which can be also found in Table 4 (average R^2). Next to GRNN, RNN also performs relatively better compared to the other two models. In fact, ANN performs poorly in all cases, whereas LSTM was found to perform better for the smallest forecasting horizon (1 hour). But as the forecasting horizon starts to increase, with respect to the R^2 index, LSTM performs poorly compared to the other two algorithms.

As one of the major tasks of the model is to correctly forecast thus only fitting (R^2) should not be enough to justify a model as the best model. Therefore along with the fitting performance, forecasting accuracy (Mean of Mean Absolute Percentage of Error of forecasting horizon) and mean of model total fitting error (Mean Absolute Percentage Error of curve fitting) should also be considered. The related information can be found in Table 4 also. Using the proposed algorithm for EP forecasting, the least mean MAPE (fitting) and least mean MAPE (forecasting horizon) for a 1-hour forecasting horizon fitting horizon has been found using RNN (5.2514 for mean MAPE (fitting) and 5.8835 for Mean MAPE (forecasting horizon)). Due to the real-time online application of the proposed algorithm along with the mentioned fitting-based and error-based indexes, training time and training cycle-based accuracy should be considered also. Lower training time provides the scope for the algorithm to train under practical time constraints. Whereas a lower number of training cycles proves the model's generalization capability. If the error of a reappeared model does not exceed a certain percentage of error from the previous forecasting task, the model will not go under the retraining cycle. Less number of retraining cycles shows the robustness of the model, also from the practical perspective lower number of training/retraining cycles provides more profit for the stakeholders, as the chances of market participation become higher. Therefore considering the training time-based index (average training time) and training cycle-based index (no of training cycles) RNN (0.1007 seconds) and GRNN (51 training/retraining cycles out of 872 cycles) have performed best respectively, among the compared models for 1 hour ahead forecasting task. Therefore, it can be summarized that for 1 hour ahead forecasting task in terms of different indexes different algorithms have shown their superiority over other models. Therefore using the indexes for 1 hour ahead forecasting for choosing the best model is a difficult task and to some extent is not a reliable method. Therefore to choose the best ML model using the proposed algorithm two more multi-horizon forecasting tasks have been performed. The results from the multi-horizon forecasting tasks can provide a clear direction for choosing the best model. The results for the other two horizons (12 hours and 24 hours ahead) are discussed below:

For 12 hour ahead forecasting task, GRNN has the lowest fitting mean MAPE (5.1071%), fastest training time (0.5863 seconds), and highest fitting accuracy (0.9094). Whereas mean MAPE for forecasting horizon RNN has the best accuracy (10.8535%) and LSTM performs the least number of training/retraining tasks (112 out of 722). Similarly

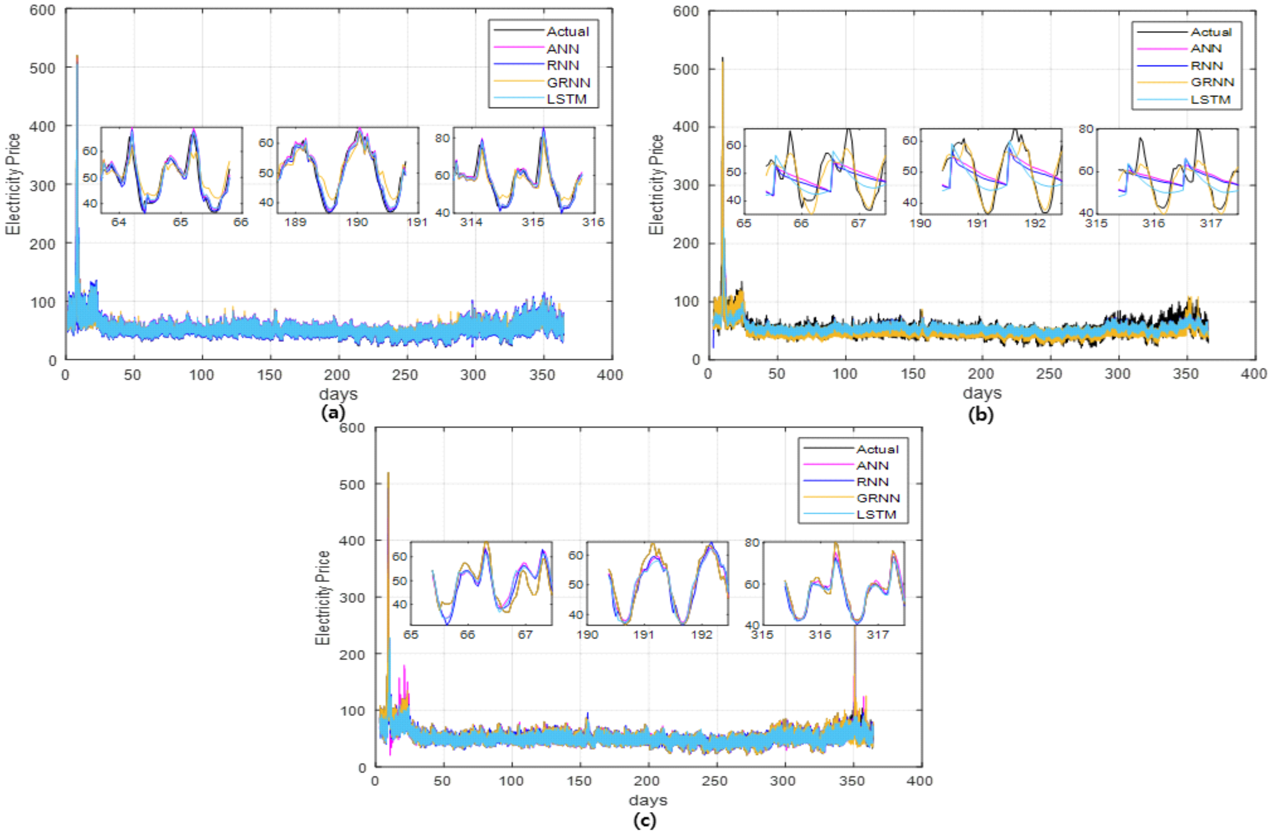

FIGURE 5. Electricity price forecasting model performance for (a) 1 hour (b) 12 hour (c) 24 hour ahead forecasting.

TABLE 4. Performance analysis of different neural network architectures for electricity price forecasting.

Forecasting Horizon (hour)	Forecasting Model	Mean MAPE (fitting)	No of Training/ Retraining Cycles	Avg. Training Time (sec)	Avg. R^2	Mean MAPE (forecasting Horizon)	Maximum Training Time in a cycle (sec)
1	ANN	182.7352	656	0.6583	0.5858	63.1855	2.4017
	RNN	5.2514	66	0.1007	0.9528	5.8835	0.9828
	GRNN	6.7661	51	0.1074	0.9614	9.0834	0.31288
	LSTM	6.0158	64	5.5387	0.9386	6.1585	101.4567
12	ANN	172.8778	562	1.3334	0.4058	70.6879	19.6194
	RNN	9.0489	142	10.5472	0.7911	10.8535	296.0894
	GRNN	5.1071	154	0.5863	0.9094	10.9911	1.7043
	LSTM	15.0789	112	38.3690	0.5749	16.6870	938.8834
24	ANN	149.4132	294	1.5665	0.5548	47.7467	7.7716
	RNN	6.0128	118	15.1849	0.9057	10.1715	184.2713
	GRNN	2.3479	147	0.8385	0.9851	7.7386	2.7496
	LSTM	10.6840	92	41.1876	0.6898	11.4256	485.4107

comparing the results for the 24-hour forecasting horizon task, it can be found that GRNN has the lowest fitting mean MAPE (2.3479%), fastest training time (0.8385 seconds), highest fitting accuracy (0.9851), and the lowest mean MAPE (forecasting horizon) (7.7386%). Whereas, LSTM has the lowest number of training/retraining cycles (92 out of 361). Another important performance index is the maximum time needed for a training/retraining cycle. This information provides a good idea about the computational complexity associated with the training algorithm. If the training/retraining time exceeds the data sampling period or takes an excessively long time, thus any algorithm will fail to apply in a real system. For the developed framework considering the results found in Table 4 for any type of forecasting horizon for any model, the maximum training/retraining time never exceeded the data sampling interval. Maximum training time found for 1 hour, 12 hours, and 24 hours ahead are 101.4567, 938.8834, and 485.4107 seconds using LSTM. And for all the forecasting horizons the minimum of the maximum training time for any specific model was found using GRNN. The times are 0.31288, 1.7043, and 2.7496 seconds for 1 hour, 12 hours, and 24 hours respectively. This provides a clear understanding that considering the time complexity GRNN is clearly ahead of all the algorithms and can be applied in the real system without violating the time-bound constraint. Therefore from the above three forecasting tasks, it can be clearly seen that for most of the indexes GRNN has outperformed the other renowned algorithms. In brief, out of a total of 18 indexes for 3 different horizon forecasting tasks GRNN has shown the best performances for 12 indexes. It should be noted that, during the execution of the proposed algorithm, the hyperparameter set for all the models kept the same as the initial set values. This is because, it is considered that, the proposed algorithm is completely automated and only performs training/retraining based on the correlation analysis while keeping the hyperparameter setting fixed. This reduces the complexity of the algorithm. Under this scenario, the GRNN outperformed other considered models. An additional approach with a change in hyperparameters during the execution cycles may provide better results for other deep learning algorithms, but such chances are low. Because GRNN has only one hyperparameter compared to the hundreds to thousands of hyperparameters of ANN, RNN, and LSTM. Due to the real-time nature of the algorithm, finding the best hyperparameter combinations would be difficult compared to the GRNN. Thus, even if other deep learning models were selected they would perform with limited accuracy due to the above-mentioned reasons.

As GRNN has been judged as the best model for the proposed algorithm thus related results that MAPE (fitting) and mean MAPE (forecasting horizon) have been shown in figure 6. Also, the training/retraining cycle figure according to the occurrence of different input combinations (model) is shown in figure 7. In Figure 7, 1 indicates the training of a model, and 2 indicates the retraining of a model but the retraining does facilitate the improvement of the model in terms of error accuracy. 0 indicates the no training condition, which means the model successfully passed the forecasting accuracy condition. The training cycle figures show that for 1 hour ahead forecasting, data become stable after around 100 days, and afterward, changes happen rarely. Also, those changes are the repetition of the previously occurred models. As the trained models show robustness against changes (error remains within the bounded condition), thus reappearing of the model does not initiate the retraining of the existing model. Hence, for 1 hour ahead forecasting task, no retraining cycles arose after the training found around the 100^{th} days. For 12-hour and 24-hour ahead forecasting, the number of models that appeared was found to be more than the 1-hour ahead forecasting. This is because, as the forecasting horizon increases, finding a stable dataset describing system dynamics becomes difficult. Because the forecasting horizon data structure is not an apriori thus the current data relation found from the correlation analysis is assumed to prevail in the future also. But as time passes by, due to the long range of forecasting horizon those relations are bound to change. Thus more changes in the model or the types of input combinations are found. Although the number of models has increased similar to 1 hour ahead forecasting task, frequent changes in the model appearance become less after 100-150 days. Therefore the number of training/retraining cycles also becomes less evident after the $100 - 150^{th}$ day. Less occurrence of new models and less occurrence of the training/retraining cycles also show the stable data condition and robustness of the trained models against the changes in data size, pattern, and characteristics.

C. IMPACT OF MULTILEVEL CORRELATION IN THE ALGORITHM

As mentioned in the algorithm section, the proposed algorithm is an adaptive online algorithm and can configure the appropriate input variables along with their respective delays by performing the multi-level correlation analysis. Thus with the growth of the data from the environment, the algorithm will encounter different combinations of inputs. Each combination may have a different number of input variables for each combination as shown in figure 8.

An interesting characteristic of the input combinations and change in the number of inputs can be found in figure 8. It can be found that at the initial stage, the number of input variables including the lagged variables is more, even almost close to 150 for a 12 and 24-hour prediction horizon. This is because, in the beginning, the system has high correlations with all the variables including the lagged variable, thus more variables are selected through the correlation analysis. As more data starts gathered as time goes by, correlation with the data becomes more evident, and only variables with the high correlated variables are chosen. This number of input variables decreases over time. This is significant, because more input variables make the learning model complex, and take a long time to train, meaningful model creation becomes difficult, and considering the time constraints, the training also may become impossible. Thus the lower number of input variables is good to develop a meaningful, less complex, and more accurate generalized model. Also from the multilevel correlation analysis, it can be found that only 1 level of operation can not reduce the model, as it doesn't consider the multicollinearity between the variables. Thus if only primary correlation analysis is used, the operator may not have an optimal number of input variables, and the training process will still remain a complex task. Using the secondary and tertiary correlation analysis the number of input variables reduces to its minimum possible value. In a tertiary analysis as mentioned in the intervariable correlation is considered,

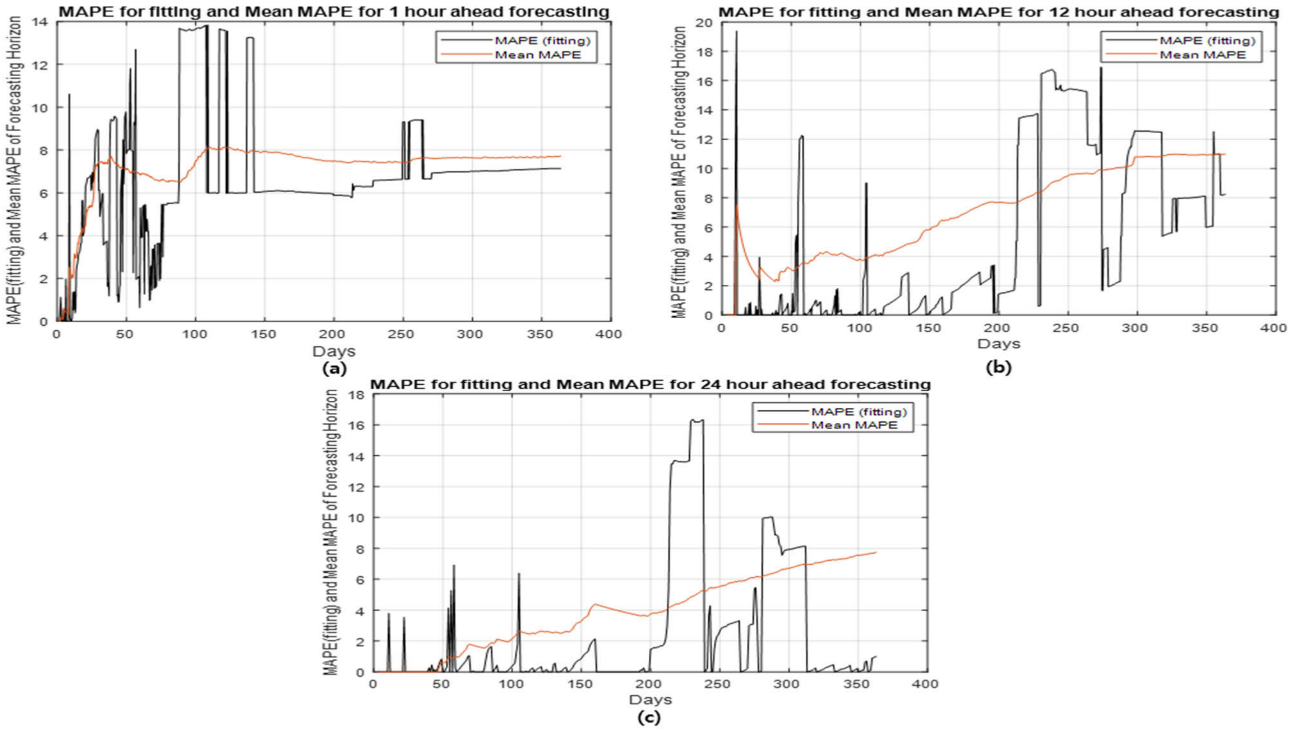

FIGURE 6. Forecasting performance of GRNN for electricity price forecasting MAPE (fitting) and mean hour ahead MAPE for (a) 1 hour (b) 12 hour (c) 24 hour ahead task.

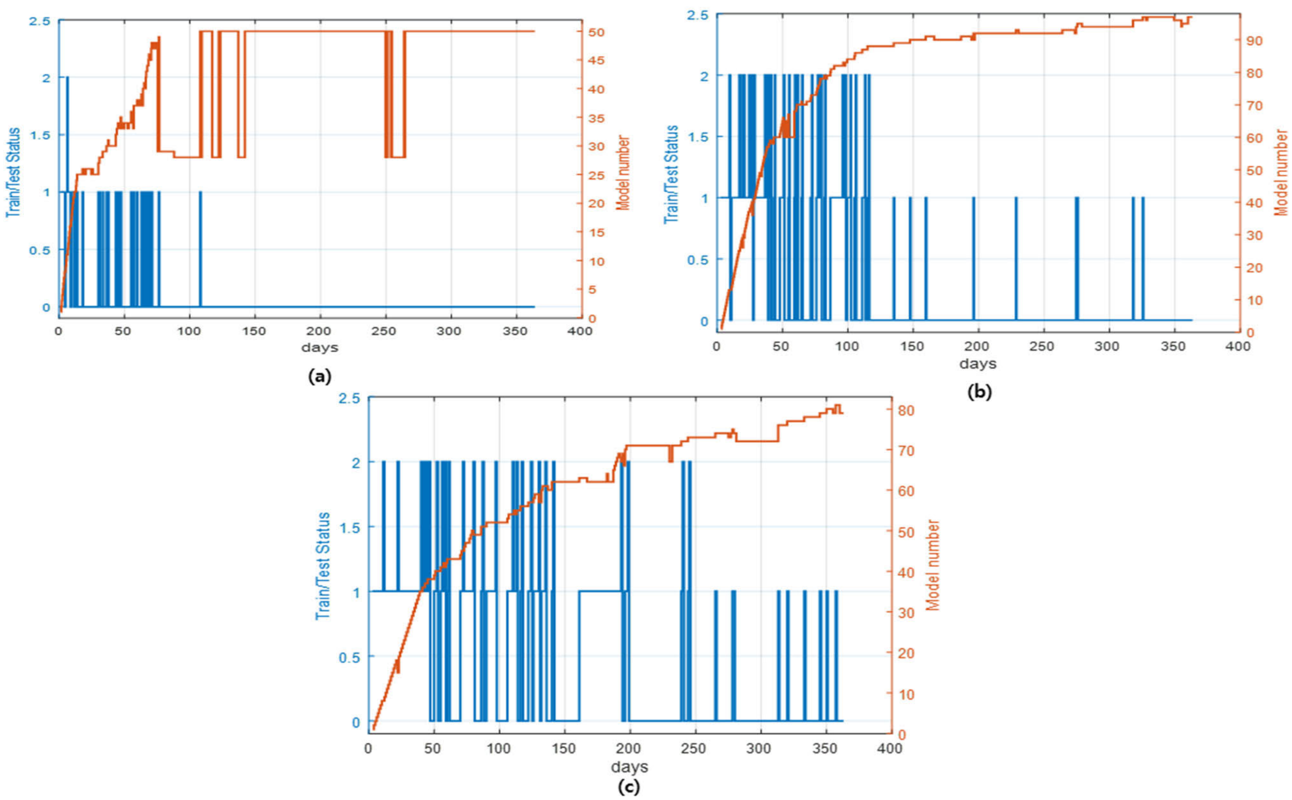

FIGURE 7. Forecasting model training/testing status of GRNN and model appearance for Electricity Price Forecasting (a) 1 hour (b) 12 hour (c) 24 hour ahead task.

therefore will omit the variables within the other variables. For this specific dataset, as found in figure 8, this relation is rarely found and found mostly before the 100^{th} day. Afterward, the number of input variables from secondary and

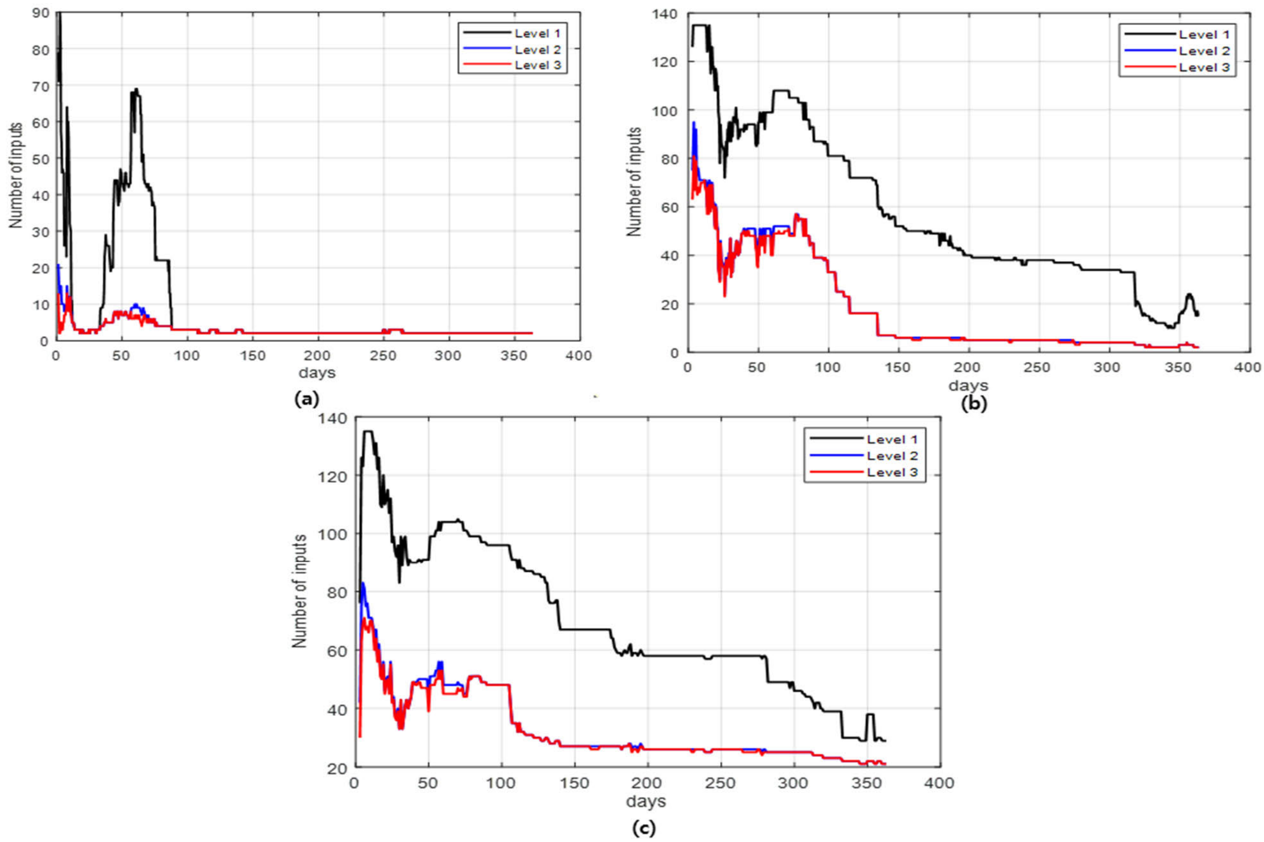

FIGURE 8. Change in number of inputs with respect to the level of correlation analysis for electricity price forecasting for (a) 1 hour (b) 12 hour and (c) 24 hour ahead forecasting horizon.

tertiary becomes the same. but to keep the system robust the tertiary analysis should continue to perform as the future is uncertain.

Also, from figure 7, although many models have occurred only a few of them are dominant modes. That means those modes remain for a longer period of time compared to other modes. For example for 1 hour, ahead the dominant model number was 50 and remained for 560 cycles out of a total of 872 cycles (64%). For the other two forecasting horizons, the dominant model numbers are 92 and 62 respectively. The number of times they have appeared is 133 and 40 out of a total of 722 and 361 cycles respectively. Percentage-wise they have appeared at 18.4% and 11% respectively. The input combination including the number of lagged variables corresponding to those combinations is shown in Table 5. From Table 5 it can be seen that, as the forecasting horizon increases, the total number of input variables including the delayed variables increases. For example, for 1 hour ahead forecasting task the total number of input variables is 2, whereas for 12 hours and 24 hours ahead these numbers are 5 and 27 respectively. The lagged combination of variables also provides insight into the developed model. For 1-hour and 12 hours ahead models the maximum lagged value is 1, but for 24 hours the lagged values across the variables are found between 1-5. Higher lagged combinations put a greater emphasis on the forecasting task. For the 24-hour forecasting model, variable 15 (Electricity Price) provides a lagged combination 5. Variables 6 (System load) and 11 (Price from the same hour the previous day) provide a lagged combination of 4, whereas variables 7 (previous day's average load), 8 (load from the same hour the previous day), and 10 (previous day's average price) provide a lagged combination of 3. Observing the variables carefully it can be understood that these variables with higher lagged combinations contain information about the historical load and price data. This is quite intuitive as the electricity historical price and load information have an impact on electricity price. Thus, it can be said that the developed framework not only provides the model with optimal input combinations but also provides explainable input combinations.

Also, interestingly it can be found that for all three forecasting purposes, variable 2 (dew point temperature) and variable 4 (day of the week) have no delays. Thus these variables can be neglected for long-term forecasting model development. Such decision-making not only effectively reduces the database size but reduces the cost related to sensors also.

Thus from the above discussion in this section, it can be said that using multi-criteria index-based analysis, GRNN along with the proposed algorithm can be selected as the

TABLE 5. Characteristics of the dominant modes for different horizons.

Forecasting Horizon (Hour)	Combination Number (total appearance)	Input Variable (numbers) with delays														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	50 (560)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
12	92 (133)	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0
24	62 (40)	0	0	1	0	0	4	3	3	1	3	4	1	1	1	5

beat ML model for self-supervised adaptive multi-horizon electricity forecasting algorithm.

D. ANALYZING GRNN PERFORMANCE: IMPACT OF FEATURE ENGINEERING, HYPERPARAMETER SENSITIVITY, AND CROSS-DATASET VALIDATION

It was found in section B that GRNN outperformed other models in most indexes. But to show the impact of the proposed algorithm forecasting should be performed by not considering the feature engineering. That provides an insight into why feature engineering should be applied. The choice of hyperparameters affects the performance of the model greatly, thus sensitivity analysis should be performed by varying hyperparameters. Also important to find out the robustness of the GRNN using the developed algorithm for the other dataset. Therefore, to demonstrate the importance of the proposed algorithm experiment without feature engineering is performed. Sensitivity has been analyzed by varying the hyperparameter (sigma) between 0.00009∼0.9. To confirm the robustness, the proposed algorithm has been applied to a new dataset. All these experiments are conducted for an hour ahead forecasting, but the same approach can be considered and can be discussed for the intraday and day ahead or any forecasting horizon. The results are organized in Table 6.

1) WITHOUT FEATURE ENGINEERING

A robust multi-level correlation analysis to develop a robust forecasting module is proposed in this work. On the contrary, to show its importance it is also required to discuss the performance of the forecasting model when feature engineering is not considered. To do that an hour ahead of the electricity price forecasting a single delay for all the variables was considered. To keep the same comparing ground, the hyperparameter and error weightage were kept the same as the model with feature engineering discussed in the previous section. Under these conditions, as feature engineering was not considered, a huge degradation in the performance was found. Without feature engineering, it is found that MAPE for fitting and forecasting horizon becomes really high compared to the result found when feature engineering was applied. The average R^2 was also found to be very small (0.2387) compared to the model with feature engineering (0.9614). Due to the same setting of the error weightage (1000) of the model with feature engineering, the model went under the training cycle only once, but that could not help achieving better forecasting accuracy. Therefore, it can be said that, for a proper forecasting model development, feature engineering must be performed.

2) SENSITIVITY ANALYSIS BY VARYING THE HYPERPARAMETER

GRNN has only one hyperparameter (sigma), thus finding the optimal parameter is easier compared to the other machine/deep learning algorithms. But again, the choice of this hyperparameter has a significant impact on the performance of the model, which can be found in Table 6. In the previous section for the optimal model hyper parameter was set as 0.009. For sensitivity analysis, a range of values centering on this value have been considered. The selected values are increase/decrease in order of 10 from this value. By varying the hyperparameter it was found that the mean MAPE (fitting and forecasting horizon) increases, and the mean R^2 decreases if the sigma deviates from the chosen optimal value (0.009). For MAPE, the increment is larger when sigma is increased (0.09-0.9). For R^2, a large decrease in R^2 is found when sigma is increased. There is no such significant difference in training time is found. However, lowering the hyperparameter can increase the number of training/retraining cycles (63 for sigma with 0.00009). Thus, it can be said that a proper choice of hyperparameter is key to the development of an accurate forecasting model.

3) PERFORMANCE ON A NEW DATASET

To validate the robustness of the proposed framework, the algorithm has been applied to the European wholesale electricity price dataset [59]. The chosen dataset contains hourly electricity price information for Switzerland. The dataset is univariate which means only contains information about the price of electricity and contains data from January 2015- June 23, 2022. But for the robustness analysis, 1 year of data has been chosen from 2015-2016. The approach can be easily adopted in the following years and more. For this dataset, the value of the sigma was kept the same as the previous dataset, and the weightage error parameter has been chosen as 0.1 for better accuracy. Under the above setting, the results found were satisfactory. The mean MAPE for fitting and forecasting was found to be 6.0911%, and 6.2299% respectively, with an average training time of 0.8198 seconds and a maximum training time of 4.8410 seconds which is far less than the data sampling period (1 hour). The average R^2 is also found to be satisfactory with 0.9317. As the dataset is univariable, capturing the dynamics was easier compared to the multivariate dataset. Thus, the above satisfactory results were obtained only by having 17 training/retraining sessions. Therefore, it can be said that the proposed algorithm can successfully perform for other datasets too.

TABLE 6. Performance analysis of GRNN for electricity price forecasting under different conditions.

Condition		Mean MAPE (fitting)	No of Training/ Retraining Cycles	Avg. Training Time (sec)	Avg. R^2	Mean MAPE (forecasting Horizon)	Maximum Training Time in a cycle (sec)
Without feature engineering		20.6241	1	0.4509	0.2387	23.3717	1.8964
Variable hyperparameter (sigma)	0.00009	7.4621	63	1.4711	0.9172	9.6767	3.6457
	0.0009	7.3470	51	0.7737	0.9261	8.8733	1.9938
	0.009	6.7661	51	0.1074	0.9614	9.0834	4.0317
	0.09	15.5211	50	0.2980	0.8064	16.4742	0.8480
	0.9	23.7836	50	0.3076	0.8228	26.4035	0.9786
New dataset		6.0911	17	0.8198	0.9317	6.2299	4.8410

V. CONCLUSION

In this research, a real-time self-supervised adaptive multi-horizon electricity price algorithm has been developed using multilevel correlation analysis and GRNN. Where, MIC has been used for feature engineering for detecting the appropriate relationship between the variables, and GRNN has been used for the accurate forecasting model development. Using the algorithm, different stakeholders (producers and consumers) can optimally forecast the electricity price with an appropriate system describing the model. Observations from the proposed research experiment can be summarized as follows:

1. To demonstrate the effectiveness of the algorithm for real-time electricity price forecasting, real-life 1-year data has been considered. The proposed algorithm with the GRNN-based machine learning model can successfully forecast the electricity price. The proposed algorithm can effectively train, forecast, and relearn within the considered forecasting horizon and outperformed some eminent machine learning algorithms namely ANN, RNN, and LSTM. The superiority of the machine learning model has been found by multi-index based (Mean MAPE (fitting), number of real-time training/retraining cycles, average training time, average R^2, Mean MAPE (forecasting horizon)) analysis for each forecasting model. Among 18 performance indexes across 3 different forecasting horizons in 12 indexes GRNN outperformed other machine and deep learning models. Considering hour ahead to day ahead forecasting horizon, GRNN mean MAPE (fitting) varies between 2.3479%∼6.7661%, and mean MAPE (forecasting horizon) varies between 7.7386%∼10.9911%. The number of training cycles varies between 51∼154, average R^2 varies between 0.9094∼0.9851. The mean and maximum training/retraining time in a cycle varies between 0.1074∼0.8385 seconds, and 0.31288∼2.7496 seconds respectively. All these can be considered as satisfactory results.

2. Additionally the impact of not considering feature engineering, in other words, the importance of considering feature engineering has been validated through experiments. The sensitivity analysis provides insight into the choice of hyperparameters for a better result. Finally, the robustness of the algorithm has been validated by applying to a different dataset (Switzerland).

3. The proposed algorithm is completely self-supervised; thus no human intervention is required during the whole process. Also, due to the use of the GRNN, model development becomes faster, thus the approach can be used for any smaller time horizon.

4. Finding the optimal number of input variables is accomplished using multilevel correlation analysis, which is one of the key features of the algorithm. This results in a machine-learning model that is less complex and requires fewer input variables. The reduced number of input variables can result in the reduction of related costs (database/cloud size, number of sensors/sensor network size, etc.).

As a result of the proposed algorithm, power system operators will not only be able to effectively forecast electricity prices, but they will also be able to maximize their return on investment. In the future, and inclusively, the research could be oriented as follows:

1. Economic profit-based optimization tasks relating to power system operation can be successfully implemented using the algorithm. Self-supervised training of the forecasting algorithm can reduce any uncertainty that arises as a result of the algorithm being real-time in nature.

2. Upon applying the proposed algorithm, as training can be performed only when it is necessary, therefore the forecasting model can be used for intended operation/forecasting most of the time, and therefore more profits can be earned.

The proposed algorithm is open to improvement in the future by utilizing various forefront artificial intelligent models and optimization techniques. Also, the proposed algorithm can be effectively applied to any other forecasting tasks (share market, biological systems, pandemic forecasting) and explore the potential for real-time forecasting-based operation.

REFERENCES

[1] M. Narajewski, "Probabilistic forecasting of German electricity imbalance prices," *Energies*, vol. 15, no. 14, p. 4976, Jul. 2022, doi: 10.3390/en15144976. [Online]. Available: https://www.mdpi.com/1996-1073/15/14/4976

[2] K. Xu, B. Sun, P. Wang, Z. Zhu, and H. Tang, "Electricity market price forecasting for a high renewable penetrated power system via random forest," in *Proc. IEEE/IAS 58th Ind. Commercial Power Syst. Tech. Conf. (I&CPS)*, Las Vegas, NV, USA, May 2022, pp. 1–7, doi: 10.1109/ICPS54075.2022.9773839.

[3] J. Mei, D. He, R. Harley, T. Habetler, and G. Qu, "A random forest method for real-time price forecasting in New York electricity market," in *Proc. IEEE PES Gen. Meeting Conf. Expo.*, National Harbor, MD, USA, Jul. 2014, pp. 1–5, doi: 10.1109/PESGM.2014.6939932.

[4] A. Ciarreta, B. Martinez, and S. Nasirov, "Forecasting electricity prices using bid data," *Int. J. Forecasting*, vol. 39, no. 3, pp. 1253–1271, Jul. 2023, doi: 10.1016/j.ijforecast.2022.05.011.

[5] B. Uniejewski and R. Weron, "Regularized quantile regression averaging for probabilistic electricity price forecasting," *Energy Econ.*, vol. 95, Mar. 2021, Art. no. 105121, doi: 10.1016/j.eneco.2021.105121.

[6] A. Yousaf, R. M. Asif, M. Shakir, A. U. Rehman, F. Alassery, H. Hamam, and O. Cheikhrouhou, "A novel machine learning-based price forecasting for energy management systems," *Sustainability*, vol. 13, no. 22, p. 12693, Nov. 2021, doi: 10.3390/su132212693.

[7] J. Deng, W. Song, and E. Zio, "A discrete increment model for electricity price forecasting based on fractional Brownian motion," *IEEE Access*, vol. 8, pp. 130762–130770, 2020, doi: 10.1109/ACCESS.2020.3008797.

[8] Z. Shao, Q. Zheng, S. Yang, F. Gao, M. Cheng, Q. Zhang, and C. Liu, "Modeling and forecasting the electricity clearing price: A novel BELM based pattern classification framework and a comparative analytic study on multi-layer BELM and LSTM," *Energy Econ.*, vol. 86, Feb. 2020, Art. no. 104648, doi: 10.1016/j.eneco.2019.104648.

[9] A. Rafi, S. Mujeeb, and N. Javaid, "ERBFNN based electricity load and price forecasting," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Islamabad, Pakistan, Dec. 2019, pp. 191–1915, doi: 10.1109/FIT47737.2019.00044.

[10] T. Ashfaq and N. Javaid, "Short-term electricity load and price forecasting using enhanced KNN," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Islamabad, Pakistan, Dec. 2019, pp. 266–2665, doi: 10.1109/FIT47737.2019.00057.

[11] J. Pórtoles, C. González, and J. Moguerza, "Electricity price forecasting with dynamic trees: A benchmark against the random forest approach," *Energies*, vol. 11, no. 6, p. 1588, Jun. 2018, doi: 10.3390/en11061588.

[12] O. Abedinia, N. Amjady, and H. Zareipour, "A new feature selection technique for load and price forecast of electrical power systems," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 62–74, Jan. 2017, doi: 10.1109/TPWRS.2016.2556620.

[13] T. Pinto, T. M. Sousa, and Z. Vale, "Dynamic artificial neural network for electricity market prices forecast," in *Proc. IEEE 16th Int. Conf. Intell. Eng. Syst. (INES)*, Jun. 2012, pp. 311–316, doi: 10.1109/INES.2012.6249850.

[14] Z. Shao, S. Yang, F. Gao, K. Zhou, and P. Lin, "A new electricity price prediction strategy using mutual information-based SVM-RFE classification," *Renew. Sustain. Energy Rev.*, vol. 70, pp. 330–341, Apr. 2017, doi: 10.1016/j.rser.2016.11.155.

[15] E. Abramova and D. Bunn, "Forecasting the intra-day spread densities of electricity prices," *Energies*, vol. 13, no. 3, p. 687, Feb. 2020, doi: 10.3390/en13030687.

[16] S. S. Torbaghan, A. Motamedi, H. Zareipour, and L. A. Tuan, "Medium-term electricity price forecasting," in *Proc. North Amer. Power Symp. (NAPS)*, Champaign, IL, USA, Sep. 2012, pp. 1–8, doi: 10.1109/NAPS.2012.6336424.

[17] T. Jonsson, P. Pinson, H. A. Nielsen, H. Madsen, and T. S. Nielsen, "Forecasting electricity spot prices accounting for wind power predictions," *IEEE Trans. Sustain. Energy*, vol. 4, no. 1, pp. 210–218, Jan. 2013, doi: 10.1109/TSTE.2012.2212731.

[18] K. He, Y. Xu, Y. Zou, and L. Tang, "Electricity price forecasts using a curvelet denoising based approach," *Phys. A, Stat. Mech. Appl.*, vol. 425, pp. 1–9, May 2015, doi: 10.1016/j.physa.2015.01.012.

[19] M. Bernardi and F. Lisi, "Point and interval forecasting of zonal electricity prices and demand using heteroscedastic models: The IPEX case," *Energies*, vol. 13, no. 23, p. 6191, Nov. 2020, doi: 10.3390/en13236191.

[20] W.-M. Lin, H.-J. Gow, and M.-T. Tsai, "Electricity price forecasting using enhanced probability neural network," *Energy Convers. Manage.*, vol. 51, no. 12, pp. 2707–2714, Dec. 2010, doi: 10.1016/j.enconman.2010.06.006.

[21] T. V. D. Heijden, J. Lago, P. Palensky, and E. Abraham, "Electricity price forecasting in European day ahead markets: A greedy consideration of market integration," *IEEE Access*, vol. 9, pp. 119954–119966, 2021, doi: 10.1109/ACCESS.2021.3108629.

[22] J. Lago, F. De Ridder, P. Vrancx, and B. De Schutter, "Forecasting day-ahead electricity prices in Europe: The importance of considering market integration," *Appl. Energy*, vol. 211, pp. 890–903, Feb. 2018, doi: 10.1016/j.apenergy.2017.11.098.

[23] G. Memarzadeh and F. Keynia, "Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm," *Electr. Power Syst. Res.*, vol. 192, Mar. 2021, Art. no. 106995, doi: 10.1016/j.epsr.2020.106995.

[24] P.-H. Kuo and C.-J. Huang, "An electricity price forecasting model by hybrid structured deep neural networks," *Sustainability*, vol. 10, no. 4, p. 1280, Apr. 2018, doi: 10.3390/su10041280.

[25] L. Tschora, E. Pierre, M. Plantevit, and C. Robardet, "Electricity price forecasting on the day-ahead market using machine learning," *Appl. Energy*, vol. 313, May 2022, Art. no. 118752, doi: 10.1016/j.apenergy.2022.118752.

[26] Z. Zhang and M. Wu, "Real-time locational marginal price forecasting using generative adversarial network," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Tempe, AZ, USA, Nov. 2020, pp. 1–6, doi: 10.1109/SmartGridComm47815.2020.9302938.

[27] H. Ur-Rehman, S. Mujeeb, and N. Javaid, "DCNN and LDA-RF-RFE based short-term electricity load and price forecasting," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Islamabad, Pakistan, Dec. 2019, pp. 71–715, doi: 10.1109/FIT47737.2019.00023.

[28] M. Ghayekhloo, R. Azimi, M. Ghofrani, M. B. Menhaj, and E. Shekari, "A combination approach based on a novel data clustering method and Bayesian recurrent neural network for day-ahead price forecasting of electricity markets," *Electr. Power Syst. Res.*, vol. 168, pp. 184–199, Mar. 2019, doi: 10.1016/j.epsr.2018.11.021.

[29] M. Afrasiabi, J. Aghaei, S. Afrasiabi, and M. Mohammadi, "Probability density function forecasting of electricity price: Deep Gabor convolutional mixture network," *Electr. Power Syst. Res.*, vol. 213, Dec. 2022, Art. no. 108325, doi: 10.1016/j.epsr.2022.108325.

[30] H. Yang and K. R. Schell, "QCAE: A quadruple branch CNN autoencoder for real-time electricity price forecasting," *Int. J. Electr. Power Energy Syst.*, vol. 141, Oct. 2022, Art. no. 108092, doi: 10.1016/j.ijepes.2022.108092.

[31] D. Wang, I. Gryshova, M. Kyzym, T. Salashenko, V. Khaustova, and M. Shcherbata, "Electricity price instability over time: Time series analysis and forecasting," *Sustainability*, vol. 14, no. 15, p. 9081, Jul. 2022, doi: 10.3390/su14159081.

[32] J. Dong, X. Dou, A. Bao, Y. Zhang, and D. Liu, "Day-ahead spot market price forecast based on a hybrid extreme learning machine technique: A case study in China," *Sustainability*, vol. 14, no. 13, p. 7767, Jun. 2022, doi: 10.3390/su14137767.

[33] C. Xiao, D. Sutanto, K. M. Muttaqi, M. Zhang, K. Meng, and Z. Y. Dong, "Online sequential extreme learning machine algorithm for better predispatch electricity price forecasting grids," *IEEE Trans. Ind. Appl.*, vol. 57, no. 2, pp. 1860–1871, Mar. 2021, doi: 10.1109/TIA.2021.3051105.

[34] M. F. Azam and M. S. Younis, "Multi-horizon electricity load and price forecasting using an interpretable multi-head self-attention and EEMD-based framework," *IEEE Access*, vol. 9, pp. 85918–85932, 2021, doi: 10.1109/ACCESS.2021.3086039.

[35] M. Massaoudi, H. Abu-Rub, S. S. Refaat, A. A. Al-Kuwari, and T. Huang, "Locational marginal electricity price forecasting-based self-attention mechanism and simulated annealing optimizer using big data," in *Proc. 10th Int. Conf. Renew. Energy Res. Appl. (ICRERA)*, Istanbul, Turkey, Sep. 2021, pp. 391–396, doi: 10.1109/ICRERA52334.2021.9598604.

[36] Z. Zhang and M. Wu, "Locational marginal price forecasting using convolutional long-short term memory-based generative adversarial network," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Washington, DC, USA, Jul. 2021, pp. 1–5, doi: 10.1109/PESGM46819.2021.9637861.

[37] K. Bhatia, R. Mittal, J. Varanasi, and M. M. Tripathi, "An ensemble approach for electricity price forecasting in markets with renewable energy resources," *Utilities Policy*, vol. 70, Jun. 2021, Art. no. 101185, doi: 10.1016/j.jup.2021.101185.

[38] L. Sun, L. Li, B. Liu, and S. Saeedi, "RETRACTED: A novel day-ahead electricity price forecasting using multi-modal combined integration via stacked pruning sparse denoising auto encoder," *Energy Rep.*, vol. 7, pp. 2201–2213, Nov. 2021, doi: 10.1016/j.egyr.2021.04.009.

[39] A. K. Srivastava, A. S. Pandey, R. M. Elavarasan, U. Subramaniam, S. Mekhilef, and L. Mihet-Popa, "A novel hybrid feature selection method for day-ahead electricity price forecasting," *Energies*, vol. 14, no. 24, p. 8455, Dec. 2021, doi: 10.3390/en14248455.

[40] M. Yi, W. Xie, and L. Mo, "Short-term electricity price forecasting based on BP neural network optimized by SAPSO," *Energies*, vol. 14, no. 20, p. 6514, Oct. 2021, doi: 10.3390/en14206514.

[41] H. Yang and K. R. Schell, "HFNet: Forecasting real-time electricity price via novel GRU architectures," in *Proc. Int. Conf. Probabilistic Methods Appl. Power Syst. (PMAPS)*, Liege, Belgium, Aug. 2020, pp. 1–6, doi: 10.1109/PMAPS47429.2020.9183697.

[42] A. Brusaferri, M. Matteucci, D. Ramin, S. Spinelli, and A. Vitali, "Probabilistic day-ahead energy price forecast by a mixture density recurrent neural network," in *Proc. 7th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, vol. 1, Prague, Czech Republic, Jun. 2020, pp. 523–528, doi: 10.1109/CoDIT49905.2020.9263898.

[43] J. Zhang, Z. Tan, and Y. Wei, "An adaptive hybrid model for short term electricity price forecasting," *Appl. Energy*, vol. 258, Jan. 2020, Art. no. 114087, doi: 10.1016/j.apenergy.2019.114087.

[44] W. Yang, J. Wang, T. Niu, and P. Du, "A novel system for multi-step electricity price forecasting for electricity market management," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 106029, doi: 10.1016/j.asoc.2019.106029.

[45] W. Qiao and Z. Yang, "Forecast the electricity price of U.S. using a wavelet transform-based hybrid model," *Energy*, vol. 193, Feb. 2020, Art. no. 116704, doi: 10.1016/j.energy.2019.116704.

[46] K. Wang, C. Xu, Y. Zhang, S. Guo, and A. Y. Zomaya, "Robust big data analytics for electricity price forecasting in the smart grid," *IEEE Trans. Big Data*, vol. 5, no. 1, pp. 34–45, Mar. 2019, doi: 10.1109/TBDATA.2017.2723563.

[47] D. Bissing, M. T. Klein, R. A. Chinnathambi, D. F. Selvaraj, and P. Ranganathan, "A hybrid regression model for day-ahead energy price forecasting," *IEEE Access*, vol. 7, pp. 36833–36842, 2019, doi: 10.1109/ACCESS.2019.2904432.

[48] R. K. Agrawal, F. Muchahary, and M. M. Tripathi, "Ensemble of relevance vector machines and boosted trees for electricity price forecasting," *Appl. Energy*, vol. 250, pp. 540–548, Sep. 2019, doi: 10.1016/j.apenergy.2019.05.062.

[49] O. Karabiber and G. Xydis, "Electricity price forecasting in the Danish day-ahead market using the TBATS, ANN and ARIMA methods," *Energies*, vol. 12, no. 5, p. 928, Mar. 2019, doi: 10.3390/en12050928.

[50] A. Naz, M. Javed, N. Javaid, T. Saba, M. Alhussein, and K. Aurangzeb, "Short-term electric load and price forecasting using enhanced extreme learning machine optimization in smart grids," *Energies*, vol. 12, no. 5, p. 866, Mar. 2019, doi: 10.3390/en12050866.

[51] M. Zahid, F. Ahmed, N. Javaid, R. Abbasi, H. Z. Kazmi, A. Javaid, M. Bilal, M. Akbar, and M. Ilahi, "Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids," *Electronics*, vol. 8, no. 2, p. 122, Jan. 2019, doi: 10.3390/electronics8020122.

[52] Wikipedia Contributors. *Point-Biserial Correlation Coefficient*. Accessed: Dec. 12, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Point-biserial_correlation_coefficient&oldid=983535923

[53] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011, doi: 10.1126/science.1205438.

[54] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, 1991, doi: 10.1109/72.97934.

[55] Wikipedia Contributors. (2022). *General Regression Neural Network—Wikipedia, The Free Encyclopedia*. Accessed: Jun. 19, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=General_regression_neural_network&oldid=1079466016

[56] F. Xie, M. Huang, W. Zhang, and J. Li, "Research on electric vehicle charging station load forecasting," in *Proc. Int. Conf. Adv. Power Syst. Autom. Protection*, Beijing, China, Oct. 2011, pp. 2055–2060, doi: 10.1109/APAP.2011.6180772.

[57] A. Deoras. (2022). *Electricity Load and Price Forecasting Webinar Case Study*. Accessed: Dec. 27, 2022. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/28684-electricity-load-and-price-forecasting-webinar-case-study

[58] J. Heaton. (2017). *The Number of Hidden Layers*. [Online]. Available: https://www.heatonresearch.com/2017/06/01/hidden-l

[59] Ember Climate Data Catalogue. *European Wholesale Electricity Price Data*. Accessed: Feb. 20, 2024. [Online]. Available: https://ember-climate.org/data-catalogue/european-wholesale-electricity-price-data/

MUHAMMAD AHSAN ZAMEE (Member, IEEE) was born in Dhaka, Bangladesh, in 1987. He received the B.Sc. degree in electrical, electronic and communication engineering from the Military Institute of Science and Technology, Dhaka, in 2009, the M.Sc. degree in electrical and electronic engineering from the Islamic University of Technology, Gazipur, Bangladesh, in 2015, and the Ph.D. degree in electrical and computer engineering from Inha University, Incheon, South Korea, in 2020. He was a Postdoctoral Fellow with the Inha Power System Laboratory, Inha University. He is currently a Research Associate with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. His research interests include the application of artificial intelligence and nature-inspired optimizations in power and control systems.

YEONGSANG LEE was born in Republic of Korea, in 1996. He received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea in 2020, where he is currently pursuing the M.S. degree in electrical engineering. His research interests include microgrids and peer-to-peer trading.

DONGJUN WON (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 1998, 2000, and 2004, respectively. He was a Postdoctoral Fellow with the APT Center, University of Washington, Seattle. He is currently a Professor with the School of Electrical Engineering, Inha University, Incheon, South Korea. His research interests include power quality, microgrids, renewable energy, electric vehicle, and energy storage systems.