Received April 11, 2022, accepted April 22, 2022, date of publication April 29, 2022, date of current version May 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3171270

A Performance Comparison of Machine Learning Algorithms for Load Forecasting in Smart Grid

THAMER ALQUTHAMI[1], (Member, IEEE), MUHAMMAD ZULFIQAR[2], MUHAMMAD KAMRAN[2], AHMAD H. MILYANI[1], AND MUHAMMAD BABAR RASHEED[3,4], (Senior Member, IEEE)

[1]Electrical and Computer Engineering Department, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[2]Department of Electrical Engineering, University of Engineering and Technology Lahore, Lahore 54890, Pakistan
[3]Department of Electronics and Electrical Systems, The University of Lahore, Lahore 54000, Pakistan
[4]Escuela Politécnica Superior, Universidad de Alcalá, Alcalá de Henares, 28805 Madrid, Spain

Corresponding authors: Muhammad Babar Rasheed (muhammad.rasheed@uah.es) and Thamer Alquthami (tquthami@kau.edu.sa)

This work was supported in part by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia, under Grant RG-34-135-42; and in part by the European Union Horizon 2020 Research and Innovation Program through the Marie Sklodowska-Curie under Grant 754382, GOT ENERGY TALENT.

ABSTRACT With the rapid increase in the world's population, the global electricity demand has increased drastically. Therefore, it is required to adopt efficient energy management mechanisms. Since the energy consumption trends are rather dynamic. Therefore, precise energy demand estimation and short and/or long-term forecasting results with higher accuracy are required to develop the optimization and control mechanism. Consequently, the machine learning (ML) techniques along with distributed demand response programs are being adopted to predict the future energy demand requirement with satisfactory results. In this paper, different state-of-the-art ML algorithms such as logistic regression (LR), support vector machines (SVM), naive Bayes (NB), decision tree classifier (DTC), K-nearest neighbor (KNN), and neural networks (NNs), have been implemented to analyze their performance. The main objective of this paper is to present a comparative analysis of ML algorithms for short-term load forecasting (STLF) regarding accuracy and forecast error. Based on the implementation and analysis, we have identified that, among other algorithms, the DTC provides comparatively better results. Therefore, we devised the enhanced DTC (EDTC) by integrating fitting function, loss function, and gradient boosting in DTC mathematical model for fine-tuning the control variables. The implementation results show that the proposed EDTC algorithm provides better forecast results (i.e., 99.9 % recall, 100% F1, 100% precision, 99.21 % training accuracy, and 99.70% testing accuracy.)

INDEX TERMS Smart grids, electric load forecasting, machine learning algorithms, logistic regression, decision tree.

I. INTRODUCTION

The tremendous increase in the world economy and population, along with rapid development in urbanization, can enhance the need for energy usage in the years ahead. Electricity, a vital energy source, can be generated from various sources, including water, wind, solar cells, fossil fuels, and thermal & nuclear reactors. Furthermore, as our population grows and progresses, the demand for electricity rises, prompting the need for increased energy production. The essential concerns in energy management (EM) are electricity generation, transmission, and distribution. The electric grid (EG) is a well-known interconnected network that connects customers to energy providers and transports energy from producer to consumer. It consists of power plants that generate electricity, substations that regulate electrical voltage based on usage, transmission lines (the transporter of electricity), and distribution lines that link customers [1]. As described above, classical EGs use a centralized network with thousands of units. Enhancing the EG load introduces the potential for generating overhead, resulting in power quality issues. As a result, the installation of new plants becomes necessary. On the other hand, these grids lack a reliable forecast system for predicting intermittent power failures,

The associate editor coordinating the review of this manuscript and approving it for publication was Zhouyang Ren.

their reasons, reaction latency, memory space, and resource utilization [2]–[5].

Scientists determined that the current electrical power system (PS) has remained unchanged for several decades [6]–[9]. With the population growth, there is a massive electricity demand. The shortcomings of the traditional PS include a lack of visibility, mechanical switches, which results in a slower response time, and a lack of monitoring and power control. Metamorphoses in climatic conditions, component failure, the demand for energy, population growth, demand for fossil fuels, a drop in electric power output, a shortage of energy storage, unilateral communication, and various other issues all contribute to the need for new grid technology. Hence, a new grid framework is crucial to handle such issues. The smart grid (SG), a next-generation energy infrastructure, appears as a critical technology to meet high-priority demands and enhance modern human life quality [10]. According to a comparison, the conventional EG provides one-way communication confined to energy users, whereas SG provides vast two-way communication. In the traditional EG, power quality concerns are resolved slowly; however, in the case of SG, a rapid self-healing facility is proffered. The traditional EG system is more vulnerable to cyber-attacks and natural calamities, responding considerably more slowly. The SG, on the other hand, is significantly more resistant to natural disasters and cyber threats. The conventional EG system responds gradually to system disturbances, whereas the SG automatically detects and responds to problems and has a far lower impact on customers. Power flow control is completely limited in the traditional EG system; however, it is immense in SG [11], [12].

Both classical (time-series) and computational intelligence methods are used in the literature for ELF [13]. Both approaches have their drawbacks. The limitations of the previous classical methods in dealing with non-linear data are cited. On the other hand, computational intelligence approaches are chastised for their handcrafted features, restricted learning ability, ineffectual learning, erroneous appraisal, and insufficient motivating importance. Despite this, specific current ML models are used for ELF, which partially solves the problems mentioned above and improves performance due to an innovative technique [14]. Since low prediction accuracy results in significant economic loss, a proper system must overcome the concerns mentioned above. A 1% increase in forecast error results in a 0.01 billion rise in overall utility costs. As a result, electric utility companies are attempting to design an STLF model that is fast, accurate, robust, and accessible. Furthermore, explicit forecasting might help spot failures and ensure reliable grid operation. Aside from forecasting accuracy, forecasting stability, or the ability of the forecasting model to maintain a constant degree of forecasting accuracy, is also critical for a forecasting model to ensure the safe operation of the energy system. However, forecasting stability is frequently overlooked based on previously proposed forecasting models [15]. Artificial intelligence and its subset ML algorithms [16]–[19] can be used to predict faults in SG, which aids in the implementation of precautionary measures The most major technical challenge, predicting an SG's stability, is considered in this study because it determines the effective energy transfer in over 60% of the SGs. [20]. The SG environment with AI technology is depicted in Fig. 1.

A. MOTIVATIONS

With the rapid increase in the world's population and industrial revolutions, the global electricity demand has increased drastically [21]. Consequently, the increased load demand is fulfilled by combining traditional and distributed energy generation technologies such as photovoltaic energy, energy storage system, and electric vehicles [19]. However, their integration into the primary grid or residential premises has posed severe problems regarding prediction and forecasting. This is also due to dynamic load demand and consumption trends [22]. Furthermore, numerous works are being presented involving load demand management through active or passive involvement of prosumers. The main objective was to reduce the energy consumption cost and customer dissatisfaction without considering SG stability and control [23]. The first solution is to invest in the generation and transmission infrastructure. At the same time, the second solution is to manage the demand requirements through customer engagement and ML-based optimal control strategies [24]. This former is a long-term solution and requires more investment cost. However, the latter is a short-term solution without more investment costs on infrastructure [25]. However, to optimally utilize the full potential of ML algorithms for stability and control, it is a prior requirement to identify the most suitable and reliable algorithm for prediction, forecasting, and estimation before any decision making. In this context, the present work has explored different state-of-the-art ML algorithms such as LR, SVM, NB, DTC, KNN, and NNs, for STLF. The main focus of this paper is to identify the best suitable algorithms for STLF through implementation and comparison. Through implementation, we determine that the DTC algorithm has provided significant results compared to other counterpart algorithms. Furthermore, recent research has concentrated on feature engineering and classical methods such as decision tree (DT), ARIMA, and ANN. Despite overfitting issues, the DT outperforms training and performs poorly in forecasting, whereas ANN has low generalization power. It is not an easy task to control the rate of convergence. Furthermore, these learning models are not ideal for large amounts of data because their performance degrades as the size of the data increases. Moreover, the optimization module must be integrated with the forecaster to achieve exceptional performance. A DT was used in several ML and data mining tasks as a classifier. In this study, we discuss several recent works about the DT in Table 2. The selection and adjustment of hyper-parameters have a significant impact on the forecast accuracy of ML systems [23]. Therefore, optimal and accurate hyper-parameters tuning is a substantial challenge with ML models [26]. These individual/single ML algorithms are not helpful in all aspects (accuracy, convergence rate, stability) because every individual approach has its imperfections and inherent limitations [27], [28]. They depend on random weights, biases [29], thresholds [30], and hyper-parameters tuning [31]. These problems influence ELF and cause unstable performance. These shortcomings deprive individual methods of achieving all objectives (accuracy, convergence rate, stability) simultaneously [32]–[34]. However, by properly tuning hyper-parameters and optimizing random weight and bias initialization, the DTC model can be assumed to be promising and effective in improving forecast accuracy [22], [35]–[37]. Furthermore, none of these models evaluated accuracy, stability, and convergence rate simultaneously [38], [39]. As a result, a robust, enhanced model is required to address the shortcomings of existing models while simultaneously improving forecast accuracy and stability with a fast convergence rate.

B. REAL CONTRIBUTIONS

With this motivation, in this work, a novel, robust, enhanced forecasting algorithm is developed by integrating fitting function, loss function, and gradient boosting analysis with DTC, called the EDTC forecasting model. The novelty and significant technical contributions are highlighted below.

- The state-of-the-art ML algorithms such as SVM, KNN, NN, DTC, and LR are compared for predicting the forecast accuracy. While DTC gives more effective and efficient results with comparatively high accuracy, good speed, and low memory usage among these classifiers.
- Since hyperparameters highly affect the stable performance of ML algorithms in ELF. It is a challenge to select and modify these parameters for accurate and stable performance. To overcome the hard-to-tune hyper parameters problem of the ML algorithm, we proposed a novel and enhanced DTC EDTC) by optimizing random weights and bias initialization of the DTC. The proposed EDTC improves the accuracy by adding fitting function, loss function, and gradient boosting in DTC mathematical model for fine-tuning the control variables.
- Experiments on real datasets acquired from the New York ISO (NYISO) are performed to validate the effectiveness of the proposed methodology. Experimental results show that our proposed EDTC model outperforms other benchmark ML models such as SVM, KNN, NN, LR, and DTC in accuracy, stability, and convergence rate.

1) NOVELTY ASPECT

This paper has compared different ML algorithms from a load forecasting perspective. Firstly, SVM, KNN, LR, ANN, and DTC are implemented to analyze the results in terms of features and classification parameters, as depicted in the Table 1. Results show that, among other algorithms, the DTC gives more accurate and efficient output with comparatively high speed. One of the main reasons for the high speed and accuracy of DTC is the optimal memory utilization to store the rule-set in the form of smaller trees. Furthermore, the classification process in DTC has occupied lower memory than other techniques because it generates fewer rules to develop the optimal output. The accuracy is also higher as the error rate is low on unseen cases due to the developing pruned trees. Therefore, we have identified that DTC could generate more accurate results if the control parameters are further optimized based on comparative analysis. This paper has proposed an enhanced EDTC to perform feature selection, cross-validation, reduced error pruning, and model complexity to optimize the error ratio. The feature selection is used for dimensionality reduction. Where it minimizes the attribute space of the feature set, it is also analyzed that the classification accuracy can be further increased if the model complexity is increased. Therefore, by applying the reduced error pruning technique, the overfitting problem of the DTC is solved. Results show that the proposed EDTC has improved the accuracy by 1-2%. The classification error rate is also reduced compared to the existing algorithms. Eqs. 20-24 in the revised version describe the mathematical formulation that was used to modify the DTC algorithm. Whereas in DTC, only the Eqs. 18 and 19 are used to handle the control variable, causing reduce accuracy and slow convergence.

C. PAPER ORGANIZATION

The following is how the paper's material is structured. Section I provides a concise introduction and motivation for ELF and ML approaches for ELF difficulties. Section II is devoted to studying the existing literature on the application of ML approaches to ELF. In section III, ML-models are depicted. Section IV characterizes the devised methodologies. Whereas section V confers the simulation results and discussions. Section VI explains the critical analysis. Section VII concludes by clarifying the research and its outcomes and concluding remarks.

II. LITERATURE SURVEY

This section contains an overview of various research solutions, methodology, outcomes, and limitations of existent works on SGs. The SG, which will replace the definitive energy grid, will be capable of two-way communication. To distribute the electricity transferred from various sources, a convoluted system is employed with the assistance of EVs [41]. Since it regulates its features to maximize performance, this adds new overhead to the SG modeling. As a result, stability, robustness, efficiency, and dependability must be monitored frequently under various operating conditions. Researchers have employed ML methods such as LR, KNN, SVM, ANN, random forest (RF), ridge regression (RR), gradient boosting (GB), extra trees regressor (ETR), stochastic gradient descent (SGD), and gradient boosting (GB) to evaluate load in SG.

SGs are transitioning to demand-based power supply services for customers. As a result, forecasting consumer load is obligatory. An endeavor is made to confine if the

TABLE 1. Characteristics comparison between proposed and existent ML algorithms.

Characteristic	EDTC	LR	DTC	SVM	NN
Learning type	Eagerly learner	Eager learner	Lazy	Eager learner	Eager learner
Speed	Very fast	Fast	Slow	Fast with active	Slow
Accuracy	Good in many domains	Good in many domains	High	Significantly high	-
Scalability	Efficient for small dataset	efficient for large datasets	-	-	Low
Interpretability	Good	Good	Average	-	Worse
Transparency	Rules	Not rules (Black box)	Rules	Black box	Black box
Missing Val interpretation	Missing Values	Missing Values	Missing Values	Sparse data	-

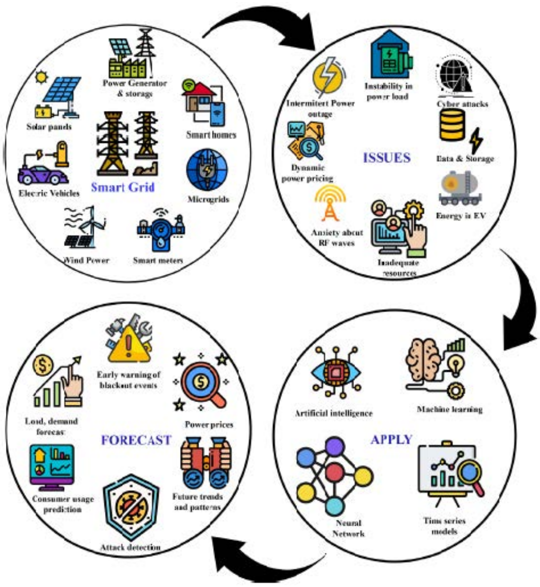


FIGURE 1. AI technology embedded into the SG context.

existent short-term load forecasting (STLF) framework or anthropological-structural data accurately foretells individual consumer household load [42]. An STLF framework was developed based on anthropological structural data from the residential consumers to identify the optimal LF framework for an individual load. The devised model can forecast deviated loads using a particular instance at different time series. The researchers used back-propagation (BP), NN, and SVM to predict the proposed STLF framework. According to the results, devised STLF is 7% more accurate than SLTF and reduces inaccuracy by 60%. The study affirmed the enhancement of the SLTF model by employing anthropological structural data correctness. The week ahead household data were used to learn an ANN to forecast the energy consumption hourly for the subsequent day.

Hernandez *et al.* devised an ELF based ANN technique in SGs that involved three main stages: segmentation using K-means classification, a self-organizing map (SOM) approach uses pattern recognition, and demand forecasting (DF) in individual clusters [43]. Realtime data from a Spanish corporation was used to validate the ANN model. Periodic values were used to train the model (weekahead and month ahead). This framework outperformed benchmarks based on generalized regression NN and radial basis function NN. The sustainability of the SG is reliant on the ability

TABLE 2. DT algorithms based accuracy in latest literature survey.

References	Techniques used	Accuracy achieved
[35]	DT, KNN, LR, SVM, and NB	DT: 98.93%, KNN: 98.23%, LR: 92.13%, SVM: 89.12%, and NB: 80.12%
[36]	DT	DT: 89.14%
[37]	DT and KNN	DT: 98.12%, and KNN: 97%
[40]	DT (XGboost) and RF	DT: 83.12% and RF: 81.78%
[22]	DT and KNN	DT: 98.12% and KNN: 97.98%
Proposed	SVM, KNN, NN, LR, and EDTC	EDTC:99.7%, KNN: 94.50%, NN: 98.9%, LR: 98.5%, and SVM: 97.5%

to generate uninterrupted electricity based on usage. Chen and Ahmad used three diverse ML frameworks for MTLF and LTLF in the SG [44]. They use nonlinear ANN comprising of ada boost, multivariate linear regression (MLR), and auto-regressive exogenous multivariate inputs framework (AEMIF). The researchers diverged the load into three intervals based on aggregated exhaustive consumption metrics: one month ahead, seasonal perspective, and one year ahead. The models enhanced predictability while accurately defining energy differences, modifications, and coming energy prediction prospects. Because of its superior prediction ability, the Ada boost model outperformed the other models.

Khan *et al.* presented a comprehensive review of dynamic pricing (DP) and EL in the SG [45]. The study focused on the relationship between real-time pricing (RTP), time of use (TOU), and critical peak price (CPP). Computational and AI models were presented as procedures to LF. In AI frameworks, ANN, RNN, auto-regressive integrated moving average (ARIMA)-SVM, generalized regression neural networks (GRNN), SVM, wavelet transformation (WT)-ANN, wavelet transformation error correction (WTEC)-ANN, probabilistic NN (PNN), Expert systems, and fuzzy logic (FL) were used. According to the study results, forecasting algorithms based on AI outperformed other stochastic approaches.

Muhammad *et al.* recently completed a survey on projecting yields of photo-voltaic (PV) [45]. Because the overwhelming majority of studies endeavored to predict PV output using conventional, analytic, and AI approaches. This study discovered that ANN might provide more accurate forecasts than traditional and quantitative models. According to the survey, the precision of any prediction technique varies depending on the day, seasonal variance, infusion factors, and other appraisal matrices. In the recent SG [45], Muhammad and Abbas explored AI-ELF algorithms. The performance of these frameworks were determined by its structure, input factors, activation function, and ML algorithms used for training and creating predictive errors. It was discovered that the BP training method was often used to train NN, despite the fact that it posed various problems. ANN, on the other hand, was better suited for ELF and produced better results than BP. It was eventually able to confirm that integrated techniques had more significant effects.

It is well known that regular power outages cause a slew of catastrophic failures. To address this issue in SG, improved surveillance of blackout situations has become necessary. Gupta *et al.* indicated prior blackout incidents using a time series model, particularly SVM, which they verified using an IEEE 30-bus testing ground [46]. SVM was learned to use a historical database that was created by evaluating system efficiency in static and transient modes. This resource documented both normal and aberrant events (cascading failure conditions). Pan and Lee conducted a comparison between ANN and SVM in the SG [47] midterm LF. ANN was typically used for forecasting; recently, researchers adapted SVM. The parameters for the day ahead, the week ahead, the month ahead, and the year ahead LF were analyzed. Mitchell *et al.* utilized these hyper-parameters for ELF on several loads [48]. SVM produced the global minimum on some occasions, according to the results. Both systems performed severely, with more than 4% deflection on intermittent load, while 2.3% variation on steady load.

Climate change, seasonal variability, sea-level rise, and natural calamities all impact ELF. As a result, SG demand management (DM) determines its dependability and stability in satisfying the consumer's regular power requests. The demand schedule can be efficiently derived from a reasonable projection of users' electricity usage patterns. Ali and Azad used ML approaches for DM and LF, such as LR, SVM, and MLP [49], which beat the other frameworks. The restricted quadratic optimization problem was used in support vector regression (SVR), which transferred the input characteristics into high dimensional space using a kernel.The support vector regression (SVR), which used a kernel to transfer the input characteristics into high-dimensional space, is employed to solve the restricted quadratic optimization (RQO) problem. SVR beat the BP-trained NN and other LR approaches. It also produced high-quality results when time series data was unavailable. As a result, the study advised SVR for LF. In other forecasts, SVM, like DM, performed admirably. When estimating lake water levels, SVM outperformed ANN and regression techniques, particularly the seasonal auto-regressive (SAR) paradigm, demonstrating coherence and a great outcome (long-term prognosis) [50]. SVM fused with the time series forecasting (TSF) module is employed for financial analysis, which assisted in overcoming two typical issues, namely noisy and non-stationary data [51], [52]. Furthermore, SVM with sparse representation outperformed statistical models and GRNN [53]. The fusion of chaotic GA and SVR algorithms improved the precision of chaotic LF [54]. Furthermore, SVM outperformed MLP in forecasting wind speed. Alazab *et al.* created a multi-directional LSTM (MDLSTM) framework to anticipate the SG's resilience, and the results show that MDLSTM outperforms classical LSTM, RNN, and gated recurrent units (GRUs) [19].

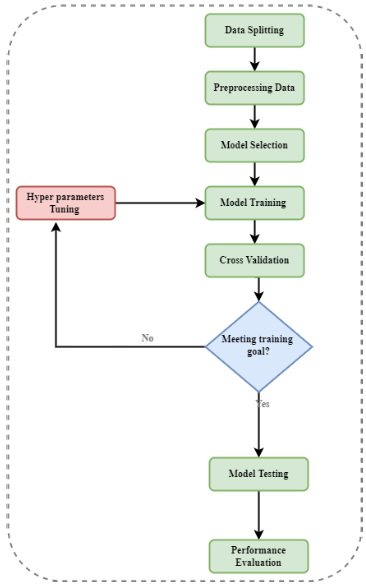


FIGURE 2. Flow chart of ML frameworks.

It is clear from the prior studies that ML approaches are pretty valuable for LF. Statistical models are simple to build and require few computations, yet they are inaccurate. However, a vast amount of data and calculations are necessary to create these models. LF is influenced by various factors, including building material, size, individual loads, the number of loads, occupant behavior, and weather, among others. Furthermore, regardless of data set size, it is clear that most SG research has been conducted utilizing DL methods. However, due to the magnitude of the SG data set, ML techniques coped better than DL models. Thus, the current effort focuses on the application of ML techniques to the SG data set. The problems were described in an overview of the survey. Table 3 shows the frameworks employed and the conclusions drawn from their approach.

III. ML MODELS

This study covers the supervised ML algorithms based on regression. In this study, all the ML models are used to forecast the hourly load consumption. The general flow adopted in all the algorithms is depicted in Fig. 2.

A. LR

The LR algorithm is a supervised ML technique that conducts regression. It only determines the linear relationship between the input and output variables. When their relationship is nonlinear, it is considered as multi-variable regression (MVR). The following is how the LR is defined:

$$\mathbf{y} = \theta_0 + \theta_1 \cdot \mathbf{x} \tag{1}$$

Here,

$\mathbf{x}$ - input training data

TABLE 3. Literature review on SGs.

Identified problem	Devised frameworks	Generalizations
Individual consumer's household load for DF [42]	1.3 layered ANN (60 − 20 − 1 neuron) 2. SVM 3. Regressive framework	The STMLF was not evaluated to some other model, nor was the effect of data during EP forecasting, while the relationship was not discussed.
STLF within micro-grids [43]	1. SOM 2. K-Means clustering 3. MLP	The system was not tested in an MG context. In addition, the number of patterns examined for review when compared to other models
Forecasting long and medium-terms energy demand in SG at different level [44]	1. ANN-NAEMI 2. MLR3 3. Ada-Boost	Various forecasting errors with big data and setting input parameters
Linking between EL and EP signals of the non-linear patterns [55]	1. LSSVM-MIMO 2. ANN 3. Optimization based QOABC algorithm 4. GMI	The system may anticipate load power signals without using previous data on a certain forecast day, although it may not be particularly precise.
To avoid catastrophic loss, an alert notification for predicting blackout episodes in SG was developed [46]	1. SVM	The SG robustness can be evaluated.
Midterm LF [47]	1. SVM 2. ANN	NA
STLF [48]	1. SVM 2.ANN	Achieving global minimum repeatedly By SVM. Deviation in prediction on constant and erratic loads.
Demand and LF in SG [49]	1. SVM, SVR 2. LR 3. NN with BP algorithm for training 4. MLP	SVR showed to be the best option for the context (even with the big dataset, it was suitable with the other methods), while MLP has more computational cost.
Lake water Levels prediction [50]	1. SVM 2. ANN 3. SAG framework	In the long-term, SVM produced more compatible and superior outcomes than other algorithms.
Financial TSF [51]	1. Fuzzy based SVR	In noisy and quasi data, the model yielded better results.
Non-stationary financial TSF [52]	1. SVM with regularized risk function	The framework used few SVs than the classical one. The framework should be explored with a sophisticated loss function.
ERP [53]	1. Linear and non-linear SVM 2. GA for feature selection	SVM outperforms traditional NN, and the structural risk minimization approach was used to reduce over-fitting.
ELF in SG [56]	1. LSTM-based RNN. 2. Reduction of recursive feature. 3. ETR as benchmark ML framework. 4. Ga for recognizing layers in LSTM.	On a multiple datasets, RNN-based LSTM may not function adequately.
A thorough examination of LF and dynamic electricity pricing [57]	1. ANN, WT-ANN 2. ARIMA-SVM, Regression 3. WTEC-ANN, FL 4. Stochastic frameworks	AI-based forecasting algorithms outperform analytical models.
An exploration of AI-based LF models [23]	1. ANN 2. BP, gradient descent	To improve forecast accuracy, integrated approaches can be utilized to train the NN. ANN outperforms BP in terms of performance.
STLF [58]	1. SVM with PBK and RBK 2. PCA 3. Rough set theory	Individual kernel SVM functions executed ineffectively.
Cyclic ELF [59]	1.SVR 2.Chaotic GA	ARIMA and Tensor Flow SVR simulated annealing models outperformed

$\mathbf{y}$ -output (supervised learning)

θ_0 - intercept

θ_1 - coefficient of $\mathbf{x}$ (2)

The optimal values of θ_0 and θ_1 are discovered to have the best-fit forecast line. The cost function is then determined using these values. This model essentially seeks the most appropriate value of y with the most minor difference between the valid and forecasted values. As a result, updating the values of θ_0 and θ_1 is required to minimize the mistake. In Eq. 3, the cost function $\digamma$ is defined often known as gradient descent (GD):

$$\digamma = \frac{1}{n}\sum_{i=1}^{n}(\mathsf{z}_i - \mathsf{y}_i)^2 \tag{3}$$

In this case, y_i represents the actual value, and z_i is the anticipated value. As can be seen, this technique essentially returns the z and y values of RMSE. The values of θ_0 and θ_1 are chosen at random here, and they are changed with each iteration, reducing the RMSE value and determining the best fit line for the model via GD.

B. NN

After executing certain mathematical computations, the NN translates the input values to the suitable output unit [60]. The NN is composed of two layers: a layer that accepts attributes considered as input, while layers reside between the input and output levels taken as hidden layers. The hidden layer should converge the input layer parameters and depart the yielded values to activation functions. Forecasting is assembled at the final layer, which is suggested as an output layer. Fig. 1 illustrates the general architecture of NNs. The working of hidden layer is represented in Eq. 4:

$$\mathcal{H}_n = \phi_1 + \left(\sum_{i=1}^{k}\mathcal{W}_{mn} + \theta_n\right) \tag{4}$$

n^{th} input and m^{th} hidden layers having weight $\mathcal{W}_{mn}$ and θ_n is the value of bias factor.

C. KNN

KNN is frequently used for classification and regression in data recognition and consistency. KNN is a supervised ML in a family of algorithms. It is a non-parametric strategy used to consider statistics. In both cases, the input is obtained from a training block, which is the training input, and then a corresponding target and output model are formed. Because inferences are produced directly from training examples, K-NN is a type of memory-based learning. The neighbors are derived from a known class's set of objects. If $K = 1$, the class is assigned to the class's single nearest neighbor. A straight line will always be formed under a standard clustering algorithm when there is the shortest distance between two peers, and this distance is known as the Euclidean distance [61]. The result of KNN regression is the mean of its KNNs. The disadvantages of the KNN algorithm include that it is not the fastest algorithm, works with a limited number of inputs, requires homogeneous features, and is sensitive to local data alignment. Equations for KNN are as follows: Euclidean equation ($\mathfrak{E}_{\mathfrak{e}}$) is represented in Eq. 5:

$$\mathfrak{E}_{\mathfrak{e}} = \sqrt{\sum_{j=1}^{H}(\alpha_j - \beta_j)^2} \tag{5}$$

Manhattan equation ($\mathfrak{M}_{\mathfrak{e}}$) is presented in Eq. 6:

$$\mathfrak{M}_{\mathfrak{e}} = \sum_{j=1}^{H} |\alpha_j - \beta_j| \tag{6}$$

Minkows equation ($\mathfrak{M}\mathfrak{k}_{\mathfrak{e}}$) is presented in Eq. 7:

$$\mathfrak{M}\mathfrak{K}_{\mathfrak{e}} = \left(\sum_{j=1}^{H}(|\alpha - \beta_j|)^p\right)^{1/p} \tag{7}$$

D. SVM

The classification method is described mathematically as:

$$\mathfrak{f}\,(u, v) = \sum_{j=1}^{\mathcal{D}} v_j \mathfrak{X}_j\,(u) + \mathfrak{k}, \tag{8}$$

where $v_j^{\infty}(j = 1, 2, 3, \ldots)$ are the forecaster parameters computed The dimensional space is marked by $\mathcal{D}$, and $\mathfrak{k}$ is determined by the distribution of data and classification variables. SVM tries to define a hyper-plane that separates data points in a $\mathcal{D}$-dimensional subspace. In this study, the hyper-plane is defined by Eq. 8. The regularized risk function $\mathfrak{R}_f$ is therefore defined as follows:

$$\mathfrak{R}_f\,(v) = \frac{\sum_{j=1}^{\mathcal{D}} \left|\mathfrak{L}_j^{\mathsf{a}} - \mathfrak{f}\,(u, v)\right|_{\varkappa} + \sigma v^2}{\mathcal{D}}, \tag{9}$$

where σ is the feature selection regulating threshold, $\varkappa$ is the insensitive loss function parameter, and $\mathfrak{L}_j^{\mathsf{a}}$ is the targeted load consumption pattern. The parameter v must be obtained through minimization of this $\mathfrak{R}_f$. The robust error function $\mathfrak{u}$ is calculated as follows:

$$\mathfrak{u} = \begin{cases} 0 if & \left|\mathfrak{L}_j^{\mathsf{a}} - \mathfrak{f}\,(u, v)\right| < \varkappa \\ \left|\mathfrak{L}_j^{\mathsf{a}} - \mathfrak{f}\,(u, v)\right| & otherwise. \end{cases} \tag{10}$$

Eq. 10 employs a function to minimize Eq. 9 and can be modeled as follows:

$$\mathfrak{f}\left(u, \pi, \pi^*\right) = \sum_{j=1}^{M} \left(\pi^* - \pi\right)\mathfrak{K}^*\left(u, u_j\right) + \kappa \tag{11}$$

where $\pi^* \geq 0$ for all I values. $\mathfrak{K}^*\,(u, z)$ is the SVM kernel function that shows the multiplication of radial basis KPCA in the feature space $\mathfrak{f}^*$ as:

$$\mathfrak{K}^*\,(u, z) = \sum_{j=1}^{\mathcal{D}} \mathfrak{X}_j\,(u)\mathfrak{X}_j\,(z) \tag{12}$$

In an infinite feature space, the $\mathfrak{K}^*$ eliminates the requirement for $\mathfrak{X}_j$ feature will be calculated. By maximizing the quadratic form, the π and π^* can be obtained:

$$\mathcal{R}\left(\pi^*, \pi\right) = -\varkappa \sum_{j=1}^{M}\left(\pi_j^* + \pi_j\right) + \sum_{j=1}^{M} \mathfrak{L}_j^{\mathsf{a}}\left(\pi_j^* - \pi_j\right) \tag{13}$$

$$-\frac{1}{2}\sum_{j,k=1}^{M}\left(\pi_j^* + \pi_j\right)\left(\pi_j^* - \pi_j\right)\mathfrak{K}^*\left(u_j, z_j\right). \tag{14}$$

The generalized versions of kernel functions are as follows, where $\mathfrak{P}$ indicates the principal component:

(i). Linear kernel function: It is a function that is used in conjunction with SVM to provide identical data points in a dataset [62].

$$\mathfrak{K}\,(r,\ z) = \langle r,\ z\rangle \tag{15}$$

(ii). Logistic Sigmoid based kernel function: It is also known as the Hyperbolic tangent kernel, which developed in the NN research area. The Sigmoid function has been employed as an activation function for NNs in the majority of cases.

$$\mathfrak{K}\,(r,\ z) = \tanh\left(\mathsf{a}_0\langle r,\ z\rangle^d + \mathsf{a}_1\right) \tag{16}$$

(iii) Radial basis kernel: It is an ubiquitous kernel function that is commonly utilized in a wide range of kernel-based ML techniques. It is indeed commonly used for SVM classification tasks.

$$\mathfrak{K}\,(r,\ z) = \exp\left(-\theta\|r - z\|^2\right) \tag{17}$$

E. DTC

DTC is one of the few classification approaches that allow us to comprehend the whole reasoning that the classifier applies when making a specific classification [63]. DTC displays a graphical representation of all possible decision options based on certain circumstances. It, like a tree, starts with a root and then spreads to various viable answers. The training data set is added to the tree by the root node, and each node subsequently asks a true or false question about one of the features. The dataset is now separated into two independent subsets. Eq. 18 gives the equation for entropy:

$$\mathrm{E} : \boldsymbol{I}\,(p_1, p_2, \ldots p_n) = \sum_{i=1}^{n}(p_i \log\left(1/p_i\right) \tag{18}$$

$(p_1, p_2, \ldots p_n)$ indicates the class label probabilities The Gini-index ($\mathfrak{G}\mathfrak{I}$) was used to partition the data in the proposed model. It is determined by subtracting the sum of each class's squared probabilities from one. It prefers larger partitions that are simple to implement, whereas information gain prefers smaller partitions that have different values. Mathematically, $\mathfrak{G}\mathfrak{I}$ is presented in Eq. 19:

$$\mathfrak{G}\mathfrak{I} = 1 - \sum (P(x = k))^2 \tag{19}$$

where $P(x = k)$ is the probability that a target feature takes a specific value, k.

F. PROPOSED EDTC

A DT that incorporates fitting functions, loss functions, and gradient descent analysis is known as an enhanced DTC [9]. The DT in this work creates starting values for fitting functions with multiple regression, which handles with a large number of input variables. The errors between observed data sets and output values are then determined using a loss function. Furthermore, popular loss functions include square-error, absolute-error, and unfavorable binomial log-likelihood functions. The gradient boosting (GB) approach is then utilized to find the fitting function with the lowest predicted loss function value. The preceding phase is repeated to acquire the best fitting procedure.

Following the input vector p and the output variable q, which contains training samples $(\mathsf{p}_m,\ \mathsf{q}_m)$ are provided, when the value of loss function $(\mathfrak{L}(\mathsf{q}, \mathfrak{F}(\mathsf{p})))$ is reduced, a fitting function $(\mathfrak{F}_\mathsf{p})$ is chosen. $\mathfrak{F}_\mathsf{p}$ is a linear combination of a cluster of basis functions $\mathfrak{f}_r(\mathsf{p})$ after R iterations, which is depicted in Eq. 20:

$$\mathfrak{F}(\mathsf{p}) = \sum_{r=1}^{R} \delta_r \mathfrak{f}_r(\mathsf{p}) + c \tag{20}$$

where c is the constant value.The gradient boosting algorithm uses the gradient descent analysis to approximate targeted Eq. 20. The specific method is described as follows: Let $\mathfrak{f}_0(\mathsf{p}) = 1$ then find the constant coefficient δ_0 from a sum of the minimum loss function $\mathfrak{L}(\mathsf{q}_j, \delta)$:

$$\mathfrak{F}_0(\mathsf{p}) = \delta_0 \mathfrak{f}_0(\mathsf{q}) = \delta_0 = \arg\min_{\delta} \sum_{j=1}^{J} \mathfrak{L}(\mathsf{q}_j,\ \delta) \tag{21}$$

After getting $\mathfrak{f}_0(\mathsf{p})$, we can use the recursive idea to solve this problem. $\mathfrak{F}_0(\mathsf{p})$, $\mathfrak{F}_1(\mathsf{p})$, $\mathfrak{F}_{r-1}(\mathsf{p})$ are derived by the Eq. 22:

$$\mathfrak{F}_r(\mathsf{p}) = \mathfrak{F}_{r-1}(\mathsf{p}) + \delta_r \cdot \mathfrak{f}_r(\mathsf{p}) \tag{22}$$

We derive δ_r as a sum of minimum loss function

$$\delta_r = \sum_{j=1}^{J} \mathfrak{L}(\mathsf{q}_j,\ [\mathfrak{F}_{r-1}(\mathsf{p}_j) + \delta \mathfrak{f}_r(\mathsf{p}_j)]) \tag{23}$$

while $\mathfrak{f}_r(\mathsf{p})$ is the sum of the negative gradient of $\mathfrak{F}_{r-1}(\mathsf{p}_j)$:

$$\mathfrak{f}_r(\mathsf{p}) = -\sum_{j=1}^{J} \Delta_{\mathfrak{F}} \mathfrak{L}(\mathsf{q}_j,\ \mathfrak{F}_{r-1}(\mathsf{p}_i)) \tag{24}$$

The procedure of E-DTC is described in algorithm 1.

Algorithm 1 EDTC

Require:

1. $\mathfrak{F}_0(\mathsf{p}) = \arg\min_{\delta} \sum_{j=1}^{J} \mathfrak{L}(\mathsf{q}_j,\ \delta)$
2. J: Number of data sets

Ensure: $\mathfrak{F}(\mathsf{p}) = \mathfrak{F}_R(\mathsf{P})$

3. R: Iteration times

For $r = 1 to R$

4. $\mathfrak{f}_r(\mathsf{p}) = -\sum_{j=1}^{J} \Delta_{\mathfrak{F}} \mathfrak{L}(\mathsf{q}_j,\ \mathfrak{F}_{r-1}(\mathsf{p}_i))$
5. $\delta_r = \sum_{j=1}^{J} \mathfrak{L}(\mathsf{q}_j,\ [\mathfrak{F}_{r-1}(\mathsf{p}_j) + \delta \mathfrak{f}_r(\mathsf{p}_j)])$
6. $\mathfrak{F}_r(\mathsf{p}) = \mathfrak{F}_{r-1}(\mathsf{p}) + \delta_r \cdot \mathfrak{f}_r(\mathsf{p})$

end

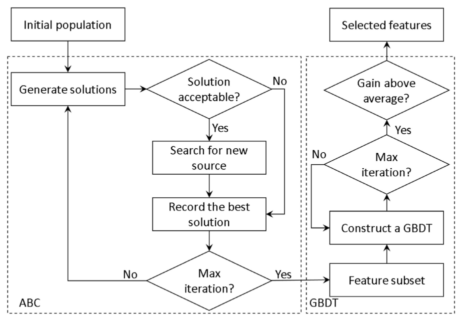


FIGURE 3. **EDTC algorithm processes.**

1) PROPOSED EDTC DESCRIPTION

Algorithm 1 is an improved top-down algorithm of DTC, and it uses information gain presented in Eq. 19 as splitting criteria to build a DT. The criteria of EDTC is Gain Rations which is a modification of the information gain. The benefit of EDTC is noticeably low error rates, less memory, and high optimization. Therefore, ETDC algorithm is more accurate and much faster. EDTC has tree like structures, prunes the original dtc, and creates DT in the way of "divide and rule". In addition, the most improvement in DTC is through gradient boosting technique. Algorithm 1 illustrates that objective function of proposed EDTC is to get boosting factor δ from the minimum loss function $\mathfrak{L}(\mathsf{q}_j, \delta)$ as depicted in Eq. 21. The principle of algorithm is repeatedly calling weak learners and giving these weak learners high weight vote value. By doing so, the training process can focus more on the cases that caused error, which tends to reduce bias. With respect to EDTC, the most critical feature of EDTC is boosting technique, and another is the construction of a cost-sensitive as formulated in Eqs. 21 and 23. The proposed EDTC algorithmic process is depicted in Fig. 3.

Definition 1: Let the input data which is expected to classify correctly.

Proof: Input to algorithm consists of a collection of training cases, each having a tuple of values for a fixed set

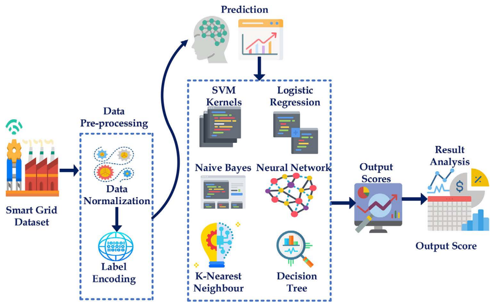


FIGURE 4. The work-flow of the devised framework.

of attributes (or independent variables) $A = \{A_1, A_2, \ldots, A_k\}$ and a class attribute (or dependent variable). An attribute A_2 is described as continuous or discrete, if it is the EDTC algorithm. In case of the DTC, Attributes are of type only numerical or nominal. The class attribute (target attributes) C is discrete and has values $C_1, C_2, \ldots, C_x$ □

Remark 1: EDTC which classifies the data correctly.

IV. PROPOSED METHODOLOGY

The designed system comprises four major components, as illustrated in Fig. 4. The first section contains the datasets, which consist of four years of publicly available NYISO datasets (2017-2020). Pre-processing is the second stage, in which we attempt to clean the data. Pre-filtering is critical for enhancing the quality of data and the importance of ML algorithms, both of which can aid in successful forecasting. The third component features engineering, which improves classification accuracy, reduces data dimensions to avoid complexity, and speeds up the processing time [64]. Two ML methods are used for feature engineering in the proposed system. DTC determines the relevance of characteristics first. RFE is then used to remove low-importance features from the datasets. The vital components are kept, which can help with accuracy. Before being fed to classifiers, data is separated into training and testing sets in a 3:1 ratio. The first nine months of data from each year are used as the training set, with the remaining three months serving as the testing set. The classifiers employed are SVM, LR, KNN, DTC, ANN, and E-DTC. For LF, we improved DTC and proposed EDTC.

A. DATA-SET DESCRIPTION

Python is used to carry out the experiments. Four years of NYISO data are used [65]. The data contains sixteen features and 1095 instances, as well as system load data for each day and a variety of additional parameters. We extract the essential characteristics in the first stage because all aspects are incompatible.

B. DATA SETTING AND PREPROCESSING

Pre-processing is critical for increasing the quality of the data as well as the effectiveness of ML algorithms [66]. Normalization and data transformation are two typical pre-processing strategies utilized in any ML model. The variables in an SG dateset are distributed across several ranges, which typically results in a bias favoring values with higher weights, reducing the effectiveness of the devised framework. Since attribute normalization aids in the convergence rate and numerical stability of NN training, a zero-mean normalization strategy is used in the study for data normalization on the load and temperature variables. Normalization is performed according to Eq. 25:

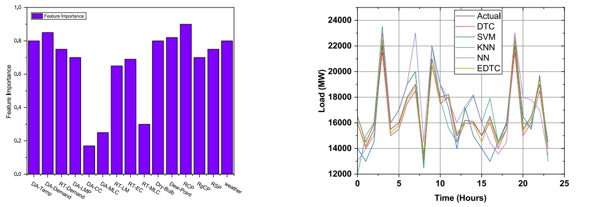
$$y'_j = \frac{x_j - \nu}{\kappa} \tag{25}$$

where y'_j refer to the zero average score of the j instance, ν is the mean in time series dataset, while κ correspond to standard deviation of the dataset, respectively. To normalize the test data, the mean and standard deviation of the training data were employed. For final predictions, test data outcomes are normalized. we used dunce variables to control for these characteristics, as most previous ELF research has been done and described in the literature [24], [67], [68].

Machines process data using mathematical formulae; therefore, data must be quantitative. Since the majority of the dataset comprises both categorical and numerical information, data encryption is achieved during data pre-processing [69], which converts quasi inputs to numeric ones before supplying them to ML frameworks. After that, data is divided into training and testing datasets. The training dataset is used to prepare the ML algorithms, then considered with a new dataset to enhance their performance. In this work, 30% is used to evaluate the performance of the produced ML algorithms, while 70% of the dataset is used to understand the ML algorithms, while diverse ML techniques, rather than DL-based algorithms, are used for classification depending on the magnitude of the dataset. To classify the SG dataset, conventional ML schemes such as LR, KNN, NN, SVM, and DTC were used in this work. F1-score, accuracy, receiver operating characteristic (ROC), and precision are then used to test the efficiency of the ML algorithms. The acquired results are similar to previous work on the SG datasets. The ML methods employed in this work are detailed in the following subsections.

C. FEATURE ENGINEERING

In the AEMO (NWS) data-set, DTC-based FE is used to select the essential traits and reduce outliers; feature patterns are considered vectors. The feature values in these vectors have separate timestamps. Features for EL are regarded as load demand, and those with a minor impact on EL forecasting may be removed. DT, the most advanced and efficient feature extraction technique, determines the importance of features. The DT method assigns a score to each characteristic individually. Then, features are selected using recursive features elimination (RFE) depending on their score. The score given by DT is shown in Fig. 5a. Let 0.5 be the feature selection threshold ($\mathfrak{T}_{\mathfrak{fs}}$). When the grades of features are greater than the $\mathfrak{T}_{\mathfrak{fs}}$, these features may be kept and be used



(a) Importance of features calculated by devised DTC.

(b) Trend of forecasting for different techniques.

FIGURE 5. Importance of features calculations by devised DT & forecasting trends of different techniques.

TABLE 4. The devised DTC based FS with different $\mathfrak{T}_{\mathfrak{fs}}$ values. (Threshold: $\mathfrak{T}_{\mathfrak{fs}}$, Time: $\mathfrak{T}$, Error: $\mathfrak{E}$, Dropped features: $\mathfrak{F}_{\mathfrak{d}}$.)

Parameters	Observations			
$\mathfrak{T}_{\mathfrak{fs}}$	0.70	0.60	0.55	0.50
$\mathfrak{T}$	89s	98s	105s	121s
$\mathfrak{E}$	3.1%	3.0%	1.9%	1.4%
$\mathfrak{F}_{\mathfrak{d}}$	DA-MLC	DA-MLC	DA-MLC	DA-MLC
	RT-MLC	RT-MLC	RT-MLC	RT-MLC
	DA-CC	DA-CC	DA-CC	DA-CC
	RT-CC	RT-CC	RT-CC	RT-CC
	RSP	RSP	RSP	RSP
	RT-EC	RT-EC	RT-EC	
	RT-LM	RT-LM		
	Rgcp	Rgcp		
	RT-Demand			
	DA-LMP			
	RCP			

for further processing. Besides, the grade of features smaller than the fixed $\mathfrak{T}_{\mathfrak{fs}}$ value may be pitched. Distinguishable $\mathfrak{T}_{\mathfrak{fs}}$ weights are employed to restrain the FS process for considering the significance of feature extraction. The results are depicted in Table 4. Simultaneously, it boosts the $\mathfrak{T}_{\mathfrak{fs}}$ values from 0.5, resulting in dropping more features. It is evident from the results that increment in $\mathfrak{T}_{\mathfrak{fs}}$ value is directly proportional to the more feature drop ($\mathfrak{F}_{\mathfrak{d}}$). It would enhance the training momentum but lessen the forecasting accuracy.

With feature extraction, the retrieved features are sent to classifiers; however, the data was split into testing and training sets before feeding the data to classifiers. The information for nine months of each year is retained in the training set, while the data for the remaining three months is preserved in the evaluation set. With that, the models are trained using a training set. The fundamental reason for writing this work is to compare basic ML methods with one upgraded DTC for LF. These five strategies (SVM, LR, NN, KNN, and EDTC) are often employed for LF, but it is unclear which one is best suited. After training, the performance of classifiers was assessed using a testing set.

D. PERFORMANCE METRICS

Four statistical measures assess classifier accuracy: RMS, MSE, MAPE, and MAE. MSE and MAE have built-in functions, and RMS and MAPE have defined roles. The study also considers performance aspects, including recall, F1 score, precision, and accuracy, to evaluate the ML algorithms. In the most delinquent research, 70% of the SG data-set can be used for training, whereas 30% for verification and validation. The parameters mentioned above and the ROC curve are used to evaluate the ML frameworks, increasing the confirmation of the outcomes.

V. SIMULATION RESULTS AND DISCUSSION

A. PREDICTION TRENDS AND STATISTICAL MEASURES RESULTS

The forecasted trend is depicted in Fig. 5b. When the predicting trend is compared to the actual trend, it is obvious that DTC and EDTC closely reflect the real trend. This signifies that these two algorithms are operating admirably. Furthermore, in terms of trend following, EDTC outperforms DTC. Furthermore, the comparison is shown using other statistical indicators. MAE is depicted in Fig. 6a. EDTC has a very low MAE, which indicates it causes very little error when compared to other approaches. Moreover, in our scenario described in Tables 1 and 7, the EDTC processing time is relatively short. MAE for EDTC is ranked on first and is better than DTC. MAPE is also calculated for the STLF as shown in Fig. 6b. In MAPE, the error rate of KNN is more unpredictable than that of EDTC. In this case, EDTC outperforms all other approaches. The performance of KNN may improve as the number of instances rises. Tables 1 and 7 show that EDTC works well and has a very quick emergence time. RMS performance is depicted in Fig. 6c. RMS follows the same pattern as MAPE. EDTC's performance is superior to those of its competitors. DTC is ranked second. A function is defined to calculate RMS using the conventional RMS formula. Finally, the MSE score is illustrated in Fig. 6d. For MAE and MSE, sklearn's built-in functions are employed. The MSE for NN is close to 0, indicating that it has an extremely low MSE. Once again, LR comes in second. DTC's score is nearly comparable to EDTC's, but EDTC outperforms DTC in this category as well. Based on the foregoing extensive explanation of outcomes, it is obvious that DTC and EDTC are excellent for STLF due to their ease of use, rapid emergence, and high accuracy. Furthermore, DTC and EDTC can capture non-linear and noisy LF data quite effectively. Furthermore, as time passes, the data load increases, but the performance of EDTC does not decline, but rather improves. As a result of our experiments, we recommend using DTC and EDTC for STLF among the previously stated ML algorithms.

B. PERFORMANCE METRICS RESULTS

1) NN RESULTS

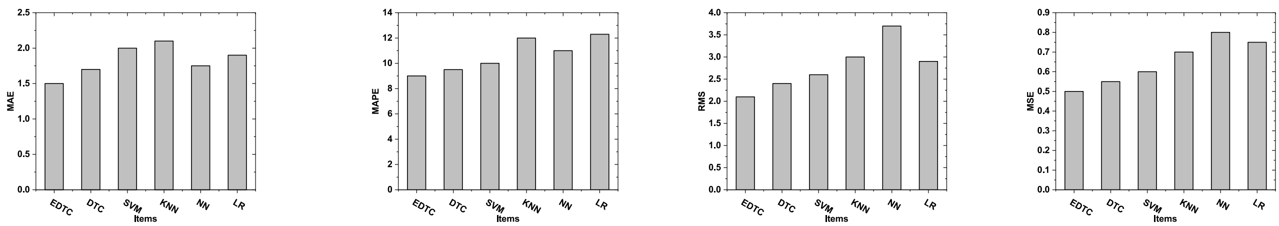
The confusion metric (CM) for NN classification in Table 5 shows that we achieved 97.5% accuracy with 1057 records out of 1084 and 2.5% false positive rate (FPR) in case of stable class, while in unstable/faulted case, the accuracy and FPR attained are 98.00% with 1887 out of 1916 records and 1.5% respectively. The classification report (CR) for NN in Table 6 represents that we got 98.5%, 97.60%, and 99.50% F1 score, precision, and recall respectively in stable class, while

TABLE 5. Confusion metrics (CM) for the proposed EDTC and other ML algorithms.

CM for NN			CM for KNN			CM for DTC			CM for SVM			CM for the proposed EDTC		
Labels	Stable	Unstable	Labels	Stable	Unstable	Labels	Stable	Unstable	Labels	Stable	Unstable	Labels	Stable	Unstable
Stable	1057 97.5%	27 2.5%	Stable	701 64.7%	383 35.3%	Stable	1050 96.8%	34 3.13%	Stable	977 90.1%	107 9.9%	Stable	1084 100%	0
Unstable	29 1.5%	1887 98%	Unstable	275 14.4%	1641 85.6%	Unstable	66 3.44%	1850 96.6%	Unstable	157 8.2%	1759 91.8%	Unstable	1 0.1	1915 99.9%

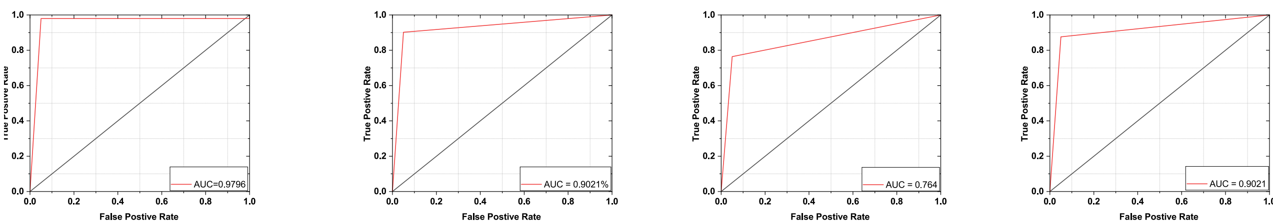
TABLE 6. Classification report (CR) for proposed EDTC and existing ML models.

ML Models	Labels	Precision	Recall	F1	Support	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss
SVM	Stable	0.830	0.901	0.864	0.361	97.20	0.07	97.50	0.07
	Unstable	0.941	0.896	0.918	0.639	97.20	0.07	97.50	0.07
NN	Stable	0.99	0.957	0.973	0.361	98.45	0.07	98.90	0.07
	Unstable	0.976	0.995	0.985	0.639	98.45	0.07	98.90	0.07
KNN	Stable	0.718	0.647	0.681	0.639	94.23	0.08	94.50	0.08
	Unstable	0.811	0.856	0.833	0.639	94.23	0.08	94.50	0.08
DTC	Stable	0.959	0.920	0.932	0.361	97.98	0.05	98.50	0.05
	Unstable	0.961	0.943	0.969	0.639	97.98	0.05	98.50	0.05
Proposed EDTC	Stable	0.999	1.000	1.000	0.361	99.21	0.02	99.70	0.02
	Unstable	1.000	0.999	1.000	0.639	99.21	0.02	99.70	0.02



(a) MAE score (b) MAPE score. (c) RMS error score. (d) MSE score.

FIGURE 6. MAE, MAPE, RMS, & MSE scores.



(a) ROC curve for NN. (b) ROC for SVM. (c) ROC for KNN. (d) ROC for DTC.

FIGURE 7. ROC for NN, SVM, KNN, & DTC.

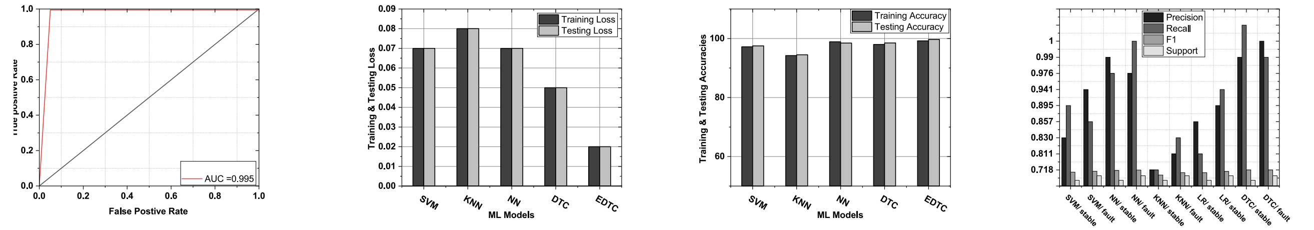
in the unstable category, F1-Measure scores, the precision, and recall are 97.30%, 99.00%, and 95.70%, and 97.30%, respectively. The ROC for the NN classifier is depicted in Fig. 7a, where the area under the curve (AUC) is 97.96%.

2) SVM RESULTS

The CM for SVM classifier in Table 5 shows that we achieved 90.1% forecasting accuracy with 977 records out of 1084 and 9.9% FPR in case of stable class, while in unstable/faulted case, the accuracy and FPR attained are 91.80% with 1759 out of 1916 records and 8.2% respectively. The classification report (CR) for SVM in Table 6 represents that we got 86.4%, 83.00%, and 90.10% F1 score, precision, and recall respectively in stable class, while in the unstable category, F1-Measure scores, the precision, and recall are 91.80%, 94.10%, and 89.60%, respectively. The ROC for the SVM classifier is depicted in Fig. 7b, where the AUC is 90.21%.

3) KNN RESULTS

The CM for KNN classifier in Table 5 shows that we achieved 64.7% forecasting accuracy with 701 records out of 1084 and 35.3% FPR in case of stable class, while in unstable/faulted case, the accuracy and FPR attained are 86.60% with 1641 out of 1916 records and 14.4% respectively. The classification report (CR) for KNN in Table 6 represents that we got 68.1%, 71.80%, and 64.70% F1 score, precision, and recall respectively in stable class, while in the unstable category, F1-Measure scores, the precision, and recall are 83.30%,



(a) ROC for EDTC (b) Training and testing loss (c) Training and testing accuracy (d) Metrics performance in CR.

FIGURE 8. ROC for EDTC, Training/testing loss, Training/testing accuracy, & Metrics performance in CR.

TABLE 7. Analysis of proposed and existent ML models based on classification parameters.

Parameters	EDTC	LR	KNN	DTC	NN
Accuracy in general	Excellent	Good	Good	Very Good	Good
Speed of learning with respect to number of attributes and the number of instances	Excellent	Very Good	Good	Average	Average
Speed of classification	Excellent	Good	Very Good	Good	Average
Tolerance to missing values	Good	Good	Excellent	Very Good	Very good
Tolerance to irrelevant Attributes	Good	Average	Good	Excellent	Good
Dealing with discrete /binary /continuous attributes	Excellent	Very Good	Average	Very Good	Excellent
Tolerance to noise	Excellent	Good	Good	Good	Very Good
Dealing with Over fitting	Excellent	Average	Excellent	Average	Average
Attempts for incremental learning	Excellent	Good	Very Good	Good	Good
Explanation ability/transparency of knowledge/classifications	Good	Good	Average	Good	Very Good

81.10%, and 85.60%, respectively. The ROC for the KNN classifier is depicted in Fig. 7c, where the AUC is 76.45%.

4) DTC RESULTS

The CM for DTC classifier in Table 5 shows that we achieved 96.8% forecasting accuracy with 1050 records out of 1084 and 3.13% FPR in case of stable class, while in unstable/faulted case, the accuracy and FPR attained are 96.50% with 1850 out of 1916 records and 3.44% respectively. The classification report (CR) for LR in Table 6 represents that we got 95.9%, 92.10%, and 93.20% F1 score, precision, and recall respectively in stable class, while in the unstable category, F1-Measure scores, the precision, and recall are 96.10%, 94.30%, and 96.90%, respectively. The ROC for the DTC classifier is depicted in Fig. 7d, where the AUC is 90.21%.

5) EDTC RESULTS

The CM for EDTC in Table 5 shows that we achieved 100% forecasting accuracy with 1084 records out of 1084 and 0% FPR in case of stable class, while in unstable/faulted case, the accuracy and FPR attained are 99.90% with 1915 out of 1916 records and 0.1% respectively. The classification report (CR) for EDTC in Table 6 represents that we got 100%, 99.9%, and 100% F1 score, precision, and recall respectively in stable class, while in the unstable category, F1-Measure scores, the precision, and recall are 100%, 100%, and 99.00%, respectively. The ROC for the EDTC is depicted in Fig. 8a, where the AUC is 99.95%.

C. TRAINING AND TESTING LOSS AND ACCURACY

Training and testing accuracy along with loss of ML algorithms are described in Table 6. Graphical representation in Fig. 8c shows that SVM training accuracy and loss is 97.20% and 0.07 respectively, while its testing accuracy is 97.50% with 0.07 data loss. NN classifier having training and testing accuracy 98.45% and 98.90% with data loss 0.07 respectively.In the case of LR, data loss is 0.05 for both training and testing, and accuracy is 97.98% and 98.50%, respectively. The effectiveness of KNN training and testing is 94.23% and 94.80%, respectively. For both training and testing with KNN, the data loss is 0.08. EDTC achieved 99.07 % accuracy and 0.02 loss for training and testing. The EDTC obtained 1.98% higher accuracy compared to SVM, KNN, NN and LR.

D. METRICS PERFORMANCE IN CLASSIFICATION REPORT

Table 6 shows the effectiveness of ML models in terms of precision, recall and F1-score. The EDTC achieved a 100.00% F1-score, 99.00% recall and 100% precision, which is more remarkable than the ML models stated above for stable class. Similarly, in terms of precision, recall, and F1-score, the EDTC model beat the standard models in the unstable class.

E. ACCURACY ACHIEVED BY PROPOSED EDTC

The proposed EDTC framework has the highest forecast accuracy of 99.89% compared to other classifiers employed in this work, as shown in Table 9. ETDC outperforms different algorithms used in this work regarding prediction accuracy, recall, precision, and F1-Measure because it is a probability-based algorithm. while Table 10 depicts the comparison of proposed EDTC with other existing frameworks. We observed that Authors of [12], [70], [71] achieved 99.01%, 97.82% and 95.37% AUC using MLSTM, EKNN and Adaboost respectively while proposed EDTC achieve 99.42% AUC.

Remark 2: The following conclusions can be drawn from the results:

(i): Compared to DL models, ML techniques are better suited for classifying the SG dataset due to their small size.

(ii): Because the number of attributes is relatively low, the EDTC outperforms the other ML methods considered.

VI. CRITICAL ANALYSIS

Aside from performance, ML algorithms have several advantages over other AI and classical models for LF, including noise tolerance, pattern generation rather than assumption, handling non-linearity, and ease of use [72]. We employed five strategies in particular (EDTC, SVM, LR, KNN,

TABLE 8. Advantages EDTC over other ML Algorithms.

Proposed EDTC	SVM	DTC	KNN	NN
Easily Observed & develop generated rules	Fast, highly scalable model building (parallelism) and scoring.	Robust to noisy training data and effective if the training data is large.	More accurate than DT classification but less than EDT.	High tolerance of noisy data and ability to classify patterns for untrained data.

TABLE 9. The accuracy achieved after a comparison of ML algorithms and devised EDTC.

Classifier	Achieved accuracy
SVM	92.43
KNN	79.45
LR	84.76
NN	93.79
DTC	98.12
Proposed EDTC	99.43

TABLE 10. Comparison of proposed EDTC with other existing works in terms of accuracy.

Frameworks	Achieved accuracy
MLSTM [12]	99.12
E-KNN [70]	95.11
Adaboost [71]	98.16
Proposed EDTC	99.42

and NN). Tables 1, 7, and 8 provide a brief explanation of the logic behind the use and performance of each strategy.

VII. CONCLUSION AND FUTURE SCOPE

The stability of the SG is essential for efficient power distribution to the control stations. ML techniques play an integral part in signifying the resilience of the SGs. With the emergence of different Ml algorithms, the foremost challenge is to find the most appropriate algorithm to predict the stability of the SG. To accomplish this, a comprehensive survey of the state-of-the-art ML algorithms has been performed to predict the stability of SGs. In this work, a novel EDTC model is introduced to predict the stability of the smart grid. The proposed model has experimented on the smart grid dataset from NYISO. The performance of EDTC is compared with traditional ML models like SVM, KNN, NN, LR, and DT. The experimental results proved that the DTC algorithm outperforms SVM, KNN, LR, and NN. The comparative analysis proves the superiority of the proposed model concerning the accuracy, precision, loss, and ROC curve metrics. The proposed model achieved 99.07% training and testing accuracy, which is 3% times higher than other traditional ML algorithms. As part of the future work context aware paradigm, dynamic power requirements can be met while also making the SGs more reliable.

REFERENCES

[1] B. Zhao, L. Zeng, B. Li, Y. Sun, Z. Wang, M. Shahzad, and P. Xi, ''Collaborative control of thermostatically controlled appliances for balancing renewable generation in smart grid,'' *IEEJ Trans. Electr. Electron. Eng.*, vol. 15, no. 3, pp. 460–468, Mar. 2020.

[2] K. Berk, A. Hoffmann, and A. Müller, ''Probabilistic forecasting of industrial electricity load with regime switching behavior,'' *Int. J. Forecasting*, vol. 34, no. 2, pp. 147–162, Apr. 2018.

[3] J. R. Cancelo, A. Espasa, and R. Grafe, ''Forecasting the electricity load from one day to one week ahead for the Spanish system operator,'' *Int. J. Forecasting*, vol. 24, no. 4, pp. 588–602, 2008.

[4] M. Djukanovic, S. Ruzic, B. Babic, D. J. Sobajic, and Y. H. Pao, ''A neural-net based short term load forecasting using moving window procedure,'' *Int. J. Electr. Power Energy Syst.*, vol. 17, no. 6, pp. 391–397, Dec. 1995.

[5] L. J. Soares and M. C. Medeiros, ''Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data,'' *Int. J. Forecasting*, vol. 24, no. 4, pp. 630–644, Oct. 2008.

[6] R. Hu, S. Wen, Z. Zeng, and T. Huang, ''A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm,'' *Neurocomputing*, vol. 221, pp. 24–31, Jan. 2017.

[7] A. Kavousi-Fard, H. Samet, and F. Marzbani, ''A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting,'' *Expert Syst. Appl.*, vol. 41, no. 13, pp. 6047–6056, 2014.

[8] Z. H. Osman, M. L. Awad, and T. K. Mahmoud, ''Neural network based approach for short-term load forecasting,'' in *Proc. IEEE/PES Power Syst. Conf. Expo.*, Mar. 2009, pp. 1–8.

[9] N. Zeng, H. Zhang, W. Liu, J. Liang, and F. E. Alsaadi, ''A switching delayed PSO optimized extreme learning machine for short-term load forecasting,'' *Neurocomputing*, vol. 240, pp. 175–182, May 2017.

[10] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, ''Smart grid technologies: Communication technologies and standards,'' *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[11] S. K. Desai, A. Dua, N. Kumar, A. K. Das, and J. J. P. C. Rodrigues, ''Demand response management using lattice-based cryptography in smart grids,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[12] M. Alazab and R. Broadhurst, *Cyber-Physical Security: Protecting Critical Infrastructure at the State and Local level*, 1st ed, M. C. Robert and H. Simon Eds. Cham, Switzerland: Springer, 2017, pp. 251–266.

[13] L. Hernandez, C. Baladron, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, J. Lloret, and J. Massana, ''A survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings,'' *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1460–1495, 3rd Quart., 2014.

[14] A. Rahman, V. Srikumar, and A. D. Smith, ''Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks,'' *Appl. Energy*, vol. 212, pp. 372–385, Feb. 2018.

[15] L. Xiao, W. Shao, C. Wang, K. Zhang, and H. Lu, ''Research and application of a hybrid model based on multi-objective optimization for electrical load forecasting,'' *Energy*, vol. 180, pp. 213–233, Oct. 2017.

[16] N. Deepa, B. Prabadevi, P. K. Maddikunta, T. R. Gadekallu, T. Baker, M. A. Khan, and U. Tariq, ''An ai-based intelligent system for healthcare analysis using Ridge–Adaline stochastic gradient descent classifier,'' *J. Supercomput.*, vol. 77, no. 2, pp. 1998–2017, 2021.

[17] Anamika and N. Kumar, ''Market clearing price prediction using ANN in Indian electricity markets,'' in *Proc. Int. Conf. Energy Efficient Technol. Sustain. (ICEETS)*, 2016, pp. 454–458, doi: 10.1109/ICEETS.2016.7583797.

[18] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. R. Maddikunta, and G. Srivastava, ''Deep neural networks to predict diabetic retinopathy,'' *J. Ambient Intell. Hum. Comput.*, pp. 1–14, Apr. 2020, doi:10.1007/s12652-020-01963-7.

[19] H. Patel, D. S. Rajput, G. T. Reddy, C. Iwendi, A. K. Bashir, and O. Jo, ''A review on classification of imbalanced data for wireless sensor networks,'' *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 4, 2020, Art. no. 1550147720916404.

[20] W. Tian, "A review of smart grids and their future challenges," in *Proc. MATEC Web Conf.*, vol. 173. Les Ulis, France: EDP Sciences, 2018, p. 02025.

[21] G. K. Venayagamoorthy, "Dynamic, stochastic, computational, and scalable technologies for smart grids," *IEEE Comput. Intell. Mag.*, vol. 6, no. 3, pp. 22–35, Aug. 2011.

[22] M. Li, H. Xu, and Y. Deng, "Evidential decision tree based on belief entropy," *Entropy*, vol. 21, no. 9, p. 897, 2019.

[23] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, Oct. 2015.

[24] X. Liu, Z. Zhang, and Z. Song, "A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning," *Renew. Sustain. Energy Rev.*, vol. 119, Mar. 2020, Art. no. 109632.

[25] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, "Deep belief network based electricity load forecasting: An analysis of Macedonian case," *Energy*, vol. 115, pp. 1688–1700, Nov. 2016.

[26] P.-S. Yu, T.-C. Yang, S.-Y. Chen, C.-M. Kuo, and H.-W. Tseng, "Comparison of random forests and support vector machine for real-time radar-derived rainfall forecasting," *J. Hydrol.*, vol. 552, pp. 92–104, Sep. 2017.

[27] J. Wang, Y. Song, F. Liu, and R. Hou, "Analysis and application of forecasting models in wind power integration: A review of multi-step-ahead wind speed forecasting models," *Renew. Sustain. Energy Rev.*, vol. 60, pp. 960–981, Jul. 2016.

[28] S. Wang, X. Wang, S. Wang, and D. Wang, "Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting," *Int. J. Electr. Power Energy Syst.*, vol. 109, pp. 470–479, Jul. 2019.

[29] F.-J. Yang, "An extended idea about decision trees," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, 2019, pp. 349–354.

[30] D. Liu, Y. Shi, and Y. Tian, "Ramp loss nonparallel support vector machine for pattern classification," *Knowl.-Based Syst.*, vol. 85, no. 5, pp. 224–233, 2015.

[31] T.-Y. Wang and H.-M. Chiang, "Solving multi-label text categorization problem using support vector machine approach with membership function," *Neurocomputing*, vol. 74, no. 17, pp. 3682–3689, Oct. 2011.

[32] H. Akbaripour and E. Masehian, "Efficient and robust parameter tuning for heuristic algorithms," Dept. Ind. Eng., Tarbiat Modares Univ., Iran, Tehran, Tech. Rep., 2013.

[33] M. Gavrilas, "Heuristic and metaheuristic optimization techniques with application to power systems," Power Syst. Dept., Tech. Univ. Iasi, Iasi, Romania, Tech. Rep., Oct. 21, 2010.

[34] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," *Int. J. Soft Comput. Eng.*, vol. 2, pp. 137–151, May 2012.

[35] S. Nandhini and K. S. J. Marseline, "Performance evaluation of machine learning algorithms for email spam detection," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng. (ic-ETITE)*, 2020, pp. 1–4.

[36] I. Ramadhan, P. Sukarno, and M. A. Nugroho, "Comparative analysis of K-nearest neighbor and decision tree in detecting distributed denial of service," in *Proc. 8th Int. Conf. Inf. Commun. Technol. (ICoICT)*, 2020, pp. 1–4.

[37] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Machine learning-based fast intra mode decision for HEVC screen content coding via decision trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1481–1496, May 2020.

[38] G.-C. Liao, "Hybrid improved differential evolution and wavelet neural network with load forecasting problem of air conditioning," *Int. J. Electr. Power Energy Syst.*, vol. 61, pp. 673–682, Oct. 2014.

[39] Y. Dai and P. Zhao, "A hybrid load forecasting model based on support vector machine with intelligent methods for feature selection and parameter optimization," *Appl. Energy*, vol. 279, Dec. 2020, Art. no. 115332.

[40] V. M. E. Batitis, M. J. G. Caballes, A. A. Ciudad, M. D. Diaz, R. D. Flores, and E. R. E. Tolentino, "Image classification of abnormal red blood cells using decision tree algorithm," in *Proc. 4th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Mar. 2020, pp. 498–504.

[41] M. Alazab, S. Khan, S. S. R. Krishnan, Q.-V. Pham, M. P. K. Reddy, and T. R. Gadekallu, "A multidirectional LSTM model for predicting the stability of a smart grid," *IEEE Access*, vol. 8, pp. 85454–85463, 2020.

[42] F. Javed, N. Arshad, F. Wallin, I. Vassileva, and E. Dahlquist, "Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting," *Appl. Energy*, vol. 96, pp. 150–160, Aug. 2012.

[43] L. Hernández, C. Baladrón, J. M. Aguiar, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, "Artificial neural networks for short-term load forecasting in microgrids environment," *Energy*, vol. 75, pp. 252–264, Oct. 2014.

[44] T. Ahmad and H. Chen, "Potential of three variant machine-learning models for forecasting district level medium-term and long-term energy demand in smart grid environment," *Energy*, vol. 160, pp. 1008–1020, Oct. 2018.

[45] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "On recent advances in PV output power forecast," *Sol. Energy*, vol. 136, pp. 125–144, Oct. 2016.

[46] S. Gupta, R. Kambli, S. Wagh, and F. Kazi, "Support-vector-machine-based proactive cascade prediction in smart grid using probabilistic framework," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2478–2486, Apr. 2015.

[47] X. Pan and B. Lee, "A comparison of support vector machines and artificial neural networks for mid-term load forecasting," in *Proc. IEEE Int. Conf. Ind. Technol.*, Mar. 2012, pp. 95–101.

[48] G. Mitchell, S. Bahadoorsingh, N. Ramsamooj, and C. Sharma, "A comparison of artificial neural networks and support vector machines for short-term load forecasting using various load types," in *Proc. IEEE Manchester PowerTech*, Jun. 2017, pp. 1–4.

[49] A. B. M. S. Ali and S. Azad, "Demand Forecasting in Smart Grid," in *Smart Grids Green Energy and Technology*. A. Ali, Eds. London. U.K.: Springer, 2013, pp. 135–150, doi: 10.1007/978-1-4471-5210-1_6.

[50] M. S. Khan and P. Coulibaly, "Application of support vector machine in lake water level prediction," *J. Hydrol. Eng.*, vol. 11, no. 3, pp. 199–205, May 2006.

[51] R. Khemchandani and S. Chandra, "Regularized least squares fuzzy support vector regression for financial time series forecasting," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 132–138, Jan. 2009.

[52] F. E. H. Tay and L. J. Cao, "Modified support vector machines in financial time series forecasting," *Neurocomputing*, vol. 48, nos. 1–4, pp. 847–861, Oct. 2002.

[53] W.-C. Hong, "A hybrid support vector machine regression for exchange rate prediction," *Int. J. Inf. Manage. Sci.*, vol. 17, no. 2, pp. 19–32, 2006.

[54] Q. Zhang, Y. Sun, and Z. Cui, "Application and analysis of ZigBee technology for smart grid," in *Proc. Int. Conf. Comput. Inf. Appl.*, Dec. 2010, pp. 171–174.

[55] H. Shayeghi, A. Ghasemi, M. Moradzadeh, and M. Nooshyar, "Simultaneous day-ahead forecasting of electricity price and load in smart grids," *Energy Convers. Manage.*, vol. 95, pp. 371–384, May 2015.

[56] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies* vol. 11, no. 7, p. 1636, Jun. 2018.

[57] A. R. Khan, A. Mahmood, A. Safdar, Z. A. Khan, and N. A. Khan, "Load forecasting, dynamic pricing and DSM in smart grid: A review," *Renew. Sustain. Energy Rev.*, vol. 54, pp. 1311–1322, Feb. 2016.

[58] W. Li, N. Yan, and Z.-G. Zhang, "Study on long-term load forecasting of mix-SVM based on rough set theory," *Power Syst. Protection Control*, vol. 38, no. 13, pp. 31–34, 2010.

[59] W.-C. Hong, Y. Dong, L.-Y. Chen, B. K. Panigrahi, and S.-Y. Wei, "Support vector regression with chaotic hybrid algorithm in cyclic electric load forecasting," in *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS)* (Advances in Intelligent and Soft Computing), vol. 130, K. Deep, A. Nagar, M. Pant, and J. Bansal, Eds. India: Springer, Dec. 2011, pp. 833–846, doi: 10.1007/978-81-322-0487-9_79.

[60] A. R. Javed, M. U. Sarwar, S. Khan, C. Iwendi, M. Mittal, and N. Kumar, "Analyzing the effectiveness and contribution of each axis of tri-axial accelerometer sensor for accurate activity recognition," *Sensors*, vol. 20, no. 8, p. 2216, Apr. 2020.

[61] N. Deepa, Q.-V. Pham, D. C. Nguyen, S. Bhattacharya, B. Prabadevi, T. R. Gadekallu, P. K. R. Maddikunta, F. Fang, and P. N. Pathirana, "A survey on blockchain for big data: Approaches, opportunities, and future directions," 2020, *arXiv:2009.00858*.

[62] A. P. Gopi, R. N. S. Jyothi, V. L. Narayana, and K. S. Sandeep, "Classification of tweets data based on polarity using improved RBF kernel of SVM," *Int. J. Inf. Technol.*, pp. 1–16, Jan. 2020.

[63] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu, "Interpreting CNNs via decision trees," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6261–6270.

[64] H. Chitsaz, P. Zamani-Dehkordi, H. Zareipour, and P. P. Parikh, ''Electricity price forecasting for operational scheduling of behind-the-meter storage systems,'' *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6612–6622, Nov. 2018.

[65] *ISO New England Data*, ISO Geneva, Switzerland, USA. Accessed: May 5, 2022. [Online]. Available: https://bigdata.seas.gwu.edu/data-set-18-iso-new-england-load-data-set/

[66] B. K. Tripathy, D. P. Acharjya, and V. Cynthya, ''A framework for intelligent medical diagnosis using rough set with formal concept analysis,'' 2013, *arXiv:1301.6011*.

[67] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, ''Short-term load forecasting with deep residual networks,'' *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, Jul. 2019.

[68] Z. Guo, K. Zhou, X. Zhang, and S. Yang, ''A deep learning model for short-term power load and probability density forecasting,'' *Energy*, vol. 160, pp. 1186–1200, Oct. 2018.

[69] B. K. Tripathy and K. R. Arun, ''A new approach to soft sets, soft multisets and their properties,'' *Int. J. Reasoning-Based Intell. Syst.*, vol. 7, nos. 3–4, pp. 244–253, 2015.

[70] V. Arzamasov, K. Böhm, and P. Jochem, ''Towards concise models of grid stability,'' in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2018, pp. 1–6.

[71] I. Pisica and M. Eremia, ''Making smart grids smarter by using machine learning,'' in *Proc. 46th Int. Univ. Power Eng. Conf. (UPEC)*. VDE: Soest, Germany, 2011, pp. 1–5.

[72] M. Hayati and Y. Shirvany, ''Artificial neural network approach for short term load forecasting for Illam region,'' *World Acad. Sci., Eng. Technol.*, vol. 28, pp. 280–284, Jul. 2007.

THAMER ALQUTHAMI (Member, IEEE) received the Doctor of Philosophy (Ph.D.) degree in electrical engineering with a minor in mathematics from the Georgia Institute of Technology. He is an Experienced Assistant Professor with a demonstrated history of working in higher education and industry. He has strong education professional. His research interests include smart grids, renewable energy, power system operation and control, complex system modeling and simulation, energy audit, energy efficiency and savings, and data analytics. He is skilled in PSCAD/EMTDC, Python, PTI/PSSE, RTDs, building automation implementation, R-Statistics, and C++.

MUHAMMAD ZULFIQAR received the B.Sc. and M.S. degrees in electrical engineering from Bahauddin Zakariya University, Multan, Pakistan. He is currently pursuing the Ph.D. degree with the UET, Lahore. He is also working as a Lecturer with the Department of Telecommunication systems, Bahauddin Zakariya University. His research interests include optimization, planning, energy management, and machine learning applications in smart/micro grids. He is a Lifetime Professional Engineer from the Pakistan Engineering Council.

MUHAMMAD KAMRAN joined the UET Lahore, in 1994, after attaining industrial experience by serving PEL, SIEMENS, and NESPAK. He has been a Professor with the Electrical, Electronics and Telecommunication Department, UET Lahore, New Campus, since December 2007. He is currently the Dean of the Faculty of Electrical Engineering. His responsibilities include monitoring academic activities in electrical engineering, computer engineering, computer science and biomedical engineering programs in all campuses of UET. Moreover, his responsibilities include teaching graduate and undergraduate courses and management of faculty. Further, his academic issues related to B.Sc. electrical and biomedical engineering technology are also dealt which include curriculum modifications and NTC accreditation. He is a HEC Approved Supervisor for M.Sc. and Ph.D. degree program. He is actively taking courses at the Center for Energy Research and Development at new Campus. He is a Nominated Curriculum Committee Member of the Higher Education Commission in Electronics. He is a member of selection board of various national universities. He has successfully supervised two Ph.D. candidates and 33 master's students. He is also supervising almost ten scholars in postgraduate studies in which three are Ph.D. students. He is looking forward to engaging students in various industrial and research-oriented projects.

AHMAD H. MILYANI received the B.Sc. (Hons.) and M.Sc. degrees in electrical and computer engineering from Purdue University, in 2011 and 2013, respectively, and the Ph.D. degree in electrical engineering from the University of Washington, in 2019. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include power systems operation and optimization, renewable and sustainable energy, power electronics, electric vehicles, and machine learning.

MUHAMMAD BABAR RASHEED (Senior Member, IEEE) received the master's and Ph.D. degrees from COMSATS University, Islamabad, in 2013 and 2017, respectively. He is currently working as a GET-COFUND Marie Curie Fellow at the UAH, Spain. Previously, he was working as an Associate and an Assistant Professor with the Department of Electronics and Electrical Systems, The University of Lahore, Pakistan. After that, he obtained Postdoctoral Fellowships from Durham University U.K., and King Abdulaziz University (KAU), Saudi Arabia, in the years 2019 and 2020. He has authored over 40 peer-reviewed papers in well-reputed journals and conference proceedings and supervised/supervising more than ten students in their final year projects and theses. His research interests include LP, NLP, and heuristic optimizations, machine learning, smart grids, electric vehicles, and demand response. He is an Active Reviewer of many esteemed journals and conferences, including the IEEE Transactions, IEEE Access, the IEEE Transactions on Industry Application Systems, *Applied Energy*, and *Energies*.