A Comparative Analysis of Machine Learning Algorithm for Short Term Load Forecasting in Smart Grid

Dr. P. Phanindra Kumar Reddy Department of Artificial Intelligence and Data Science, Annamacharya Institute of Technology and Sciences(Autonomous), Rajampet, Andhra Pradesh, India-516126.

E. Srimayee, P. Snehalatha, B. Dilip Kumar, B. Lokendra, Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences(Autonomous), Rajampet, Andhra Pradesh, India-516126

ABSTRACT:

The demand for electricity around the world has risen dramatically due to the rapid increase in world population. So, effective management techniques must be used, developing the optimization and control mechanism requires accurate energy demand estimation and results from short- and/or long-term forecasting with higher accuracy. In order to predict the future energy demand requirement with satisfactory outcomes, distributed demand response programmes and machine learning (ML) techniques are being used. Different cutting-edge machine learning (ML) algorithms have been implemented to analyse their performance, including logistic regression (LR), support vector machines (SVM), decision tree classifier (DTC), K-nearest neighbour (KNN). This paper's major goal is to give a comparative comparison of machine learning (ML) methods for short-term load forecasting (STLF) in terms of forecast error and accuracy. Among other algorithms, the DTC gives noticeably better outcomes based on implementation and analysis. The results of the implementation demonstrate that the suggested EDTC algorithm produces better forecast outcomes (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.

INTRODUCTION:

Due to the enormous growth in the global economy and population as well as the rising urbanisation of the world. Water, wind, solar cells, fossil fuels, thermal and nuclear reactors, and other sources of energy can all be used to produce electricity, a crucial source of energy. Additionally, as our population expands and advances, a greater amount of energy must be produced to meet the rising need for electricity.

Using a centralised network made up of thousands of units, traditional EGs operate as previously stated. The possibility of overhead generation is introduced by increasing the EG load, which could lead to problems with the quality of the electricity. In order to remedy this, fresh plants must be installed. On the other hand, these grids lack a trustworthy forecasting system for identifying the causes of intermittent power outages as well as their reaction times, memory requirements, and resource usage. The current electrical power system (PS) hasn't altered in a number of decades, according to scientists. A huge demand for electricity has been created by population growth. The classic PS has a number of drawbacks, including as poor visibility, slower-responding mechanical switches, and a lack of power control and monitoring. The need for new grid technology is influenced by a variety of factors, including changes in climatic conditions, component failure, the demand for energy, population growth, the use of fossil fuels, a decline in electric power output, a lack of energy storage, unilateral communication, and other problems. In order to address these problems, a new grid architecture is essential. In order to satisfy urgent needs and improve the quality of modern living, the smart grid (SG), a next-generation energy infrastructure, is thought to be essential. The typical EG offers limited one-way communication to energy users, while SG offers extensive two-way communication, according to a

comparison. However, in the case of SG, a rapid self-healing facility is available, whereas power quality difficulties in the standard EG are rectified slowly.

A. MOTIVATION:

Global electricity demand has significantly expanded as a result of the rapid population growth and industrial revolutions. The combination of traditional and distributed energy generation technologies, including photovoltaic energy, energy storage systems, and electric vehicles, is used to meet the increasing load demand as a result. They have, however, created significant issues for prediction and forecasting due to their integration into the main grid or domestic settings. Additionally, this is a result of changing consumption and load demand patterns. The management of load demand through active or passive participation of prosumers is also the subject of various works that have been presented. Without taking SG stability and control into account, the major goal was to lower energy consumption costs and customer unhappiness.

Additionally, to obtain remarkable performance, the forecaster and optimization module must be integrated. Several ML and data mining tasks employed a DT as a classifier. The decision about and alterations to hyper-parameters have a significant effect on the forecasting accuracy of ML systems. Consequently, tweaking hyper-parameters using ML models presents a significant challenge in terms of accuracy and efficiency. Due to the flaws and intrinsic constraints of each individual technique, these separate/single ML algorithms are not beneficial in all areas (accuracy, convergence rate, stability). They are dependent on arbitrary weights, biases, thresholds, and hyper-parameter tweaking. Performance is unsteady as a result of these issues with ELF.

B. REAL CONTRIBUTIONS

With this as a driving force, the EDTC forecasting model—a novel, reliable, and improved forecasting algorithm—is created in this study by fusing DTC with fitting function, loss function, and gradient boosting analysis. The originality and significant technological contributions are highlighted here.

- Modern machine learning (ML) methods like SVM, KNN, NN, DTC, and LR are compared for forecast accuracy. While among these classifiers, DTC provides more effective and efficient results with comparably high accuracy, good speed, and minimal memory consumption.
- Since hyperparameters have a significant impact on the ML algorithms' stable performance in ELF. The selection and modification of these parameters for precise and reliable performance is difficult. By improving random weights and bias initialization of the DTC, we developed a novel and upgraded DTC EDTC) to solve the hard-to-tune hyper parameters problem of the ML method. By incorporating a fitting function, a loss function, and gradient boosting into the DTC mathematical model for fine-tuning the control variables, the proposed EDTC increases accuracy.

1) NOVELTY ASPECT

A load forecasting perspective was used to compare various ML algorithms in this work. In order to examine the data in terms of features and classification parameters, the following algorithms are first implemented: SVM, KNN, LR, ANN, and DTC. The DTC produces output that is more precise and effective than those produced by other algorithms, according to the results, and does so relatively quickly. DTC's superior memory use to store the rule-set in smaller trees is one of the primary factors contributing to its high speed and accuracy. Due to the fact that it generates fewer rules to get the best output than other techniques, the classification process in DTC has also used less memory than those other techniques. Due to the growing pruned trees, the accuracy is also higher because there are less errors in cases that are not observed. As a result, we have determined that, if the control parameters are further adjusted based on comparative analysis, DTC could produce findings that are more accurate. In order to improve the error

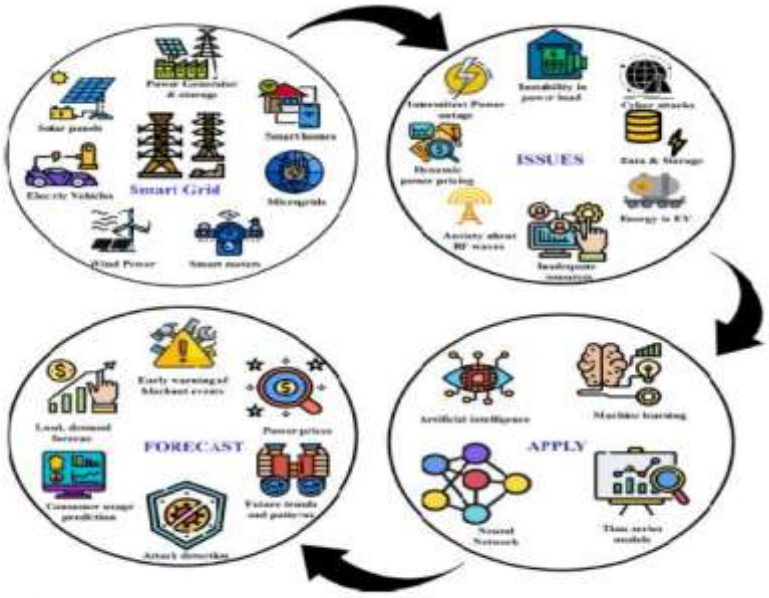


FIGURE 1. AI technology embedded into the SG context.

ratio, this work suggests an updated EDTC that combines cross-validation, model complexity, reduced error pruning, and feature selection. Dimensionality reduction uses the feature selection method.

C. PAPER ORGANIZATION

The paper's information is organised as follows. The motivation for using ELF and ML techniques to solve ELF difficulties is briefly explained in Section I. The analysis of the current research on the use of ML techniques for ELF is the focus of Section II. ML-models are shown in section III. Describes the developed approaches in Section IV. While part V presents the findings and debates from the simulation

II. LITERATURE SURVEY

This section gives a summary of the many study approaches, procedures, findings, and restrictions of the SGs literature that has already been published. A two-way communication system will be available on the SG, which will take the place of the final energy grid. With the aid of EVs, a complicated mechanism is used to distribute the electricity that has been transported from multiple sources. The SG modelling incurs more expense because it manages its features to maximise performance. Therefore, it is essential to regularly check stability, robustness, efficiency, and reliability under a range of operational circumstances. Researchers have used ML techniques like LR, KNN, SVM, ANN, random forest, ridge regression, gradient boosting, additional trees regressor, stochastic gradient descent, and gradient boosting to assess load in SG.

SGs are switching to providing their consumers with demand-based power supply services. Forecasting consumer load is necessary as a result. A goal of this study is to determine if the short-term load forecasting (STLF) framework currently in use or anthropological-structural data can accurately predict individual consumer home load. To find the best LF framework for a specific load, an STLF framework was created using anthropological structural data from home consumers. The created model has the ability to predict deviated loads using a certain instance at various time series. Back-propagation (BP), NN, and SVM were employed by the researchers to forecast the suggested STLF architecture. The findings show that the developed STLF reduces inaccuracy by 60% and is 7% more accurate than the SLTF. The study confirmed that using anthropological data will improve the SLTF model. The household data for the upcoming week was used to train an ANN to predict the daily energy use on an hourly basis.

Hernandez et al. developed an ELF-based ANN strategy in SGs that included three primary stages: segmentation using K-means classification, a self-organizing map (SOM) that employs pattern recognition, and demand forecasting (DF) in separate clusters. The ANN model was validated using real-time data from a Spanish company. The model was trained using weekly and monthly periodic values.

Benchmarks based on radial basis function NN and generalised regression NN underperformed this framework. The ability of the SG to provide uninterrupted electricity based on use is essential to its sustainability. DT algorithms based accuracy in newest literature survey. For MTLF and LTLF in the SG, Chen and Ahmad utilised three different ML frameworks. They employ a nonlinear ANN architecture that combines auto-regressive exogenous multivariate inputs, multivariate linear regression (MLR). Based on aggregated exhaustive consumption indicators, the researchers divided the load into three intervals: one month ahead, seasonal perspective, and one year ahead. While precisely defining energy differences, adjustments, and future energy forecast possibilities, the models improved predictability. LF was given instructions for computational and AI models. Wavelet transformation (WT)-ANN, wavelet transformation error correction (WTEC)-ANN, probabilistic NN (PNN), expert systems, and fuzzy logic (FL) were employed in AI frameworks along with ANN, RNN, ARIMA-SVM, GRNN, SVM, and generalised regression neural networks (GRNN). The study's findings showed that artificial intelligence-based forecasting algorithms performed better than other stochastic methods.

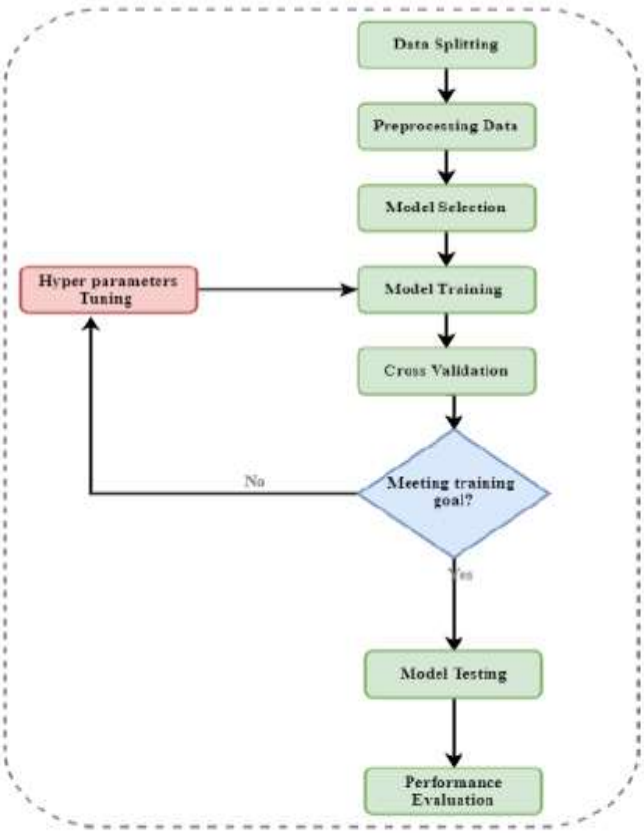


FIGURE 2. Flow chart of ML frameworks.

In order to address two common problems, noisy and non-stationary data, SVM is combined with the time series forecasting (TSF) module and utilised for financial analysis. Statistical models and GRNN were outperformed by SVM with sparse representation. SVR and chaotic GA combined to increase the accuracy of chaotic SVM performed better in predicting wind speed than MLP, too. The results reveal that the multi-directional LSTM (MDLSTM) framework developed by Alazab et al. outperforms traditional LSTM, RNN, and

gated recurrent units (GRUs) in predicting the SG's resilience. Prior research have shown that ML methods are generally beneficial for LF.

ML MODELS:

The supervised ML techniques built on regression are the subject of this paper. To predict the hourly load consumption in this study, all ML models are employee depicts the common flow used by all methods.

A. LR

Regression is carried out through the LR algorithm, a supervised ML method. The variables' linear relationship with one another is all that is determined. Multi-variable regression is used when the relationship between the two is nonlinear (MVR).

The LR is described in the manner shown below:

$$y = \theta 0 + \theta 1 \cdot x \qquad (1)$$

Here,

x - input training data
y - output(supervised learning)
θ0 - intercept
θ1 - coefficient of x (2)

It is found that the best-fitting forecast line has the ideal values of 0 and 1. Then, using these, the cost function is calculated. In essence, this model looks for the y value that has the smallest possible difference between the actual and predicted values. In order to reduce the error, it is necessary to update the values of 0 and 1. In Eq. 3, calculating costs Gradient descent (GD) is a common way to define z as follows:

Here, y_i stands for the value that really occurred, and z_i for the value that was anticipated. It is clear that this method merely returns the RMSE's z and y values. Here, the values of 0 and 1 are chosen at random, and they are modified with each iteration in order to lower the RMSE and, using GD, identify the best fit line for the model. In this instance, z_i is the expected value and y_i is the actual value. As can be seen, this method basically returns the RMSE's z and y values. Here, the values of 0 and 1 are randomly selected, and they are modified

$$F = \frac{1}{n}\sum_{i=1}^{n}(z_i - y_i)^2 \qquad (3)$$

with each iteration in order to lower the RMSE value and find the model's best fit line using GD.

C. KNN:

In data identification and consistency, KNN is often used for regression and classification. KNN belongs to a family of algorithms that uses supervised machine learning. This method of thinking about statistics is non-parametric. The training input is taken from a training block in both scenarios, after which a corresponding

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

target and output model is created. K-NN is an example of memory-based learning since conclusions are drawn directly from training instances. A recognised class's collection of objects serves as the basis for the neighbours. If K D 1, the class's solitary closest neighbour is given the assignment. The mean of its KNNs is the output of KNN regression. According to the following equations, KNN: Eq. 5 illustrates the Euclidean equation.

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

Manhattan equation m_e is presented in Eq.6:

$$\mathfrak{M}_{\mathfrak{e}} = \sum_{i=1}^{H} |\alpha_j - \beta_j| \qquad (6)$$

Minkows equation mk_e is presented in Eq.7:

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

D.SVM

Mathematically the classification method is represented as:

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

where the forecaster's inputs are represented by v_j ^∞(j=1,2,3,……) computed The distribution of the data and the classification variables define k, and the dimensionality of the space is denoted by D. Using SVM, a hyper-plane that separates a subspace of D dimensions filled with data points. In this study, the Eq. 8 de neds the hyper-plane. This leads to the following definition of the regularized risk function Rf:

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

The planned load consumption pattern is represented by L_j^a, and is the insensitive loss function parameter. It is necessary to minimize this R_f in order to acquire the parameter v. Following are the calculations for the robust error function u.

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

To minimise Eq. 9, Eq. 10 uses a function that can be described as follows:

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

When all I values are 0, The SVM kernel function displays how the radial basis KPCA is multiplied in the feature space f* as follows:

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

The K does away with the need to calculate Xj features in an unlimited feature space. It is possible to obtain the and by maximising the quadratic form:

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

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

The generalized versions of kernel functions are as follows:
(i). Linguistic kernel function: This function is used to give identical data points in a dataset I in conjunction with SVM.

$$\Re(r,\ z) = \langle r,\ z \rangle \qquad (15)$$

(ii). Logistic Sigmoid based Kernel function: The Hyperbolic tangent kernel, also referred to as the Logistic Sigmoid based kernel function, was created in the NN research field. In the vast majority of instances, NNs have been activated via the sigmoid function.

$$\Re(r,\ z) = \tanh\left(\mathrm{a}_0\langle r,\ z\rangle^d + \mathrm{a}_1\right) \qquad (16)$$

(iii). The ubiquitous concept known as the radial basis kernel (iii) is frequently used in a wide spectrum of ML methods that use kernels. SVM classification jobs are performed using it.

$$\Re(r,\ z) = \exp\left(-\theta\|r - z\|^2\right) \qquad (17)$$

E. DTC

One of the few classification methods that enables us to understand the full justification used by the classifier to make a particular classification is DTC. Depending on specific conditions, DTC presents a graphic representation of every choice that could be made. It begins with a root, then, like a tree, grows to include a variety of feasible solutions. The root node adds the training data set to the tree, and each following node then queries a feature with a true/false question. Two distinct subsets of the dataset have now been created. The entropy equation is given in Eq. 18.

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

The class label probabilities are shown by the notation (p_1, p_2, ... p_n). To segment the data in the suggested model, the Gini-index (GI) was utilised. It can be calculated by deducting the squared odds of each class from one. In contrast to information gain, which favours smaller divisions with a range of values, it favours larger, easier-to-implement partitions. Eq. 19 in the mathematics presents GI:

$$\mathfrak{GJ} = 1 - \sum (P(x = k))^2 \qquad (19)$$

where the chance that a target feature would take the value k is given by P(x = k).

F. PROPOSED EDTC:

An upgraded DTC is a DT that uses fitting functions, loss functions, and gradient descent analysis. The DT in this work generates initial values for multiple regression fitting functions, which can handle a high number of input variables. Then, a loss end function is used to calculate the errors between the observed data sets and the output values. Square-error, absolute-error, and unfavourable binomial log-likelihood functions are further common loss functions. Then, using the gradient boosting (GB) method, the titling function with the lowest predicted loss function value is found. To find the optimal fitting process, the prior phase is repeated.
When the value of the loss function (L(q; F(p)) is lowered, a fitting function (Fp) is selected after the input vector p and the output variable q, which contain training samples (pm; qm), are given. After R iterations, Fp is a linear combination of a collection of basis functions fr (p), as shown in Eq. 20:

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

Where the value c is a constant. To approach the required Eq. 20, the gradient boosting algorithm employs gradient descent analysis. As follows is a description of the specific technique: Using the least loss function L(qj,$^{\delta}$) as a starting point, we may find the constant coefficient $^{\delta}$0 as follows:

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

Using the recursive concept, we may solve this issue after obtaining f0(p). Eq. 22 is used to get F0(p), F1(p), and Fr1(p).

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

The sum of minimum loss function is derived as $^{\delta}_{r}$:

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

Algorithm 1 EDTC

Require:

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

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

3. R: Iteration times

For $r = 1 to R$

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

end

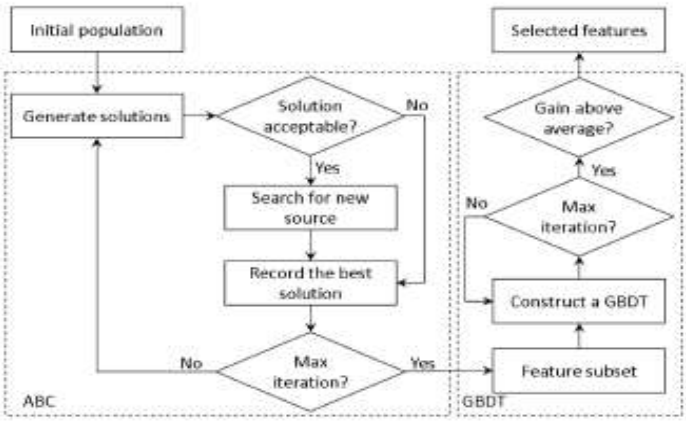


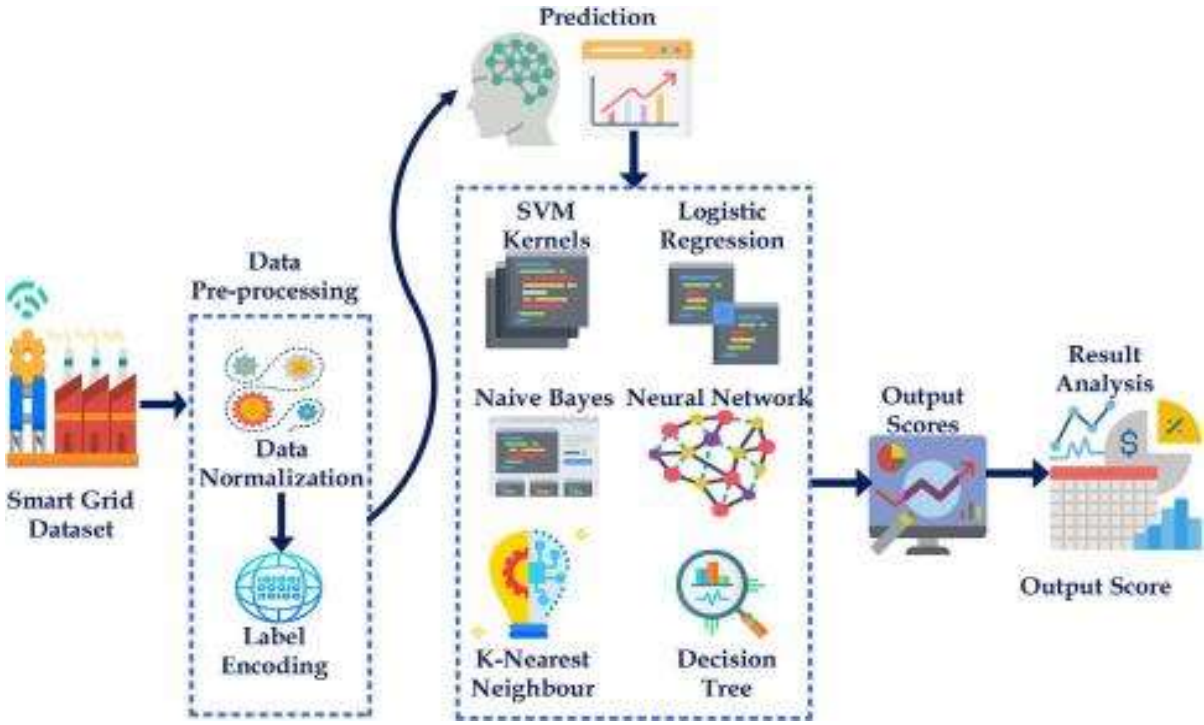
FIGURE 3. EDTC algorithm processes.

Fr_1 (p_j negative)'s gradient is added together to form fr (p):

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

PROPOSED EDTC DESCRIPTION

Using the information gain described in Eq. 19 as split- ting criteria to construct a DT, Method 1 is an enhanced top-down algorithm of DTC. Gain Rations, a modified version of information gain, serves as the EDTC criteria. Low error rates, little memory usage, and high optimization are advantages of EDTC. ETDC algorithm is consequently significantly faster and more precise. DT is created using the "divide and rule" method, and EDTC has structures akin to trees, prunes the original DT, and does so. Furthermore, the gradient boosting technique is what has improved DTC the most. Algorithm 1 shows that the purpose of the suggested EDTC is to obtain a boosting factor from the minimal loss function L(qi), as shown in Eq. 21. The basic idea behind the algorithm is to summon weak students repeatedly and give them high weight vote values. This has the effect of reducing bias by allowing the training process to concentrate more on the cases that led to error. The building of a cost-sensitive model as described in Eqs. 21 and 23 is another important aspect of EDTC. The boosting technique is EDTC's most important feature. Illustrates the suggested EDTC algorithmic procedure. Let the input data that is anticipated to be accurately classified, according to definition 1.



Proof: Each training case contains a tuple of values for a fixed set of attributes (or independent variables) A=A1,A2,…..,Ak and a class attribute. This data is used as the algorithm's input (or dependent variable). If attribute A2 is the EDTC, then it is classified as either continuous or discrete algorithm. The DTC only accepts attributes of the numerical or nominal kind.

IV. PROPOSED METHODOLOGY:

The four main parts of the designed system are shown in Fig. 4 together. The datasets are found in the first part and consist of four years' worth of freely downloadable NYISO datasets (2017-2020). We try to clean up the data in the second stage, known as pre-processing. Pre- filtering is essential for boosting the quality of the data and the significance of ML algorithms, both of which can help with successful forecasting. Engineering is

the third element, which increases classification accuracy, minimises data dimensions to prevent complexity, and shortens processing times. For feature engineering in the suggested system, two ML techniques are employed. Initially, DTC decides whether a characteristic is relevant. The datasets are then cleaned up using RFE to exclude characteristics with little significance. Accuracy can be improved by keeping the essential elements. Data are split into training and testing sets in a 3:1 ratio before being given to classifiers.

DATA-SET DESCRIPTION:

The experiments are performed with Python. The NYISO data are used for four years. In addition to system load information for each day and a number of other factors, the data includes 16 characteristics and 1095 instances. Due to the incompatibility of all factors, we first isolate the important qualities.

DATA SETTING AND PREPROCESSING

The success of ML algorithms depends on the quality of the data, which can be improved through pre-processing. The two pre-processing techniques that are typically used in ML models are normalisation and data transformation. The variables in an SG dataset are dispersed throughout multiple ranges, which frequently leads to a bias in favour of values with greater weights and lessens the efficacy of the developed framework. In the study, the load and temperature variables' data are normalised using a zero-mean normalisation approach since attribute normalisation speeds up convergence and increases numerical stability in NN training. Following Eq. 25, normalisation is carried out:

$$y'_j = \frac{x_j - \nu}{\kappa} \qquad (25)$$

If y'_j refers to the j instance's zero average score is the time series dataset's mean, and corresponds to the dataset's standard deviation, respectively. The mean and standard deviation of the training data were used to standardize the test data.

FEATURE ENGINEERING

Utilizing DTC-based FE, which treats feature patterns as vectors, the AEMO (NWS) data set's important features are chosen and outliers are reduced. These vectors have unique timestamps for each of the feature values. Features for EL are viewed as load demands, thus any features that have a negligible impact on EL forecasting may be eliminated. The relevance of features is determined using DT, the most sophisticated and effective feature extraction method. Each characteristic is given a score individually by the DT technique. Afterward, features are chosen using RFE (recursive features elimination) based on their score. displays the rating provided by DT. The feature selection threshold will be set at 0:5. (Tfs). Features may be maintained and used for additional processing when their grades are higher than the Tfs. Additionally, the grade of features that are smaller than the fixed Tfs value may be pitched.

Parameters	Observations			
$\mathfrak{T}_{fs}$	0.70	0.60	0.55	0.50
$\mathfrak{T}$	89s	98s	105s	121s
$\mathfrak{E}$	3.1%	3.0%	1.9%	1.4%
$\mathfrak{F}_0$	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			

The FS process is constrained such that the significance of feature extraction can be taken into account by distinct TFS weights.

PERFORMANCE METRICS

RMS, MSE, MAPE, and MAE are four statistical variables that gauge classification accuracy. The roles of RMS and MAPE are defined, whereas MSE and MAE have built-in functions. In order to assess the effectiveness of the ML algorithms, the study also takes performance factors like as recall, F1 score, precision, and accuracy into account. In the most irresponsible studies, training can be done with 70% of the SG data set while verification and validation can be done with the remaining 30%. In order to evaluate the ML frameworks and strengthen the confidence in the results curve and the aforementioned parameters are used.

V. SIMULATION RESULTS AND DISCUSSION

A. PREDICTION TRENDS AND STATISTICAL

MEASURES RESULTS

It is clear that DTC and EDTC closely reflect the true trend when the forecasting trend is compared to the actual trend. The excellent performance of these two algorithms is indicated by this. Also, EDTC performs better than DTC when it comes to trend following. In addition, several data indicators are used to illustrate the comparison. In comparison to other approaches, EDTC has a very low MAE, meaning it produces very little error. Additionally, the EDTC processing time is not too long in the scenario DTC is inferior to MAE for EDTC, which is ranked first. Compared to EDTC, KNN's error rate in MAPE is more erratic. EDTC performs better than every other strategy in this situation.

B. PERFORMANCE METRICS RESULTS

1) SVM RESULTS

The CM for SVM classifier demonstrates that in the stable class, we were able to achieve 90.1% forecasting accuracy with 977 records out of 1084 and 9.9% FPR, while in the unstable/failed class, we were able to achieve 91.80% accuracy and FPR with 1759 out of 1916 records and 8.2%, respectively. According to the classification report (CR) for SVM, the stable class had an F1 score, precision, and recall of 86.4%, 83.00%, and 90.10%, respectively, whereas the unstable class had F1-Measure scores, precision, and recall of 91.80%, 94.10%, and 89.60%, respectively. The ROC for the SVM classifier, with an AUC of 90.21%.

2) KNN RESULTS

The CM for KNN classification, we were able to forecast with an accuracy of 64.7% in the stable class (701/1084) and an FPR of 35.3% in the unstable/failed case (86.60%/1641/ 1916), respectively. According to the classification report (CR) for KNN, we achieved F1 scores of 68.1%, 71.80%, and 64.70% in the stable class, while these values were 83.30%, 81.10%, and 85.60% in the unstable group.

3) DTC RESULTS

The CM for DTC classifier demonstrates that in the stable class case, we achieved 96.8% predicting accuracy with 1050 records out of 1084 and 3.13% FPR, while in the unsteady/failed situation, the accuracy and FPR achieved are 96.50% with 1850 out of 1916 records and 3.44% respectively. The report (CR) for LR, we achieved F1 scores, precision, and recall of 95.9%, 92.10%, and 93.20% in the stable class, while these values were 96.10%, 94.30%, and 96.90% in the unstable group, with an AUC of 90.21 percent, shows the ROC for the DTC classifier.

4) EDTC RESULTS

In the stable class, as shown by the CM for EDTC, we were able to forecast with 100% accuracy using 1084 records out of 1084 and 0% FPR, while in the unstable/failed class, we were only able to reach 99.90% accuracy using 1915 out of 1916 records and 0.1% FPR. The classification report (CR) for EDTC. We obtained 100%, 99.9%, and 100% F1 scores, precision, and recall in the stable class, compared to 100%, 100%, and 99.00% in the unstable group for F1-Measure scores, precision, and recall. With an AUC of 99.95.

TRAINING AND TESTING LOSS AND ACCURACY

SVM testing accuracy is 97.50% with 0.07 of a data loss, while training accuracy and loss are respectively 97.20% and 0.07 in Fig. 8c. Training and testing accuracy for the NN classifier are 98.45% and 98.90%, respectively, with a data loss of 0.07 percent. For LR, accuracy is 97.98% and 98.50%, respectively, with data loss of 0.05 for training and testing. 94.80% and 94.23%, respectively, of KNN training and testing are effective. Using KNN, the data loss is 0.08 for both training and testing. For training and testing, EDTC obtained a precision of 99.07% and a loss of 0.02. When compared to SVM, KNN, NN, and LR, the accuracy achieved by the EDTC was 1.98 percent greater.

C. METRICS PERFORMANCE IN CLASSIFICATION REPORT

In terms of precision, recall, and F1-score, for stable class, the EDTC outperformed the ML models mentioned above with a 100.00% F1-score, 99.00% recall, and 100% precision. The EDTC model outperformed the conventional models in the unstable class in terms of precision, recall, and F1-score as well.

D. ACCURACY ACHIEVED BY PROPOSED EDTC

Considering prediction accuracy, recall, precision, and F1-Measure, ETDC surpasses other algorithms employed in this paper. This is because ETDC is a probability-based method. The authors respectively used MLSTM, EKNN to reach 99.01%, 97.82%, and 95.37% AUC, while the proposed EDTC achieved 99.42% AUC.
Observation 2: Based on the findings, the following interpretations can be made:
Because ML techniques are more compact than DL models, they are better suited for classifying the SG dataset.

VI. CRITICAL ANALYSIS

Beyond performance, ML algorithms provide a number of advantages over other forms of AI and traditional models for LF, including the ability to handle noise, generate patterns rather than rely on assumptions, handle non-linearity, and be simple to apply. We used a variety of techniques, but (EDTC, SVM, LR, KNN, and NN).

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

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

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

VII. CONCLUSION AND FUTURE SCOPE

Effective power distribution to the control stations depends on the SG's stability. The durability of the SGs is fundamentally represented by ML approaches. Finding the ideal algorithm to forecast the stability of the SG is the main difficulty posed by the emergence of various ML algorithms. To achieve this, a thorough analysis of cutting-edge ML algorithms has been conducted to forecast the stability of SGs. In order to forecast the stability of the smart grid, unique EDTC model is presented in this work. On the NYISO smart grid dataset, the proposed model has undergone testing. Traditional ML models including SVM, KNN, NN, LR, and DT are used to compare the performance of EDTC. . The experimental findings demonstrated the superior performance of the DTC algorithm over SVM, KNN, LR, and NN

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 behaviour,'' 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 neuralnet 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, ``Modelling 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 fruity optimization algorithm,'' Neurocomputing, vol. 221, pp. 24_31, Jan. 2017.
7. A. Kavousi-Fard, H. Samet, and F. Marzbani, ``A new hybrid modified 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. format., vol. 7, no. 4, pp. 529_539, Nov. 2011.