2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-3509-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCNT56998.2023.10307585

Electricity Demand Forecasting Using an Ensemble Model with Feature Selection

Banoth Prashanth
Electrical Engineering Department
National Institute of Technology Warangal
Warangal, India
banothpse@gmail.com

N.V. Srikanth
Electrical Engineering Department
National Institute of Technology Warangal
Warangal, India
nvs@nitw.ac.in

***Abstract*—Accurate forecasting empowers electricity stakeholders to anticipate and arrange operational requirements for electricity production, ensuring reliable power. In this paper, a novel ensemble forecasting model proposed using the outcomes of Random Forest (RF) and Bidirectional Gated Recurrent Unit (Bi-GRU) model are fed to an XGBoost model as input. The input data contains temporal and weather variables,by applying filter, wrapper, and embedded methods a set of common selected features has been identified. With appropriate data pre-processing, it improves the forecasting accuracy. The performance of the proposed model is compared with that of the RF, LightGBM, and XGBoost models, using evaluation metrics as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of determination (R^2). The performance metrics of proposed ensemble model is R^2 98.84% , RMSE 0.5401, and MAE 0.3738 excels the other comparable models.**

***Keywords*—Short-term load forecasting (STLF), Ensemble model, Feature Selection, Bi-GRU.**

I. Introduction

Growth in Electricity demand is one of the key indicators of nation's economic and social development. Electric load forecasting can be classified based on the timeframe into short-term (a few hours to a week), medium-term (a few weeks to several months), and long-term (up to several years) [1]. Accurate forecasting aid the utilities of power industry in planning, scheduling, operation, expansion etc. long term and medium term mainly aid the distribution companies in making power purchase agreements (PPA) and generation and transmission companies for future expansions. Short-term load forecasting (STLF) helps in understanding temporal and seasonal load patterns, efficient resource utilization like renewables, and decision-making in Day ahead and spot electricity markets.

Traditional and statistical methods like ARIMA, SARIMA, etc. for forecasting are not capable of handling the larger data and unable to identify the temporal patterns in demand. To overcome drawbacks of traditional and statistical methods, Machine learning (ML) models provides a better insights for capturing temporal and seasonal variation in data, and can able to handle larger data having nonlinearities [2]. To enhance the performance of predictive ML models, researchers have proposed hybrid or ensemble approaches that combine multiple models, integrating the strengths of each model [2]. Ensemble methods can be broadly classified into three types: Bagging, Boosting, and Stacking, with homogeneous and heterogeneous methods utilizing the same or different base models respectively [2].

Huan et al. proposes a STLF model that utilizes a Bi-GRU model by considering weather data [3]. Y. Xuan et al. the authors employed a multi-model fusion combining CNN, Bi-GRU model, utilizing random forest as a feature selector [4]. Siva et al. utilize a hybrid approach that combines filter and wrapper techniques. It uses RReliefF and mutual information filters to eliminate uninformative features, while wrapper-based RFE with SVR as the evaluator fine-tunes the selected features to reduce overfitting and identify the most optimal feature subset. [5]. Gao et al. proposed stacking ensemble model of LightGBM and LSTM algorithms and also compared model performance before and after fine tuning [6]. Ghareeb et al.illustrated averaging the prediction results of models, it improved STLF performance compared to individual models [7].

To improve short-term load forecasting accuracy, researchers frequently employ weather data and apply efficient data pre-processing and hybrid modelling techniques. These strategies can effectively enhance the accuracy of load forecasting models. The objective of this work is to perform precise data pre-processing by considering the temporal and weather variables. The proposed approach combines the strengths of both the RF and Bi-GRU models by merging their outputs used as input to the XGBoost model.

The paper is organized as follows: In Section II, describes proposed model and all the methodologies used in paper. Section III outlines the data and pre-processing steps performed. In Section IV, we present the experimental results. Finally, Section V provides the conclusion of our work.

II. Proposed Methodology

Both Bagging and Boosting are ensemble learning techniques used in machine learning to improve the accuracy and robustness of models. Bagging (Bootstrap Aggregating) [8] involves building multiple independent models in parallel trained from random subsets obtained by sampling of rows from the training data and then averaging the predictions of each model to obtain the final prediction. Boosting, on the other hand, involves building a sequence of models that

are trained iteratively on modified versions of the data. In each iteration, the model is trained on the data that was not accurately predicted in the previous iteration. Random Forest (RF), LightGBM and XGBoost are ensemble models that use homogeneous learners based on Decision Trees. These models are known for their high accuracy and performance in a wide range of machine learning tasks, including classification and regression, these models are capable of handling large datasets, missing values, robust to outliers, noisy data and provide feature selection capabilities. RF uses an out-of-bag error estimate, while LightGBM and XGBoost use regularization techniques and early stopping to reduce overfitting.

Random Forests [9] is an ensemble learning method based on Bagging. It employs row and feature sampling with replacement from the training data to train multiple Decision trees, outputs of Decision Tree are aggregated to obtain final prediction, it utilizes two types of randomness to improve the model performance, to prevent overfitting and avoid correlation. Firstly, a random sample is drawn from the original data to build the tree. Secondly, at each node of the tree, a subset of features is randomly chosen to identify the optimal split.

XGBoost [10] is an implementation of boosting with decision trees as learners, known for its computational speed, efficiency and auto pruning of regression trees. Widely used in many machine learning competitions. XGBoost is a regularized model that prevents overfitting and incorporates techniques like shrinkage and feature subsampling to further prevent it. It also handles sparse data and instance weights through sparsity aware algorithms and weighted quantile sketches. XGBoost is scalable for real-world problems due to its parallel and distributed computing and out-of-core computation capabilities, requiring minimal resources.

LightGBM [11] is proposed in 2017, it incorporates Gradient-based One-Side Sampling, Exclusive Feature Bundling, and with its unique leaf-wise growth strategy and depth constraints, LightGBM is an excellent choice for higher accuracy, less memory usage and training large data with high feature dimensions. LightGBM has faster training speed due to histogram-based splitting, LightGBM is based on boosting technique utilizing decision trees as the base learner.

GRU [12] is a type of recurrent neural network (RNN) that shares similarities with the LSTM architecture. Unlike the LSTM, separate cell state is not present in GRU, making it computationally less expensive, easier to train, as it has fewer parameters.The GRU consists of two gates: namely the reset gate (r_t) and the update gate (z_t), illustrated in Figure 1. The reset gate controls the amount of previous hidden state to forget, while the update gate regulates the inclusion of new input in the current hidden state. With inputs including the previous hidden state (h_{t-1}) and the current input (x_t). The GRU computes the new hidden state (h_t) by processing them through the reset and update gates as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{1}$$

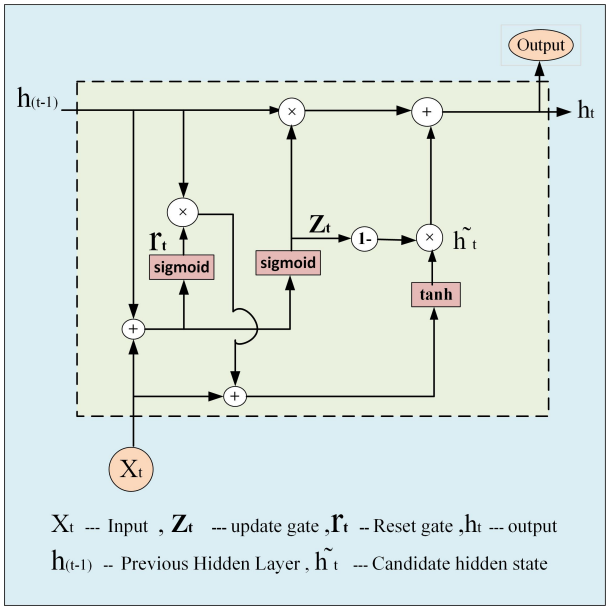

Fig. 1. A GRU Cell.

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{2}$$

$$\tilde{h} = \tanh(W x_t + U(r_t \cdot h_{t-1})) \tag{3}$$

$$h_t = z_t h_{t-1} + (1 - z_t)\tilde{h} \tag{4}$$

σ is a sigmoid activation function, U and W refer to the weights associated with the hidden and input, respectively.

Bidirectional Gated Recurrent Unit (Bi-GRU), is a variation of GRU where two independent GRUs are trained on the input sequence, one in a forward direction and the other in a backward direction. This allows the model to take into account not only the past information but also the future information when predicting the next output. Whereas a regular GRU only has access to information from the past. By incorporating this bidirectional approach, the model can capture complex temporal dependencies in the data, resulting in more precise and reliable predictions.

The proposed novel ensemble model employs an architecture in which the outputs from both the RF and Bi-GRU model are concatenated together through a Merging layer. The merged output is then fed as input to the XGBoost algorithm. The overall structure of the ensemble model is visually represented in Figure2. Before feeding the data into the Random Forest (RF) and Bidirectional Gated Recurrent Unit (Bi-GRU) models, the input data contains both temporal variables (such as day, time, week, etc.) and weather variables,these variables are brought into a structured format through a data preprocessing step. The data has been divided into two sets, a training set consisting of 80% of the data, which will be used to train the model, and a test set consisting of remaining 20% of the data, which will be used to evaluate the performance of

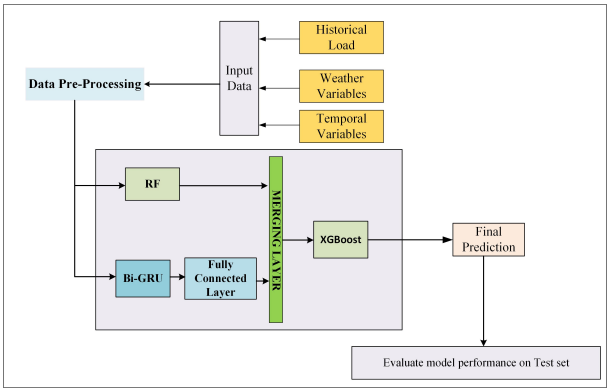

Fig. 2. An Ensemble Model.

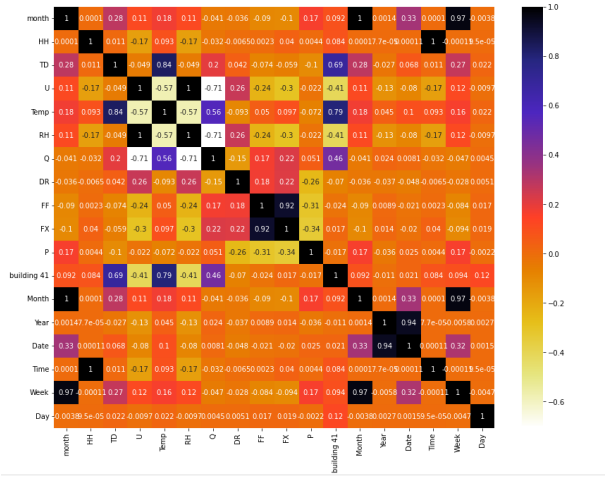

Fig. 3. Heat map.

the model.The training data underwent a data pre-processing stage. The Bi-GRU model is configured with 64 units and employs the hyperbolic tangent (tanh) activation function. The Fully connected layer, which follows the Bi-GRU model, contains a single unit with a linear activation function. A batch size of 32 was used to train the Bi-GRU model over 100 epochs. Table I provides a list of hyperparameters used in the RF, and XGBoost models. The models in this study were developed in Python 3, the implementation was carried out on a Windows 11 desktop with a 2.6 GHz Intel Core i5 processor and 16 GB RAM.

III. Data Transformation and Preparation

Data Transformation and Preparation entails the transformation of raw data into a structured format by handling missing values, addressing outliers, normalizing data, and performing feature engineering. EDA, on the other hand, involves visualizing and analysing data to extract insights, patterns, and relationships between variables using statistical and visualization techniques [6].

A. Data

The dataset used for this study was sourced from [13], and it contains electrical load data for the Netherlands country, along with the corresponding weather data from The Royal Netherlands Meteorological Institute (KNMI) for the years 2016, 2017, and 2018. The weather data comprises various variables such as Temperature (Temp), Humidity (U), Dew-point Temperature (TD), Rainfall Quantity (RH), Rainfall Duration (DR), Solar Irradiance (Q), Wind Speed (FF), Wind Gust (FX), and Pressure (P). Furthermore, in addition to the load data, relevant temporal information such as day, date, month, year, and time was extracted for further analysis.

B. Feature selection

Feature selection is the process of selecting a subset of relevant features from a larger set. Feature selection methods such as, Filter, Wrapper [5], embedded methods [14] are applied. Dimensionality reduction method reduce the number of features by transforming the data into a lower-dimensional space while preserving the most important information [15].Various feature selection techniques such as Pearson correlation, Recursive feature elimination with cross validation(RFECV), and embedded methods with Decision Tree, RF, and XGBoost are employed. By comparing the results obtained from various methods, we can identify the most consistent features that are selected across all methods.

Filter methods employ statistical tests to evaluate the relationship between features and the target variable. We utilized the Pearson correlation coefficient to perform feature selection. This involved computing the correlation between each feature and the target variable [16], [17]. Highly correlated features were removed to avoid overfitting and improve model interpretability. We visualized the Pearson correlation values of our data using a heat map, as shown in Figure 3, based on these correlation values highly correlated features can be obtained. Wrapper methods use a model to evaluate the performance of subsets of features, RFECV with Decision Tree is a feature selection method that iteratively removes unimportant features while utilizing cross-validation to evaluate model performance [18]. Embedded feature selection is a technique that enables the automatic identification and selection of the most relevant features during the training of machine learning model, such as Random Forest, and XGBoost [15], [19] . Figures 4, and 5 display the feature importance of the Random Forest, and XGBoost models respectively. To assess the relevance of the selected features, dimensionality reduction techniques can be employed. If the model's performance improves after reduction, it may indicate redundant or noisy features, while deterioration may suggest important information was lost during reduction.

C. Normalization

Normalization mitigates potential issues that may arise due to differences in feature scales by rescaling the data, StandardScaler is a widely used scaling technique that rescales the data such that the mean of the data becomes 0 and the

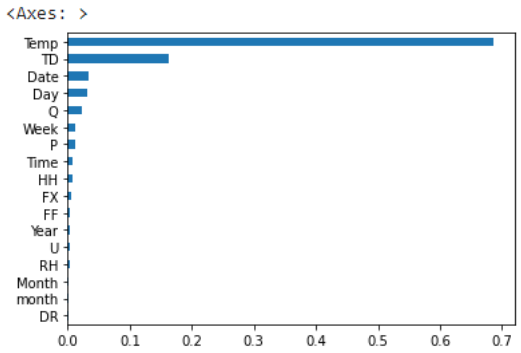

Fig. 4. Random Forest feature importance.

standard deviation become 1, making it insensitive to outliers. This is achieved by applying the formula in equation 5.

$$Scaled\ data = \frac{X - \mu}{\sigma} \tag{5}$$

Where X is actual data, μ, σ represents the mean and standard deviation of X respectively.

Outliers can significantly impact the accuracy of the model, which can arise from measurement errors, data entry errors. One common method to identify and remove outliers is the Interquartile Range (IQR) approach, where data points beyond 1.5 times the IQR above the upper quartile or below the lower quartile are removed [1]. The Yeo-Johnson power transformation is applied, which brings in the normality of the distribution and removes skewness.

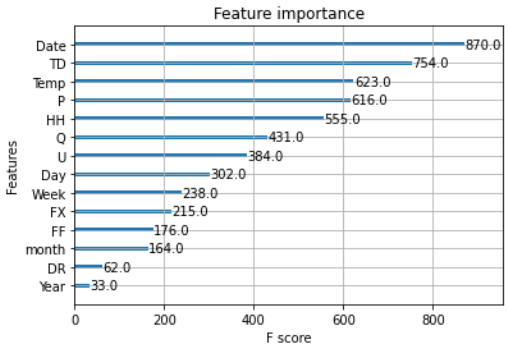

Fig. 5. XGBoost feature importance.

D. Hyper parameter tuning

Hyper parameter tuning helps to find the best set of hyper parameters for a given model on a given dataset, to avoid overfitting and under fitting issues. Grid Search involves an exhaustive search over a predefined hyper parameter space, despite its computational cost, Grid Search remains a popular method for hyperparameter tuning, especially for smaller hyperparameter spaces and datasets. To optimize the search process, multiple iterations are run with variations in the search space values, allowing for the selection of the best parameters. However, for larger hyperparameter spaces or massive datasets, the computational expense of Grid Search can become a significant challenge [1].

IV. Results and Discussions

Evaluating the performance of models, using metrics such as the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Coefficient of determination (R^2). RMSE, MAE, R^2 are defined in Equation 6, 7 and 8 respectively as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_p)^2} \tag{6}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_p| \tag{7}$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y_p)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_p)^2} \tag{8}$$

Where y_i and y_p represents actual and predicted load values, n is the total number of observation.

TABLE I
The Optimal Set of Hyper Parameters for the Models

Models	Hyperparameters
RF	Max_features=3,Max_depth=32,Min_samples_leaf=1, Min_samples_split=2,n_estimators=398
XGBoost	Learning_rate=0.066,Max_depth=11,n_estimators=799, Min_child_weight=5, reg_alpha=0.5,reg_lambda=1
LightGBM	Learning_rate=0.102,Max_depth=17,n_estimators=979, reg_alpha=1,num_leaves=67

TABLE II
Performance of Models Before(B) and After(A) Tuning Parameters

Models	**RMSE_ B/ A**	**MAE_ B/ A**	**R^2_ B/ A**
RF	1.189/ 1.160	0.774/ 0.769	0.9438/ 0.9468
XGBoost	1.087/ 0.946	0.752/ 0.619	0.9530/ 0.9644
LightGBM	1.199/ 0.9626	0.848/ 0.6456	0.9428/ 0.9632

TABLE III
Performance of Models on the Test Data and Model's Training Time

Models	**RMSE**	**MAE**	**R^2**	**Training Time**
Random Forest	1.1609	0.7693	0.9468	18 sec
XGBoost	0.9465	0.6195	0.9644	28 sec
LightGBM	0.9626	0.6456	0.9632	14 sec
Ensemble Model	0.5401	0.3738	0.9884	180 sec

Table II presents a comparison of the performance of the models before and after hyper parameter tuning. The results clearly demonstrate a significant improvement in the model's performance after the hyper parameter fine-tuning. The outcomes of the models on the test set are presented in Table

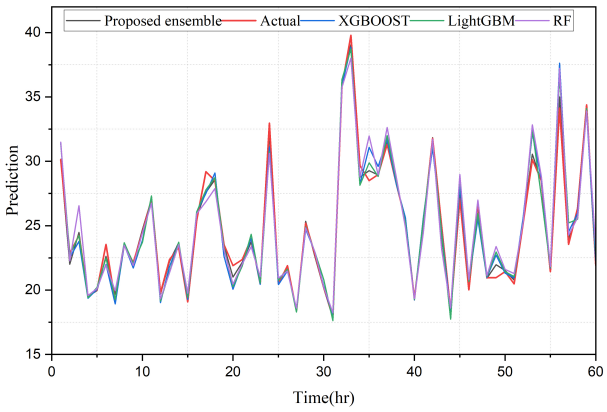

Fig. 6. Performance of models on test data.

III and Figure 6. Based on comparison of metrics Ensemble model outperformed other algorithms.

V. Conclusion

In this paper, we conducted a performance analysis of proposed ensemble model in comparison to individual models, and our results demonstrated that proper data pre-processing such as normalization, feature selection, and outlier removal is crucial to improve prediction accuracy. Specifically, the ensemble model exhibited significantly higher prediction accuracy than individual models, with RMSE, MAE and R^2 values of 0.5401, 0.3738, and 98.84% respectively. Utilizing the Grid Search technique allowed us to optimize the hyperparameters of the model's and ultimately enhance its performance. In summary, our paper provides valuable insights into the use of ensemble models, data Normalization, and hybrid feature selection techniques to improve load forecasting accuracy in the field of energy prediction.

References

[1] Vardhan, B.V.S.; Khedkar, M.; Srivastava, I.; Thakre, P.; Bokde, N.D. A Comparative Analysis of Hyperparameter Tuned Stochastic Short Term Load Forecasting for Power System Operator. Energies 2023, 16, 1243. https://doi.org/10.3390/en16031243.

[2] Bento, P.M.R.; Pombo, J.A.N.; Calado, M.R.A.; Mariano, S.J.P.S. Stacking Ensemble Methodology Using Deep Learning and ARIMA Models for Short-Term Load Forecasting. Energies 2021, 14, 7378. https://doi.org/10.3390/en14217378.

[3] He, Huan & Wang, Haomiao & Ma, Hongliang & Liu, Xuesong & Jia, Yilin & Gong, Gangjun. (2020). Research on Short-term Power Load Forecasting Based on Bi-GRU. Journal of Physics: Conference Series. 1639. 012017. 10.1088/1742-6596/1639/1/012017.

[4] Y. Xuan et al., "Multi-Model Fusion Short-Term Load Forecasting Based on Random Forest Feature Selection and Hybrid Neural Network," in IEEE Access, vol. 9, pp. 69002-69009, 2021, doi: 10.1109/ACCESS.2021.3051337.

[5] Siva Sankari Subbiah, Jayakumar Chinnappan,"Deep learning based short term load forecasting with hybrid feature selection",Electric Power Systems Research,Volume 210,2022,108065,ISSN 03787796,https://doi.org/10.1016/j.epsr.2022.108065.

[6] Gao, W., Huang, X., Lin, M., Jia, J. and Tian, Z. (2022), "Short-term cooling load prediction for office buildings based on feature selection scheme and stacking ensemble model", Engineering Computations, Vol. 39 No. 5, pp. 2003-2029. https://doi.org/10.1108/EC-07-2021-0406

[7] A. Ghareeb, H. Al-bayaty, Q. Haseeb and M. Zeinalabideen, "Ensemble learning models for short-term electricity demand forecasting," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 2020, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325623.

[8] Breiman, L. Bagging predictors. Mach Learn 24, 123–140 (1996). https://doi.org/10.1007/BF00058655

[9] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

[10] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[11] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: a highly efficient gradient boosting decision tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 3149–3157.

[12] Cho, K., Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. (2014)

[13] https://www.kaggle.com/datasets/anasanand/energy-data

[14] Jiaqi Shi, Chenxi Li, Xiaohe Yan,"Artificial intelligence for load forecasting: A stacking learning approach based on ensemble diversity regularization",Energy,Volume 262, Part B,2023,125295,ISSN 0360-5442,https://doi.org/10.1016/j.energy.2022.125295.

[15] Dimitrios Effrosynidis, Avi Arampatzis,"An evaluation of feature selection methods for environmental data",Ecological Informatics,Volume 61, 2021,101224,ISSN 1574-9541,https://doi.org/10.1016/j.ecoinf.2021.101224.

[16] Abumohsen, M.; Owda, A.Y.; Owda, M. Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms. Energies 2023, 16, 2283. https://doi.org/10.3390/en16052283

[17] Woohyun Kim, Yerim Han, Kyoung Jae Kim, Kwan-Woo Song,"Electricity load forecasting using advanced feature selection and optimal deep learning model for the variable refrigerant flow systems",Energy Reports,Volume 6,2020,Pages 2604-2618,ISSN 2352-4847,https://doi.org/10.1016/j.egyr.2020.09.019.

[18] Zoufal, C., Mishmash, R., Sharma, N., Kumar, N., Sheshadri, A., Deshmukh, A., Ibrahim, N., Gacon, J. & Woerner, S. Variational quantum algorithm for unconstrained black box binary optimization: Application to feature selection. *Quantum*. **7** pp. 909 (2023,1), https://doi.org/10.22331

[19] Ahmad Alsahaf, Nicolai Petkov, Vikram Shenoy, George Azzopardi,A framework for feature selection through boosting,Expert Systems with Applications,Volume 187,2022,115895,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2021.115895.