Big Data Analytics based Short Term Electricity Load Forecasting Model for Residential Buildings in Smart Grids

Inam Ullah Khan[1,*], Nadeem Javaid[2], C. James Taylor[1], Kelum A. A. Gamage[3] and Xiandong Ma[1]
[1]*Engineering Department, Lancaster University, Bailrigg, Lancaster LA1 4YW, UK*
[2]*Department of Computer Science, COMSATS University Islamabad, Islamabad, 44000, Pakistan*
[3]*School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK*
*Correspondence: i.u.khan@lancaster.ac.uk

***Abstract*—Electricity load forecasting has always been a significant part of the smart grid. It ensures sustainability and helps utilities to take cost-efficient measures for power system planning and operation. Conventional methods for load forecasting cannot handle huge data that has a nonlinear relationship with load power. Hence an integrated approach is needed that adopts a coordinating procedure between different modules of electricity load forecasting. We develop a novel electricity load forecasting architecture that integrates three modules, namely data selection, extraction, and classification into a single model. First, essential features are selected with the help of random forest and recursive feature elimination methods. This helps reduce feature redundancy and hence computational overhead for the next two modules. Second, dimensionality reduction is realized with the help of a t-stochastic neighbourhood embedding algorithm for the best feature extraction. Finally, the electricity load is forecasted with the help of a deep neural network (DNN). To improve the learning trend and computational efficiency, we employ a grid search algorithm for tuning the critical parameters of the DNN. Simulation results confirm that the proposed model achieves higher accuracy when compared to the standard DNN.**

***Index Terms*—Big data, Electricity load forecasting, Feature engineering, Classification, Smart grid**

I. INTRODUCTION

Electricity is an expensive commodity and its consumption must be synchronized with the generation to avoid wastage. Today's technologies do not allow to store or queue extra energy in an economical manner. Also, due to the limited transmission capacity of the existing power network, it cannot be transported to other regions and hence makes electricity characteristics local and time-varying in multiple aspects among different regions. Electricity load in its nature is one of the volatile and unpredictable commodities, and it can rise to tens and sometimes hundreds of times to its average value. The under/overestimation of power generation/consumption can pose severe challenges to the power system network. In other words, accurate forecasting and reducing the mean absolute percentage error (MAPE) only by 1% is so impressive and meaningful to have the impact of 3–5% on the generation side. The overall impact of this decrease in generation can reduce the generation cost of about 0.1% to 0.3% [1]. For this reason, various Artificial Intelligence (AI) and Machine Learning (ML) forecasting models are proposed to achieve better accuracy in the power market.

In a deregulated environment of the power industry, the role of electricity demand forecasting has become increasingly important in the smart grid. The primary purpose of price/load prediction is to minimize power demand peaks and to balance the supply-demand gap. Among numerous forecasting methods, short term load forecasting (STLF) aims to predict the load from several minutes up to hours and weeks into the future. An accurate and stable STLF strategy brings an unprecedented level of flexibility for its management and creates a win-win situation for both its generation and consumption side stakeholders. The electricity load is affected by many factors such as generation capacity, fuel prices, renewable generation, etc., and most of the factors vary within short intervals. Accurate forecasting is essential, but due to the more extensive data, it is challenging to increase accuracy. Smart meters continuously monitor the associated factors such environment, RES generation, temperature, etc., all in real-time; however, the amount of data available for forecasting is considerably large and hence difficult to handle, especially for STLF [2].

Residential buildings account for 20 – 40% of total energy demand, and hence making buildings energy efficient is essential for sustainable development of electric power systems. Apart from a major source of energy consumption, buildings are also identified for a substantial amount of energy wastage. Hence, the role of STLF is critical to minimize energy wastage at the building level and mitigating uncertainties for the reliability of the grid [3].

Since the early 1990s, AI techniques have been widely explored for STLF, and one of the popular AI methods is Neural Network (NN). The NNs prediction models provide promising prediction results, and that is the reason that they are extensively used in different applications. However, NNs undergoes a number of weaknesses, which includes overfitting issue, estimation of connection weight, model construction, and consideration of extensive data for model training. Due to these reasons, it is challenging to employ NNs for STLF problems [4]. In 1995, turkey *et al.* [5] proposed an inno-

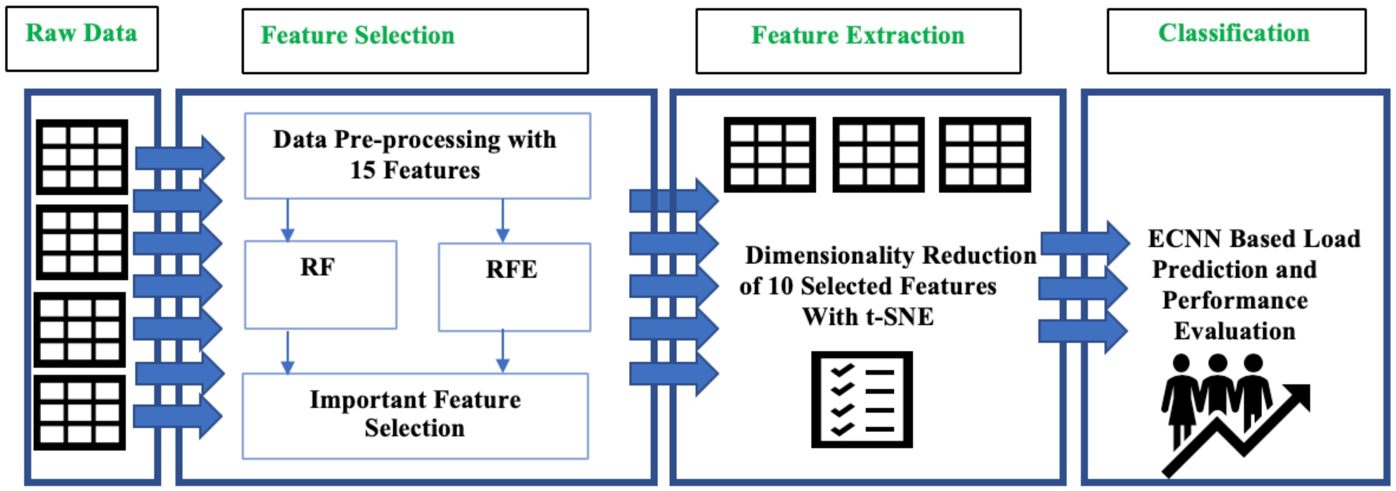
Raw Data
Feature Selection
Feature Extraction
Classification
Data Pre-processing with 15 Features
RF
RFE
Important Feature Selection
Dimensionality Reduction of 10 Selected Features With t-SNE
ECNN Based Load Prediction and Performance Evaluation

Fig. 1: **Proposed system model**

vative AI technique they called support vector machine and support vector regressor to address the shortcoming of NNs. These methods employ empirical risk minimization principles to improve the training process and find globally optimal solutions in the search space. However, these methods are computationally costly and hence make the algorithm difficult to converge. Also, these methods are not suitable for large data sets and difficult to perform when the values of the training class are overlapping.

For STLF strategies, most of the work is based either on selection or classification methods where Decision Tree (DT) algorithms and Artificial Neural Networks (ANNs) have gained much attention. Both methods have limited capabilities such as DT faces overfitting problems, which means that model performance is good in training but not in prediction. Similarly, ANN models have limited generalization capabilities, limited control over convergence/stability, and limited capabilities to deal with the uncertainty. Furthermore, the learning-based model does not take into account the big data characteristics, and the performance evaluation criterion is based only on price/load data, which is not large. With the consideration of big data characteristics, the forecasting accuracy needs to be further improved.

The remaining sections of this paper are organized as follows. The survey of the proposed load forecasting framework is described in sectionII. Feature selection and feature extraction methodologies are described in section III and section IV, respectively. In section V, the Enhanced Convolutional Neural Network (ECNN) classifier is demonstrated. Section VI shows the experimental results for verifying our proposed framework. The paper is concluded in section VII finally.

II. SYSTEM FRAMEWORK

Inspired from [5], the Fig. 1 shows the framework of the proposed system that is based on three modules, namely feature selection, extraction, and classification. The first part of Fig. 1 corresponds to the feature selection which starts with the standardization of the raw data. Standardization is very crucial because it later affects the overall performance of the classifier. After applying min-max standardization, data is fed into the feature selector, which is based on Random Forest (RF) and Recursive Feature Elimination (RFE) algorithms. Feature selector decides whether a feature needs to be reserved or removed before fed into feature extractor. A feature is kept only in the feature selector index if selected from both RF and RFE algorithms. To remove redundant features, the t-Stochastic Neighbourhood Embedding (t-SNE) algorithm is applied in the second stage. Finally, extracted features are fed into the CNN classifier for building the forecast model. Since CNN performance is controlled by many hyperparameters, we use the grid search algorithm (GSA) to assign optimal values to the parameters for better efficiency. The following three sections describe the details of these modules.

III. FEATURE SELECTION

We propose a combined method based on two algorithms to control the feature selection process. In this way, more accurate features are selected to improve the forecasting mechanism. First of all, RF is applied, which is an ensemble learning technique and has a higher computational capability. As the name suggests, it consists of RF with hundreds of decision trees trained with the bagging method. RF grows on bootstrap data sets to divide the data into feature bagging and out of bag (OOB) data to best separate the samples. The OOB data is used to calculate feature importance in the data set. RF guarantees that all trees are decorrelated and, therefore, reduce variance and overfitting problems of the decision tree method. During the training process, each feature impact on Gini impurity is calculated. A feature has more importance if it decreases the Gini impurity. The final significance of the variable is determined with high cardinality. Fig. 2 shows that combined importance scores add up to 100%, and clearly, 10 out of 15 features are the most prominent features contributing (>0.80) to the creation of the model.

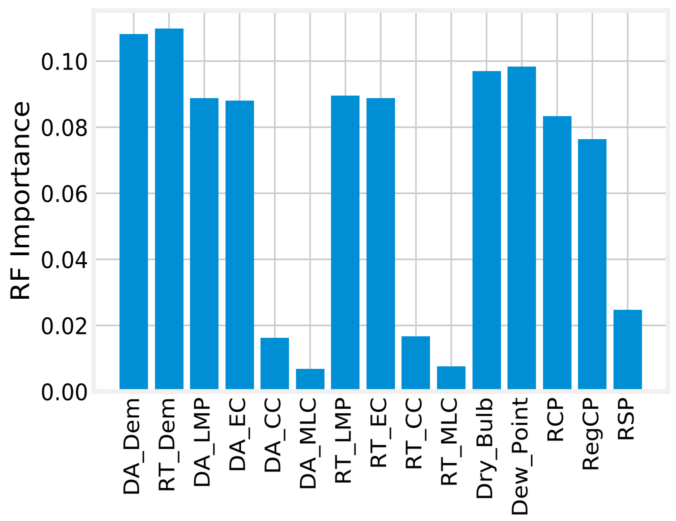

Fig. 2: **RF grades of each feature (DA_CC, DA_MLC, RT_CC, and OTHER have low grades obviously)**

The second method employed for finding an optimal number of features is RFE with Cross-Validation (RFECV). Contrary to the RF method, RFECV recursively eliminates highly correlated in the data set. Highly correlated features give the same results and bring high computational complexity during classification. With the help of the feature selection process, much computational overhead is reduced to train the model. Fig. 3 shows that the RFECV achieves (>0.85) score when six informative features are found. The performance of the curve gradually decreases when non-informative features are added to the model. The shaded area in the curve shows the variability of cross-validation above and below the mean score. Initially, 15 features are fed, and their cumulative score jumps low to high when 6–8 features are found and declined again from the optimal number of features. Both feature selectors work independently and can be deployed distributedly to achieve computation efficiency. To select the best ten features, we introduce a threshold ($T_{RF} \geq 0.07$) for RF. The RFECV provides the list of ten best features. A combination of RF and RFE selects the most important features. There exists a redundancy among ten best-selected features for which they are sent to the t-SNE algorithm for feature extraction.

IV. FEATURE EXTRACTION

Feature extraction is useful to remove redundant features, and a model generalizes better when appropriate features are used during the fitting process. To reduce the redundancy among features, PCA, and classical multidimensional scaling are the most common methods for feature extraction. However, these techniques assume a linear mapping from high to low dimension space [6]. Fig. 4 clearly shows that PCA makes the clusters of non-linear data that are entirely overlapping and results in high dimension mapping. Data in electricity load forecasting needs to be non-linear mapped for appropriate embedding into low dimension.

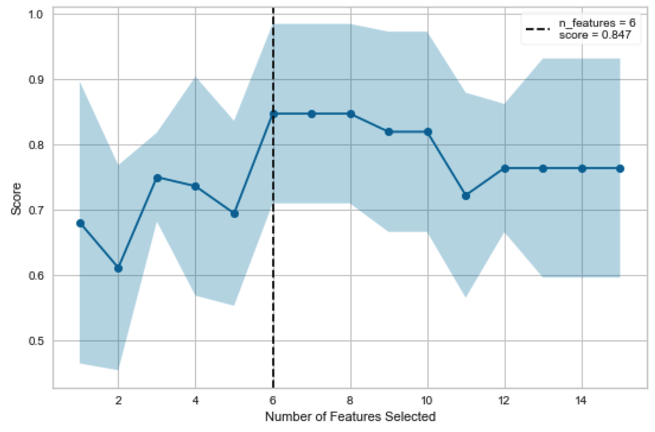

Fig. 3: **Number of optimum features selected by RFE**

To addressed non-linear data mapping issues, Kernel PCA (KPCA) is used, which is an extension of PCA. However, KPCA requires multiple hyperparameters of the kernel functions to be tuned, which increases computation time and hinders the performance. Moreover, KPCA is not as interpretable as PCA because it is not possible to determine how much variance is explained by individual dimensions [7].

To address the above-mentioned issues in PCA and KPCA, we employ t-SNE [8] to perform non-linear mapping and dimension reduction of data altogether. The t- SNE uses "stochastic neighbours," which means not to have a clear border to distinguish how multiple data points are neighbours of the other locations. This is a significant advantage of t-SNE to take both local and global structures into considerations. Finding local and global structure simultaneously create a well-balanced dimensionality reduction map. The aim is to preserve the maximum possible useful high dimensional data points into the low dimension map. Fig. 5 shows how the data points from the different clusters are well separated in the two-dimensional space. The ten best-selected features are used as an input of t-SNE and the output matrix is expressed as,

$$X = (x_1, x_2, x_3, ..., x_N)^T \tag{1}$$

where x_i is the ith feature of electricity load. In the t-SNE algorithm, two essential steps are performed. First, in high dimensional data space, a probability distribution P is constructed. Given a set of N high dimensional objects, a data point x_i would pick x_j as its neighbour if its probability is in proportionate to the probability density of a Gaussian centred on x_i. The conditional probability$(p_{j|i})$ for picking a nearby data point is relatively high, whereas, for faraway data points, it is almost negligible. Mathematical expression for construction P distribution is given by,

$$p_{j|i} = \frac{exp^{-||x_i - x_j^2||}/(2\sigma_i^2)}{\sum_{k \neq i} exp^{-||x_i - x_j^2||}/(2\sigma_i^2)} \tag{2}$$

such that the probability of selecting the pair x_i and x_j is,

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N} \tag{3}$$

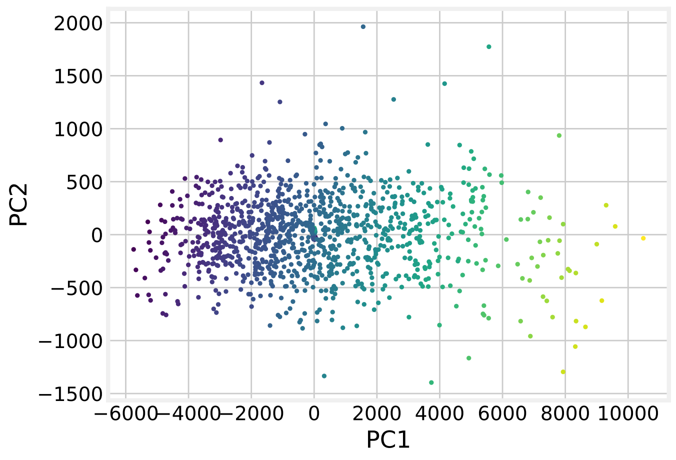

Fig. 4: **Performance of PCA on dimensionality reduction**

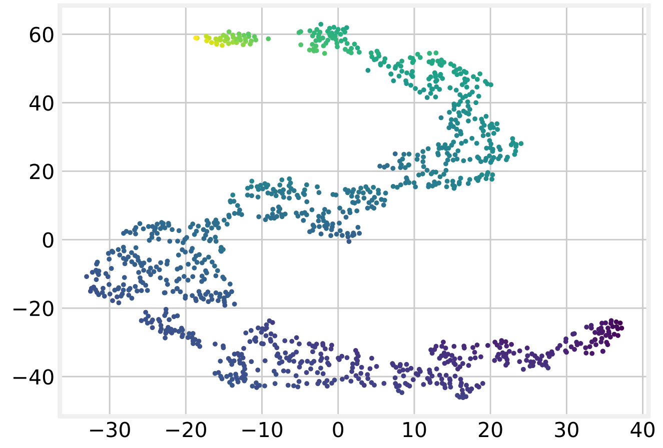

Fig. 5: **Performance of t-SNE on dimensionality reduction**

The probabilities $p_{ij} = 0$ for $i = j$. In Eq. 2, σ represents the bandwidth of the Gaussian kernel to set the perplexity of the conditional distribution. Perplexity indicates how well the bandwidth of local and global aspect is adapted according to the density of data. The perplexity value has a complex effect on prediction and model fitting of a sample. To achieve a target perplexity, the value of bandwidth σ_i is adjusted according to the data density.

For the construct of d-dimensional map $y_i, ..., y_N$ where $y_i \in R^d$, second phase of t-SNE defines probability density distribution,Q, through perfect replication of high dimensional data points (x_i, x_j) into low dimensional data points (y_i, y_j). Mathematically, q_{ij} is defined as following,

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k=l}(1 + ||y_k - y_l||^2)^{-1}} \tag{4}$$

The Student's t-distribution is used to measure the similarities of high dimensional data in q_{ij}. To obtain the y_i, the Kullback Leibler divergence between high and low dimensional space is minimized as follows,

$$KL(P||Q) = \sum_{i \neq j} p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{5}$$

In fact, this result reflects the similarities between the high-dimensional inputs very well. After describing feature selection and feature extraction, we propose the ECNN classifier in the next section to perform the final electricity load forecasting.

V. OPTIMAL CLASSIFICATION

Since CNN is robust and efficient enough in electricity load data, we choose CNN as the classifier. In this section, the classification problem is investigated first. After that, the GSA based CNN is proposed to optimize this problem. The main goal of this work is to minimize the cross-entropy loss function of CNN. However, there is a strong link between the loss function and value of CNN super parameters. It is very challenging to obtain the optimal value of these super parameters to achieve better efficiency and higher accuracy. In this work, we employ GSA to tune these parameters.

In essence, CNNs are a special kind of neural network, which processes data that has grid topology. In this perspective, images are formed because of 2D grids, and time-series data such as electricity load and price data are viewed as a 1D grid. Among multiple layers, at least one layer of CNNs is dedicated to performing convolutions for specific linear operation. The output of the convolution layer for multidimensional input is calculated with the following equation,

$$S = (x * w) \tag{6}$$

where x is the input function, and w denotes the weighting function, also called the $filter$ or $kernel$ of a CNN. The output is in the form of a feature map, denoted by S. The inputs and weights of a CNN are multidimensional arrays. During the course of iterations, random weights are assigned to each input for training purposes. The convolution operation for a two-dimensional input can be expressed as:

$$S(i,j) = (I * K)(i,j) = \sum_l \sum_m I(l,m)K(i+l, j+m) \tag{7}$$

where I and K represent two-dimensional input and kernel, S is the resulting feature map after applying the convolution operation. In reality, there are three phases to complete the operation of the convolutional layer. As a first step, a feature map is obtained after performing a convolution operation. Then, a nonlinear activation function is applied to all the elements of the feature map. The Rectified Linear Activation function (ReLU) is the preferred function to faster the training process.

Finally, to achieve a modified and desired feature map, a pooling function is employed. The purpose of pooling operation is to reduce the dimensionality and amount of parameters, thus making the network less susceptible to small variations in the input. In this work, we use the max-pooling method

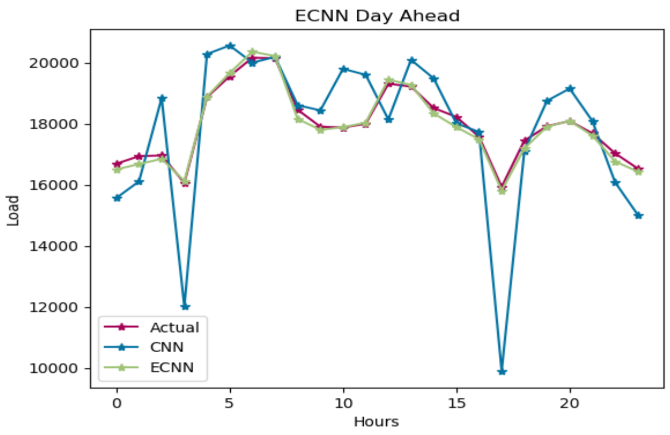

Fig. 6: **Performance on load forecasting using CNN and ECNN**

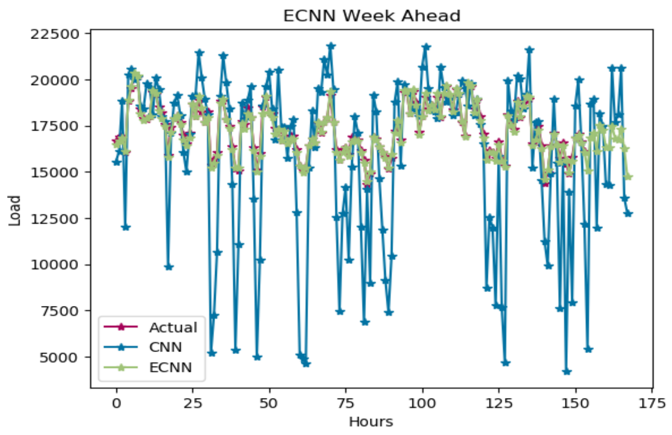

Fig. 7: **Performance on load forecasting using CNN and ECNN**

to avoid overfitting and computational complexity. In max pooling, the operation chooses the maximum value within a matrix and discards the lower value to provide an abstracted form of representation.

As stated, the designed framework can be formed with one or more convolutional layers. In the end, the produced outputs of the convolutional layer(s) are sent to one or more fully connected layers to extract the features. In principle, fully connected layers are the same as hidden layers in a traditional multi-layer perceptron neural network. The output of fully connected layers in the form of a flatten matrix is given to the output layer for classification. The function of the output layer is similar to the output layer in a standard ANN. The final convolution involves backpropagation for the learning process to weigh the end product accurately.

A. *The Grid Search Algorithm*

In the proposed framework, we employ GSA to choose optimal values for the dropout rate (0.2–0.8), learning rate (0.2–0.8), epochs, and the number of neurons in the standard CNN. The main reason for choosing these parameters is that little variations in the values can affect the performance of CNN many folds. Among different optimization techniques, GSA is seen as one of the fundamental tools to find the best combinations of parameters as a search problem. GSA tries all candidate solutions on a grid and chooses the best one in terms of the fitness function. It is a simple and straight forward method to reduce the computational overhead. The optimization problem in GSA is defined as,

$$\begin{aligned} &max\ F(\theta_1, \theta_2, ..., \theta_n) \\ &\qquad\qquad s.t\ \theta_{min,i} \le \theta_i \le \theta_{max,i}, (i = 1, 2, ..., n) \end{aligned} \tag{8}$$

where $F(*)$i denotes fitness function and θ_i is the i-th decision variable. There are two main steps in a standard grid search method, namely, grid creation and grid validation. First, a set of grids parameters are generated as the candidate solutions in the form of dictionary. These candidates solutions contain an equal interval $[d_i = \frac{\theta_{max,i} - \theta_{min,i}}{m_i}]$ for the decision variable i, where m_i represents sum total of all candidates. Similarly, the j-th candidate solution for variable i, $\theta_{i,j}$, is expressed as follows,

$$\theta_{i,j} = \begin{cases} \theta_{min,i} & (j = 1) \\ \theta_{min,i} + m_i d_i & (j = 2, 3, ..., m_i) \end{cases} \tag{9}$$

As a second step, all candidate solutions are tried on the created grids to find the optimal solution $\theta_1^*, \theta_2^*, ..., \theta_n^*$. It reaches the best set of parameters from a set of values. In this work, the fitness function is designed as follows,

$$F = \frac{1}{3}\ ac(Tr) + \frac{1}{3}\ ac(Vs) + \frac{1}{3}\frac{1}{|ac(Tr) - ac(Vs)|} \tag{10}$$

where $ac(Tr)$ and $ac(Vs)$ represent the average prediction accuracy for the training and validation datasets of the ECNN model, respectively. According to Eq. 10, an optimal value of hyperparameters needs to guarantee accurate prediction and avoid the overfitting problem during learning. In the proposed framework, the GSA searches the optimal values of the defined hyper parameters in an array.

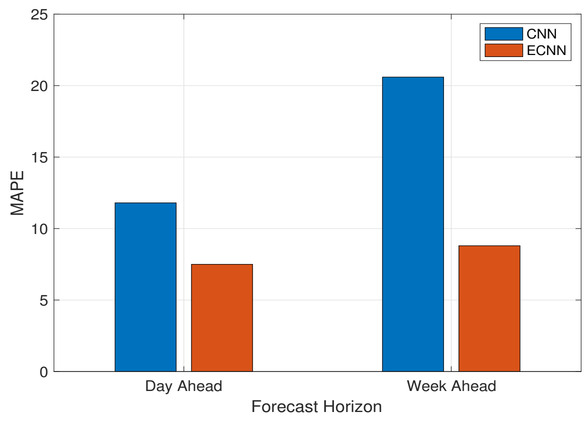

Fig. 8: **MAPE comparison between CNN and ECNN**

VI. SIMULATION SETUP AND RESULTS

In this section, we evaluate the performance of the proposed framework. The python simulator is developed according to the system framework devised in section II. For this framework, input data contains energy generation data and hourly electricity load data of the ISO New England Control Area (ISO NE-CA) from 2010 to 2015 [9]. This record consists of over 50000 real-world electricity price records. The simulation results are organized as follows:

*1) **Performance of Hybrid Feature Selection**:* Important features in ISO NE-CA are roughly selected from hourly electricity load data from 1-1-2015 to 31-12-2017. During the feature selection process, every feature sequence takes the form as a vector. The feature value in different timestamps is represented as components of this sequence. Since our goal is to predict the electricity load, which is named "System load" in the data and those features that have little effect on the load are removed. First of all, RF is applied to calculate the feature importance, as shown in Fig. 3. The optimum number of features graded by RFE method is shown in Fig. 4, which indicates that 6–8 most important features achieve above 84% score. We drop five features with obvious low grade, i.e., features DA_CC, RT_MLC, RT_CC, DA_MLC, and RSP. It is pertinent to mention here that with the increase in the threshold value, more features are dropped, resulting in the increase of training speed and the decrease of accuracy.

*2) **The t-SNE Performance Comparision with PCA**:* In order to eliminate the redundant information within the features, two principal components PC1 and PC2 are extracted with t-SNE and PCA. PCA is a linear algorithm, and it does not interpret the complex polynomial relationship between features, while t-SNE captures the exact relationship between data points. PCA performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximised. As shown in Fig. 4, PCA concentrates on placing dissimilar data points far apart in a lower dimension representation with higher ranges. The t-SNE extracts most of the principal components, as shown in Fig. 5 within a low range. Thus, we select the t-SNE to guarantee the accuracy of forecasting. The data points of t-SNE distribute along coordinate axes, i.e., extract the principal components that are more representative than the PCA.

*3) **ECCN Performance Comparision with Standard CNN**:* We compare the performance of ECNN with benchmark CNN classifier to forecast day-ahead electricity load. To comprehensively understand the characteristic of the proposed method, we calculate the MAPE as a performance indicator. This is expressed as the following,

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} |\frac{(y_i - \hat{y}_i)^2|}{y_i} \times 100\% \qquad (11)$$

In Eq.11, y_i and $\hat{y}_i$ are the actual and forecasting values, respectively. As shown in Figs. 6 and 7, the ECNN is demonstrated as an improved model both for the day-ahead and week-ahead load forecasting strategies. Fig. 8 clearly shows that the MAPE values of CNN are much higher for both day-ahead and week-ahead forecasting as compared to the ECNN values in the same scenarios. The GSA helps optimise the super parameters of CNN jointly; therefore, ECNN performs better in terms of the accuracy of electricity load forecasting than the CNN.

VII. CONCLUSION

In this work, we investigated the short term electricity load forecasting problem, while considering feature engineering and classifier parameters adjustment. We proposed a two-stage electricity load forecasting framework, which is based on feature processing and enhanced CNN classifiers to solve forecasting accuracy problems. Specifically, to select the critical features, a new combined two-stage model is employed to process the n-dimensional time sequence data as input. Additionally, to enhance CNN classifier efficiency in terms of accuracy and speed, we apply t-SNE for feature extraction with less redundancy. Moreover, the GSA automatically and efficiently obtains the appropriate super parameters for ECNN to boost classification performance. The numerical results confirm that the proposed framework shows better results in terms of accuracy when compared to the standard CNN. Furthermore, the work suggests GSA offers a flexible and powerful tool for certain types of optimzation problem. In a different context, for example, GSA has a strong potential to be used for research into robot control systems for nuclear decommissioning and mobile robot path planning, which the present authors are also investigating.

Acknowledgements The authors acknowledge funding support from COMSATS University Islamabad and Lancaster University UK to support this project. The work was in part supported by the UK EPSRC grant EP/R02572X/1.

REFERENCES

[1] Gao, Wei, et al. "Different states of multi-block based forecast engine for price and load prediction." International Journal of Electrical Power & Energy Systems 104 (2019): 423-435.
[2] Mohammadi, Mohsen, et al. "Small-scale building load forecast based on hybrid forecast engine. " Neural Processing Letters 48.1 (2018): 329-351.
[3] AK Srivastava, Ajay Shekhar Pandey, and Devender Singh. "Short term load forecasting methods: A review " .In: Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES), International Conference on. IEEE. 2016, pp. 130138.
[4] Medha Joshi and Rajiv Singh. "Short-term load forecasting approaches: A review" .In: International Journal of Recent Engineering Research and Development (IJRERD) 01.3 (2017), pp. 917.
[5] Wang, Kun, et al. "Robust big data analytics for electricity price forecasting in the smart grid." IEEE Transactions on Big Data 5.1 (2017): 34-45.
[6] Wang, Yifei, Xiandong Ma, and Malcolm J. Joyce. "Reducing sensor complexity for monitoring wind turbine performance using principal component analysis. " Renewable energy 97 (2016): 444-456.
[7] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.Nov (2008): 2579-2605.
[8] Amarasinghe, Kasun, Daniel L. Marino, and Milos Manic. "Deep neural networks for energy load forecasting." 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). IEEE, 2017.
[9] ISO New England Energy Offer Data [Online]. Available: www.iso-ne.com/isoexpress/web/reports/pricing/dayahead- energy-data, 2018.