Khan, I.U., Javaid, N., Taylor, C.J., Gamage, K. A.A. and Ma, X. (2020) Big Data Analytics Based Short Term Load Forecasting Model for Residential Buildings in Smart Grids. In: IEEE International Conference on Computer Communications (IEEE INFOCOM - 2020), Virtual Conference, 6-9 July 2020, ISBN 9781728186955 (doi:10.1109/INFOCOMWKSHPS50562.2020.9163031)

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

http://eprints.gla.ac.uk/220696/

Deposited on 16 July 2020

Enlighten – Research publications by members of the University of Glasgow
http://eprints.gla.ac.uk

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 Gamage[3] and Dr. 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 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 to reduce feature redundancy and hence computational overhead for the next two modules. Second, dimensionality reduction is realized with the help of a t-stochastic neighborhood embedding algorithm for best extraction. Finally, the electricity load is forecasted with the help of a deep neural network. To improve the learning trend and computational efficiency, we employed a grid search algorithm for tuning the critical parameters of DNNs. Simulation results confirm that the proposed model has better results when compared to the benchmark schemes.**

***Index Terms*—Big data, Smart grid, Load forecasting, Classification, Feature selection**

I. Introduction

A. Background

Electricity is an expensive commodity and its consumption must be synchronized with generation to avoid wastage. Today's technologies do not allow to store or queue extra energy in an economical manner. Also, due to limited transmission capacity of existing power network, it cannot be transported to other regions and hence makes electricity characteristics really local and time varying in multiple aspects among different regions [1]– [4]. 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 applications of conventional forecasting methods such as Autoregressive Integrated Moving Average (ARIMA) and Linear Regression have larger errors, hence advanced load forecasting models need to be proposed. The under/overestimation of power generation/consumption can pose severe challenges to the power system network. In other words, by accurate forecasting and reducing the mean absolute error (MAPE) only by 1%, is so impressive and meaningful to have the impact of 3–5%. The overall impact of this decrease can reduce the generation cost of about 0.1% to 0.3% [5]. For this reason, various Artificial Intelligence (AI) and Machine Learning (ML) based forecasting models are proposed recently 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. The primary purpose of price/load prediction in the smart grid is to minimize power peak demand and balance supply-demand gap [7]. A precise forecasting method not only reduces the demand-supply gap but also helps to develop a stable and efficient power management system. 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 [8]. An accurate and stable STLF brings an unprecedented level of flexibility for its management and create a win-win situation both for its generation and consumption side stakeholders. On one end, it helps the utility to address uncertain power generation challenges specifically when penetration of RES is increasing. Besides, it brings higher reliability and aims to achieve available energy sources economically and rationally in an effective manner. As for as customers and end-users are concerned, they are eager to know electricity prices, which is mainly based on generation patterns and power demand for a specific period. Customers have a predefined power price threshold, and based on forecasting results; they can decide to control power demand for a specific time to get financial benefits in terms of energy cost savings. Due to this reason, energy suppliers, as well as consumers, require electricity price/load classification. 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, and hence the amount of data available for forecasting is considerably large and hence difficult to handle, especially for STLF [9]– [10].

B. *Related Work*

The need for accurate STLF strategies can be traced back in the 1960s, and perhaps one of the first comprehensive studies on STLF was conducted by Heinman et al. in 1966 [11]. The authors used regression analysis to investigate the relationship between temperature and energy consumption during summer. Since then, many other approaches and methods are proposed for STLF with variations in the degree of success. The STLF methods are broadly classified into two categories; classical statistical methods and artificial intelligence (AI) methods. The statistical methods determine the mathematical relationship between the exogenous factors (independent) and the load (dependent). Many statistical methods are discussed in the literature, such as multiple linear regression, time series analysis, adaptive filtering, and exponential smoothing [12]. The regression method comprises to assume a linear/non-linear relationship between dependent and independent variables (price, weekdays, weather variables, etc.).

The statistical methods identify the load pattern, and, based on the obtained pattern, the time series analysis approaches are utilized to provide the future value of the measurements. Regression analysis is then applied to determine the coefficients of the independent variables in the assumed model. For instance, Amral et al. [13] used a multiple linear regression model to forecast the electric load up to 24 hours ahead for the Sulewesi Island Indonesia, by selecting the current and previous hourly values of the temperature as independent variables. Time series models, on the other hand, achieve accurate prediction by performing correlation analysis of past observation. Some of the most widely used time series models are the auto-regressive integrated moving average (ARIMA). These models have shown good performance measures based on the box and Jenkins methodology. Ozozen et al. [14]. proposed an ARIMA based Algorithm to capture the linear component of the load time series. However, the existence of various outliers, computational burden and building a model with raw data tend to make the forecasting accuracy unstable [15].

Since the early 1990s, the AIs techniques have been widely explored methods for prediction. One of the popular AIs methods is neural networks (NNs). In an artificial neural network (ANN), the prediction is based on assuming a non-linear relationship between historical data and external variables. 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 [16]. In 1995, turkey et al. [17] proposed an innovative AI technique they called Support Vector Machine (SVM) and Support Vector Regressor (SVR) to address the shortcoming of NNs. These methods employ empirical risk minimization (ERM) principle to improve the training process and find global optimal solutions in the search space. However, these methods are computational very expensive and hence make the algorithm difficult to converge. Also, these methods are not suitable for large data sets and performs under when training class values are overlapping.

For short-term load forecasting 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 [18]. Similarly, ANN models have limited generalization capabilities, limited control over convergence/stability, and limited capabilities to deal with the uncertainty [19]. 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 can further be improved [20].

C. *Key contributions*

In this work, we examine the load forecasting issue for a smart building. 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 [24]. Our objective is to predict the electricity load of a smart building accurately, considering the numerous factors (big data) form the smart grid. To overcome challenging accuracy objectives, we propose a convolution neural network (CNN) [25] reinforced framework that forecasts the electricity load accurately. CNN uses multiple convolution layers to transform input into the output. Three types of layers, namely convolution, pooling, and fully connected layers, are used to build a CNN architecture. Linear and non-linear operations are performed to transform input tensor into output tensor in each layer of CNN. Although CNN is a promising approach, the following challenges are necessary to address for accurate electricity load forecasting.

- *High computationalcomplexity*: CNN's like any neural network model is computationally expensive. It is because multi-layered architecture, along with multiple layer abstraction, are involved in learning complex relationships and patterns between inputs and outputs in deep learning. Hence, redundant and irrelevant features can affect the training process of CNN because of great computational complexity and can drop the forecasting accuracy.
- *Hard to tune parameters*: Numerous super parameters are used in the training process of CNNs such as learning rate, weight decay, batch-size, activation function, etc. The values of these super parameters greatly affect the performance of CNN in the forecasting process.

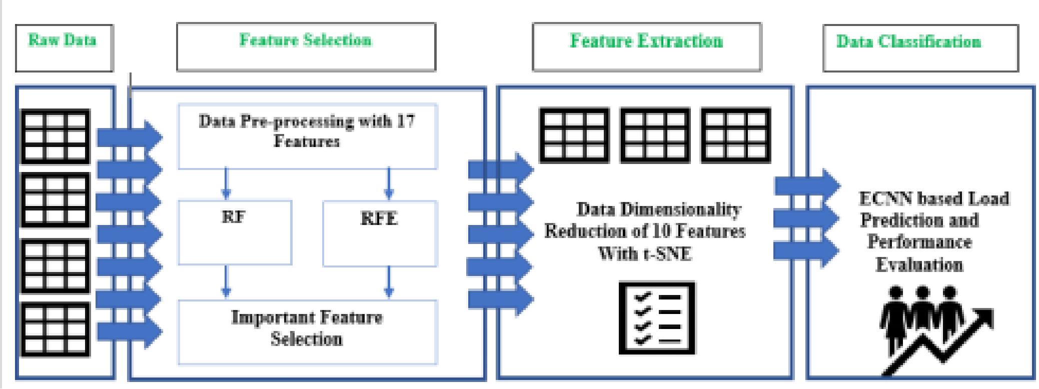
Raw Data
Feature Selection
Feature Extraction
Data Classification
Data Pre-processing with 17 Features
RF
RFE
Important Feature Selection
Data Dimensionality Reduction of 10 Features With t-SNE
ECNN based Load Prediction and Performance Evaluation

Fig. 1: Proposed System Model.

To achieve higher accuracy and better efficiency, cross-validation needs to be performed to get the optimal value of the super parameters. Optimal hyperparameter values differ depending on the application, and the common method used to adjust these parameters are manual search, random search, and Bayesian optimization. However, these methods are computationally expensive and may cause the algorithm unable to converge.

To address the challenges, as mentioned above and motivated from [26], we propose a parallelized framework for electricity load forecasting. As shown in Fig. 1, the three components of the parallelized framework are hybrid feature selection, feature extraction, and regression (HFER). Hybrid feature selection is based on two paralell and completely independent analysis; the feature extraction process is achieved with a t-distributed stochastic neighboring embedding (t-SNE) algorithm and grid search base CNN's regression. First, feature engineering is performed with important feature selection and dimensionality reduction of electricity load data. We employ two features selector algorithms to achieve an appropriate set of features reflecting larger impacts on the electricity load profile. Instead of using Principal Component Analysis (PCA) and kernel (PCA), we choose t-SNE for dimensionality reduction of high dimensional nonlinear data. PCA cannot capture complex polynomial relationship between features and thus not suitable for nonlinear data [27]. Similarly, KPCA is a probabilistic approach with higher variance and eventually perform uncertainly to handle larger nonlinear data. The t-SNE is very suitable for high dimensional data because it captures local and global structures, as a result of which divergence between data points is minimized. To achieve higher accuracy and computational efficiency, our main contributions are listed below:

1) To achieve higher accuracy, an integrated framework based on three modules is proposed. Due to cascaded effect, smart grid big data is efficiently handled and analysed.
2) To achieve this, we first combine Random Forest (RF) and Recursive Feature Elimination (RFE) methods to calculate feature importances independently to perform feature selection. Among the selected features, we employ t- SNE algorithm to achieve lower dimensionality and redundancy of the data. Redundancy analysis is quite a new study in electricity load forecasting. We also employ a grid search algorithm (GSA) to tune the hyper parameters of CNN. The ECNN has higher accuracy and computational efficiency when compared to the recent techniques in proposed area.
3) For performance evaluation, extensive simulations on real world data traces of grid load have been considered. The numerical results show that the proposed model shows better performance statistics than benchmark approach.

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

II. System Framework

In Fig 1, the framework of the proposed system is shown that is based on three modules namely feature selection, extraction, and classification.

A. Design Goals

The main goal of the proposed framework is to forecast electricity load efficiently and correctly. For this purpose, raw data is first processed to identify important features and then perform classification. The following metrics are important to measure the performance of proposed framework.

- *Accuracy of classification*: The core goal of the proposed framework is to achieve maximum accuracy.

- *Dimensional reduction rate*: Proposed feature engineering method increases classification performance to acheive better reults.
- *Time-efficiency*: The proposed framework must work fast when using in electricity load forecasting.

B. Framework Overview

The primary objective of electricity load forecasting is to achieve maximum accuracy. However, various factors influence the training process and hence make the forecasting process difficult. For this purpose, we develop an effective feature selection method, t-SNE based feature extraction method, and GSA-CNN based classification. The first part of Fig. 1 corresponds to the feature selection part, which starts with the standardization of the raw data. Standardization is very crucial because it later affects the overall performance of the classifier. Secondly, data is fed into the feature selector, which is based on RF and 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 it is selected both from RF and RFE algorithms. To remove redundant features, the t-SNE algorithm is applied in the third stage. Finally, data is used to be fed into CNN regressor for building the forecast model. Since CNN performance is controlled by many hyperparameters, we use GSA to assign optimal values to the parameters for better efficiency. A list of abbreviations used in this work are introduced in Table I. We assume a matrix of electcricity load data as follows,

$$\text{X=} \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & 22 & \dots & x_{2n} \\ \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \tag{1}$$

where, rows and columns represent the time stamps and the feature index of the data, respectively. Hence, the predicted component of i hour load of data's jth component can be represented as x_{ij}. The matrix can be also formulated as,

$$\text{X=} \begin{bmatrix} \overrightarrow{t_1} \\ \overrightarrow{t_2} \\ . \\ . \\ . \\ \overrightarrow{t_m} \end{bmatrix} \tag{2}$$

where,

$$\overrightarrow{t_k} = [a_{k1}, a_{k2}, ...a_{kn}]\, k \in [1, m]. \tag{3}$$

The following three sections describe the details of these modules.

III. Feature Selection

This section describes the details of feature selection methodology to identify the most relevant features. Instead of relying on a single algorithm, we propose a combined methodology based on two algorithms to control the feature selection process. In this way, more accurate features are selected to improve the forecasting mechanism. The RF and RFE algorithms independently give feature importance, and their combination selects an important set of features. Both features selection steps are important and give a good predictive performance. 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 de correlated and, therefore, reduces 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 importance of the variable is determined with high cardinality. Fig. 2 shows that combine importance scores add up to 100%, and clearly, 10 out of 15 features are the most prominent features contributing (>0.80) for creation of the model. 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 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 6 informative features are found. The performance of the curve gradually decreases when non-informative features are added into 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 decline 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. Combination of RF and RFE select most important features that are more accurate. There exists a redundancy problem among ten best-selected features for which they are sent to the t-SNE algorithm for feature extraction.

IV. Feature Extraction

This section describes the feature extraction methodology for the proposed framework. Feature extraction is useful to remove redundant features, and models generalize better when appropriate features are used during the fitting process. To reduce the redundancy among features, PCA [1], and classical

TABLE I: List of Abbreviations

Abbreviation	Description
Hour	The index of hours
DA_Demand	The day-ahead demand, which consists of fixed demand bids, price-sensitive demand bids, decrement bids, and increment offers.
Demand	The Non-PTF Demand for ISO-NE CA and the load zones as measured by metering.
DA_CC	One day ahead locational marginal price.
RT_ LMP	The dynamic locational marginal price.
RT_ EC	The energy part of the dynamic load.
RT _CC	The congestion part of the dynamic load.
RT_MLC	The marginal loss part of the dynamic load.
Dry Bulb	The dry bulb temperature.
Dew Point	The dew point temperature.
SYS Load	The actual system load measured by metering.
COAL	The energy supplied by coal.
GAS	The electricity supplied by natural gas.
HYDRO	The electricity supplied by pondage, pumped Storage, reservoir, and run of River.
DA_EC	The energy part of the day ahead load.
DA_MLC	The marginal loss part of the day ahead load.

multidimensional scaling [28] are the most common methods for feature extraction. However, these techniques assume a linear mapping from high to low dimension space [11]. Fig. 4 clearly shows that PCA makes the clusters of nonlinear data that are entirely overlapping and results in high dimension mapping. However, electricity load forecasting data needs to be embedded into dimensional embedding with proper nonlinear mapping.

To addressed nonlinear 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 increase computation time and hinder 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.

To address above mentioned issues in PCA and KPCA, we employ t-SNE to perform nonlinear mapping and dimension reduction of data altogether. The t- SNE uses "stochastic neighbours" that means not to have clear border to distinguish how multiple data points are neighbours of the other points. This is a major advantage of t-SNE to consider both local and global structure into considerations. Considering local and global structure simultaneously create a well-balanced dimensionality reduction map. The aim is to preserve 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. The output matrix can be represented as,

$$X = (x_1, x_2, x_3, ..., x_N)^T \tag{4}$$

where x_i is the ith variable of electricity load.

In t-SNE algorithm, two important 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 Guassian centered on x_i. The conditional probability($p_{j|i}$) for picking a nearby data point is relatively high, whereas for far away data points it is almost infinitesimal. Mathematical expression for construction P distribution is given by [[10]].

$$p_{j|i} = \frac{exp^{-||x_i - x_j^2||}/(2\sigma_i^2)}{\sum_{k \neq i} exp^{-||x_i - x_j^2||}/(2\sigma_i^2)} \tag{5}$$

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

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}. \tag{6}$$

The probabilities $p_{ij} = 0$ for $i = j$. In Eq. 5, σ represents the bandwidth of Gaussian kernel to set the perplexity of 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

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{7}$$

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:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{8}$$

In fact, this result reflects the similarities between the high-dimensional inputs very well.

After describing feature selection and feature extraction sections, we propose the ECNN classifier in the next section to perform the final electricity load forecasting.

V. Optimal Classification

After the two-stage feature selection and extraction, unimportant and redundant features have been dropped. This section describes our proposed approach that accomplishes the final electricity price forecasting via the processed data. 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 loss function and value of CNN super parameters. It is very challenging to get 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

$$S = (x * w) \tag{9}$$

where x and w denote the input and weighting function of a CNN and the output in the form of feature map is 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:

$$S(i,j) = (I*K)(i,j) = \sum_{l}\sum_{m} I(l,m)K(i+l,j+m) \tag{10}$$

where I and K represent two dimensional input and kernel and S is the resulting feature map after the applying the convolution. In real, there are three phases to complete the operation of convolutional layer. As a first step, a feature map is obtained after performing convolution operation. Then, a nonlinear activation function is applied on all the elements of feature map. The rectified linear activation function ?? is usually used in this stage ??. Finally, to achieve modified and desired feature map, a pooling function is employed. The pooling operation makes the representation less susceptible to small variations in the input. Out of various pooling techniques, in the presented work, max pooling method is used. In max pooling, the operation returns the maximum value of a predefined rectangular neighborhood. Other pooling techniques such as average pooling, min pooling and weighted average pooling have been used in literature ??.

As mentioned, the designed network can consist of one or more convolutional layers. Once the convolutional layer(s) produce their outputs, the output is sent to one or more fully connected layers. Fully connected layers can be thought of as hidden layers in a standard neural network. The output layer is placed after the fully connected layers. The output layer performs a similar function to an output layer in a standard ANN. Learning process of the CNN is carried out using back propagation.

A. *Grid Search Algorithm*

Among various optimization algorithms, grid search method can be seen as the most basic and fundamental tool. Generally, the grid search method holds two main unique merits, i.e., simple process and effective function. In particular, the basic idea of grid search method is to simply try all candidates on grids and find the best one as the optimal solution in terms of the highest fitness function. With sufficient enough grids, the grid search method can theoretically reach the optimal solution. Therefore, this paper especially introduces the simple but efficient grid search based optimization algorithm to select the optimal cutoff value in credit risk assessment. A typical optimization problem can be described as follows:

$$\begin{aligned} &max\ F(\theta_1, \theta_2, ..., \theta_n) \\ &\qquad s.t\ \theta_{min,i} \leq \theta_i \leq \theta_{max,i,}(i = 1, 2, ..., n) \end{aligned} \tag{11}$$

where $F(*)$is the fitness function, θ_i represents the i-th decision variable with a minimum $\theta_{min,i}$ and a maximum $\theta_{max,i}$.
Generally, the grid search method contains two main steps: grid creation and grid checking. First, a set of grids are generated as the candidate solutions with an equal interval $[d_i = \frac{\theta_{max,i} - \theta_{min,i}}{m_i}]$ for decision variable i, where m_i is the total number of candidates. Accordingly, the j-th candidate solution for variablei, $\theta_{i,j}$, can be described as follows:

$$\theta_{i,j} = \begin{cases} \theta_{min,i} & (j = 1) \\ \theta_{min,i} + m_i d_i & (j = 2, 3, ..., m_i) \end{cases} \tag{12}$$

Second, the grid search method tries all candidate solutions on grids, and finds the optimal solution $\theta_1^*, \theta_2^*, ..., \theta_n^*$ with the best fitness utility, by enumerating method [?].
In this paper, the fitness function for cutoff selection is designed as follows:

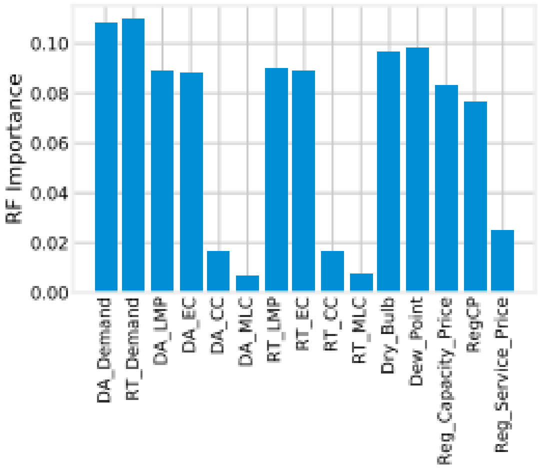

Fig. 2: RF Grades of Each Feature (DA CC, DA MLC, RT CC, and OTHER have low grades obviously).

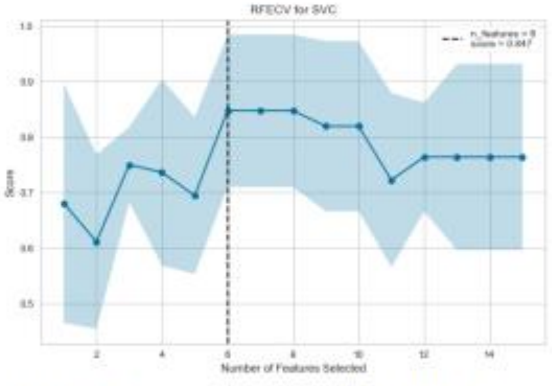

Fig. 3: Number of Optimum Features Selected by RFE.

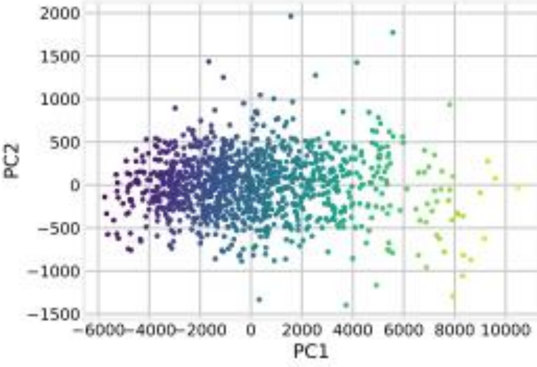

Fig. 4: Performance of PCA on Dimensionality Reduction.

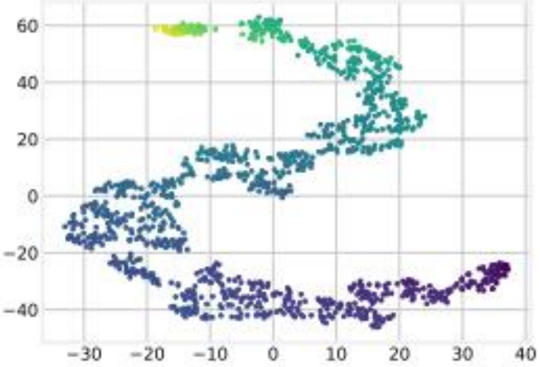

Fig. 5: Performance of t-SNE on Dimensionality Reduction.

$$F = \frac{1}{3}\ accuracy(TR) + \frac{1}{3}\ accuracy(VS) + \frac{1}{3}\frac{1}{|accuracy(TR) - accuracy(VS)|} \quad (13)$$

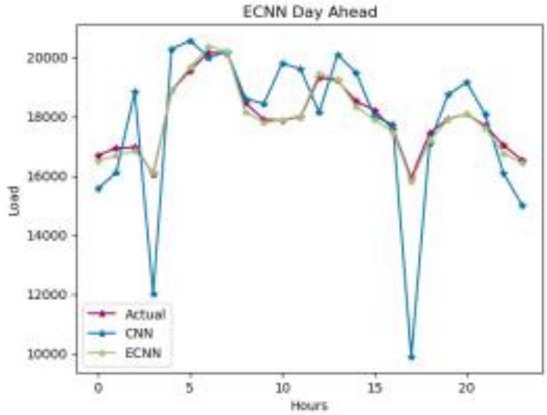

Fig. 6: Performance on Load Forecasting Among CNN and ECNNN.

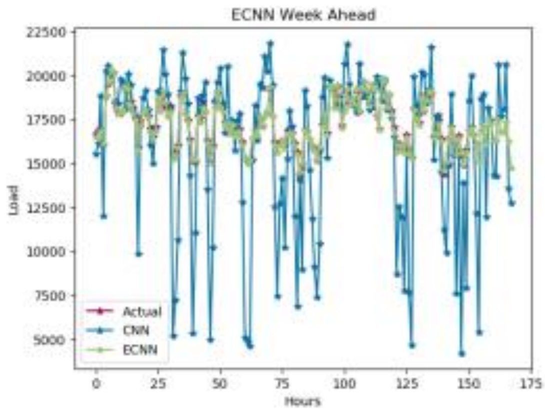

Fig. 7: Performance on Load Forecasting Among CNN and ECNNN.

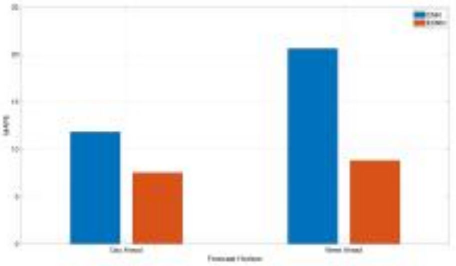

Fig. 8: Comparison on Error Among CNN and ECNN Frameworks.

where $accuracy(TR)$ and $accuracy(VS)$ represents the average prediction accuracy by the ECNN model respectively for the training dataset and validation dataset. According to Eq. (8), an optimal cutoff should not only guarantee accurate prediction results for both training and validation datasets (see the second two parts), but also avoid the overfitting problem in any dataset (see the third part). In this paper, the grid method searches the optimal cutoff value on the range of $[-1, 1]$, i.e., between the lower and upper boundaries of risk scoring, with the searching interval of 0.001.

VI. Simulation Setup

In order to investigate the capability of our proposed framework, we develop a simulator with Python according to the system framework designed in Section 2. During the simulation, the simulator is running on the platform with MAC i7, 16GB RAM, and 128GB hard disk. Hourly electricity price data and energy generation data of the ISO New England Control Area (ISO NE-CA) from 2010 to 2015 are taken as the input for this framework [5], which consists of over 50000 real-world electricity price records. The data includes attributes shown in Table. I. Simulation results are organized as follows: (1) Feature selection performance based on two independent algorithms; (2) The t-SNE performance compared with PCA for redundant features; (3) ECNN performance compared with benchmark CNN algorithm in terms of MAPE error.

A. Simulation Results

1) Performance of Hybrid Feature Selection: HFS is applied to roughly select features form hourly electricity load data during 2010-1-1 to 2015-12-31 in ISO NE-CA. In feature selection, every feature sequence has a form as a vector. The components of this sequence represent the feature values in different timestamps. Since our goal is to predict the electricity load, which is named regulation clearing price (RegCP) in the data, features that have little effect on the load can be removed. First of all, RF is applied to calculate the feature importance as shown in Fig. 3. The optimum number of features grade by RFE method is shown in Fig. 4 which indicates that 6–8 most important features achieve above 84% score. We drop four features with obvious low grade, i.e., features DA_CC, RT_MLC, RT_CC, TDA_MLC and RSP. It is pertinent to mention here that with increased in threshold value, more features are dropped, resulting in the increase of training speed and the decrease of accuracy.

2) The t-SNE Performance Compared with PCA to Reduce Dimension: In order to eliminate the redundant information within the features, t-SNE and PCA are used to extract the principle components. PCA is a linear algorithm, it does not interpret the complex polynomial relationship between features while t-SNE captures 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 maximized. 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 principle components, as shown in Fig. 5 within 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 principle components that are more representative than the PCA.

3) ECCN Performance Metrics Comparision with Benchmark CNN Algorithm: We compare the performance of ECNN with benchmark classifiers CNN to forecast day ahead electricity load. To comprehensively understand the characteristic of the proposed method, we calculate MEAN Absolute Percentage Error (MAPE) as a performance indicator. This is expressed as following,

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{(y_i - \hat{y}_i)^2|}{y_i} \times 100\% \tag{14}$$

In Eq.14, y_i and $\hat{y}_i$ are the actual and forecasting values respectively. The ECNN is shown to be the best model, yielding MAPE values of 8% when compared to the MAPE value of CNN 14%. The results are shown in Fig. 7 and Fig.8, respectively. The accuracy of ECNN achieves higher accuracy as its curve fits well with the real value. The CNN have some outliers due to which it deviates from original value. The GSA optimizes the super parameters of CNN jointly. Therefore, ECNN performs better at the accuracy of electricity load forecasting than CNN.

VII. Conclusion

In this paper, we have investigated the electricity load forecasting problem in smart grid via joint consideration of feature engineering and classifier parameters adjustment. An electricity load forecasting framework which consists of two-stages feature processing and enhanced CNN classifier has been proposed to solve this problem. Specifically, to select those important features, a new combined two stage model is employed to process the n-dimensional time sequence as an input. Additionally, t-SNE is applied to extract new features with less redundancy, which boosts CNN classifier in accuracy and speed. Moreover, the GSA algorithm obtains the appropriate super parameters for ECNN automatically and efficiently. The numerical results have shown that our proposed framework is more accurate than the benchmark CNN.

Acknowledgements The authors acknowledge funding support from COMSATS University Islamabad, Lahore campus and Lancaster University UK to support this project.

References

[1] K. Wang, Y. Wang, X. Hu, Y. Sun, D. J. Deng, A. Vinel, and Y. Zhang, "Wireless big data computing in smart grid," IEEE Wireless Commun., vol. 24, no. 2, pp. 28, Apr. 2017.

[2] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," IEEE Trans. Power Syst., vol. 28, no. 4, pp. 43564364, Nov. 2013.

[3] Robust Big Data Analytics for Electricity Price Forecasting in the Smart Grid.

[4] P. D. Diamantoulakis, V. M. Kapinas, and G. K. Karagiannidis, "Big data analytics for dynamic energy management in smart grids," Big Data Res., vol. 2, no. 3, pp. 94101, 2015.

[5] 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.

[6] Abedinia Oveis, Amjady Nima, Ghadimi Noradin. Solar energy forecasting based on hybrid neural network and improved metaheuristic algorithm. Comput Intell 2017.

[7] Mohammadi Mohsen, et al. Small-scale building load forecast based on hybrid forecast engine. Neural Process Lett 2017:123.

[8] G. Heinemann, D. Nordmian, and E. Plant. "The Relationship between Summer Weather and Summer Loads - A Regression Analysis" . In: IEEE Transactions on Power Apparatus and Systems PAS-85.11 (1966), pp. 11441154. issn: 0018-9510. doi: 10.1109/TPAS.1966. 291535.

[9] 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.

[10] 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.

[11] Hamidi, Omid, et al. "A comparative study of support vector machines and artificial neural networks for predicting precipitation in Iran." Theoretical and applied climatology 119.3-4 (2015): 723-731.

[12] Fard, Abdollah Kavousi, and Mohammad-Reza Akbari-Zadeh. "A hybrid method based on wavelet, ANN and ARIMA model for short-term load forecasting." Journal of Experimental & Theoretical Artificial Intelligence 26.2 (2014): 167-182.

[13] Anamika and N. Kumar, "Market clearing price prediction using ANN in indian electricity markets," in Proc. Int. Conf. Energy Efficient Technol. Sustainability, Apr. 2016, pp. 454458.

[14] Wu, Ying-Chih, Wei-Ping Lee, and Ching-Wei Chien. "Modified the performance of differential evolution algorithm with dual evolution strategy." International conference on machine learning and computing. Vol. 3. 2011.

[15] A. Shiri, M. Afshar, A. Rahimi-Kian, and B. Maham, "Electricity price forecasting using support vector machines by considering oil and natural gas price impacts," in Proc. IEEE Int. Conf. Smart Energy Grid Eng., 2015, pp. 15.

[16] Anamika and N. Kumar, "Market clearing price prediction using ANN in indian electricity markets," in Proc. Int. Conf. Energy Efficient Technol. Sustainability, Apr. 2016, pp. 454458.

[17] Y. Bengio, Y. Lecun, and Y. Lecun, Convolutional Networks for Images, Speech, and Time-Series. 1995.

[18] A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short-Term Load Forecasting.

[19] F. Khan, D. Kari, I. A. Karatepe, and S. S. Kozat, "Universal nonlinear regression on high dimensional data using adaptive hierarchical trees," IEEE Trans. Big Data, vol. 2, no. 2, pp. 175188, Apr.-Jun. 2016.

[20] Y. Bengio, Y. Lecun, and Y. Lecun, Convolutional Networks for Images, Speech, and Time-Series. 1995.

[21] A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short-Term Load Forecasting.

[22] Wu, Ying-Chih, Wei-Ping Lee, and Ching-Wei Chien. "Modified the performance of differential evolution algorithm with dual evolution strategy." International conference on machine learning and computing. Vol. 3. 2011.

[23] A. Shiri, M. Afshar, A. Rahimi-Kian, and B. Maham, "Electricity price forecasting using support vector machines by considering oil and natural gas price impacts," in Proc. IEEE Int. Conf. Smart Energy Grid Eng., 2015, pp. 15.

[24] Anamika and N. Kumar, "Market clearing price prediction using ANN in indian electricity markets," in Proc. Int. Conf. Energy Efficient Technol. Sustainability, Apr. 2016, pp. 454458.

[25] Building Energy Load Forecasting using Deep Neural Networks

[26] Applications of Convolutional Neural Networks

[27] F. Khan, D. Kari, I. A. Karatepe, and S. S. Kozat, "Universal nonlinear regression on high dimensional data using adaptive hierarchical trees," IEEE Trans. Big Data, vol. 2, no. 2, pp. 175188, Apr.-Jun. 2016.

[28] Y. Bengio, Y. Lecun, and Y. Lecun, Convolutional Networks for Images, Speech, and Time-Series. 1995.

[29] A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short-Term Load Forecasting.

[30] Hochreiter Sepp, and Jurgen Schmidhuber. "LSTM can solve hard long time lag problems." In *Advances in neural information processing systems*, (1997) pp. 473-479.