An Evaluation of Deep Neural Network Models on Short-Term Energy Consumption Forecasting for Smart Home Management System

Syed Nazir* | Azlan Abd Aziz | Jakir Hossen, Jaya Ganesan, Nor Azlina Aziz, G. Ramana Murthy

Correspondence
Syed Nazir, Faculty of Engineering and Technology, Multimedia University, Malaysia.
E-mail address: syednazir13@gmail.com

Funding information
Telecom Malaysia (TM) Research & Development.
Grant/Award Number: MMUE/190007.02

Abstract—The smart home Electricity Management System (EMS) enables users to obtain crucial information about the entire household electricity consumption. In this paper, a similar smart home EMS named beLyfe developed by Telecom Malaysia (TM) has been used in which we intend to forecast a day ahead of hourly electricity consumption time-series data. The short-term electricity consumption time-series data forecasting is complex due to the non-linear form of data, recently the Deep Neural Network (DNN) models have gained considerable attention to perform this task. This study aims to determine an effective DNN model for forecasting our (EMS) hourly data by evaluating four popular DNN models including the Convolutional Neural Network (CNN), the Long Short Term Network (LSTM), the hybrid CNN-LSTM, and the Multilayer Perceptron (MLP). The significant hyper-parameters, such as filter size, neuron size, and optimizer algorithm, have been determined through hyper-parameter tuning and performed a comparison experiment between each of these model on four different week sample datasets. The evolutionary experimental results indicate that the 1-D CNN model outperformed the remaining models concerning prediction accuracy and computational training complexity.

Index Terms—1-D Convolutional Neural Network, Electricity Management System (EMS), Time-Series Data

I. INTRODUCTION

Smart grids have recently drawn growing interest due to their stability, durability, sustainability, and performance. A traditional smart grid comprises different components such as smart meters, electricity management systems, energy storage systems, and renewable energy resources [1], [2]. This research focuses on the electricity management system (EMS), in which our aim to forecast a day ahead electricity data in this system. Discovering the daily consumption of energy would allow the user to manage electricity costs [3]. Forecasting electricity time-series data can be classified into three main categories: short-term, medium-term, and long-term [4]. Short-term ranges from hourly, daily, and weekly. For medium-term ranges from weekly to monthly. For long-term ranges from yearly or more. The short and medium terms of electricity forecasting are mainly required for cost management and demand and response management [5], whereas the long-term forecasting is usually done to identify a specific power generation plant's production capacity in the future [6]. The selection of an appropriate forecasting model has a significant influence on forecasting ranges. In long-term forecasting, statistical models such as ARIMA and SARIMA are frequently studied due to predicting data containing trends and seasonal components. These models could have reasonable prediction outcomes where data patterns involve linearity and are less computationally expensive than current machine learning models. In short-term electricity forecasting, data trends involve non-linear abrupt changes that are complex to forecast [7]. The deep neural network models can forecast time-series data, including non-linear patterns, through automatic learning of temporal dependency and mapping from input to output [8].

In recent studies, the deep neural network model, particularly the recurrent neural network RNN, primarily developed to predict sequence data, has received considerable attention to predicting short-term electricity load time-series data [9], [10]. The RNN contains recurrent layers that process inputs sequentially, making it flexible to process all types of sequence data, especially time-series data, through which temporal dependence can be learned. RNN is often suffered due to vanishing and exploding gradient problem; however, this issue has been overcome by the improved version of the RNN model named long short term memory LSTM model. The LSTM model works well in sequence data prediction and could be used for different purposes such as text translation, sentiment analysis, time-series forecasting.

In the last few years, the LSTM model has been frequently discussed in the short term load forecasting STLFF. Kong et al. [11] proposed an LSTM-based forecasting framework to forecast residential loads accurately by considering residential activities’ volatile behavior. Kong et al. [12] proposed a similar LSTM based forecasting framework; in this study, a density-based clustering approach is used to determine the aggregated load and individual load profiles for consistency analysis. Choi et al. [10] proposed a novel framework by combining Residual network ResNet with LSTM model to improve forecasting accuracy by incorporating latent features of input historical load data.
through ResNet. Yan et al. [13] proposed a hybrid framework by incorporating the stationary wavelet transformation SWT with LSTM models to forecasting household 5 min incremental electricity load data. The framework identifies the appropriate SWT level through the Pearson correlation coefficient and assigns each high pass filter data to the separate LSTM model for forecasting. Each forecasting outcome is integrated using Inverse SWT. The hybrid CNN-LSTM has also been widely utilized in several research articles on STLF [7], [14]. The CNN-LSTM model can extract spatial and temporal features from electricity load time-series data for prediction. Kim et al. [15] recently proposed a novel hybrid CNN-LSTM model for forecasting residential house energy consumption. In this study, the significant variable that influences prediction has been utilized. A significant drawback of this work is the discovery of optimal hyper- parameters of the CNN-LSTM model. The 1-D CNN model with some combination of other architecture and algorithm has also been studied in several research articles on electricity forecasting [16]–[19]. Muralitharan et al. [16] used two different algorithms to identify a CNN model’s optimal weight, such as Genetic Algorithm GA and Particle Swarm Optimization PSO, to forecast energy demand. Kim et al. [17] proposed a recurrent inception convolution neural network (RICNN) that coupled RNN and 1-D CNN models to enable STLF for energy management systems (EMS). Lang et al. [18] use the CNN model to predict 36 hours of energy load data. The author scans significant hyper parameters such as dense size, filter size, and kernel size in this study to obtain high accuracy. The researchers also used the MLP model with support classification and reduction algorithms to forecast electricity load data [20], [21]. Protasiewicz et al. [20] performed a simplification and reduction in the structure of a multilayer perceptron (MLP) neural network to increase the MLP performance of prediction using two pruning algorithms such as Optimal Brain Damage (OBD) and Optimal Brain Surgeon (OBS). Wahid et al. [21] performed statistical-based classification using MLP to identify a residential building’s electricity demand.

This paper aims to forecast short-term hourly a day ahead of electricity consumption time-series data for our smart home electricity management system (EMS). Four deep neural network models, such as CNN, LSTM, CNN-LSTM, and MLP, are evaluated to determine which model on our electricity datasets performs reasonably better in predictive accuracy and even computation time. The significant hyper-parameter selection scanning is performed manually since neural network performance depends on data characteristics [18]. We consider the most significant hyper-parameter in this tuning process that directly affects the performance of the DNN models: neuron size, filter size, activation functions, optimization algorithms, and batch size. The numerical input values are scaled to a standard range between 0 and 1; this is the normalization scaling technique that scales each input values independently within the range 0-1 through which the DNNs models can achieve most accuracy [22]. The four different weeks sample datasets have been used in the comparison experiment to confirm each of these DNNs models performance. The experiment results indicate that the 1-D Convolutional Neural Network CNN outperformed the remaining deep neural network models in predictive accuracy, and the training computational time is also consistent in four different weeks sample datasets.

The rest of the paper is organized as follows. Section 2 provides a data description of our smart home electricity management system (EMS). Section 3 discussed the architecture of 1-D Convolutional Neural Network (CNN) used to forecast time-series data. Section 4 presents the experiment setup, and Section 5 discusses the experimental results. Finally, the conclusion is drawn.

II. DATA DESCRIPTION

The dataset contains 23,259 rows and four different columns like input date, sensor id, time, and active power. Our interest columns are time and active power that we will use to forecast the day-ahead prediction. To identify a day ahead of hourly electricity consumption for smart home EMS, we utilized historical data of one week, such as 24x7 = 168 records. These records will be used to train each DNN models on which we will perform a comparison experiment. The input data used in this experiment is normalized and scaled in the range between 0 and 1. The equation (1) shows the scaling process of inputs.

\[
y = \frac{(x - \text{min})}{(\text{max} - \text{min})}
\]

In equation (1), min represents the minimum value, max represents the maximum value of the input data, \(x\) is the input value and \(y\) is the scaled value.

III. CONVOLUTIONAL NEURAL NETWORK

The fundamentals of 1-D CNN is used to forecast and classify fixed-length windows in time series. The CNN model can be applied to 1-D time-series data, in which the same pattern that appears in various regions can be captured. The CNN architecture comprises several layers such as convolutional layers, pooling layers, and fully connected layers [7]. A convolutional layer and a pooling layer are typically the first two layers of a convolutional neural network: both perform smoothing. Since they are part of the same function to forecasts, smoothing parameters are explicitly optimized to perform well in a predictive task by optimizing the neural network loss. Later layers use smoothed raw data to handle the central part of the time series forecasting or classification problem.

![Fig. 1. Convolutional Neural Network CNN (1D) architecture](image)
Figure 1 shows 1-D convolutional operation, in which each convolutional layer consists of several convolution units, a backpropagation algorithm optimizes each convolution unit’s parameters. The CNN differ from MLP by uses the concept of weight sharing. In Figure 1, the inputs are $x_1$ to $x_6$, and $c_1$ to $c_4$ are the feature maps after 1-D convolution [23]. In general, features with a broad dimension are usually obtained from the convolutional layer, which needs to be reduced in dimension. The outputs of neuron clusters are merged in the next layer into a single neuron. The final result is determined using a fully connected layer that integrates all local features into global features. The CNN benefit is that preparation is reasonably simple since weights are smaller than full-connected architecture.

IV. EXPERIMENT

The experiment is conducted to identify a useful DNN model to forecast a day ahead of hourly electricity consumption time-series data for our EMS. The DNN models used in this comparative analysis are CNN, LSTM, CNN-LSTM, and MLP. In this experiment, hyper-parameters tuning was initially performed to identify an efficient combination of hyper-parameters that yields high accuracy in forecasting. In this tuning process, the chosen hyper-parameters for which tuning has been done are the filter size, the neuron size, and the optimizer algorithm. The selected filter sizes for tuning are 16, 32 and 64, the neuron sizes are 50 and 100, and the optimizer algorithms are ADAM, RMSprop and SGD. The batch size and activation function remain the same for each model during tuning. The selection of an effective combination between these hyper-parameters relies on the least prediction error, measured with the root mean square error RMSE, as shown in equation (2).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$  \hspace{1cm} (2)

While evaluating each models performance on different hyper-parameter combination sets, we observed that the filter size of 16 and the neuron size of 50 with root mean square propagation RMSprop optimization algorithm will provide better results such as least prediction errors. For CNN, and CNN-LSTM, using a filter size of 16 with a neuron size of 50 and RMSprop as an optimization algorithm provides the least prediction errors compared to filter sizes 32 and 64. Figure (2) shows the prediction errors of each model on two separate hyper-parameters combinations such as number of neurons and optimization algorithm. Subsequently an analysis of models performance on different hyper parameter combinations, It is observed that the suitable combination between filters, neurons, an optimization algorithm is 16, 50, RMSprop, respectively.

![Fig. 2. The RMSE error score on two different hyper-parameters such as neuron size and optimization algorithm.](image)

Since the chosen hyper-parameters are sufficient for all models on our smart home EMS electricity consumption time series dataset, our next step is to conduct a comparative experiment utilizing the same hyper-parameter configuration between each of these models. Each model has been trained using 1000 epochs. The epochs are essential parameters in training
deep neural network DNN models that defines how many times the training data passed to the neural network to achieve the best fit condition. An early stopping technique is also used in this experiment to avoid overfitting and stop training once the model performance has stopped improving [24]. Each model input is sampled into input and output sub samples to transform time-series data into supervised learning. A recursive forecast technique has been applied in this experiment, in which the models make one-step predictions, and the output is fed as inputs for the subsequent predictions. The code of this procedure is mention below.

\[ n\text{Steps} = 24 \]
\[ n\text{Features} = 1 \]
\[ \text{prediction} \rightarrow \text{list[1]} \]
\[ \text{tempList} \rightarrow \text{list[ndarray = \text{next24hoursData}]} \]

\[ i = 0 \]
\[ \text{while } i < n\text{Steps} : \]
\[ \text{if } \text{tempList.length} > n\text{Steps} : \]
\[ \text{next24hoursData} = \text{np.array(tempList[1:])} \]
\[ \text{next24hoursData} = \text{next24hoursData.reshape((1, nSteps, nFeatures))} \]
\[ \text{yhat} = \text{model.predict(next24hoursData, verbose = 0)} \]
\[ \text{tempList.append(yhat)} \]
\[ \text{tempList = tempList[1 :]} \]
\[ \text{prediction.append(yhat)} \]
\[ \text{else :} \]
\[ \text{next24hoursData} = \text{next24hoursData.reshape((1, nSteps, nFeatures))} \]
\[ \text{yhat} = \text{model.predict(next24hoursData, verbose = 0)} \]
\[ \text{tempList.append(yhat)} \]
\[ \text{prediction.append(yhat)} \]
\[ i = i + 1 \]

The code represents the recursive forecast strategy of 24 hour time intervals where \text{nSteps} represents the range of the next future predicted values, \text{nFeatures} represents the number of feature in the time-series data. In our case, we only have one feature correspond to a time interval which is total active power. The \text{tempList} is the temporary list that contains the recent historical 24 hours observations of electricity consumption time-series data used to predict the future 24 hours data through the trained DNNs models that are trained on one-week historical data such as 24x7=168. A trained model will make a one-step prediction by transforming the shape of the 24 hours input data into a sample of (1, 24, 1) which is defined as (samples, nSteps, nFeatures). The \text{yhat} represents the predicted value which is further appended into both of the \text{tempList} and \text{predictionList}.

V. RESULTS AND DISCUSSION

In this section, we have discussed each model performance in term of prediction accuracy and computational complexity. We have taken four different weeks sample datasets of our EMS electricity data to evaluate each of these models performance, as shown in table (1).

<table>
<thead>
<tr>
<th>Data Samples</th>
<th>RMSE</th>
<th>CNN</th>
<th>LSTM</th>
<th>MLP</th>
<th>CNN-LSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td></td>
<td>94110</td>
<td>104940.8</td>
<td>81797</td>
<td>68851.5</td>
</tr>
<tr>
<td>Week 2</td>
<td></td>
<td>81655</td>
<td>110115.9</td>
<td>105940.5</td>
<td>108864.5</td>
</tr>
<tr>
<td>Week 3</td>
<td></td>
<td>150366</td>
<td>197132.5</td>
<td>142779.8</td>
<td>165924</td>
</tr>
<tr>
<td>Week 4</td>
<td></td>
<td>76313</td>
<td>100266.6</td>
<td>82953.5</td>
<td>59025</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>100611</td>
<td>128114</td>
<td>103367.7</td>
<td>100666.25</td>
</tr>
</tbody>
</table>

The prediction results show that the 1-D CNN model outperformed all the remaining model used in this comparative study. Figure (3) show the comparative results of four different week’s samples.
We also measured the training computational time during the evaluation of each model on four different week samples. A boxplot graph is drawn to identify the spread of each model training time in seconds of four different week samples. This graph reveals that the 1-D CNN model generates predictions comparatively less than the remaining models with the least prediction errors measure through RMSE.

Nevertheless, the study outcomes depend on manually tuning to identify the appropriate hyper-parameter for the neural network sequence prediction models. In deep neural networks, the hyper-parameters setting depends specifically on the input data's characteristics [18]. The forecasting of energy consumption has received much interest from the residential sector to optimizing the cost of electricity through effective approaches [25]. In previous short-term electricity forecasting research, the researcher utilizes meteorological factors, and residential behaviors as input features to obtain high predictive performance [26]. These approaches somehow improve the overall performance, but we could not utilize them in general. In this comparative study, we conclude that the 1-D CNN could be used as a prediction model for our EMS hourly data that contains non-linear abrupt shifts that are quite difficult to forecast accurately by the traditional statistical model.

VI. CONCLUSION

The prediction of short-term electricity consumption time-series data that contains non-linear abrupt changes requires an efficient DNN model to learn crucial features from dataset by mapping input to output. This paper evaluates four deep neural network models such as CNN, LSTM, CNN-LSTM, and MLP to forecast a day ahead of hourly electricity consumption time-series data for our smart home (EMS). The system is used to monitor the current
electricity consumption to manage electricity cost-efficiently. A Hyper-parameter tuning is performed prior to the evaluation experiment to determine appropriate hyper-parameters combination on which each model produce high performance. The experiment results show that the 1-D CNN model provides high accuracy of predictions with minimum training computational time. In the future, we would incorporate more features in our EMS electricity data, such as meteorological factors, to enable multivariate forecasting that could improve the predictive accuracy of the CNN model.

REFERENCES


