Journal of Signal Processing Systems (2023) 95:1341–1352
https://doi.org/10.1007/s11265-022-01785-0

IoT Based Load Forecasting for Reliable Integration of Renewable Energy Sources

Levi Randall[1] · Pulin Agrawal[2] · Ankita Mohapatra[1]

Received: 10 May 2021 / Revised: 5 May 2022 / Accepted: 16 June 2022 / Published online: 6 July 2022

Abstract

Renewable energy resources have gathered substantial interest, and several nations are striving to use them as the dominant power resource. However, the power output from these energy sources is inherently uncertain due to their reliance on natural forces like wind, sunlight, tides, geothermal, etc. An accurate estimation of expected consumer load demand can assist with scheduling and coordination between various generating units, ensuring a consistent supply of power to consumers. Internet of Things (IoT) devices are becoming ubiquitous in all technological domains and making different kinds of data readily available. This data from heterogenous IoT sources can be combined and applied towards rapid, short-term load forecasting. This work proposes a Long Short-Term Memory (LSTM) based load prediction model that combines weather data, historical and current load demand to project the hour-ahead load demand. LSTMs are excellent for picking out patterns in time series data and learning long-term dependencies, allowing them to predict over a prolonged period. Using our LSTM model, we obtained a Mean Absolute Percentage Error (MAPE) of 0.62% on the hour-ahead forecast. We further enhanced this model using Wavelet Transforms (WT-LSTM) and observed an improvement of 16% over LSTM model. Both models performed significantly better than their equivalent Artificial Neural Network (ANN) model counterparts, with LSTM and WT-LSTM outperforming the ANN and WT-ANN by 50%, respectively. Short term load forecasts from models predicting on such streaming data from IoT sensors can be used to do rapid generator balancing, thus making the grid more reactive to changes and capable of providing a reliable power supply.

Keywords Load forecasting · Long Short-Term Memory · Data acquisition · Internet of Things (IoT) · Artificial Neural Networks (ANN) · Wavelet Transforms (WT)

1 Introduction

The past several decades has seen a significant shift from fossil fuels towards renewable energy sources for power generation to reduce greenhouse gas emissions and provide a cleaner, safer environment for all. To this end, several governments are encouraging ambitious policies to incorporate more green energy into the power grid. For example, in 2018 the state of California (USA) mandated that 50 percent of the state's electricity needs must be met by renewable resources by 2025 and 60 percent by 2030. It further suggested a bold goal of 100% zero-carbon powered grid by 2045. Figure 1 shows the current constitution of energy resources in California. Only 31.7% of the total energy demands are being met by renewables, with solar and wind generation as the major resources.

However, one of the major challenges to relying completely on renewable energy is that resource availability is heavily dependent on the forces of nature and cannot be controlled by humans. Often the energy supplies need to be dynamically shifted and re-balanced between the various generating units to meet the requirements of the power grid and maintain grid stability. To ensure a consistent and reliable energy supply to consumers through generator schedule planning between these resources, it is critical to accurately predict the short-term consumer electricity load demand. Electricity load forecasting has been playing a key role in the power industry for several years. The time scale of the

P. Agrawal is an independent collaborator.

✉ Levi Randall

✉ Ankita Mohapatra
amohapatra@fullerton.edu

[1] Computer Engineering Program, California State University Fullerton, Fullerton, CA, USA

[2] California, USA

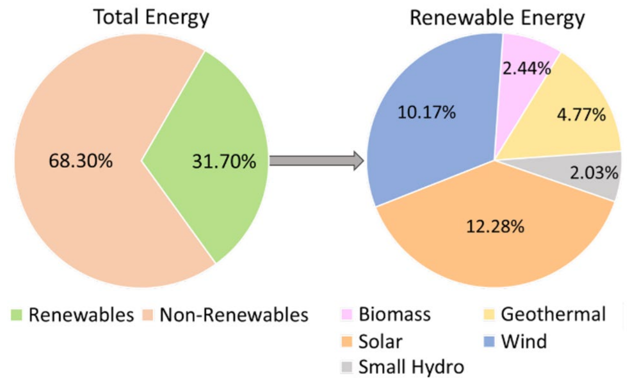

Figure 1 **(left)** California's current energy supply profile **(right)** Breakdown of renewable energy sources.

forecast can vary from a few seconds to several years. They are also vital for several other applications such as infrastructure planning and expansion, price prediction, energy trading, scheduling maintenance etc. [1].

Despite the expected irregularity of consumer usage, the overall electrical load demand is observed to be largely dependent on a combination of several factors such as temperature, humidity, dew point, solar irradiation, collective seasonal user pattern, etc. In Fig. 2, we have shown the load consumption of California for four years (2015–2019).

In Fig. 2, one can easily observe that in the summer months of June to September, the load demands are significantly higher than the rest of the year, which correlates with higher temperatures during these months. Most prediction models rely on real-time measurements of meteorological variables and current and past consumer behavior as inputs to forecast future demand. To this end, Internet-of-Things (IoT) compatible devices integrated with the power grid have been a gamechanger for smarter, faster, and efficient power management [2, 3]. An array of IoT sensors connected to the grid can be used for effectively gathering real-time weather information. On the consumer side, IoT-enabled smart meters can measure the immediate load consumption (Fig. 3). The energy consumption and weather data sets are sent for analysis and computation of load demand estimate for the next hour, shown in Stage 1 in Fig. 3. The forecasted load estimates can then be dispatched to the load scheduling service to coordinate various renewable and non-renewable energy sources, shown in Stage II in Fig. 3. In this paper, we have proposed a hybrid Wavelet Transform LSTM (WT-LSTM) network to implement Stage 1 by combining weather data, current and past load demand to predict next-hour load forecast. We compared the performance of this network with a regular LSTM, a simple Artificial Neural Network (ANN) model, and a Wavelet Transform enhanced ANN (WT-ANN). Similar methods can be adopted to determine the optimal load scheduling in Stage 2, but this paper does not focus on load scheduling. The reader is encouraged to refer to these articles instead [4–8].

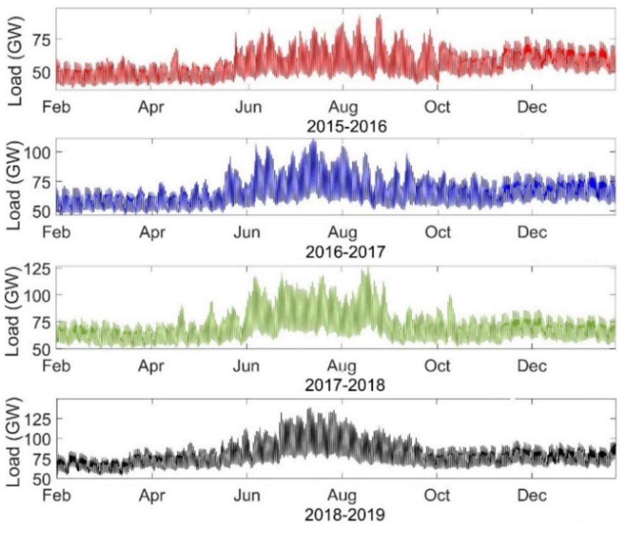

Figure 2 Electrical Load demand of 2015 to 2019 for California.

2 Background

Electrical load forecasting has been explored by many researchers since the 1950s. In the late 1980s, this field of research was again encouraged by the introduction of market deregulation and smart grid, with the current impetus driven by the assimilation of renewable energy resources into the energy sector [9, 10]. Load demand is a complex signal that typically shows highly non-linear and time-variant behavior. It has a strong correlation with a multitude of variables such as weather data, customer behavior, hour of the day, day of the week, the month, load demand in the previous hour, at the same hour on the previous day(s) or week(s) (historical data) [11]. Any prediction model should aggregate all these relevant factors as inputs to generate a reliable forecast. Even fractional percentage improvement in accuracy can translate to huge savings in electricity generation. Several different techniques have been described and implemented for forecasting electricity load, such as Autoregressive Integrated Moving Average (ARIMA), Artificial Neural Networks (ANNs), Fuzzy Logic Systems (FLS), Support Vector Machines (SVM), etc. [10]. Artificial Neural Networks (ANN) are quite possibly the most widely applied technique to load forecasting [9], with one of the earliest papers recording a Mean Absolute Percentage Error (MAPE) of 1.40% and 2.06% for hour-ahead and 24-h ahead prediction on Seattle energy sector [12]. Similar models designed for hour-ahead load forecasting by two separate research groups reported an average forecast MAPE

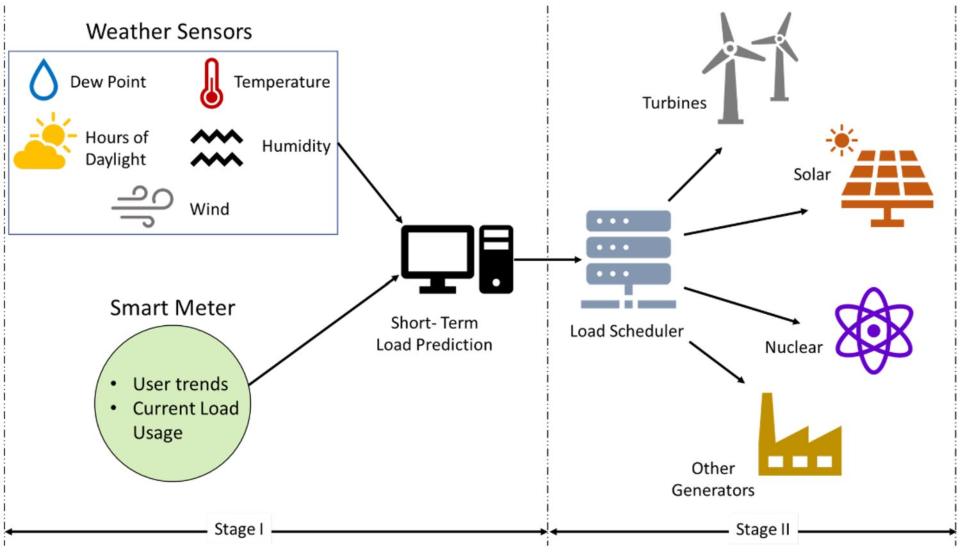

Figure 3 An example of electric load generation management system with IoT enabled devices.

of 1.12% and 1.63% for Saudi Arabia Okinawa load data, respectively [13, 14]. This era of ANN was quickly replaced by a surge of Deep Learning models for load demand prediction, like Deep Neural Networks (DNN), Deep Belief Neural Networks (DBN), Extreme Learning Machine (ELM), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Gated Recurrent Unit (GRU), Long Short-Term Memory Networks (LSTM), etc. [11]. Ryu et al. modeled two DNN with two pre-training methods (Restricted Boltzmann Machine Rectified Linear Unit) for day-ahead forecasts and obtained an MAPE of 2.27% and 2.19%, which were a significant improvement over a shallow neural network (2.98%) and ARIMA (3.29%) based models [15]. Deng et al. proposed a multi-scale CNN with periodic time encoding algorithm for single step point forecast and 48 h forecasting on Ireland's data set and reported MAPEs of 0.98% and 3.74% respectively [16]. A novel Stacked Auto-Encoder ELM with additional time series features such as fluctuation rate, smoothness, sliding trend, etc. was proposed for short term-load forecast by Chen et al., and outperformed other networks [17]. Jung et al. devised a GRU with an attention mechanism to focus on variables showing higher correlation with energy demand and demonstrated an improvement of 10% in performance with this mechanism applied to hourly forecasting [18].

Although ANNs are dependable, simple to implement, and effective at producing very accurate outputs for many practical problems, they are limited by their incapability to hold on to information, making them less effective for forecasting using time series data. RNNs were among the first networks that contained feedback connections to emulate a type of internal memory. Retaining information allows it to predict future values with previous states. Despite the advantage of using prior information, RNN is not a perfect network and suffers from an issue that reduces accuracy because of the Vanishing Gradient problem [19]. A more modern network, the Long Short-Term Memory (LSTM) model was designed to overcome the issues of the RNN network. This LSTM was built upon the framework of the RNN but did not suffer from the Vanishing Gradient problem [20]. LSTM introduces a new kind of cell called a gated cell and uses these cells to store pertinent information for long periods (Fig. 4). This allows the LSTM model to immediately use older previous data to understand progressively higher-order patterns in data.

With the addition of these new cells, this network takes longer to train but is more accurate than RNN or ANN for predicting over longer periods. Therefore, we chose LSTM as the model for load forecasting in this study. The equations modeling an LSTM are described here. The following equations X_t, h_t, and C_t are the input, output, and memory cell state at time t. W_k and b_k represent the weights and biases at layer k. The LSTM computations start with the 'forget gate' layer that allows the model to forget unimportant data [21]:

$$f_t = \sigma\left(W_f * \left[h_{t-1}, X_t\right] + b_f\right) \tag{1}$$

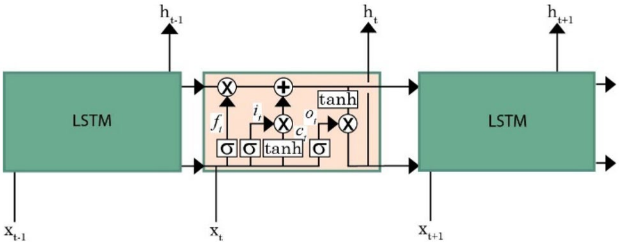

Figure 4 Long Short-Term Memory Cell with Flow of Data Displayed.

After the forget gate, the input layer consists of two computations. Of these two, one is a sigmoid layer (i_t), and the other is a tanh layer as follows:

$$i_t = \sigma\left(W_i * \left[h_{t-1}, X_t\right] + b_i\right) \tag{2}$$

$$\overline{C}_t = tanh\left(W_c * \left[h_{t-1}, X_t\right] + b_c\right) \tag{3}$$

Although the equations are similar to the forgetting gate equations, the purpose of the layers are different. In the tanh layer, the input is transformed into a vector candidate ($\overline{C}_t$) for the current state of the memory cell, while the sigmoid layer controls whether the data from the tanh layer is used in the current state of the memory cell.

$$C_t = f_t * C_{t-1} + i_t * \overline{C}_t \tag{4}$$

The current state of the memory cell is updated based on Eq. (4), which implements forgetting on the previous memory cell state and keeps the portion of the data from the current state. Lastly, the output gate calculates the overall output of the cell. The output gate is modeled below:

$$o_t = \sigma\left(W_o * \left[h_{t-1}, X_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t * tanh\left(C_t\right) \tag{6}$$

LSTM networks are considered superior to classic RNN as they solve RNN's gradient disappearance problem and learn the time dependence of data over a long duration. Muzaffar and Afshari implemented LSTM on MATLAB® to forecast 24 h, 48 h, 7 days, and 30 days ahead [22]. They reported MAPE values of 1.522%, 2.16%, 5.97%, and 9.75% for these forecast horizons. CNN and LSTM were also combined for determining electrical load prediction of Bangladesh power system, which outperformed several other models in 24 h, 48 h, and 7-day ahead forecasts, with the best MAPE of 3.22% for 24 h forecast [23].

This paper discusses the LSTM model for hour-ahead load forecasting of California energy market. The model uses historical load data, current load demand, day/week identifiers, and weather data as inputs. The current load demand and weather parameters can be obtained through IoT-enabled devices (smart meters) and sensors. We also proposed a novel LSTM modified with Wavelet Transforms (WT-LSTM). We compared the performance of both models with Artificial Neural Networks (ANN). Such models are suitable for supporting integration of renewable resources on a smart grid, where instantaneous scheduling decisions are critical.

3 Methodology

3.1 Data Preparation

For every trainable mathematical model, a series of trainable inputs must be supplied to assist the network in learning the output pattern. For the data used in this study, 4 years of hourly data ranging from February 11th, 2015 to 14th August 2019 was used for testing. The data was sourced from the California Independent System Operator (California ISO) that oversees California's power grid. The boxplots in Fig. 5a, b show the statistical distribution of the hourly load separately for each year and season. Figure 5c shows a daily variation of load recorded in a typical day.

We included previous load data as inputs to assist the sequential pattern detector in LSTM in using historical data and the day information for making predictions. Day information like the hour in the day, day of the week, and month of the year are all essential contributors to the amount of energy consumed by the public. For example, energy consumption is much higher in the summer months than in the winter months because of air conditioners, higher on weekends than weekdays, and higher during dusk than during the day. Therefore, the following five features are used as inputs to the model:

1. One-Hour-Before Load (in KWh)
2. One-Week-Before Load (in KWh)
3. Hour in the day (1–24)
4. Day of the week (1–7)
5. Month of the year (1–12)

For the models that use Wavelet Transforms (WT), we applied Wavelet Decomposition to “One-Hour-Before Load" and "One-Week-Before Load” data and used the decomposition outcomes as inputs to the models. We chose four levels of decomposition, and therefore, the set of components fed to the WT models as inputs are as follows:

6. Details Level 1 of One-Hour-Before Load
7. Details Level 2 of One-Hour-Before Load
8. Details Level 3 of One-Hour-Before Load
9. Details Level 4 of One-Hour-Before Load
10. Approximation Level 4 of One-Hour-Before Load
11. Details Level 1 of One-Week-Before Load
12. Details Level 2 of One-Week-Before Load
13. Details Level 3 of One-Week-Before Load
14. Details Level 4 of One-Week-Before Load
15. Approximation Level 4 of One Week-Before Load

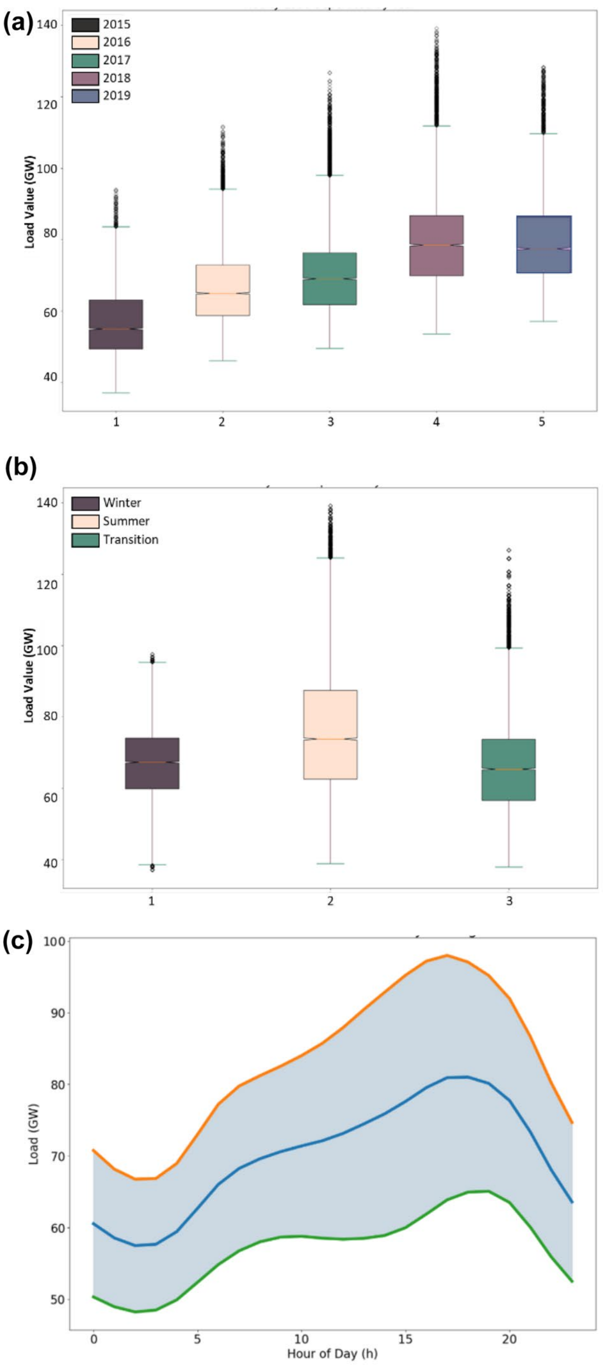

Figure 5 Box plot distribution of hourly load data for **a** Each year and **b** Each season **c** Standard Deviation Error Plot for Hourly Average Load for one typical day.

Figure 6 is a representation of the LSTM network showing the input data, which is a combination of metereological data, wavelet decomposition data and date indexes. A similar structure is used to feed data through the ANN networks by replacing the LSTM block with the ANN layers. Wavelet Decomposition is discussed in more details in Sect. 3.1.2.

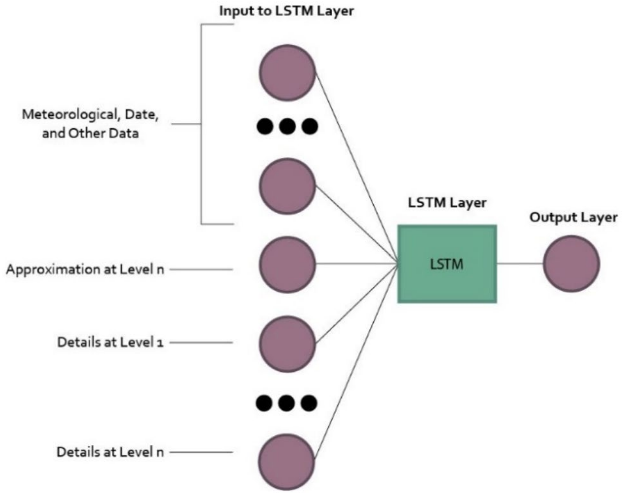

Figure 6 Diagram of LSTM Network Inputs.

3.1.1 Time-series Sequencing

Sequential models depend on receiving historical data and current data to learn the pattern behavior and form a prediction. To accomplish the time sequencing the historical data, current data and target forecast must be pre-processed and fed to the model in the same instance to enable the model to learn from the past sequences. The window size parameter decides the amount of previous data used for the current prediction and in this study, was set to use 30 prior data points. Each datapoint contains all the features for that hour. Figure 7 shows an example of an input variable x_t as input to the LSTM Layer with a window size of 7 to predict a single output h_t.

3.1.2 Wavelet Decomposition

The last type of data pre-processing implemented in this study uses Wavelets transformation. All load pattern signals are non-stationary and include a combination of variable

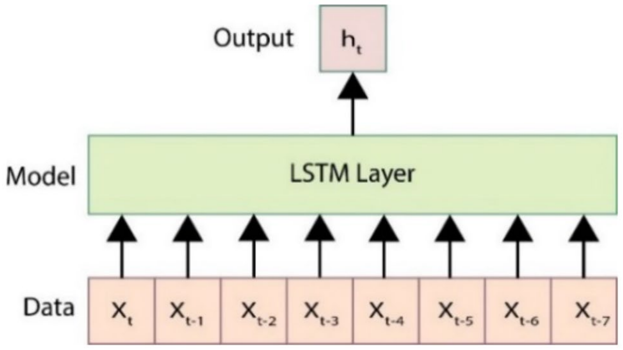

Figure 7 Example of Time Series Sequencing.

sub-patterns whose frequency is determined by several independent factors. For example, California has a few weeks with a temperature greater than 90°F that significantly hikes the consumer load demand, and this trend can be expected to re-occur annually (Fig. 2). Similarly, the weekend usage pattern can be intuitively hypothesized to have a different pattern compared to weekdays. Using frequency transforms dissociates these various frequency components before feeding them into separate prediction models, which usually improves the performance of the prediction model. We chose Wavelet Transform (WT) as this method provides both the signal's time and frequency localization characteristics. WT can be classified into Continuous Wavelet Transforms (CWT) or Discrete Wavelet Transforms (DWT). If x(t) is the data signal to be analyzed and ϕ(t) is the mother wavelet with scale factor ‘a’ and translation factor ‘b’, then the CWT is mathematically defined as [24]:

$$CWT_x(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty}\phi^*_{a,b}(t)x(t)dt \tag{7}$$

$$\phi_{a,b}(t) = \frac{1}{\sqrt{a}}\phi\left(\frac{t-b}{a}\right), a>0, -\infty<b<\infty \tag{8}$$

For the same sampled signal of known length T, the DWT is defined as Eq. (9). The DWT is more computationally efficient and easier to implement compared to CWT, which tends to produce several redundant terms:

$$DWT_x(m,n) = 2^{-m/2}\sum\nolimits_{t=0}^{T-1}x(y)\phi\left(\frac{t-n.2^m}{2^m}\right) \tag{9}$$

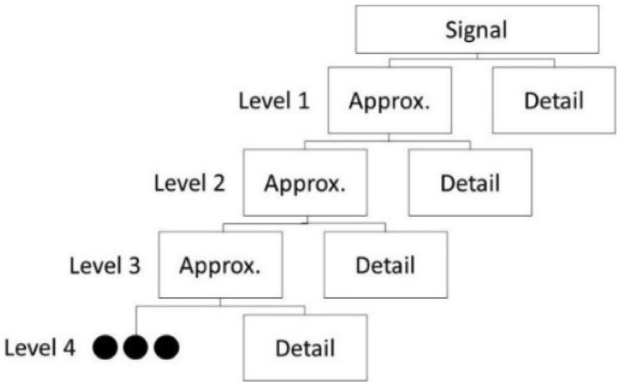

Figure 8 Wavelet Decomposition model of a signal observed at 3 levels of decomposition.

Here, a and b are discretized to 2^m and $n.2^m$ respectively. WT compares the signal to the shifted and scaled mother wavelet ϕ(t) [25] and the similarity helps in multiresolution analysis of the frequency spectrum. The wavelet decomposition filters the original signal into two groups of sub-signals: one with low frequencies (approximations) and higher frequencies (details). The approximations can be further decomposed into more levels to observe more approximations and details at lower frequencies. as shown in Fig. 8.

The choice of wavelet can drastically alter the outcome of the decomposition and must be chosen to match the signal shape as closely as possible. For example, we took a random two-week hourly load data sample from the complete California Load database and decomposed the signal using several different wavelet families, at different levels. The reconstructed signal was compared to the original signal (Table 1). It can be seen that Daubechies, Symlet and Coiflet wavelet families were almost able to re-create the original signal, while Meyer wavelet failed to do so.

Table 1 A comparison of reconstruction errors of the same signal by different mother wavelets, at 1–4 levels of decomposition.

Wavelet family	Level of Decomposition	Reconstruction Error
Meyer	1	0.29
	2	0.69
	3	1.29
	4	1.71
Daubechies (1 vanishing moment)	1	2.64×10^{-10}
	2	4.76×10^{-10}
	3	7×10^{-10}
	4	9.65×10^{-10}
Coiflet (1 vanishing moment)	1	1.91×10^{-8}
	2	9.72×10^{-8}
	3	3.80×10^{-7}
	4	6.87×10^{-7}
Coiflet (2 vanishing moment)	1	8.16×10^{-8}
	2	2.88×10^{-8}
	3	6.18×10^{-7}
	4	8.96×10^{-7}
Symlet	1	2.62×10^{-8}
	2	1.08×10^{-7}
	3	2.98×10^{-7}
	4	4.75×10^{-7}

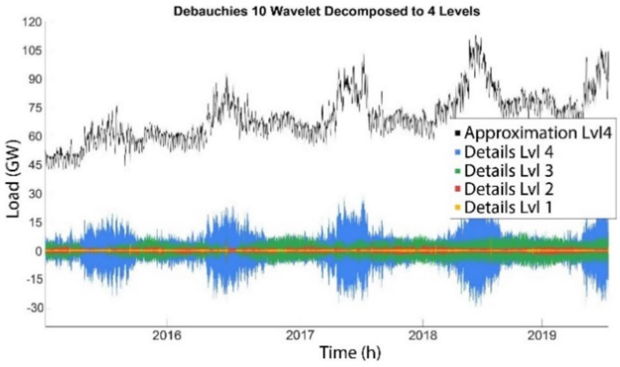

Figure 9 Wavelet Decomposition of CAISO Electrical Load Data to 4 Levels Using Daubechies 10 Wavelet.

This highlights the power of wavelet transforms in accurately breaking down the signal into high and low-frequency components, as can be seen by extremely low reconstruction errors on some wavelet families. A low reconstruction error means the components were close to natural waveforms and corresponding frequencies that exist in the original signal. However, decomposing the data into many levels does not necessarily improve the reconstruction performance as information integrity is lost beyond a certain sampling frequency. After decomposition, the data is normalized and fed into the model.

Figure 9 shows the decomposition of the load data using Daubechies wavelet. At the top are the lower frequencies or the "Approximation" at level 4 and is the signal that most resembles the original wave. The lower part of the figure shows all other higher frequency components are below the approximations due to their smaller weight compared to the lower frequencies in deciding the load demand.

3.1.3 Data Normalization

The input data was normalized between 0 and 1 to improve training performance and convergence. Doing this also prevents higher sensitivity to inputs with larger values that can cause unstable learning. The failure to normalize causes uneven training of the networks weight values, resulting in higher generalization error. The following formula was used for data normalization:

$$x_{i(n)} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{10}$$

In the equation above, x_i is the value to be normalized while x_{min} and x_{max} are the lowest and highest values within the sequence of inputs to the network.

3.2 Model Topology

For all the following models, the best values for the number of layers of LSTM, the number of units and activation function and the dropout for each LSTM were chosen using hyperparameter optimization and are reported in Sect. 4. For the first layer of LSTM in all the models returning sequences was set to true. For any second layer of the LSTM, it was set to false. All other values were set to defaults. Use of Bi-LSTM was out of scope for this work. For ANN models, all the values were left to default other than the ones that were used for hyperparameter optimization. All the values used for defaults were as specified in the Keras v2.4 [26].

3.3 Model Training

A topology of an LSTM network's training pipeline design is shown in Fig. 10. First, as a data processing step, a wavelet decomposition was performed on all the available load data.

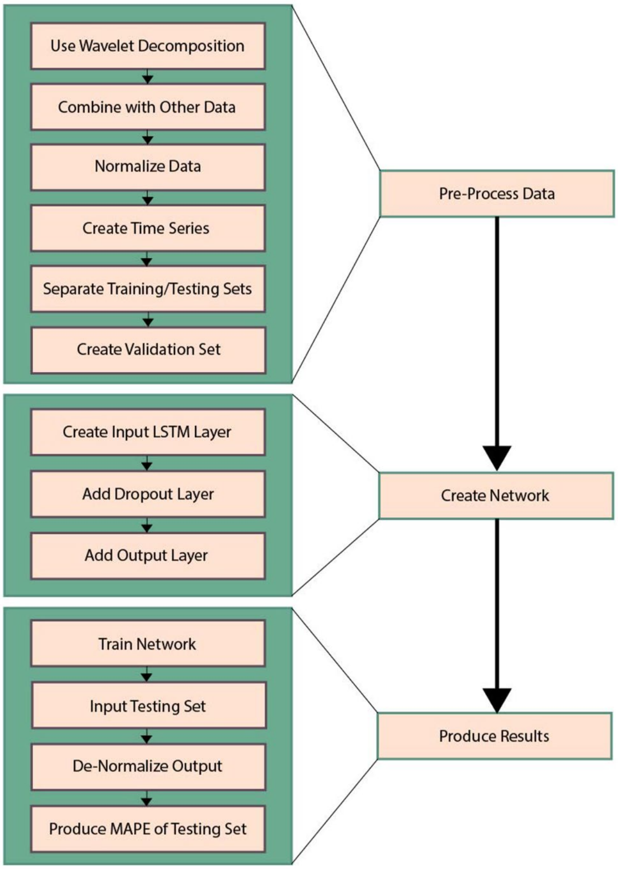

Figure 10 Flowchart of Process Execution for a general LSTM model example.

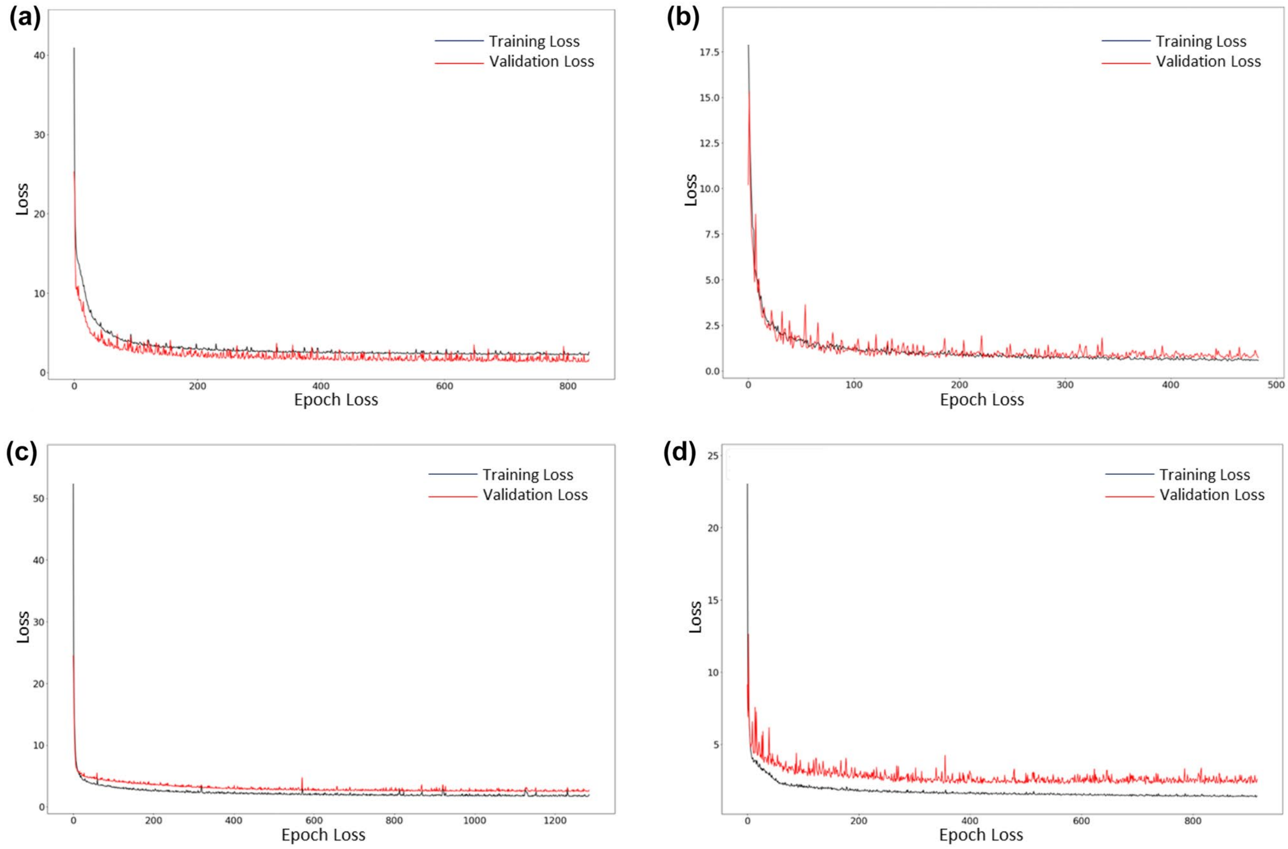

Figure 11 Training and Validation loss curves for **a** LSTM **b** WT-LSTM **c** ANN **d** WT-ANN.

The decomposed approximation and details thus obtained were then combined with the remaining meteorological data and day indicators. Each data type was normalized between 0 to 1 and put together in a time-series format. The complete data was then split into training and testing sets. Further, we split the training data into a validation set for training the neural network. We selected the most recent year's (2019) complete data as the testing set to prevent any data leakage that could occur with Wavelet decomposition. Also, testing on the entire data for one year allowed us to account for all the variability in load patterns that happened across the span of one year in our reported metrics.

This sequential training data was then fed into an LSTM Cell as described in Sect. 3.1.1. During training, the output from the LSTM was fed into a dropout layer that is used to improve the generalization performance of the network [27]. This dropout layer was turned off during testing. The predicted load from the output layer was then de-normalized before comparing it to the actual load by the MAPE metric. For training, the MAPE over a batch of input was computed as the loss value that was propagated back to the network for backpropagation training using 'Adam'. We used a batch size of 128 and maximum epochs of 1400. We also used early stopping and checkpointing with patience of 100 using a validation set. We tested four different setups of the pipeline. We designed LSTM and ANN models and then enhanced each type by hybridizing with Wavelet Transforms (WT), and the models were designated as WT-LSTM and WT-ANN, respectively. Although the flowchart above illustrates how an LSTM simulation is created, an ANN pipeline is very similar. The main differences come from the ANN not requiring a time series input, so there was no "Creation of Time Series" step, and a regular hidden layer was used in place of the drop out layer in Fig. 11 shows the training and validation losses for all models. In Figs. 12, 13, and 14 compare the actual data values vs

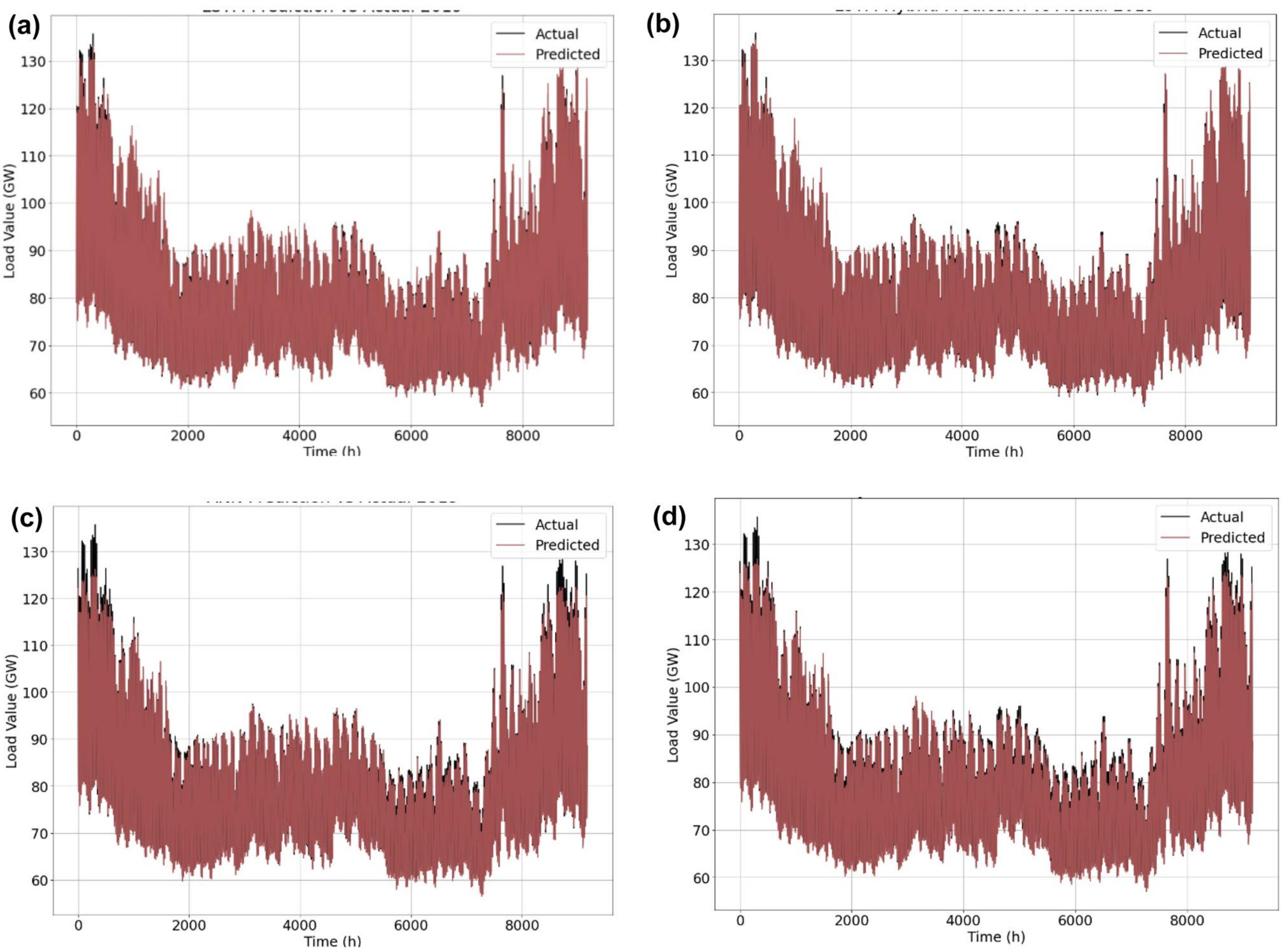

Figure 12 Predicted values vs Actual for **a** LSTM **b** WT-LSTM **c** ANN **d** WT-ANN. The prediction is hour-ahead for a year worth of data, and the horizontal axis is presented in hours starting on August 15th, 2018.

the predicted hour-ahead values by all models for an entire year, a week in summer and a week in winter, respectively. Hyperparameter optimization was done on each of the four configurations to obtain the reported results.

4 Results

A range of hyperparameters was tested with each of the four configurations (LSTM, WT-LSTM, ANN and WT-ANN). Separate optimizations were done on LSTM and WT-LSTM networks with wavelet decomposition of the signal data as inputs. For both tests, the chosen optimal hyperparameters were different, as presented in Tables 2 and 3. For the LSTM, the optimization chose a higher number of units per layer. It also chose 28.3% for dropout and "Scaled Exponential Linear Unit" as an activation function. In WT-LSTM, the optimization opted for more layers with fewer units in each, a dropout of 0% and "Sigmoid" as the activation function. The difference in hyperparameters between LSTM and WT-LSTM is most likely to compensate for the increase in input data because of the addition of wavelet amplitudes that the network needs to learn from. For the latter two configurations of ANN and WT-ANN, we saw the same pattern where WT-ANN needed more units per layer as compared to ANN. Using these parameters, the network was trained to obtain the MAPEs shown in Table 4.

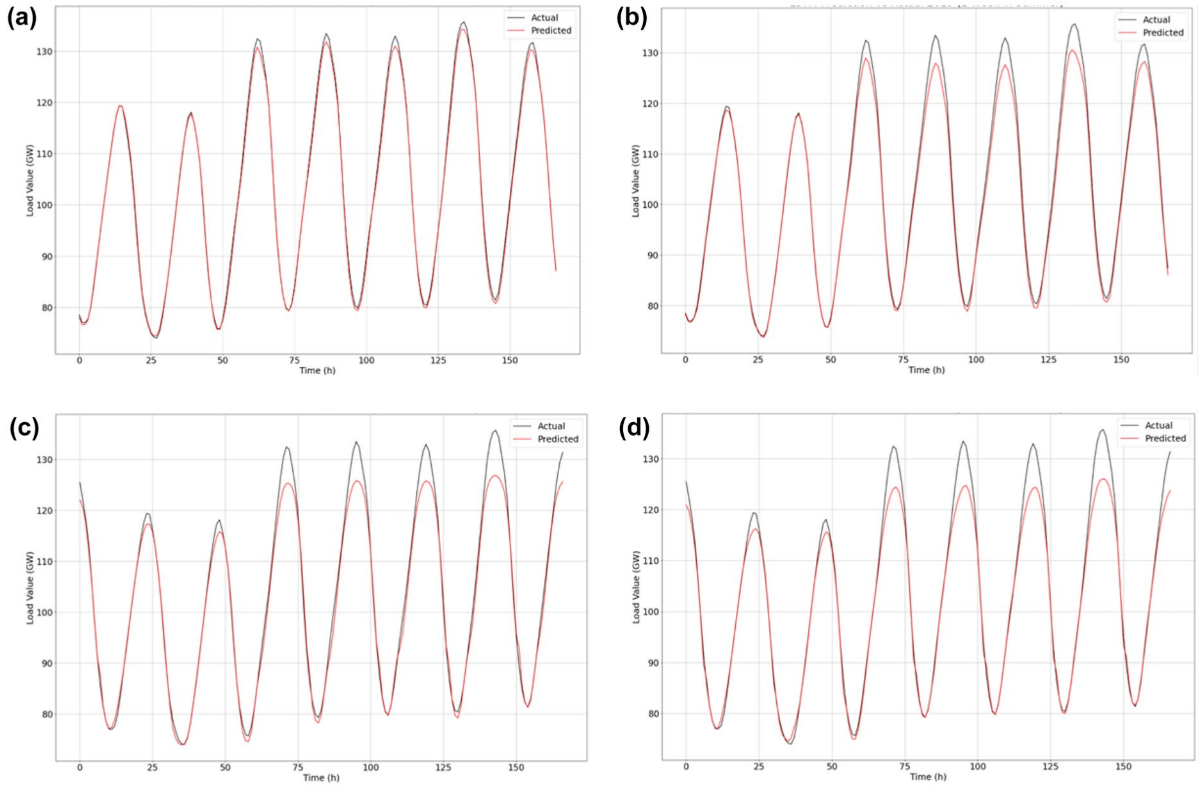

Figure 13 Predicted vs Actual for **a** LSTM **b** WT-LSTM **c** ANN **d** WT-ANN. The prediction is an hour-ahead for a week in summer.

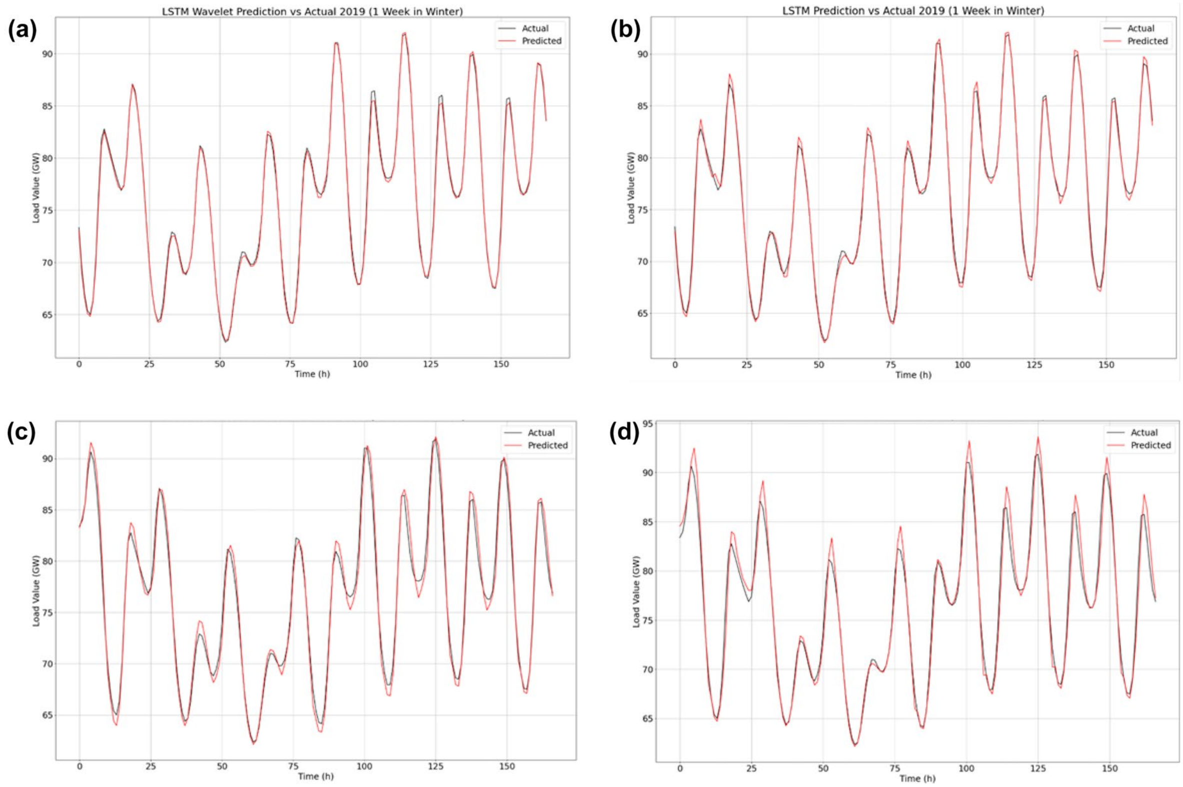

Figure 14 Predicted vs Actual for **a** LSTM **b** WT-LSTM **c** ANN **d** WT-ANN. The prediction is an hour-ahead for a week in winter.

Table 2 Hyperparameter optimization results for LSTM models.

Model	Layer No	Units/ Layer	Dropout/Layer	Activation Function
LSTM	1	106	28.3%	SELU
WT-LSTM	2	[98, 76]	[0%, 0%]	Sigmoid

Table 3 Hyperparameter optimization results for ANN models.

Model	Layer No	Units/ Layer	Dropout/Layer	Activation Function
ANN	2	[44, 110]	N/A	SELU
WT-ANN	2	[108, 41]	N/A	Tanh

Table 4 Best MAPEs of Each Network Organized by Type.

Model	MAPE (%)
LSTM	0.62
WT-LSTM	0.52
ANN	1.24
WT-ANN	1.04

5 Conclusion

In this paper, we compared many forms of data pre-processing, most notably wavelet decomposition and time-series sequencing, which increased prediction accuracy. These two methods were used in conjunction with Long Short-Term Memory Networks and Artificial Neural Networks to create a basis of prediction on the CAISO 2015–2019 dataset for California. LSTM, being more suited to time series data, performed significantly better than regular ANNs. Moreover, we found that adding wavelet decomposition to any model improved its accuracy. Both LSTM and ANN models saw an improvement in MAPE by 50% when combined with Wavelet Transforms.

In today's ever-expanding power grid that relies increasingly on renewable energy resources, the importance of predicting the load demand is vital to ensure a steady power supply. IoT sensors and devices across the grid allow us to gather crucial data that can be leveraged for making useful predictions. With the method proposed in this paper, hour-ahead load demand can be predicted very accurately using data from such sensors. Such predictions can be used to reliably schedule and manage power generation across fickle renewable energy sources.

References

1. Hong, T., & Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting, 32*, 914–938.
2. Li, L., Ota, K., & Dong, M. (2017). When Weather Matters: IoT-based Electrical Load Forecasting for Smart Grid. *IEEE Communications Magazine, 55,* 46–51.
3. Motlagh, N. H., Mohammadrezaei, M., Hunt, J., & Zakeri, B. (2020). Internet of Things (IoT) and the Energy Sector. *Energies, 13*, 1–27.
4. Jasmin, E., Ahamed, T., & Remani, T. (2018). Residential Load Scheduling With Renewable Generation in the Smart Grid: A Reinforcement Learning Approach. *IEEE Systems Journal, 13*(3), 3283–3294.
5. Morais, H., Kadar, P., Faria, P., Vale, Z., & Khodr, H. M. (2010). Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. *Renewable Energy, 35*(1), 151–156.
6. Lakshminarayan, S., & Kaur, D. (2018). Optimal maintenance scheduling of generator units using discrete integer cuckoo search optimization algorithm. *Swarm and Evolutionary Computation, 42*, 89–98.
7. Arora, I., & Kaur, M. (2016) Unit commitment scheduling by employing artificial neural network based load forecasting. In *7th India International Conference on Power Electronics, Patiala*.
8. Joy, V. M., & Krishnakumar, S. (2020). Efficient Load Scheduling Algorithm Using Artificial Neural Network in an Isolated Power System. *Inventive Computation Technologies* (pp. 615–621). Springer.
9. Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., & Zareipour, H. (2020). Energy Forecasting: A Review and Outlook. *IEEE Open Access Journal of Power and Energy, 7,* 376–388.
10. Hong, T. (2014). Energy Forecasting: past, present and future. *Foresight: The International Journal of Applied Forecasting, 32,* 43–48.
11. Mamun, A. A., Sohel, M., Mohammad, N., Sunny, M. S. H., Dipta, D. R., & Hossain, E. (2020). A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models. *IEEE Access, 8,* 134911–134939.
12. Park, D. C., El-Sharkawi, M. A., Marks, L. E. A. R. J., II., & Damborg, M. J. (1991). Electrical Load Forecasting Using an Artificial Neural Network. *IEEE Transactions on Power Systems, 6*(2), 442–449.
13. Alshareef, A., Mohamed, E. A., & Al-Judaibi, E. (2008). One Hour Ahead Load Forecasting Using Artificial Neural Network for the Western Area of Saudi Arabia. *International Journal of Electrical Systems Science and Engineering, 1,* 35–40.
14. Senjyu, T., Takara, H., Uezato, K., & Funabashi, T. (2002). One-Hour-Ahead Load Forecasting Using Neural Network. *IEEE Transactions on Power Systems, 17*(1), 113–119.
15. Ryu, S., Noh, J., & Kim, H. (2017). Deep Neural Network Based Demand Side Short Term Load Forecasting. *Energies, 10*(3), 1–20.
16. Deng, Z., Wang, B., Xu, Y., Xu, T., Liu, C., & Zhu, Z. (2019). Multi-Scale Convolutional Neural Network With Time-Cognition for Multi-Step Short-Term Load Forecasting. *IEEE Access, 7,* 88058–88071.
17. Chen, Q., Xia, M., Lu, T., Jiang, X., & Liu, Q. S. W. (2019). Short-Term Load Forecasting Based on Deep Learning for End-User

Transformer Subject to Volatile Electric Heating Loads. *IEEE Access, 7,* 162697–162707.

18. Jung, S., Moon, J., Park, S., & Hwang, E. (2021). An attention-based multilayer GRU model for multistep-ahead short-term load forecasting. *Sensors, 21*(5), 1639.
19. Chang, L., Zhijian, J., Jie, G., & Caiming, Q. (2017). *Short-Term Load Forecasting using A Long Short-Term Memory Network*. Shanghai: IEEE.
20. Jie, C., Qiang, G., & Dahua, L. (2019). *Improved Long Short-Term Memory Network Based Short Term Load Forecasting*. Tianjin: IEEE.
21. Islam, M. R., Al Mamun, A., Sohel, M., Hossain, M. L., & Uddin, M. M. (2020). *LSTM-Based Electrical Load Forecasting for Chattogram City of Bangladesh*. Pune: IEEE.
22. Muzaffar, A. A. S. (2018). *Short-Term Load Forecasts Using LSTM Networks*. Hong Kong: Energy Procedia.
23. Rafi, S. H., Masood, N., Deeba, S. R., & Hossain, E. (2021). A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network. *IEEE Access, 9*, 32436–32448.
24. Haque, A. U., Mandal, P., Meng, J., Srivastava, A. K., Tseng, T., & Senjyu, T. (2013). A Novel Hybrid Approach Based on Wavelet Trasnform and Fuzzy ARTMAP Networks for Predicting Wind Farm Power Production. *IEEE Transactions on Industry Applications, 49*(5), 2253–2261.
25. Gupta, S., Singh, V., Mittal, A., & Rani, A. (2016). *Weekly Load Prediction Using Wavelet Neural Network Approach*. New Delhi, India: University of Delhi.
26. Phyton. (2020). *Keras: Deep Learning for Humans Release 2.4.3*. MIT.
27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*(1), 1929–1958.