# Smart power consumption forecast model with optimized weighted average ensemble

**Alexander N. Ndife[1], Wattanapong Rakwichian[1], Paisarn Muneesawang[2], Yodthong Mensin[1]**
[1]School of Renewable Energy and Smart Grid Technology, Naresuan University, Phitsanulok, Thailand
[2]Department of Electrical and Computer Engineering, Naresuan University, Phitsanulok, Thailand

**Article Info**

***Article history:***

Received Sep 5, 2021
Revised Apr 20, 2022
Accepted May 19, 2022

***Keywords:***

Deep learning
Ensemble method
Forecasting
Neural networks
Power consumption

**ABSTRACT**

Smart power forecasting enables energy conservation and resource planning. Power estimation through previous utility bills is being replaced with machine intelligence. In this paper, a neural network architecture for demand side power consumption forecasting, called SGtechNet, is proposed. The forecast model applies ConvLSTM-encoder-decoder algorithm designed to enhance the quality of spatial encodings in the input feature to make a 7-day forecast. A weighted average ensemble approach was used, where multiple models were trained but only allow each model's contribution to the prediction to be weighted proportionally to their level of trust and estimated performance. This model is most suitable for low-powered devices with low processing and storage capabilities like smartphones, tablets and iPads. The power consumption comparison between a manually operated home and a smart home was investigated and the model's performance was tested on a time-domain household power consumption dataset and further validated using a real time load profile collated from the School of Renewable Energy and Smart Grid Technology, Naresuan University Smart Office. An improved root mean square error (RMSE) of 358 kwh was achieved when validated with holdout validation data from the automated office. Overall performance error, forecast and computational time showed a significant improvement over published research efforts identified in a literature review.



***Corresponding Author:***

Yodthong Mensin
School of Renewable Energy and Smart Grid Technology (SGtech)
Naresuan University, 99 Moo 9. Tha Pho Sub-district, Muang District, Phitsanulok 65000, Thailand
Email: yodthongm@nu.ac.th

## 1. INTRODUCTION

Reliable forecasts enable tracking the loads relative to proper balancing and creation of a dynamic energy pricing model and trading opportunities for energy users, using the knowledge of their anticipated power needs. Load forecasting is very useful in scheduling of devices [1], and energy trading that is becoming the centerpiece of a developing energy revolution. To analyze power consumption trends and to characterize patterns and develop forecasts, various statistical and traditional methods [2], [3] are used. However, modeling a complex real-world problem, such as power forecasting, with statistical linear models like autoregressive model (AR), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and seasonal autoregressive integrated moving average (SARIMA), is often difficult. These types of models cannot determine non-linear relationships in complex data, such as power consumption data with stochastic nature, therefore complex models, perhaps based on machine intelligence like neural networks, provide the analysis leverage necessary. Statistical tools from some industrial players like Prophet [4] from Facebook and Uber [5] that won the M4 Competition achieved some level of success because of the methodology likened to the use of dropout and its invariants in approximating a well-known probabilistic model, the gaussian process in neural networks. In contrast to statistical modelling, neural network models formulate a model based on features learned from existing data and this dependency makes them data-driven and self-adaptive, essential aspects for time series forecasting and where Big Data is involved. Although neural networks are preferable in most time series problems, they are not without their limitations. Problem of large number of trainable parameters sometimes makes neural networks models unimplementable in low-processing devices. For an instance, AlexNet, which won the 2012 ImageNet challenge, has about 60 million trainable parameters and VGGNet has a huge 138 million parameters. Although there has been continuous effort towards trainable set size reduction and overall performance optimization, more efforts are still needed. For instance, SqueezeNet was able to reduce its trainable parameters to 1.2 million while achieving a reasonable performance. These model size reduction efforts are important because real-world problems, including power consumption forecasting, requires real-time and on-device processing. It is not enough to have an accurate prediction model without ability to operate on resource-constrained low-power edge device without latency problem. Experimentally illustrated facts have shown that the model size affects its inference time [6], so the smaller the model size the faster the computational speed.

More recently, neural network methods have become very popular in time series forecasting due to the high performance achieved. Implementations in the form of deep learning algorithms have also become a turning point for both classification and regression tasks which, hitherto, have been difficult even on computers with excellent performance. Applying neural networks solution usually require training of large amounts of data to realize an appropriate machine learning model that can effectively be used in making projections. Given this, the model size obtained is normally big, requiring lengthy processing time. Therefore, a model compression technique is necessary to reduce the size and to expedite the computational process. Importantly, learning the arbitrary complex mapping from inputs to outputs has become the focus of research from which significant performance improvements have been achieved. However, a huge gap still exists between the methods of deployment and the implementation environment. Some of these gaps include a means to: capture the dominant factors in the data that need to be learned, as well as reducing the size of the model, increasing its inference time, and the selection of the model's parameters. These are the major areas that the proposed forecast model, discussed in this paper, aims to optimize.

Complex models based on deep learning, such as SGtechNet proposed in this paper, stand a better chance of addressing most of the noted difficulties of a complex real-world problem like power forecasting. It is intended that this model will be implemented in a low-powered-low-memory on-device mobile system, enabling smartphones to be used for demand-side energy management and control. It has been observed that the availability of high-speed graphics processing units (GPUs) in labs gives greater performance for models with larger trainable parameters, but these models are unusable in many real-world applications especially when implemented on resource-constrained devices. Achieving a lightweight model with a very high confidence in the predictions, was a major objective of our work. Based on this, an ensemble method together with advanced feature representation was used in combination with other improvement methods such as the layer compression technique to leverage improved forecast results. Many methods have yielded good model performance results but, in our work, we are more concerned on the scalable methods capable of optimizing the model training for quick convergence. aggregated deeep belief networks (DBNs) outputs using the support vector machine (SVM) algorithm, reported in [7], outperformed benchmark methods such as support vector regression (SVR), feedforward neural networks (FFNN), DBN and ensemble FFNN. The model compression algorithm implemented in the current work addresses the challenges of cost, power, heat and other related issues, all of which will be elaborated in the methodology discussion.

## 2. THE NETWORK ARCHITECTURE

Our proposed architecture as illustrated in Figure 1 is centered on optimizing neural networks learning process and mitigating its inherent challenges while achieving state-of-the-art forecast model. A weighted average ensemble method using multiple models with similar configurations, but different initial random weights is proposed. Those various models were trained on 3 different datasets including two load demand datasets from a household in France and the one from the smart office of SGtech, Naresuan University Thailand. However, combining predictions from multiple models can also add a bias that can make the model less sensitive to specifics in the training data, choice of training scheme and the serendipity of single training. It has been observed over time that ensemble methods, if not properly checked, might not ensure that the best-performing set of weights are used as a final model. So, our proposed method performed weighted average ensemble [8] as one of the ways of achieving a model ensemble in neural networks like voting [9] and stacking [10] and snapshot or checkpoint [11], among others, in a unique way. Here, instead of allowing equal contribution of all the models to the final prediction model, contributions were dependent on the level of trust and estimated performance, to ensure the performance of poorly performed models do not affect the overall forecast result. This method not only reduces the variance of predictions, but also reduces the generalization error.

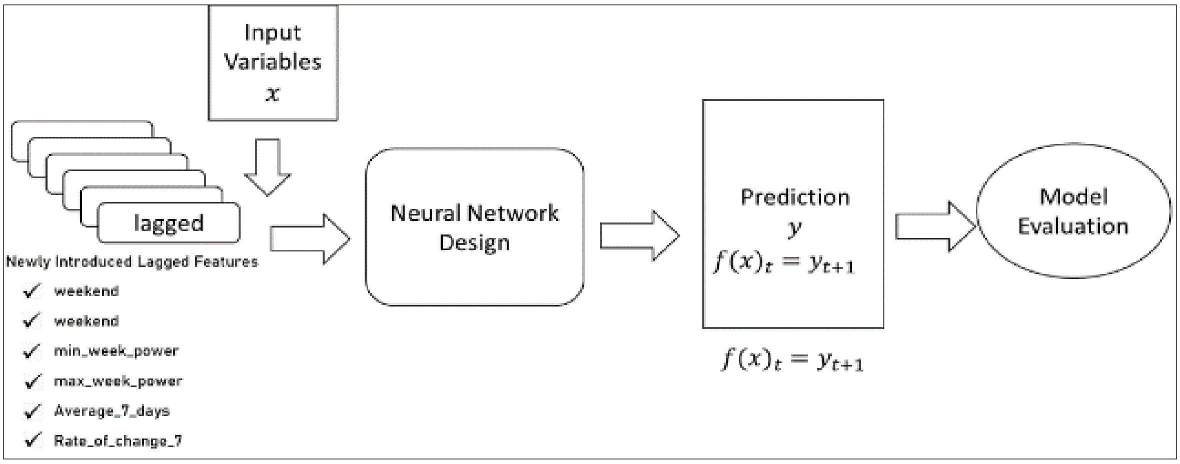



Figure 1. Proposed neural network forecast model

Aside from model improvement, the design for SGtechNet feature learning made it adaptable to different datasets including augmented power consumption dataset [12] from an automated office in such a way that it detected and analyzed the atmospheric climate changes. In our development process we considered the weather conditions all year round. To ensure that the real-time power consumption data used for both augmentation and validation of the model's performance captures this fact, we juxtaposed the power generation capacity of test environment Thailand on the load factors based on urban and rural characterization discussed in [13] to test if climate changes have any effect on the characteristics of household electricity consumption. Load factors, seasonal factors, and utilizations factor are some of the usage characteristics relevant to the power consumption of air conditioners, fans, refrigerators, water heaters and even washing machines and clothes driers. For example, especially in the case of the latter three domestic appliances, heating water or drying laundry may not in fact be necessary in a climate such as is experienced in Thailand, whereas it could be a significant use of power in cold climates. Table 1 describes Thailand's 2020 power statistics showing the monthly power generation capacity and load factors. Juxtaposing the generation capacity with the load demand as illustrated in Figure 2, sourced from [14], showed that the load factor surpassed generation capacity in March and September. This indicated the need to ensure that the validation data for the proposed model was tested on across the different seasons of the year. Also, the result of the preliminary analysis of weekday power consumption and generation/load demand discrepancy shown in Figure 3, emanating from data from the smart office that was used for the validation of the proposed model, showed the daily power consumption characteristics. These daily characteristics proved useful in determining the performance of the model.

Table 1. Thailand power statistics 2020 [14]

| | Jan. | Feb. | Mar. | Apr. | May | Jun. | Jul. | Aug. | Sep. | Oct. | Nov. | Dec. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GENERATION (GWh) | 16,138 | 15,477 | 17,618 | 15,715 | 16,899 | 15,887 | 16,390 | 16,348 | 16,195 | 15,457 | 15,292 | 14,483 |
| LOAD FACTOR (%) | 79.1 | 82.0 | 82.7 | 78.7 | 80.2 | 81.0 | 82.0 | 80.7 | 82.8 | 79.5 | 77.4 | 75.1 |

A description of time series modeling methods used by deep neural networks, for power consumption forecasting, has been introduced previously, together with discussion of the various methods identified in the literature. The organization of this paper includes, in section 2, related work, then the experimental and development methodology in section 3. section 4 presents the experimental results and discussion, and the paper is summarized in the conclusion.

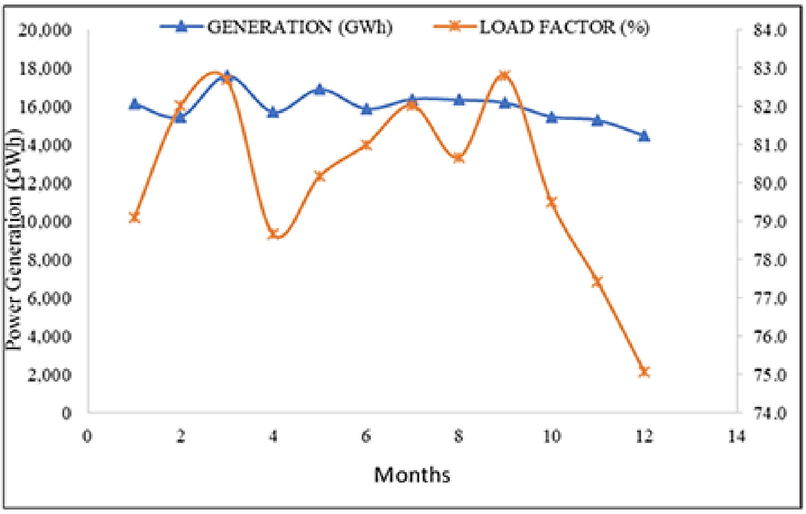



Figure 2. Annual power generation against load

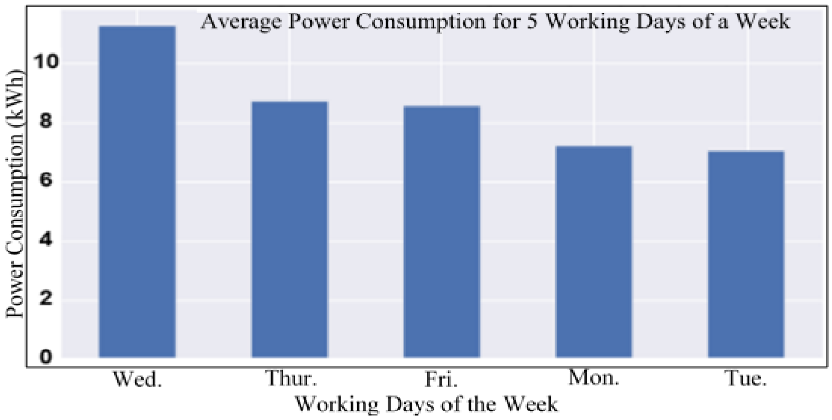



Figure 3. Daily power consumption

## 3. METHOD

Modeling power consumption of a smart home is very challenging due to its stochastic nature and non-linear relations over time. Given the sequence by sequence nature of the multivariate dataset used in this model, where an input sequence time (x = $x_1, x_2, \cdots, x_T$) with $x_t \ni$ Rn, and n is the variable dimension. Our objective was to predict the corresponding outputs (y = $y_1, y_2, \cdots, y_h$) at each time step. The expected result of this type of sequential modeling network is to obtain a nonlinear mapping of input sequence ($x$) to the prediction sequence ($y$) through optimization from the current state as:

$$(y_1, y_2, \cdots, y_h) = \frac{h}{2.m}(x_1, x_2, , \cdots, x_T) \tag{1}$$

Also, considering the neural network and its weights, the distinct forecast output gives:

$$y_i = f \sum_{i=1}^{n} w_i x_{1i} + b_i \tag{2}$$

where $x_i$ is the input to the neuron, $w_i$ is the weight of the network, $b_i$ is the bias in the network, $f()$ is the nonlinear function, while $y_i$ is the output. Therefore, our objective is to develop a network architecture capable of optimizing the mapping process. The development process started with the framing of the type of prediction we are interested in, considering the available datasets, and proceeded to how the network could be trained and validated and finally ended with the performance evaluation.

However, during the data preprocessing stage, we noticed non-stationarity and seasonality trends due to the spatiotemporal factors contained in the power consumption datasets. This prompted the decision to apply a different approach to modeling power consumption behavior for reliable forecasting. Statistical methods and neural network combinations [15]–[18] have been applied to regression problems of this nature with good results. Ordinarily, a stochastic method approach would have been the easiest to apply, particularly for power consumption forecasting, if not for its error susceptibility and inflexibility [1], [18], [19] implemented different types of neural networks for time series problems. However, as we are interested in predicting a week ahead horizon, we started our experimentation using the previous 7 days power demand as the input vector $x_i$, and the next 7-steps ahead as $y_i$ in our adaptive algorithm and continued varying the timesteps upwardly based on the hypothesis of the more the timesteps the better the prediction. Additionally, power consumption dependencies such as weather, calendar events (holidays, family social events, festival days and so on), other factors such as geographical locations, human comfortable temperature, heating/cooling technology, and type of consumers or purpose of electricity use industrial or residential, were included as additional lagged features to assist our model learn the data better.

This forecast method was implemented on deep learning encoder-decoder networks. Dropout, which have been a common technique in model regularization, were used to block out a random set of unit cells during model training to avoid overfitting. In (3) expresses the way in which this proposed model accepts multi-variant time series input variables and output 7 distinct forecasts ahead. The input parameters are the previously observed data at the scale time ($t+y_{-1}$, $t+y_{-2}$ … t). Therefore, the answer to finding the relationship between the input and output data) for the purpose of predicting the future data at the time (t+p) lies in the nonlinear functional mapping from the past observations of the time series to the future value, calculated in (3) and using (4).

$$y_t = \tilde{f}(y_{t-1}, y_{t-2}, \dots, y_{t-p}, W) + \varepsilon_t \tag{3}$$

where $w$ is a vector of all parameters and $f$ is the function determined by the network structure and the connection weights.

Using a simple feed-forward neural network architecture with 3-layers, for example, the output of the model can be computed as:

$$y_i = \alpha_0 + \sum_{j=1}^{q} \alpha_j\, g\left(\beta_{0j} + \sum_{i=1}^{p} \beta_{ij}\ y_{t-1}\right) + \varepsilon_t, \forall \tag{4}$$

At the instances $y_{t-i}$ $(i = 1, 2, 3, \dots, p)$ are the p inputs and $y_i$ is the output and p, q are the integer values of the number of input nodes and hidden nodes respectively, while $\alpha_j$ $(j = 0, 1, 2, \dots, q)$ and $\beta_{ij}(i = 0, 1, 2, \dots, p; j = 0, 1, 2, .., q)$ are the connection weights, and $\varepsilon_t$ is the random shock, $\alpha_0$ and $\beta_{0j}$ are the bias terms. For activation of this type of model, nonlinear activation functions such as the logistic sigmoid function or similar, such as linear, gaussian, hyperbolic tangent and so forth can be used. However, the estimation of the connection weights as a measure for minimizing the error function in this network can be done using the nonlinear least square method of (5).

$$F(\Psi) = \sum_t e_t^{\,2} = \sum_t (y_t - \hat{y}_t)^2 \tag{5}$$

In (5) applies an optimization technique for error minimization, where $\Psi$ is the space of all connection weights.

### 3.1. Dataset

Because of power consumption correlation to previous load consumption historical data and consumer behavior [20], this research leveraged secondary data from [12] that was augmented with remote-sensing data acquired from SGtech Smart Office. This secondary data was a multivariate time series dataset containing 2,075,259 measurements gathered from a house located in Sceaux, France, between December 2006 and November 2010 (47 months), recorded in real-time. The observations were made every minute and the temporal data captured the consumption behavior across different seasons of the year and weather conditions. Given that SGtechNet is interested in modeling power consumption behavior of a typical smart home, where all appliances are automated, we therefore validated the model performance with real-time data from a smart office. Our Smart Office data were collected through a smart means where devices in the automated office were configured to transmit data in real-time to a smart meter to enable profiling of each individual appliance, and their power consumption, and as well serving the purpose of power-quality monitoring. Figure 4 showed the ditributions of the variables, and we later added one additional variable, Sub_metering_4 as shown to Figure 5, to the original 7 independent variables that comprised the original dataset secondary data from [12]. This represents active energy for electric vehicle (EV) charging and other miscellaneous energy needs that were not accounted for in the original dataset. In the model design, additional features like weekend and weekday as shown in Figure 3, were added because Total Active Power consumed changes very much for weekdays and weekends.

Both datasets were split and 75% of each was used for model training, with the remaining 25% used for validation. The variables are obviously time dependent and can easily be influenced by changes in the weather. However, the unique characteristic of the weather suggests that location is an important determinant of a method to be applied in power forecasting. Location variation can invalidate the potency of a successful method when it is applied in another location with different weather and ambient characteristics. Therefore, a use case scenario [13] that characterizes power demand in urban areas, and rural areas across Thailand, was used for easy determination of likely energy demand in each category and to consider their various power consumption behavior. By this idea, the result of this predictions can therefore be compared with other predictions applicable to different locations and based on similar characterization**.**

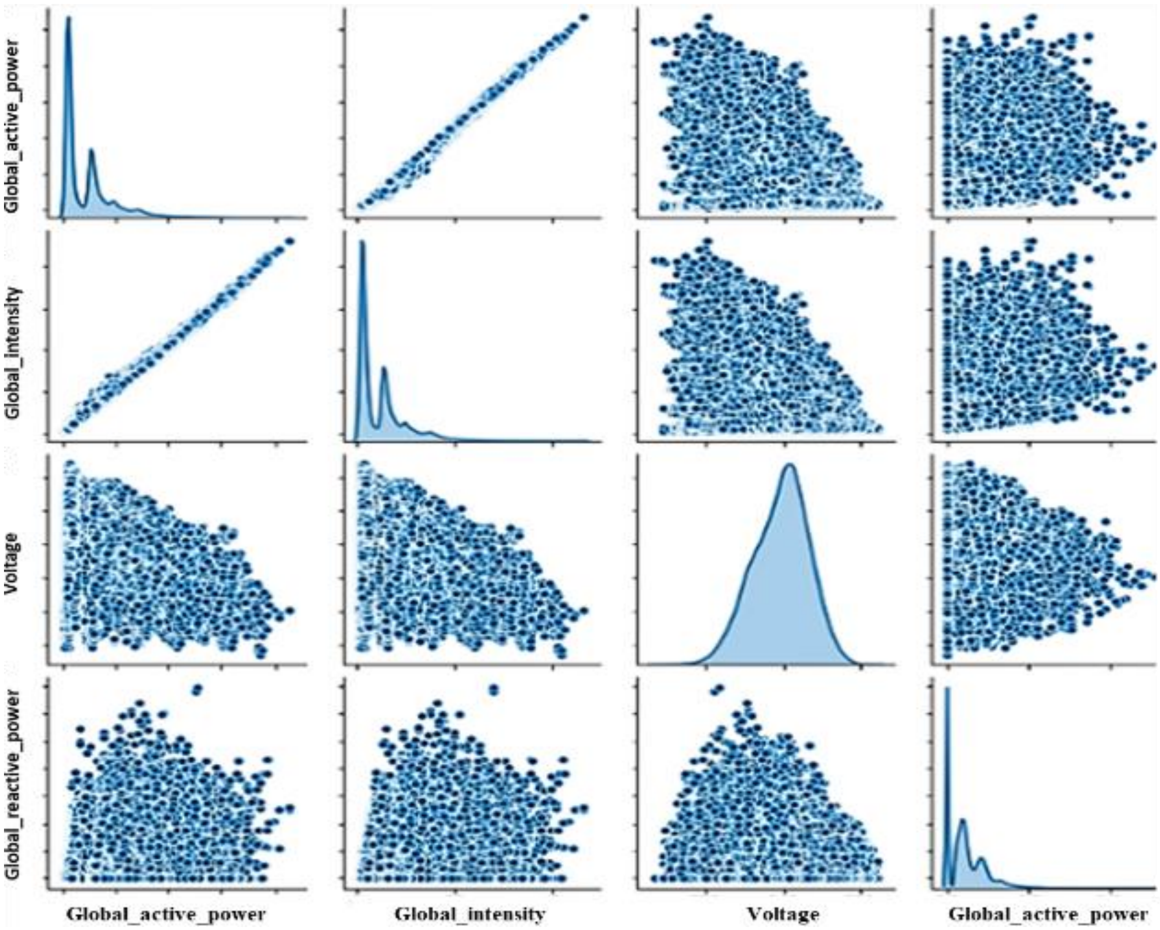



Figure 4. Plot of dataset variable distributions

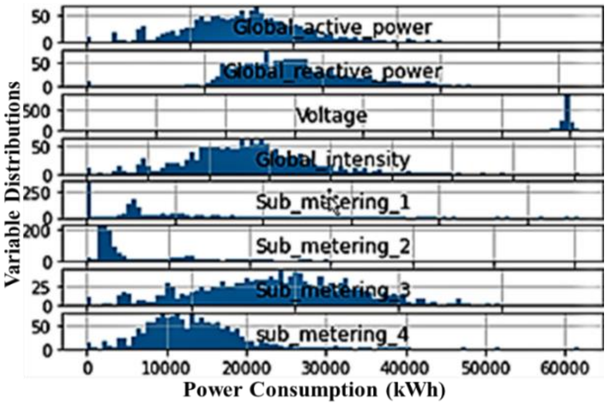



Figure 5. Individual distribution of the attributes

### 3.1.1. Date pre-processing

This public dataset was cleaned, and imputation method used to fill all missing and corrupted values using a day-wise last observation carried forward (LOCF) technique. This simply means carrying an observation from the same time the previous day. In a time-series data of this nature with seasonality trend, other methods like linear interpolation, seasonal adjustment + linear interpolation could also be applied.

From Figure 4, it can be noticed that voltage seems to have a gaussian distribution where as rest of the data seems skewed (i.e., non-symmetric), necessitating power transformation of the data before modelling. Exploratory analysis further showed that global active power expected to be predicted has strongest correlation with global intensity with a factor of 1. Therefore, this paper further investigates the extent each input variable affects the outcome of the prediction result of the global active power.

## 3.2. Model configuration

The architecture of the network has 7 input dimensions with 1 output layer, 3 convolutional and hidden layers each. This architecture consists of combination of convolutional neural network (CNN) and long sort-term memory (LSTM) deep networks. While the input transformations and feature representation take place in the convolutional layers, the resulting output is convolved and read into fully connected LSTM unit. Since the input data is a 1-D sequence, it was easy for the interpretation over the number of time steps. The LSTM has 3 hidden layers with 4 gates that handles updates and memory functions of the network. As the gates receives both the input output from the last convolutional layer obtained at previous time step ($h_{t-1}$) and the related current time ($x_t$) the forget gate takes $x_t$ and $h_{t-1}$ as input to determine the information to be retained in cell state ($C_{t-1}$) using sigmoid layer. $c_t$ and $c_{t-1}$ denotes cell states at timesteps $t$ and $t-1$ respectively. The value of $C_t$ is therefore determined by the input gate $i_t$ using $x_t$ and $h_{t-1}$. However, the function of the output gate is to regulate the output of LSTM cell based on $c_t$ using both sigmoid layer and tanh layer.

### 3.2.1. Network training

The network is trained to forecast the next consecutive 7 days a week ahead time steps using the learned features. Those additional features introduced during model design for the purposes of augmenting the data are concatenated to the vector and passed to the final prediction. Because ensemble method was used to ensure a better generalization, global optimization was consequently performed on the ensembled models to find the best coefficients for the weighted ensemble. The result of this optimization determines the individual contributions of the weight of each ensemble method to the final prediction.

### 3.2.2. Prediction/evaluation

Since various factors including atmospheric climate domain factors are some of the determinants of power consumption differences experience across different locations, SGtechNet analyzed those factors. Diverse atmospheric climate differences across locations prompted the need to validate the performance of this model using multivariate datasets collated from different locations France and Thailand precisely. Basically, to determine the effect and influence of climatic factors relative to performance for proper comparison with other forecasting methods. Therefore, series of experiments is conducted at different timesteps with the same model configuration to increase the confidence in the prediction and validity for future studies. To ascertain the effectiveness of this weighted average ensemble method against the backdrop of the limitation of poor performance resulting from allowing equal contributions from ensemble members to the final prediction model especially when some of the models are bad. And mitigate against the drawback of lengthy preference ordering calculation of individual ensemble members which often results to higher computational complexity in some ensemble techniques like voting [21], two predictions were considered: using different numbers of ensemble members in increasing levels of complexity across different timesteps and model averaging method. We started with 10 ensemble members whose contributions to the final prediction model is based on their confidence level and kept varying the numbers until we reached a standalone. It was discovered that there were no discrepancies in error when the number of ensemble members were varied. However, a significant discrepancy is reported using model averaging method where equal contribution from ensemble members was allowed. The prediction performance of the proposed model is computed based on root mean square error (RMSE) and compared against mean absolute percentage error (MAPE) and mean absolute error (MAE) errors see Figure 6 over averaging ensemble method and standalone method. These metrics are the most used performance measures for time series analysis because the error is of the same unit with the predictions and their errors can range from 0 to ∞. Figure 7 shows the validation loss across different timesteps (7, 14, 21 and 28). Walk-forward validation scheme was implemented, where the model made 1 week prediction, then utilized the actual data for the week or 2 weeks as a basis for the predicting the subsequent week.

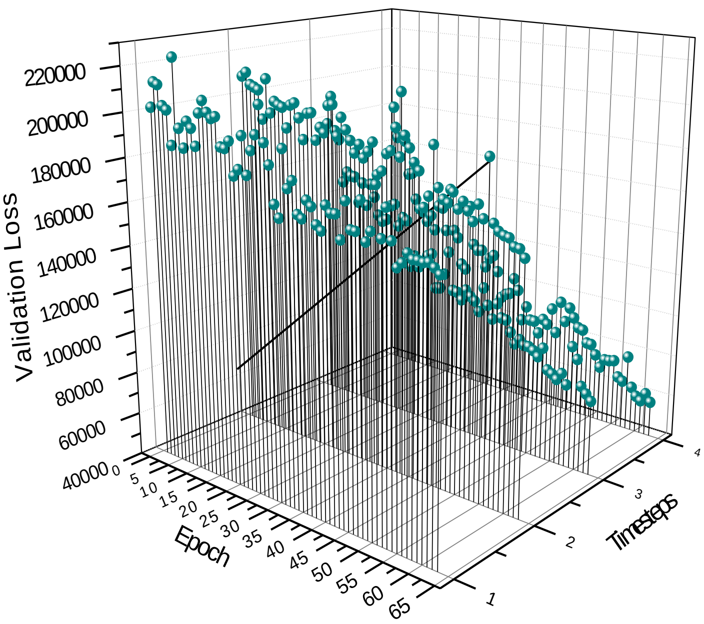



Figure 6. Power consumption across day and time

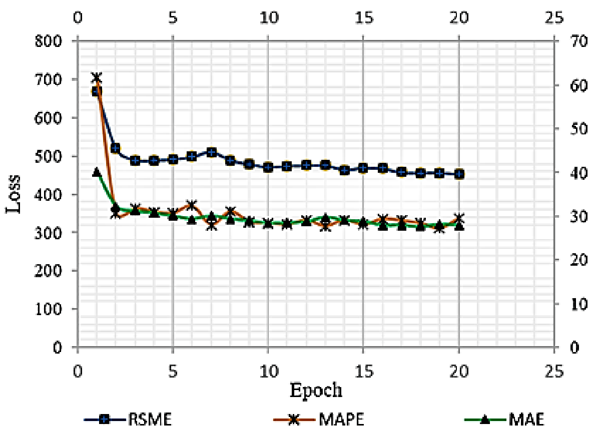



Figure 7. Validation loss across different timesteps

### 3.3. Encoder-decoder-network

We considered advanced feature representation methods, such as encoder-decoder, to preserve the hidden abstractions and invariant structures in the time series input. These have been previously applied in both reinforcement [22], supervised [23] and unsupervised learning. This unsupervised neural network method is designed for the adaptive learning of the long-term dependency and hidden correlation features of multivariate spatiotemporal data and was trained to reconstruct its own input in each layer as its output which is used as the inputs of the successive layer. In this paper, an encoder that extracts useful representative features from the time series input data was trained in such a way that the decoder could conveniently reconstruct those features from the encoded space. Specifically, the output of the convolutional layers is concatenated by Conv2D followed by LSTM layers, as achieved in [5], [24] to capture all the inherent spatiotemporal correlations in the time series input data. This proposed ConvLSTM encoder-decoder architecture has 2 sub-models: one for reading the input sequence and encoding (i.e., mapping the variable-length source sequence) this sequence into a fixed length vector, while the second part decodes the fixed-length vector and outputs the predicted sequence (i.e., mapping the vector representation back to a variable length target sequence). This output of the decoder represents the learned feature. Thereafter, a dense layer is used as the output for the network, and it uses the same weights by wrapping the dense layer in a time distributed wrapper function used in the network.

### 3.4. Model compression

On-device systems are resource-constrained, with limited memory and low computing power. However, deep learning algorithms are computational and memory intensive, so they cannot be implemented on real-world applications or other resource-constrained systems without difficulties. As deep learning models goes deeper in layers their inference time increases along with the increase in number of trainable parameters; making it difficult to be deployed on resource-constrained devices. By the parsimony concept, models with a smaller number of parameters are more likely to provide adequate representation of the underlying time series data, but models with a high number of trainable parameters requires more energy and space and are likely to overfit during training. Consequently, compression technique, as presented in [25], is required to allow the deployment of a large model on resource-constrained devices. Table 2 summarized the results from literature on the most recent efforts towards model size and trainable parameter reduction leveraging on different techniques in comparison with SGtechNet. This comparative analysis shows that SGtech has the least number of trainable parameters with a very considerate model size, hence the justification for its suitability for low-power-low-memory devices. Model size is very important as far as performance optimization of on-device system is concerned because larger models mean more memory reference and more energy [26].

Table 2. Model parameter comparison

| Model | Parameters | Size | Training Time | Inference Time |
|---|---|---|---|---|
| ENet [27] | 0.37 M | 0.7 MB | 15mins | 383ms |
| LEDNet [28] | 1.856 M | 3.8 MB | - | - |
| SegNet [29] | 29.46 M | 56.2 MB | 37mins | 286ms |
| AlexNet [30], [31] | 60 M | 232 MB | 7,920mins | - |
| VGG16 [31], [32] | 138 M | 528 MB | - | - |
| SqueezNet [25] | 0.66 M | 4.8 MB | - | - |
| ResNet152 [31] | 232 M | 60 MB | - | - |
| GoogleNet [31] | 6.8 M | 28 MB | - | - |
| **SGtechNet (Proposed)** | **128K** | **4.93 MB** | **1.3mins** | **3ms** |

Therefore, to fit the SGtechNet model on limited resourced devices, enabling the model to be usable in real-world applications, the SqueezeNet [25] concept of was used, with the modification that the 1x1 and 1x3 convolution filters were used for feature representation. As each kernel receives an input time series, the corresponding outputs are concatenated and followed by convolutional-LSTM layers which capture the long-term spatial patterns in the electricity consumption data. This method not only reduces input data dimensionality but also reduces the complexity of the data [33] leading to an improved result even though a marginal cost burden is incurred due to a slight increase in number of parameters. However, the choice of a smaller filter reduces the models inference time. Also, SqueezeNet has almost the same accuracy of AlexNet with its compression of trainable parameters, but that accuracy is a little lower than GoogleNet. SeNet [34] developed an architecture that recalibrates channel-wise feature responses and uses them to determine the interdependencies existing between two channels. Channel-wise scale and element-wise summation operations were combined into a single layer “AXPY ”using skip-connections. This resulted in considerable reductions in memory, cost, and computational burden. It is imperative to note that the application environment of most of the state-of-the-art models in Table 2 is image classification and detection, so for SGtechNet to achieve a RMSE Error of 358kwh in a regression task like power forecasting shows high level of robustness. Even though training and inference time for some of these models compared with SGtechNet was not reported in the literature, the few ones that were reported clearly put SGtechNet at advantage in terms of computational complexity.

### 3.5. Feature representation

Feature learning or representation learning in machine learning is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. Figure 8 shows the feature learning process. This is a method of finding a representation in each data the features, the distance function, and the similarity function-dictates how the predictive model will perform. Feature representation helps to reduce data complexity, so the anomalies and noise can be reduced. It also helps in reduction of the dimensional of input data, making it easier to find patterns, anomalies, and also provides a better understanding of the behavior of the data generally. Because our time series input data is 1D, a smaller kernel filters (1, 3) were used in the convolutional layers for feature learning.

Considering the spatiotemporal nature of power consumption variables, a state space representation of (6) represents the transition process expressing the discrete stochastic behavior of the variables and (7) represents the likelihood of the observations with the assumption that states are part of the model parameters.

$$\varphi_{i+1} = \varphi_i + u_t \tag{6}$$

$$y_t = \tilde{f}(x_t, \Theta_t) + V_t \tag{7}$$

where $u_t$ is the process noise, $v_t$ is the measured noise.

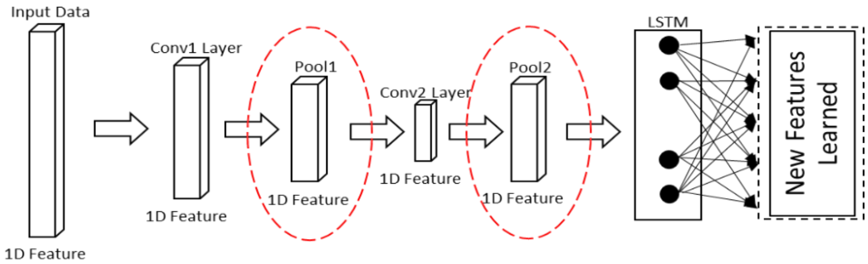



Figure 8. Feature learning process

### 3.6. Ensemble method

The stochastic nature of power consumption varying with season and time necessitated the use of a stochastic learning algorithm for dataset training. However, the neural network algorithm has the inherent limitation of randomness which results in a different final model each time it is trained on the same dataset. To address this limitation, an ensemble method of [7] with a weighted average of different trained models is used for prediction. Ordinarily, the model ensemble method allows each model an equal contribution to the final prediction which could sometimes be seen as a limitation when the contribution from poorly performed models to the final model jeopardizes the efforts of a well performed model. However, the contribution to the final model in this proposed model is purely dependent on the model's trust and estimated performance, resulting to an improved overall prediction result.

Sensitivity analysis was carried out to determine the number of ensemble members most appropriate for the forecasting problem and how impactful they could be to the test accuracy. To determine trustworthiness of ensemble models and to estimate performance, we need to find their weights. However, due to there being no analytical solution to estimation of values for the weights, we used gradient descent optimization with a unit norm weight constraint on the holdout validation set rather than on the training set. Ordinarily, a simpler way of finding each ensemble member's weights would have been to grid search values but because our holdout validation is large enough, gradient descent optimization becomes the best option.

This optimization procedure sums up all the model vector of weights to 1 i.e., $w_1, w_2, \cdots, w_k = 1$, also constrains them to positive values to allow weights to indicate the percentage of trust or expected performance of each model. The optimization process utilizes the set of information provided to it to search for weights with lower errors under defined bound (i.e., 0.0–1.0) amongst 10 ensemble members until convergence. But, before performing weight optimization, 10 single models were created, and their individual performances were evaluated on the test dataset. For the optimization, a differential-evaluation function was used to search and display the optimal sets of weights after several iterations which returned the score to be minimized and retrieved the best weights, with their performance being reported on the holdout validation data. Optimal weights of the base learners are aggregated to find the best tradeoff between bias and variance and minimize the prediction error. So, each base learner's prediction ($\hat{y}_t$) on holdout validation set, therefore gives:

$$Min\ Error\ (w_1\hat{y}_1 + w_2\hat{y}_2 + \cdots, w_k\hat{y}_k, y) \tag{8}$$

such that $\sum_{j=1}^{k} W; = 1$, when $W_j \geq 0\ \forall_j = 1, \ldots k$, where $W_j$ represents the weights corresponding to base model $j$ (j = 1,..,k), ŷ is the vector predictions of base model $j$, and y is the vector of true value. So, at any instance of training the base learner $j$, weights $W_j$ is computed from optimization (4) on the assumption that *n* is the total number of instance, $y_i$ as the true value of observation *i*, $\hat{y}_{ij}$ as the prediction of observation *i* by base model $j$.

$$Min\ \frac{1}{n}(\textstyle\sum_{i=1}^{n} y_i - \sum_{j=1}^{k} W_j\,\hat{y}_{ij}) \tag{9}$$

Such that $\sum_{j=1}^{k} W; = k, when\ W_j \geq 0\ \forall_j = 1, \ldots k$

The ensemble member contributions are evaluated based on those chosen weights. This process not only improve model performance but also saves time. Ordinarily, the search for such weights with lower error values would need to be done randomly and exhaustively, which is time demanding.

### 3.6.1. Comparing weighted ensemble and model averaging method performance

Table 3 shows the results produced by the weighted average ensemble method, which demonstrate that this method outperformed the model averaging method for individual ensemble members even though their processing time variation is insignificant. Furthermore, the model's performance is compared with baseline model see Table 4 using both secondary and primary datasets acquired from two different continents. The importance of this comparative analysis is to provide completeness of this study analysis as regards the major limitation of ensemble technique which is misleading assumption that all ensemble members are equally effective.

Table 3. Comparative analysis of weighted ensemble models and model averaging method

| Statistics | Weighted Ensemble Models | Model Averaging Method |
|---|---|---|
| Number of Iteration | 1,000 | 1,000 |
| Validation Time | ***2.053s*** | 2.185s |
| Average RMSE | ***358kwh*** | 362.617kwh |

Table 4. Comparative analysis of weighted ensemble models and baseline model on different datasets

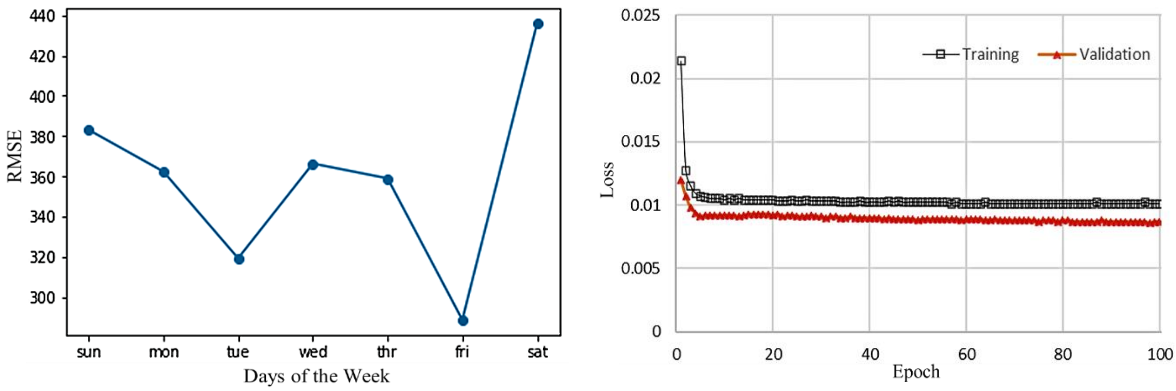
| Model Statistics | Training on HHPC Dataset France | | Training on Real-Time Dataset, SGtech, Thailand | | | |
|---|---|---|---|---|---|---|
| | Propose Model | Baseline Model | Propose Model | Persistence Model | | |
| | | | | Hourly | Daily | Weekly |
| Training Time | 114.109s | - | 78.916s | - | - | - |
| Prediction Time | 2.282s | - | 2.053s | - | - | - |
| RMSE | 3.61.885kwh | 465.294kwh | **358kwh** | 480.246kwh | 469.389kwh | 465.294kwh |

## 4. RESULTS AND DISCUSSION

A real-time experimentation using Google Colab TPU and one of the finest neural network APIs, contained in Keras® with its backend TensorFlow produced the results shown in Table 3 and Table 4. Based on the performance evaluation of the model, this model significantly outperformed the baseline model. Though an unstable training trajectory was experienced during training, which could be likened to overfitting in the training data, the overall performance is good. In the model's evaluation result of Figure 9, RMSE was found to statistically differ across the 7 days of the week as shown in Figure 9(a) while Figure 9(b) showed how the training error decreased sharply after commencement of training, before it became linear, due to the model's complexity: likewise, the validation error. A squeeze layer technique, adopted from [35], reduced the size of the model to 4.9 M without affecting its performance, making it implementable in a low-power-low-memory device; Smartphone, iPad, Tablets. One of the limitations of the model performance enhancement method being discussed in this paper is that, as the model size is reduced, the number of parameters slightly increased, resulting in a marginal increase in resources usage relative to implementation. Therefore, further work is proposed to develop a systematic method of reducing the model size without necessarily increasing the number of model parameters.

### 4.1. Comparative analysis

Evaluation of this model's performance was against the baseline model and other alternative forecasting methods even though some metrics, such as computational speed and prediction time, were not captured in all the literature reviewed. We also analyzed power consumption datasets used in validating SGtechNet along daily consumption cycles time and day as shown in Figure 10(a) and 10(b) respectively, for clearer understanding of residents habits. An experimental framework for the empirical comparison of different model performances, based on varying test conditions, was introduced. Uniqueness of weather characteristics in different locations indicated that there is no guarantee that a forecasting method that is successful at one location would be effective at a different location. The inclusion of this framework in the design accounts for diverse climatic conditions and created a valuable environment for future studies in emerging forecasting technologies. This increases the confidence in the observed results by allowing the validity of the forecasting algorithm to be tested on both the test set from France and the test set from SGtech Naresuan University Thailand, both of which are real-time data. Alternatively, to prove that the improved processing time and other improvements achieved in this model are due to pure scientific contributions rather than software and hardware differences, we experimented on different technologies. We compared the results when using an NVIDIA GeForce GTX1080 TI GPU/TPU enabled TensorFlow against those achieved when using an NVIDIA Tesla K80 GPU running on the Ubuntu Server 16.04.3. The discrepancy in the results was found to be scientifically insignificant. The result of this model is further compared with model averaging and standalone methods as shown in Table 5. SGtechNet model size is 4.93 MB which means it can easily be put in an on-chip Static random access memory (SRAM) cache.

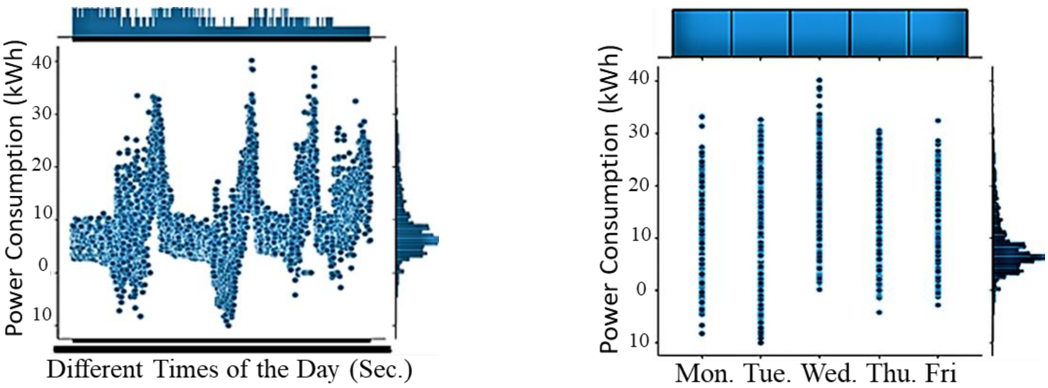



(a) (b)

Figure 9. SGtechNet performance evaluation result showing (a) RMSE across the 7 days power consecutive days forecasted and (b) Model's training and validation loss

Table 5. Comparison of the experimental results of proposed model against some existing power forecast methods

| Model Statistics | Models | | | |
|---|---|---|---|---|
| | Propose Model | Persistence Model | Model A [36] | Model B [37] |
| Training Time | 114.109s | - | - | - |
| Prediction Time | 2.282s | - | - | - |
| Size | ***4.935MB*** | - | - | - |
| RMSE | ***358kwh*** | 465.294kwh | 530kwh | 450.5kwh |




(a) (b)

Figure 10. Shows the plot of power consumption across different (a) Time and (b) Weekday

## 5. CONCLUSION

Nationwide lockdown due to covid-19 pandemic is causing a rise in domestic power consumption, making energy conservation, and planning more relevant than ever. In our research, we demonstrated the effectiveness of combining atmospheric climate domain knowledge of factors determining power consumption differences based on location, with empirical data captured from automated systems for future energy forecasting. This forecast model SGtechNet developed to optimize the data learning and prediction process leveraged on a multivariate dataset to make a multi-step time series 7 days ahead forecast. SGtechNet, is based on ConvLSTM-Encoder-Decoder algorithm explicitly designed to optimize the quality of spatiotemporal encodings throughout the feature extraction process. The validation report of this model showed a significant improvement on the forecast result when a real-time dataset from an automated office was used for model validation which was compared against a manually operated home/office represented by the secondary data. This implies, aside from the social behavioral factor that propels the users 'choice of time of use (ToU) electricity, that environmental and real-time control factors are also contributory factors that determine the consumption rate and therefore cost of power that is consumed domestically or in an office workplace. The RMSE of 361 kwh recorded was compared with 465 kwh on the persistence model and an improved RMSE of 358 kwh was achieved when validated in holdout validation data from the automated office. Overall performance on error rate, forecast time and inference time were later compared with published research, and the comparison showed that our model, the SGtechNet, provided significant improvements in these factors. One of the most significant achievements of SGtechNet is its adaptiveness to other forecast problems and different datasets in such a way that it detected and analyzed the atmospheric climate changes over different locations.

## REFERENCES


[1] K. M. U. Ahmed, M. Ampatzis, P. H. Nguyen, and W. L. Kling, "Application of time-series and Artificial Neural Network models in short term load forecasting for scheduling of storage devices," Sep. 2014, doi: 10.1109/upec.2014.6934761.

[2] G. K. F. Tso and K. K. W. Yau, "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks," *Energy*, vol. 32, no. 9, pp. 1761–1768, Sep. 2007, doi: 10.1016/j.energy.2006.11.010.

[3] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: Literature survey and classification of methods," *International Journal of Systems Science*, vol. 33, no. 1, pp. 23–34, Jan. 2002, doi: 10.1080/00207720110067421.

[4] S. J. Taylor and B. Letham, "Forecasting at Scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, Jan. 2018, doi: 10.1080/00031305.2017.1380080.

[5] L. Zhu and N. Laptev, "Deep and Confident Prediction for Time Series at Uber," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 103–110, doi: 10.1109/icdmw.2017.19.

[6] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the Computational Efficiency of Training Neural Networks," *dvances in neural information processing systems*, vol. 27, 2014.

[7] X. Qiu, L. Zhang, Y. Ren, P. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," Dec. 2014, doi: 10.1109/ciel.2014.7015739.

[8] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging Weights Leads to Wider Optima and Better Generalization," Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.05407.

[9] O. Gokalp and E. Tasci, "Weighted Voting Based Ensemble Classification with Hyper-parameter Optimization," in *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2019, pp. 1–4, doi: 10.1109/asyu48272.2019.8946373.

[10] J. Xie, B. Xu, and Z. Chuang, "Horizontal and Vertical Ensemble with Deep Representation for Classification," Jun. 2013, [Online]. Available: http://arxiv.org/abs/1306.2759.

[11] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot Ensembles: Train 1, get M for free," Mar.

2017, [Online]. Available: http://arxiv.org/abs/1704.00109.
[12] D. D.a.K.T., "Household Electricity Consumption Dataset," *UCI Machine Learning Repository*, 2017. http://archive.ics.uci.edu/ml.
[13] K. Poolsawat, W. Tachajapong, S. Prasitwattanaseree, and W. Wongsapai, "Electricity consumption characteristics in Thailand residential sector and its saving potential," *Energy Reports*, vol. 6, pp. 337–343, Feb. 2020, doi: 10.1016/j.egyr.2019.11.085.
[14] "Energy Statistics of Thailand," *Energy Policy and Planning Office*, 2020. http://www.eppo.go.th/index.php/en/ (accessed Mar. 05, 2021).
[15] A. Mellit, M. Benghanem, A. H. Arab, and A. Guessoum, "A simplified model for generating sequences of global solar radiation data for isolated sites: Using artificial neural network and a library of Markov transition matrices approach," *Solar Energy*, vol. 79, no. 5, pp. 469–482, Nov. 2005, doi: 10.1016/j.solener.2004.12.006.
[16] L. Marwala and B. Twala, "Forecasting electricity consumption in South Africa: ARMA, neural networks and neuro-fuzzy systems," Jul. 2014, doi: 10.1109/ijcnn.2014.6889898.
[17] C. Nichiforov, I. Stamatescu, I. Fagarasan, and G. Stamatescu, "Energy consumption forecasting using ARIMA and neural network models," Oct. 2017, doi: 10.1109/iseee.2017.8170657.
[18] Z. Haydari, F. Kavehnia, M. Askari, and M. Ganbariyan, "Time-series load modelling and load forecasting using neuro-fuzzy techniques," Oct. 2007, doi: 10.1109/epqu.2007.4424201.
[19] F. Fahmi and H. Sofyan, "Forecasting household electricity consumption in the province of Aceh using combination time series model," Oct. 2017, doi: 10.1109/iceltics.2017.8253239.
[20] B. Dhaval and A. Deshpande, "Short-term load forecasting with using multiple linear regression," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, pp. 3911–3917, Aug. 2020, doi: 10.11591/ijece.v10i4.pp3911-3917.
[21] F. Leon, S.-A. Floria, and C. Badica, "Evaluating the effect of voting methods on ensemble-based classification," Jul. 2017, doi: 10.1109/inista.2017.8001122.
[22] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *Advances in neural information processing systems*, vol. 28, Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.04214.
[23] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Multivariate time series forecasting via attention-based encoder-decoder framework," *Neurocomputing*, vol. 388, pp. 269–279, May 2020, doi: 10.1016/j.neucom.2019.12.118.
[24] R. Asadi and A. C. Regan, "A spatio-temporal decomposition based deep neural network for time series forecasting," *Applied Soft Computing*, vol. 87, p. 105963, Feb. 2020, doi: 10.1016/j.asoc.2019.105963.
[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," Feb. 2016, [Online]. Available: http://arxiv.org/abs/1602.07360.
[26] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," Feb. 2014, doi: 10.1109/isscc.2014.6757323.
[27] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," Jun. 2016, [Online]. Available: http://arxiv.org/abs/1606.02147.
[28] N. Rangappa, Y. R. V. Prasad, and S. R. Dubey, "LEDNet: Deep Learning-Based Ground Sensor Data Monitoring System," *IEEE Sensors Journal*, vol. 22, no. 1, pp. 842–850, Jan. 2022, doi: 10.1109/jsen.2021.3129173.
[29] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/tpami.2016.2644615.
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
[31] M. P. Véstias, "A Survey of Convolutional Neural Networks on Edge with Reconfigurable Computing," *Algorithms*, vol. 12, no. 8, p. 154, Jul. 2019, doi: 10.3390/a12080154.
[32] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556.
[33] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," Apr. 2015, doi: 10.1109/itw.2015.7133169.
[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," Jun. 2018, doi: 10.1109/cvpr.2018.00745.
[35] G. Nanfack, A. Elhassouny, and R. O. H. A. J. Thami, "Squeeze-SegNet: a new fast deep convolutional neural network for semantic segmentation," in *Tenth International Conference on Machine Vision (ICMV 2017)*, Apr. 2018, doi: 10.1117/12.2309497.
[36] E. Busseti, I. Osband, and S. Wong, "Deep learning for time series modeling," *Technical report, Stanford University*, pp. 1–5, 2012.
[37] H. Shi, M. Xu, and R. Li, "Deep Learning for Household Load Forecasting-A Novel Pooling Deep RNN," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018, doi: 10.1109/tsg.2017.2686012.

## BIOGRAPHIES OF AUTHORS

**Alexander N. Ndife** is a Data Scientist and Doctorate degree holder in Smart Grid Technology from Naresuan University, Thailand. He obtained a bachelor's degree in Electrical/Electronic Engineering, Anambra State University, Nigeria in 2008 with specialty in telecommunications and subsequently Masters' degree in Electronics and Computer (Communications Engineering) from Nnamdi Azikiwe University, Nigeria in 2014. His research interests include Wireless Networks, Artificial Intelligent Systems, Deep Learning, Image Processing, and Cybersecurity & Smart Grid Networks. He is a registered engineer in Nigeria and CEO Ogugbalex Engineering limited. He has published numerous scientific papers and a member of various engineering organizations including Nigerian Society of Engineer (NSE), IEEE, IAENG and belongs to Society of Wireless Networks. He can be contacted at email: alexandern60@email.nu.ac.th.

**Wattanapong Rakwichian** graduated from Srinakharinwirot University, Phitsanulok, Thailand in Physics (B. Ed.). He obtained master's degree in physics from Chiang Mai University, Thailand, and Doctor of Philosophy (Ph.D) in Bioregulation Renewable Energy from Tokyo University of Agriculture, Tokyo, Japan. He is presently the Director of School of Renewable Energy and Smart Grid Technology (SGtech), Naresuan University, Thailand. He had served in various positions in recent times including Sub-committee of the Royal Thai Project on Renewable Energy, Consultant of Ministry of Energy, Ministry of Science Technology and Energy. He has written several textbooks on Physics, Mathematics, Solar Energy, and Digital Systems. He has also written many research articles for international journals and proceedings. He has also hosted and participated in many national and international conference related to science, solar energy, and other energy sources. He can be contacted at email: wattanapong.r@gmail.com.

**Paisarn Muneesawang** received a B.Eng. Degree in Electrical Engineering from the Mahanakorn University of Technology, Bangkok, Thailand, in 1996, M.Eng. Sci. Degree in Electrical Engineering from the University of New South Wales, Sydney, NSW, Australia, in 1999, and Ph.D Degree from the School of Electrical and Information Engineering, University of Sydney, Sydney. He was a Post-Doctoral Research Fellow with Ryerson University, Toronto, ON, Canada, from 2003 to 2004, and an Assistant Professor with the College of Information Technology, University of the United Arab Emirates, Al Ain, UAE, from 2005 to 2006. He has been a Visiting Professor with Nanyang Technological University, Singapore, since 2012, and with Ryerson University, Toronto, Canada, since 2013. He was the Vice President for Administrative Affairs, Naresuan University, Phitsanulok, Thailand, where he is currently a Professor and Dean of the Graduate School. He co-authored Multimedia Database Retrieval: A Human-Centered Approach (Springer, 2006) and Unsupervised Learning-A Dynamic Approach (Wiley-IEEE Press, 2013). He co-edited Advances in Multimedia Information Processing-PCM 2009 (Springer, 2009). His current research interests include multimedia signal processing, computer vision, and machine learning. He has served as the Registration Co-Chair of the International Conference on Multimedia and Expo 2006 and the Technical Program Co-Chair of the Pacific-Rim Conference on Multimedia 2009. He can be contacted at email: paisarnmu@nu.ac.th.

**Yodthong Mensin** obtained bachelor's degree (B .Sc) in Computer Science, and Master (M .Sc) in Information Technology from Naresuan University, Thailand. He obtained Doctor of Philosophy degree in Energy, communities, and the environment from Chiang Mai Rajabhat University. He also received the Cert. in Renewable Energy Technology from the University of Applied Science, Stralsund, Germany. He is presently a deputy Director of research and academics affairs in School of Renewable Energy and Smart Grid Technology (SGtech), Naresuan University, Thailand. He is a permanent speaker for IEEE conference in the section of Power and Energy Society (PES) in Thailand. He has experience in the field of a microgrid system, automated demand response (ADR), virtual power plant (VPP), energy management system, smart grid data utilization, and energy trading platform with blockchain technology for more than 15 years. Currently, he is the advisory's team member for the mega-project implementation of Thailand utilities, ministry of energy and private sector. He can be contacted at email: yodthongm@nu.ac.th.