Comparison of Deep Learning and Incremental Learning Model for Net Load Forecasting

Charnon Chupong, **Nitikorn Junhuathon**,
Sirichai Dangeam, and **Boonyang Plangklang**[†], Non-members

ABSTRACT

This paper presents hourly net load forecasting, which is the forecasting of the difference between the hourly power demand and the hourly power profile of the Photovoltaic (PV) system, which is the load that the utility should supply to the consumer. Three forecasting models are compared. The first model represents Long Short-Term Memory (LSTM), which is based on a deep learning model. The second model is the Fully Online Sequential Extreme Learning Machine (FOS-ELM), which is an incremental learning model that does not require initial training data. Online Sequential Extreme Learning Machine (OS-ELM) is the third model that can be learned incrementally like FOS-ELM. In addition, a method for the initial training of the OS-ELM model was proposed by taking the first sample from the studies to synthesize a sufficient number of samples for the initial training of the OS-ELM model. It was found from the experiment that in the case of fixed PV penetration rate, the LSTM model had slightly lower of error in forecasting than the other two models. In the case of increasing PV penetration rate, the FOS-ELM, and OS-ELM models had significantly lower errors in forecasting than the LSTM model. When comparing only the OS-ELM model using the proposed method with the FOS-ELM model, it was found that the OS-ELM model gave lower errors in forecasting than the FOS-ELM model because it was initially trained by the synthetic sample properly.

Keywords: Net Load Forecasting, Incremental Learning, Online Sequential Extreme Learning Machine, Data Augmentation

1. INTRODUCTION

In the past decade, the world has paid great attention to the climate change problem and one of the most popular solutions is to use electricity generated from photovoltaic (PV) systems. Global PV increased from 584 GW in the year 2019 to 843 GW in the year 2022 [1] or a 44% increase over three years. The increase in PV systems, especially those with high penetration rates, will cause instability problems in the power system [2]. The high penetration rate of PV results in changing of the load profile as shown in Fig.1. The changing of the load profile from the utility point of view causes errors in the conventional load forecasting method. Error in load forecasting can cause some mistakes in power system planning and operation that can affect power system reliability [3]. From Fig. 1, the PV power supplies the load for some periods while the rest of the load is supplied by the utility called net load profile or "net-load". To reduce the impact of PV penetration on the power system, one popular approach is PV power forecasting [4]. PV power forecasting is used in conjunction with load forecasting as net load forecasting. There are two main approaches for net load forecasting [5]: The first approach is separate forecasting, which uses 2 models separately for PV power forecast and load forecast and calculates the net load as different between load and PV power. The second approach is direct forecasting, which uses only 1 model to forecast net load directly. Subsequently, research has shown that direct net load forecasting is more accurate than separate forecasting [6].

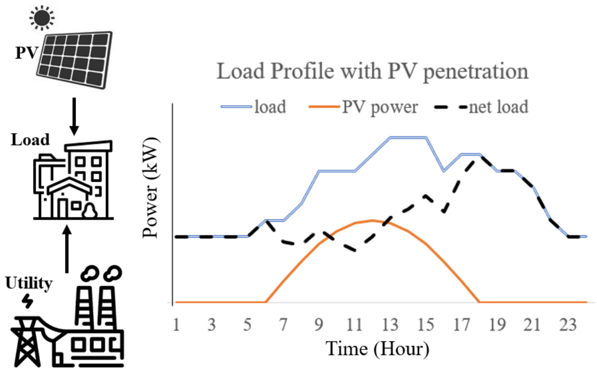

Fig. 1: *Load profile with PV penetration.*

There is some net load forecasting article presented, for example in [7], which compares net load forecasting with 7 machine learning models: Artificial Neural Network (RNN), Extreme Gradient Boosting (XGBoost), k-Nearest Neighbors (KNN), Random Forest (RF), Recurrent Neural Network (RNN), Support Vector Regression (SVR) and Naïve Persistence Models (NPM). The experimen-

Manuscript received on April 24, 2023; revised on August 15, 2023; accepted on August 19, 2023. This paper was recommended by Associate Editor Yuttana Kumsuwan.

The authors are with Department of Electrical Engineering, Faculty of Engineering, Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand.

[†]Corresponding author: boonyang.p@en.rmutt.ac.th

©2024 Author(s). This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License. To view a copy of this license visit: https://creativecommons.org/licenses/by-nc-nd/4.0/.

Digital Object Identifier: 10.37936/ecti-eec.2024221.252949

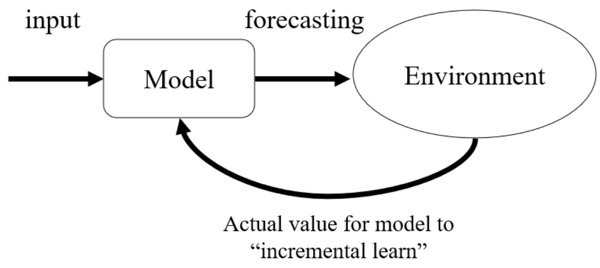

Fig. 2*: Concept of incremental learning model.*

tal results show that the RF model has the highest forecasting accuracy due to the ensemble structure and the second most accurate model is the RNN model, which works well with sequential data. There are also articles using deep learning models for load forecasting, including Long Short-Term Memory (LTSM) [8] and Gate Recurrent Unit (GRU) [9], which are models that can work with time series data better than ordinary machine learning models. All of the aforementioned models must use historical data to train the model. If there are changes in the electrical system, such as the increase of PV systems and electric vehicle chargers, which cause electricity consumption patterns to differ from the historical data used in model training, some errors will occur in forecasting. As in [10], the net load forecasting is presented by using the Adaptive Neuro Fuzzy Inference System (ANFIS) model and comparing the forecasting error when the PV system penetration ratio is increased. It was found that the error in forecasting is higher when the penetration rate is changed, but the article does not present some solution.

To address this problem, the researchers proposed an incremental learning model, a model that can learn from the newly received data and adjust itself during operation, as shown in Fig. 2. Examples of incremental models such as Incremental Support Vector Machine (ISVM) [11], Online Random Forecast (ORF) [12], Incremental Learning Vector Quantization (ILVQ) [13], Learn++ [14], Stochastic Gradient Descent (SGD) [15], and Online Sequential Extreme Learning Machine (OS-ELM) [16]. This research focuses on the OS-ELM model, because it has a low computational effort, then it can be used on a low-resource device such as microcontrollers in ordinary smart meters.

The OS-ELM model is a single hidden layer feedforward neural network that does not use iterative methods like gradient descent to adjust model parameters. This allows the OS-ELM model for fast learning, low computational cost, and high accuracy for uncomplicated tasks [17]. In reference [18], researchers use OS-ELM to forecasts PV power compared to ELM which uses empirical mode decomposition (EMD) acting as feature extraction. The result shows that the OS-ELM has higher accuracy than the ELM with EDM. In reference [19], the researchers present load forecasting by using k-MEAN clustering to group load profiles and use each OS-ELM to forecasting each group of load profiles. The result shows that the grouping of data before sending it to each OS-ELM to perform is better than using OS-ELM alone. However, the OS-ELM has one disadvantage, a sufficient amount of sample data is required for initial training [20], which makes the OS-ELM unusable in some tasks if historical electricity usage or PV power data is not collected. To address this problem, researchers have proposed a Fully Online Sequential Extreme Learning Machine (FOS-ELM) model that does not require an initial training sample [20]. But the FOS-ELM has a high error at the beginning period of forecasting because it uses an initial sample with zero value. This paper presents another approach to implementing the OS-ELM model without using an initial training sample inspired by the data augmentation method [21] and compared it with FOS-ELM and LSTM in net load forecasting tasks with varying penetration rates of PV.

The main contributions of this paper are: 1) presents a method to use the OS-ELM model without the initial training sample, 2) Comparison of net load forecasting (directed approach) accuracy between the proposed methods and the FOS-ELM and LSTM models in various PV penetration rate conditions.

2. RELATED WORKS

2.1 Incremental Learning Algorithms

The incremental learning algorithm is a method that allows a model to adjust parameters based on newly received sample data without discarding what has been learned from the past sample data. The following incremental learning algorithms are frequently discussed in numerous research studies:

- Online Random Forest (ORF) [12] is a Random Forecast algorithm when the input data differs from the previously learned past data. The 'trees' of the model will be increased to work with this data. Thus, ORF can work with changing data without forgetting the past data.
- Stochastic Gradient Descent (SGD) [15] is an algorithm in which the model parameters are changed iteratively to maximize an objective function. SGD is incremental learning because the model can adjust the parameters every time it receives new sample data to learn.
- Learn++ [14] utilizes the ensemble model approach like Ada-boost [22]. It is made up of sub-models developed through learning from past data. It is more likely that the past data where the model underperformed will be used to train the sub-model. As a result, it is claimed that the model can adapt to changing input data.
- Incremental Support Vector Machine (ISVM) [11] is like a Support Vector Machine (SVM), but some incoming data are recorded and called the Candidate Vector. Depending on the discrepancy between the newly received data and the current Support Vector, the Candidate Vector may be promoted to Support Vector.
- Incremental Learning Vector Quantization (ILVQ) [13]

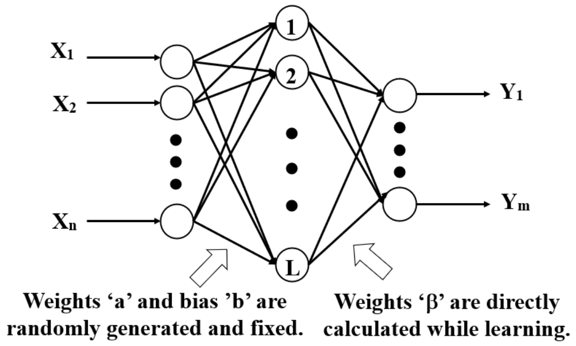

Fig. 3: *Structure of OS-ELM.*

is an incremental version of Learning Vector Quantization (LVQ) that applied the Prototyped-based learning concept [23]. A new prototype will be created if there are significant discrepancies between the received data and the previously learned data. This enables the model to operate on newly received data without losing track of learned data.

- Online Sequential Extreme Learning Machine (OS-ELM) [16] is an incremental version of Extreme Learning Machine (ELM) [24] It has a very high learning speed and a low cost of computing suitable for implementation in edge devices. The details of OS-LEM will be discussed in the next section.

2.2 Online Sequential Extreme Learning Machine (OS-ELM)

OS-ELM is an incremental learning algorithm, which was proposed by N.Y. Liang [16]. It is an incremental learning version of the Extreme Learning Machine [24]. It is applied to a single hidden layer feed-forward neural network as shown in Fig. 3.

The input layer weights and biases are randomly generated and don't change over time. The output weights are directly calculated not using an iterative approach as the gradient descent with 2 steps as follows:

1) Initial training phase: For a model with L nodes in a hidden layer and some N training samples (x_j, y_j), The connection between a and b is explained as follows:

$$\mathbf{y}_j = \sum_{i=1}^{L} \boldsymbol{\beta}_i g(\mathbf{a}_i \mathbf{x}_j + \mathbf{b}_i), \quad j = 1, 2, 3, ..., N \qquad (1)$$

where $g(...): \mathbb{R} \rightarrow \mathbb{R}$ is the hidden layer activation function $\boldsymbol{\beta}$ is the hidden layer weights $\mathbf{a}$ and $\mathbf{b}$ is the input layer weights and bias respectively. Eq. (1) can be written more simply as follows:

$$\mathbf{Y} = \mathbf{H}\boldsymbol{\beta} \qquad (2)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{g}(\mathbf{a_1 x_1 + b_1}) & \cdots & \mathbf{g}(\mathbf{a_L x_1 + b_L}) \\ \vdots & \ddots & \vdots \\ \mathbf{g}(\mathbf{a_1 x_N + b_1}) & \cdots & \mathbf{g}(\mathbf{a_L x_N + b_L}) \end{bmatrix}_{\mathbf{N \times L}}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_\mathbf{1}^\mathbf{T} \\ \vdots \\ \boldsymbol{\beta}_\mathbf{L}^\mathbf{T} \end{bmatrix}_{\mathbf{L \times m}} \text{ and } \mathbf{Y} = \begin{bmatrix} \mathbf{y}_\mathbf{1}^\mathbf{T} \\ \vdots \\ \mathbf{y}_\mathbf{N}^\mathbf{T} \end{bmatrix}_{\mathbf{N \times m}}$$

The purpose of the initial training phase is to determine the value of β by using Eq. (3), where the x and y values are known from the initial training sample.

$$\boldsymbol{\beta} = \mathbf{K^{-1} H^T Y} \text{ where } \mathbf{K} = \mathbf{H^T H} \qquad (3)$$

When the initial training samples are tiny in size or have low dimensions, the gradient descent approach will take longer to determine the value than the normal Eq. (4) [25]. Despite the input layer using random weights and bias values, numerous studies have shown that this learning algorithm can perform as well but faster than other learning algorithms [26-27].

When $\mathbf{H}^T\mathbf{H}$ is not invertible, this algorithm will encounter the numerical instability issue. There are at least two methods to solving this problem, the first is to invert the $\mathbf{H}^T\mathbf{H}$ using the Singular Value Decomposition (SVD) method [28]. The second method is adding a regularization factor λ to the $\mathbf{H}^T\mathbf{H}$ before inversion, as shown in Eq. (4) [29].

$$\mathbf{K} = \mathbf{H^T H} + \lambda \mathbf{I} \qquad (4)$$

where λ is a very small value called the regularization factor and $\mathbf{I}$ the identity matrix.

2) Incremental learning phase: the purpose of this phase is to adjust the $\boldsymbol{\beta}$ value according to the newly received sample without forgetting the past learned sample. The recursive concept is applied to Eqs. (3) and (4). In the initial phase, the OS-ELM calculates the $\boldsymbol{\beta}_0 = \mathbf{K_0^{-1} H_0^T Y_0}$ where $\mathbf{K}_0 = \mathbf{H_0^T H_0}$ which subscript 0 means round 0 of learning or the initial training. When new sample(s) data has arrived causing the value of the matrix $\mathbf{H}$ and $\mathbf{Y}$ change, the OS-ELM adjusts the $\boldsymbol{\beta}$ as follows:

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{K_{k+1}^{-1} H_{k+1}^T (Y_{k+1} - H_{k+1} \boldsymbol{\beta}_k)}, \qquad (5)$$

where $\mathbf{K}_{k+1} = \mathbf{K}_k + \mathbf{H_{k+1}^T H_{k+1}}$.

The updated $\boldsymbol{\beta}$ values can be calculated by Eq. (5), where it is observed that the updated $\boldsymbol{\beta}$ values are a function of the previous $\boldsymbol{\beta}$ values, which means the model can learn and adjust its parameters from a newly received sample without forgetting the past learned sample. The subscription term k means sample in k^{th} order and the subscription term $k+1$ means sample in $(k+1)^{th}$ order. Where $k = 0$ means the value from the initial training phase.

2.3 Fully Online Sequential Extreme Learning Machine (FOS-ELM)

The application of OS-ELM is constrained in some situations where initial sample data cannot be obtained, such as load forecasting in brand-new structures or regions that have never gathered electricity usage data. To solve this problem, FOS-ELM was introduced in reference [20] by setting $\boldsymbol{\beta}_0 = 0$ and $\mathbf{K}_0 = 0$, in other words, it is initial training by a sample that is equal to zero.

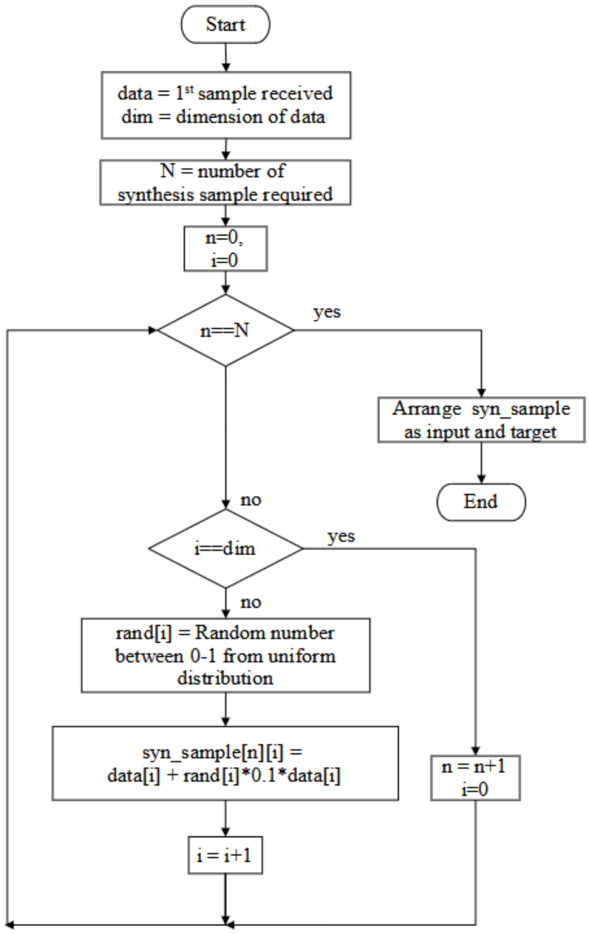

***Fig. 4**: Flowchart of the proposed method.*

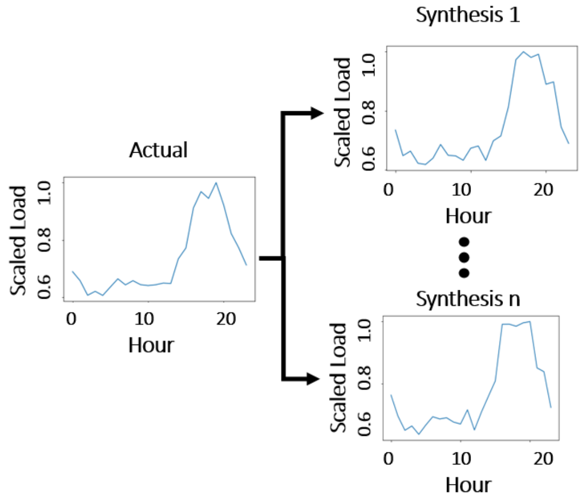

***Fig. 5**: Example of actual and synthetic samples.*

3. PROPOSED METHOD

Using OS-ELM without the initial training samples proposed here was inspired by data augmentation in the deep learning field [21]. This method starts by receiving the first sample from actual use (input and output to be predicted) and synthesizing additional samples by adding noise to received samples as shown in Fig. 4.
The flowchart in Fig.4 can be described as follow:

1) Receiving a sample from an actual working area and setting the dimensions of the desired synthetic samples. In this case, the dimension is set to 25 (24 input and one target value).

2) Setting the number of synthetic samples required for initial training of the OS-ELM model. In this case, 50 samples are required, which equals the number of nodes in the hidden layer of the OS-ELM model.

3) Randomizing a number between 0 and 1 using the uniform probability density function so that all numbers have an equal likelihood to occur.

4) Creating a new sample data by adding a random value obtained from step 3 to the sample data obtained from step 1. Before adding, such random values must be adjusted to not exceed 10% of the sample data. So that the new sample data is not too different from the original until the model predicts values incorrectly.

5) Repeating steps 3 and 4 until all dimensions of all synthetic samples are obtained.

6) Splitting the samples obtained from step 5 into input and target for initial training of the OS-ELM model.

Fig. 5 shows some parts of the synthesized sample, it is found that the synthesized samples are different from the real sample but have the same pattern.

4. METHODOLOGY

The experiment in this article compares the net load forecasting of the OS-ELM model which does not require data in the initial training method presented in the previous topic with the FOS-ELM, which does not require the initial training sample, and the LSTM model, a deep learning model without incremental learning. In this experiment, the dataset used is described in "Solar Generation and Demand Italy 2015-2016" [30], which presents the hourly electricity consumption and generation of PV systems in Italy in the period 2015-2016. The three models mentioned above were used to forecast the net hourly load for 2016. Data from 2015 was used to train the LSTM model, while OS-ELM and FOS-ELM were trained from 2016 data. In addition, this article also divides the experiment into 3 scenarios:

1) baseline penetration, in which the penetration rate is constant at about 20% based on the dataset,

2) low growth penetration, in which the penetration rate increases from the dataset by 5% every quarter from the 2^{nd} quarter, and

3) high growth penetration is the scenario where the penetration rate increases from the data set by 10% every quarter from the 2^{nd} quarter. The penetration rate is defined as in Eq. (6). In scenarios 2 and 3, we modified the PV power output in the 2016 dataset according to the penetration rate described above. The experimental methods and datasets used are summarized in Table 1.

***Table 1**: Dataset used in each scenario.*

Scenario /Model	LSTM	FOS-ELM	OS-ELM with the proposed method
Baseline penetration (Original dataset)	Train by the 2015 and test on the 2016 dataset	Test on the 2016 dataset	
Low growth penetration	Train by the 2015 and test on the 2016 dataset with some modification	Test on the 2016 dataset with some modification	
High growth penetration			

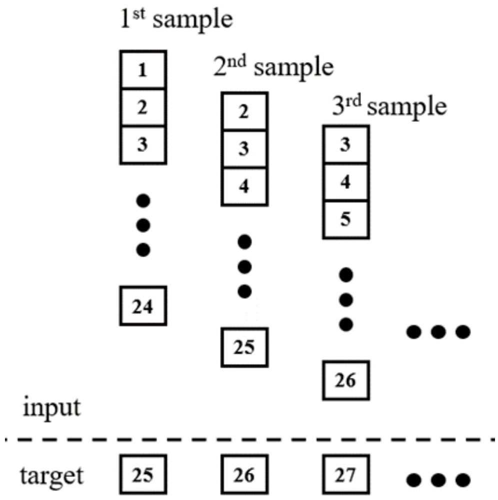

***Fig. 6**: Dataset provided as input and target samples.*

$$Penetration\ rate = \frac{Peak\ PV\ Power\ (kW)}{Peak\ Load\ (kW)} \quad (6)$$

To forecast the hourly net load, the hourly net loads of the last 24 hours were used as input. Therefore, the data in the dataset is provided as input and target samples for training the model. The target value in each sample will be the net load of 1 hour and the input value for each sample will be the hourly net load 24 hours before the target value as shown in Fig. 6.

The structure of the OS-ELM and FOS-ELM in this experiment are ensemble models consisting of sub-models with 24 nodes for the input layer and 1 node for the output layer. For the number of nodes in the hidden layer, 10, 20, 50, and 100 nodes were tested, and 10, 20, 50, and 100 sub-models were tested in the ensemble model. Therefore, a sub-model with 50 nodes in the hidden layer and 10 sub-models in the ensemble model was selected as Fig. 7, which is a compact structure with high accuracy. The OS-ELM and FOS-ELM models were implemented in Python language and run on the Spyder IDE. The LSTM model has been implemented in the *TensorFlow* platform by experimenting with the number of units 1, 5, 10, 50, and 100. The output from the LSTM passes through the 1 node in Dense Layer. The number of units was set at 50,

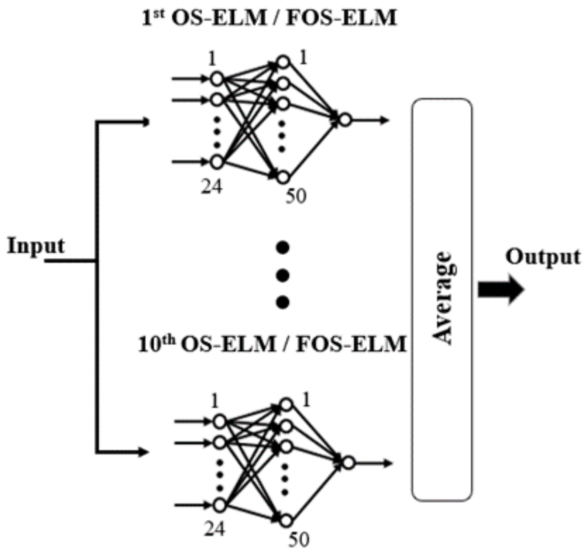

***Fig. 7**: Ensemble of OS-ELM/ FOS-ELM in this experiment.*

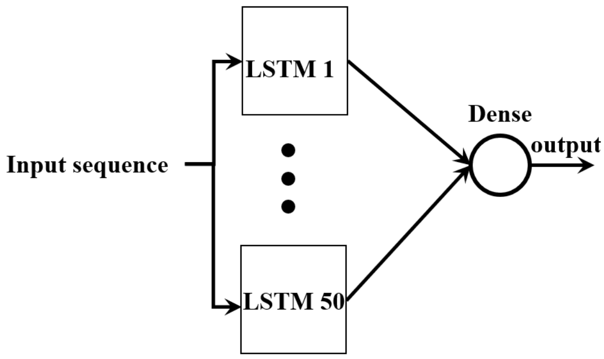

***Fig. 8**: LSTM in this experiment.*

which is a very accurate structure (see Fig. 8).

5. RESULT AND DISCUSSION

The experiments described in the previous section were performed using the Mean Absolute Percentage Error (MAPE) as the performance index of each model for net load forecasting. The MAPE can be calculated by Eq. (7). The MAPE was measured for the whole year and quarterly in the dataset to observe the trend of the model incrementally learning and adapting itself to lower MAPE values.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - F_i|}{A_i} \quad (7)$$

where A_i is the actual value, F_i the forecast value, and n the total amount of data.

Fig. 9 shows the MAPE values of the net load forecasting in a constant penetration rate scenario. It can be seen that the MAPE of the LSTM model is nearly equal across all quarters because the model was trained from the last whole year's samples. The FOS-ELM model has a high MAPE in the 1^{st} quarter because the model

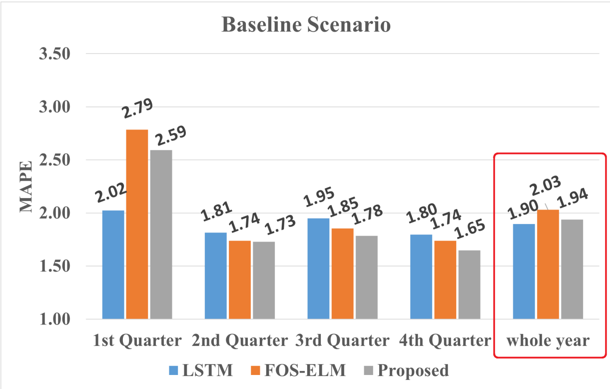

Fig. 9: *Result of baseline scenario.*

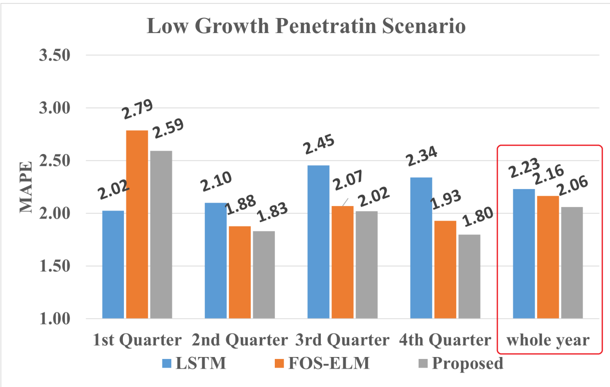

Fig. 10: *Result of low growth penetration scenario.*

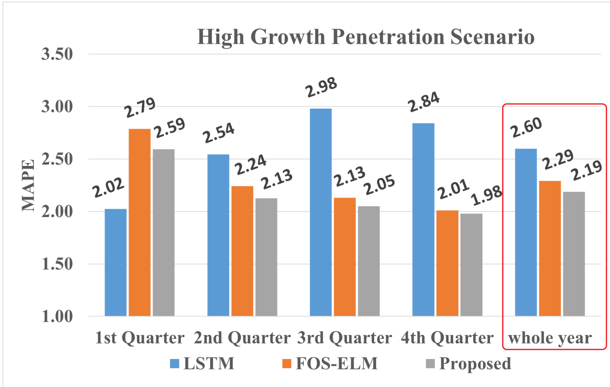

Fig. 11: *Result of high growth penetration scenario.*

works without samples for initial training causing overfit problem. In the 2^{nd} to 4^{th} quarter, the FOS-ELM model has a lower MAPE than the LSTM model due to the ability of incremental learning from the newly receive samples that are closer to the real situation than the training samples in the LSTM model. When considering the whole year, FOS-ELM still has a higher MAPE than LSTM due to the high MAPE in 1^{st} quarter. The OS-ELM model with the initial training method as proposed has MAPE similar to the FOS-ELM model i.e., high in the 1^{st} quarter and decreasing until lower than the LSTM model in the 2^{nd} to 4^{th} quarters, but MAPE for the whole year is still higher than the LSTM model. However, when comparing the FOS-ELM model and the OS-ELM model, it found that the MAPE value of the OS-ELM model is lower than the FOS-ELM model in every quarter. This is because the OS-ELM model used a sufficient number of synthetic samples for the initial training which can reduce the overfit problems.

Fig. 10 shows the MAPE values of the net load forecast in the low growth penetration scenario where the penetration rate increases by 5% every quarter (starting from the 2^{nd} quarter). It can be seen that the LSTM model has significantly increased the MAPE in the 3^{rd} to 4^{th} quarter when the penetration rate increases by 10-15%. In other words, If the data changes by 10% or more, the LSTM model is highly inaccurate. This is because the training samples are different from the actual data that the model has to forecast. The FOS-ELM model has a high MAPE in the 1^{st} quarter due to insufficient training samples. But in the 2^{nd} to 4^{th} quarters, the MAPE value will drop to significantly less than the LSTM model because the FOS-ELM model can adapt to situations where the environment changes. The OS-ELM model using the proposed initial training method has MAPE significantly lower than the LSTM model in the 2^{nd} to 4^{th} quarter, similar to the FOS-ELM model. By comparing to the FOS-ELM model, the OS-ELM model has lower MAPE values in every quarter because the initial training by enough synthetic samples allows the model to adapt faster and reduce overfit issues. In addition, from Fig. 10, it can be noticed that the MAPE of FOS-ELM and OS-ELM in the 3^{rd} quarter is higher than in the 2^{nd} quarter due to the summer season when PV systems have higher power output. This resulted in a greater change in net load, causing the model to have a high MAPE before adjusting to a lower MAPE in the following quarter.

Fig. 11 shows the MAPE values of the net load forecast in the high growth penetration scenario where the penetration rate increases by 10% every quarter (starting from the 2^{nd} quarter). It can be seen that the LSTM model significantly increases the MAPE value from the 2^{nd} to 4^{th}quarter, which is consistent with the results of the low growth penetration scenario. That is, if the data changes by 10% or more, the LSTM model will have a significantly high error in forecasting. As for the FOS-ELM and OS-ELM models, MAPE is higher in the 1^{st} quarter and decreases to lower than the LSTM model, similar to the above 2 scenarios. This is because both models have incremental learning ability even in situations where the environment highly changes from the previously learned samples. When compared with the low growth penetration scenario, it found that the MAPE of both models has higher than that of the high growth penetration scenario. This is because the higher penetration rate caused a large change in the net load then the forecasting error during the model incremental learn and adjust itself are also high. From the 3 scenarios mentioned above, it is found that the FOS-ELM and OS-

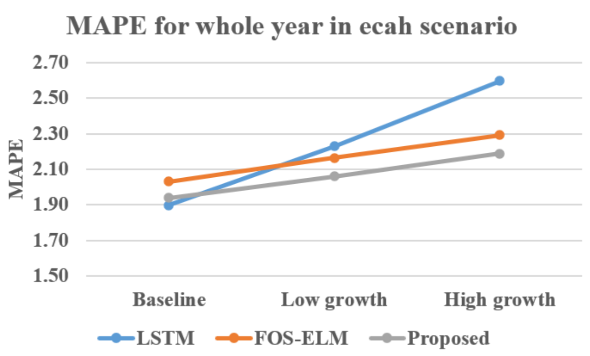

Fig. 12: *MAPE for the whole year in each scenario.*

ELM models have a high MAPE value at the beginning of working. To mitigate this problem, there is a paper that presents a method that can help the model achieve lower forecasting error at the beginning called the Re-Learning Method [31].

Fig.12 shows the annual MAPE trend for the net load forecasting of all 3 models in the 3 scenarios. It is found that the LSTM model has a significantly higher forecasting error if the penetration rate has changed. Because the model is trained by samples that are different from the data in the real situation. In the incremental learning models such as FOS-ELM and OS-ELM, the change in penetration rate has less effect on MAPE than on the LSTM model. Because the FOS-ELM and OS-ELM models can adjust themself according to environmental changes, in other words, the FOS-ELM and OS-ELM models are more robust than the LSTM models. When comparing the FOS-ELM model and the OS-ELM with the initial training method as proposed, it is found that the OS-ELM model always has a lower forecasting error because the OS-ELM model used a sufficient amount of synthetic samples based on the actual situation for initial training, so there is less overfit problem than the FOS-ELM model that starts from zero.

6. CONCLUSION

From the experiment of the net load forecasting of LSTM, FOS-ELM, and OS-ELM models with the proposed method, it was found that in the scenario of constant PV penetration, all three models gave a nearly level of error in the annual average. Although the FOS-ELM and OS-ELM models give a higher error in the beginning, they can incrementally learn and adjust themselves to reduce the error. In the scenario of an increase in the penetration rate of PV, all three models had an increase in forecasting error. But the FOS-LEM and OS-ELM models had a much lower annual average forecasting error than the LSTM model because both models have incremental learning capacity. When comparing only the OS-ELM that uses the proposed method and the FOS-ELM model, it was found that the proposed method which is the synthesis of the initial training sample helped the OS-ELM model has a lower error of forecasting than the FOS-ELM significantly. The proposed method allows the modeling for forecasting the net load forecasting with incremental learning capability, does not require the initial training samples, and low computational cost that is suitable for implementation on edge devices.

REFERENCES

[1] Statistical Review of World Energy-BP, Installed solar energy capacity, [Online]. Available: https://ourworldindata.org/grapher/installed-solar-pv-capacity, 2021.

[2] M. Islam, M. Nadarajah, and M. J. Hossain, "Short-Term Voltage Stability Enhancement in Residential Grid With High Penetration of Rooftop PV Units," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 4, pp. 2211–2222, Oct. 2019.

[3] N. Abdel-Karim, E. J. Nethercutt, J. N. Moura, T. Burgess, and T. C. Ly, "Effect of load forecasting uncertainties on the reliability of North American Bulk Power System," *2014 IEEE PES General Meeting | Conference & Exposition*, Jul. 2014

[4] K. J. Iheanetu, "Solar Photovoltaic Power Forecasting: A Review," *Sustainability*, vol. 14, no. 24, pp. 17005-17039, Dec. 2022.

[5] H. Shaker, H. Chitsaz, H. Zareipour, and D. Wood, "On comparison of two strategies in net demand forecasting using Wavelet Neural Network," *2014 North American Power Symposium (NAPS)*, Sep. 2014, pp. 1-6.

[6] G. Aburiyana, H. Aly and T. Little, "Net load forecasting model for a power system grid with wind and solar power penetration", in *2021 Global Conference on Engineering Research (GLOBCER'21)*, Turkey, 2021, pp.1-7.

[7] G. Tziolis et al., "Comparative Analysis of Machine Learning Models for Short-Term Net Load Forecasting in Renewable Integrated Microgrids," *2022 2nd International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED)*, Thessaloniki, Greece, 2022, pp. 1-5.

[8] S. Muzaffar and A.Afshari, "Short-Term Load Forecasts Using LSTM Networks," *Energy Procedia*, vol. 158, pp. 2922-2927, 2019.

[9] S. Kumar, L. Hussain, S. Banarjee and M. Reza, "Energy Load Forecasting using Deep Learning Approach-LSTM and GRU in Spark Cluster," in *Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, Kolkata, India, 2018, pp. 1-4.

[10] G. Aburiyana, H. Aly and T. Little, "Investigating the Impact of Increasing Renewable Energy Penetration Levels on the Accuracy of Net Load Forecasting," in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Halifax, NS, Canada, 2022, pp. 66-71.

[11] B. Zhang, J. Su, and X. Xu, "A Class-Incremental Learning Method for Multi-Class Support Vector Machines in Text Classification," *2006 International*

Conference on Machine Learning and Cybernetics, 2006, pp. 2581-2585.

[12] J. Li, X. Shao and H. Zhao, "An Online Method Based on Random Forest for Air Pollutant Concentration Forecasting," in *2018 37th Chinese Control Conference (CCC*), Wuhan, China, 2018, pp. 9641-9648.

[13] Jung-Hua Wang and Wei-Der Sun, "Online learning vector quantization: a harmonic competition approach based on conservation network," *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics)*, vol. 29, no. 5, pp. 642-653, Oct. 1999.

[14] R. Polikar, L. Udpa, S. S. Udpa, and V. Honavar, "LEARN++: an incremental learning algorithm for multilayer perceptron networks," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, Istanbul, Turkey, 2000, pp. 3414-3417.

[15] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the Twenty-First International Conference on Machine Learning*, Banff, Alberta, Canada, ACM, 2004, pp.116-123.

[16] N. Y. Liang, G. B. Huang, P. Saratchandran, and N. Sundarrajan, "A fast and accurate online sequential learning algorithm for feedforward networks." *IEEE Transaction on Neural Networks,* vol.17, no.6, pp.1411-1423, Nov. 2006.

[17] S. Zhang, W. Tan, and Y. Li, "A Survey of Online Sequential Extreme Learning Machine," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT),* Thessaloniki, Greece, 2018, pp. 45-50.

[18] M. Parida, M. K. Behera and N. Nayak, "Combined EMD-ELM and OS-ELM techniques based on feedforward networks for PV power forecasting," in *2018 Technologies for Smart-City Energy Security and Power (ICSESP),* Bhubaneswar, India, 2018, pp. 1-6.

[19] Y. Li, P. Guo, and X. Li, "Short-Term Load Forecasting Based on the Analysis of User Electricity Behavior," *Algorithms*, vol. 9, no. 4, p. 80, Nov. 2016.

[20] P. K. Wong, C. M. Vong, X. H. Gao, and K. I. Wong, "Adaptive Control Using Fully Online Sequential-Extreme Learning Machine and a Case Study on Engine Air-Fuel Ratio Regulation," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–11, 2014.

[21] M. Abufadda and K. Mansour, "A Survey of Synthetic Data Generation for Machine Learning," in *22nd International Arab Conference on Information Technology (ACIT),* Muscat, Oman, 2021, pp. 1-7.

[22] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[23] M. Schwabacher, H. Hirsh, and T. Ellman, "Learning prototype-selection rules for case-based iterative design," in *Proceedings of the Tenth Conference on Artificial Intelligence for Applications*, 1994, pp. 56-62.

[24] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541),* Budapest, vol.2, 2004, pp.985-990.

[25] Andrew ng, *Machine learning specialization*, Coursera, [Online] Available: https://coursera.org /specializations/machine-learning-introduction.

[26] X. Liu, S. Lin, J. Fang, and Z. Xu, "Is extreme learning machine feasible? a theoretical assessment (part I)," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 1, pp. 7-20, 2015.

[27] S. Lin, X. Liu, J. Fang, and Z. Xu, "Is extreme learning machine feasible? A theoretical assessment (Part II)", *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 1, pp. 21-34, 2015.

[28] Y. Jaradat, M. Masoud, I. Jannoud, A. Manasrah and M. Alia, "A Tutorial on Singular Value Decomposition with Applications on Image Compression and Dimensionality Reduction," in *2021 International Conference on Information Technology (ICIT),* Amman, Jordan, pp. 769-772, 2021.

[29] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *2009 IEEE Symposium on Computational Intelligence and Data Mining*, Nashville, TN, USA, 2009, pp. 389-395.

[30] A. Cedola, "Solar generation and power demand in Italy," Kaggle, [Online]. Available: http://www.kaggle.com/code/arielcedola/solar-generation-and-power-demand-in-italy.

[31] C. Chupong, N. Junhuathon, and B. Plangklang, "Short-Term Load Forecasting by FOS-ELM with Re-Learning Method," *2022 International Conference on Power, Energy and Innovations (ICPEI)* Pattaya Chonburi, Thailand, 2022, pp. 1-4.

Charnon Chupong received his B.Eng. degree in Electrical Engineering from King Mongkut's University of Technology North Bangkok in 2001, M.Eng. degree in Electrical Engineering from Rajamangala University of Technology Thanyaburi in 2012, He is currently working as an Assistance Professor at the Department of Electrical Engineering, Faculty of Engineering, Rajamangala University of Technology Thanyaburi (RMUTT). His research interest is on machine learning and its application in power systems and related fields.

Nitikorn Junhuathon received his B. Eng., M. Eng., and Ph.D. In Electrical Engineering from Suranaree University of Technology (SUT), Thailand, in 2017, 2019 and 2023, respectively. He is currently a lecturer at Department of Electrical Engineering, Faculty of Engineering, Raja Mangala University of Technology Thanyaburi (RMUTT), Thailand. His research interest is on machine learning and application, optimization technique, power forecasting, energy, power system and related fields.

Sirichai Dangeam received his B.Eng. degree in Electrical Engineering from Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand, in 1995, M.Eng. degree in Electrical Engineering from King Mongkut's University of Technology North Bangkok, Thailand, in 2003, and D.Eng. degree in Electrical Engineering at the King Mongkut's Institute of Technology Ladkrabang, Thailand in 2020. He is currently working as an Associate Professor at the Department of Electrical Engineering, Faculty of Engineering, Rajamangala University of Technology Thanyaburi (RMUTT).

Boonyang Plangklang received his B.Eng. degree in Electrical Engineering from Rajamangala Institute of Technology, Thailand, in 1996. He received a diploma in Instrumentation at Northern Alberta Institute of Technology (NAIT), Edmonton, Alberta, Canada, in 1997. He graduated Master of Science in Electronics System and Engineering Management at University of Paderborn, division Soest, Germany, with the cooperation of Bolton Institute of Higher Education, UK, by the DAAD scholarship in 2001. He received the degree of Dr.–Ing. in Electrical Engineering from Kassel University, Germany in 2005. He is now working as an Associate Professor at the Department of Electrical Engineering, Rajamangala University of Technology Thanyaburi (RMUTT).