IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 September 2022, accepted 30 September 2022, date of publication 4 October 2022, date of current version 12 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211941

RESEARCH ARTICLE

Short-Term Electricity Load Forecasting Based on Temporal Fusion Transformer Model

PHAM CANH HUY[1], NGUYEN QUOC MINH[2], (Member, IEEE), NGUYEN DANG TIEN[2], AND TAO THI QUYNH ANH[1]

[1]School of Economic and Management, Hanoi University of Science and Technology, Hanoi, Vietnam
[2]School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

Corresponding author: Nguyen Quoc Minh (minh.nguyenquoc@hust.edu.vn)

This work was supported by the Hanoi University of Science and Technology (HUST) under Project T2021-SAHEP-006.

ABSTRACT Electricity load forecasting plays an important role in the operation of power systems. Inaccurate forecast would reduce the safety of power supply and affect the economic and social activities as well as national defense and security. In addition, the forecast results also support decision-making on electricity generation and market transactions. Traditional methods such as AR, ARIMA, SARIMA have been widely used to forecast short term electricity load. Recently, load forecasting based on artificial and deep neural networks have shown significant accuracy improvement over traditional statistical models. In this research, a novel recurrent neural network named temporal fusion transformer (TFT) is used to forecast short-term electricity load of Hanoi city. The TFT is a newly developed model and it combines the advantages of several other RNN models such as LSTM and the self-attention mechanism. In addition to historical load data, we use temperature and humidity features, and time features such as calendar month, lunar month, days of the week, hours of the day and holidays. The forecast results of TFT are compared with traditional statistical models as well as well-known RNN models. The compared results show that the proposed method is better than other methods in both MAE and MAPE criteria.

INDEX TERMS Power systems, load forecasting, artificial intelligence, recurrent neural network, temporal fusion transformer.

I. INTRODUCTION

Short-term load forecasting (STLF) plays a very important role in power systems operating, controlling, planning and trading in the electricity market. Many methods have been developed to forecast electrical load, such as regression, method Box Jenkins, exponential smoothing and more recently using artificial neural network (ANN) techniques.

In [1], the author proposed an autoregressive moving average model with an exogenous weather variable (ARMAX) to forecast short-term electrical loads (next 24 hours) using cargo results show that the mean absolute percentage error (MAPE) ranges from 3.01 to 4.54%. The comparative study of electricity demand forecast of Basra, Iraq using Box-Jenkins method and artificial neural network (ANN) was reported [2]. The obtained results have shown that using ANN improves the accuracy of the forecast. The ARIMA model is used to forecast short-term electrical loads for the state of Karnataka, India [3]. The author proposed to use the ARIMA (1,1,1) model forecast the hourly load of the next month. This model obtained the MAPE of 4.46%. The combination of seasonal ARIMA and regression support vector (SVR) models is used to forecast short-term electricity load [4]. The concept of this combined model is to use SARIMA to control the linear part of the sequence and then correct the deviation using the SVR. The results show that the combined SARIMA and SVR model perform much better than the single SARIMA model with the prediction error of 4.757%. The artificial neural network is used to forecast Thailand's short term electricity load [5]. The model also takes into account the impact of weekends, holidays, other special days, and temperature on electricity demand. Based on the

The associate editor coordinating the review of this manuscript and approving it for publication was Nagesh Prabhu.

characteristics of demand, the overall data set is divided into four different subgroups and models are developed for each subgroup. The prediction error of the aggregated model ranges from 2.18% to 4.37%, depending on each day of the normal week. From previous research [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], many factors affect short-term electricity load forecast such as trend impact, seasonal effect, special impact, weather influence, random effects, etc. Furthermore, sudden changes in system demand or system failures represent another form of uncertainty regarding the load forecasting process. All above adds to the complexity of the short-team load forecasting. STLF gains a lot of attention by the scientists and power distribution companies. Recently, with the significant development of artificial intelligence and deep learning, methods using artificial neural network (ANN) have shown numerous advantages. It overcomes the limitations of previous statistical-based methods.

In this research, we propose a novel ANN model called Temporal Fusion Transformer (TFT) to forecast the short-term electricity load of Hanoi city. The proposed model takes into consideration not only weather features such as temperature, absolute humidity, wind speed, but also time features such as special holidays of the year. The results show that the TFT model can forecast the short-term load power with better accuracy than existing models.

II. DEEP LEARNING MODELS

A. ARTIFICIAL NEURAL NETWORK

Artificial Neural Network (ANN) is an information processing model, inspired by the simplified human brain's neuron systems. An ANN consists of large numbers of linkage nodes to process information in a clear relationship. It has the ability to learn by experience from training, save experiences into knowledge and apply in new data in the future. The neurons are the basic elements and engage in information processing in the neural network [16]. The neurons link to each other by weight, bias and activation function, in order to process the information. In general, an ANN includes three layers: the input layer, hidden layers and output layer. An ANN which consists of more than two hidden layer is call deep neural network or DNN.

The working principle of the ANN is illustrated in Figure 1 [7]. The value in each neuron is calculated from weighted sum of neurons from previous layer plus the bias by activation functions. This process is based on the activity of neurons and synapses in the human brain. Similar to the human brain, neural network, series of algorithms are used in ANN to identify and recognize relationships in data sets.

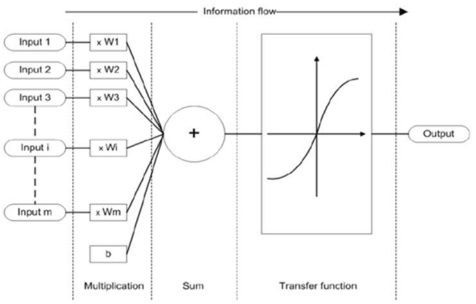

FIGURE 1. Working principle of the ANN.

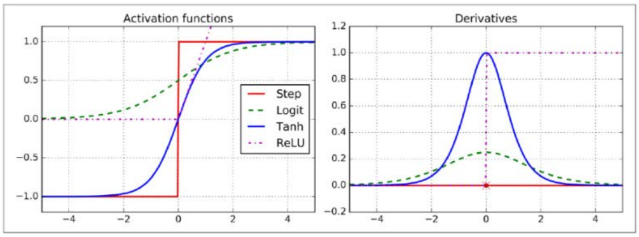

FIGURE 2. Commonly activation functions and their derivatives [7].

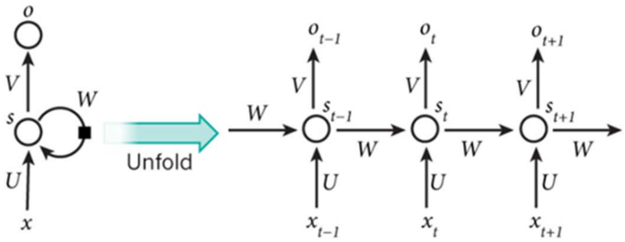

FIGURE 3. Typical RNN network structure.

The neural network training process needs to determine two components: the network architecture and the set of weight and bias values. Determining the network architecture often uses expert (predetermined) intelligence. Meanwhile, valuation of weight people used the back-propagation algorithm. Some commonly used activation functions are the step function, the logit function (or sigmoid function), the tanh function, and the Rectified Linear Unit (ReLU) function. The graphs of these activation functions and their derivatives are shown in Figure 2 [17].

B. REGRESSIVE NEURAL NETWORK RNN

The use of sequential information is the idea for the research and development of Recurrent Neural Network (RNN). RNNs are called regressive (or feedback) because they perform the same task for each element of the sequence, whose output depends on previous computation results. A typical network RNN is structured as shown in Figure 3 [8].

The variables in the RNN are defined as:

- x_t is the input at time t.
- s_t is the hidden state at time t. It is similar to network memory, s_t is calculated based on the previous hidden state and the input of the current step: $s_t = f(Ux_t + Ws_{t-1})$. The function f is usually a nonlinear function such as a tanh function or a ReLU function, st-1 is usually initialized to 0 when computing the first hidden state.
- o_t is the output at step t.

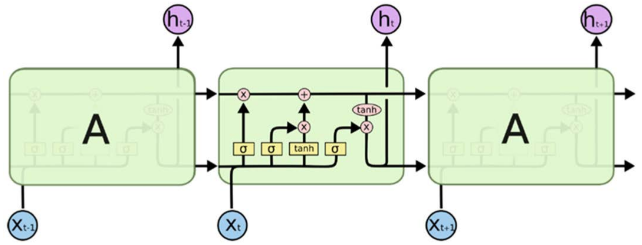

FIGURE 4. **LSTM model.**

C. LONG-SHORT TERM MEMORY NEURAL NETWORK

In general, the RNN model works well with sequential data. However, recent studies show that the RNN model might be convergence due to vanishing and exploding gradient. This limitation makes RNN ineffective for time-based data processing problems that require intermediate memory states. The Long-Short Term Memory (LSTM) model is developed to solve the limitation of RNN by introducing into the network a memory unit called memory unit or cell.

A LSTM network model is illustrated in Fig. 4. The input consists of three components: X_t is the input at the current step; h_{t-1} is the output from a previous LSTM block and C_{t-1} is the "memory" of the previous block, and this is also the most important point of the LSTM. Its output consists of h_t being the result of the current LSTM block and C_t being its memory. Thus, a single block LSTM makes a decision based on the current input, result and memory of the previous block and it generates a new output as well as its memory. LSTM a network model is illustrated in Figure 4 [19].

The LSTM block first decides what information to remove from the cell state. This decision process is performed by a sigmoid layer called the "forget gate layer". The dropout gate takes as input h_{t-1} and x_t and outputs a value between [0, 1] for cell state C_{t-1}. If the output is 1, it represents "retained information", and 0 represents "rejected information". The forget gate f_t of an LTSM cell is calculated as:

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

where:

σ: sigmoid function

W_f: the matrix of weight and bias parameters

h_t: output vector of LSTM unit

x_t: input vector of LSTM unit.

The LSTM then decides how new information should be stored in the cell state. It is composed of two parts, one is the layer sigmoid called "input gate layer" (input layer) determine the value will be updated, and a layer of satin creates a vector of new values, which can add to the cell state.

$$\begin{aligned} i_t &= \sigma\left(W_i.[h_{t-1}, x_t] + b_i\right) \\ \widetilde{C}_t &= tanh\left(W_C.[h_{t-1}, x_t] + b_C\right) \end{aligned} \tag{2}$$

where:

i_t: input gate's activation vector

$\widetilde{C}_t$: cell state.

Next, the old cell state C_{t-1} is updated at the new cell state C_t according to the formula:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{3}$$

The old memory state Ct-1 is multiplied by the value of the forget gate ft, which decide what to forget in the previous state. The value $i_t^*\widetilde{C}_t$ represents the new candidate value for the cell state determined by the specific expansion coefficient it for updating the value for each cell state.

As a final step, the LSTM block decides its output based on the cell state. Sigmoid layer is used to calculate the exported components. The cell state value is then fed into the tanh function (the result will be in the range [−1,1]) and multiplied by the output of the sigmoid gate in order to decide what should be output by the LSTM block. The calculation formula for the components of this step is as follows:

$$\begin{aligned} o_t &= \sigma\left(W_o.[h_{t-1}, x_t] + b_o\right) \\ h_t &= o_t * tanh\left(C_t\right) \end{aligned} \tag{4}$$

where:

o_t: output gate's activation vector.

An LSTM network is a combination of LSTM blocks connected in series over time. The operation of each LSTM block at a time is taken care of by the gates: the memory gate ft, the input gate it and the output gate o_t, in which the forget gate is the remarkable part of LSTM, provides the ability to use computational information from earlier time moments.

III. PROPOSED FORECAST MODEL BASED ON TEMPORAL FUSION TRANSFORMER

In this study, we propose a modern deep neural network model. This model is developed by researchers in time series forecasting at Google, named "Temporal Fusion Transformer - TFT" [20]. This is a deep neural network model architecture that is based on a focused attention mechanism for prediction [21]. This architecture, called Transformer, parallelizes by learning the feedback sequence with a centralized mechanism, and encodes the position of each element in the sequence. As a result, we have a compatible model with a significantly shorter training time.

For time series data, there are several factors which cause the data variation such as trend, seasonality, periodicity, and randomness. Trend shows the increase /decrease of the data for a long period of time. Seasonality, on the other hand, expresses the increase/decrease at a certain time (month/quarter) repeated over many years. Periodicity is defined as the variation of the data repeated with a time period greater than 1 year. Randomness is referred to irregular variation and cannot be predicted. Those above four components can be combined into the model as [24]:

$$y_t = T_t + S_t + C_t + I_t \tag{5}$$

where:

y_t is the observed value at time t.

T_t is the trend component at time t.

S_t is the seasonal component at time t.
C_t is the period component at time t.
I_t: random component at time t.

For time series data forecasting problems, sometimes the trend factor can lead to incorrect linear correlation. Machine learning and deep learning models are sometimes difficult to learn the trend, seasonality, periodicity and randomness at the same time. Therefore, the model is divided into two sub models with two different tasks: trend forecasting and data forecasting when trend is removed.

Fig. 5 illustrates the electricity load of Hanoi city from 2014 to 2015 with trend (blue curve) and without trend (green curve). The peak load of Hanoi in 2020 is 4500MW. The peak load is typically obtained in July due to its highest temperature of the year. According to the curve, the load pattern is periodical with linearly increase trend. The method used to remove the trend in the data is the differencing transform method. The load data when removing the trend at time t will be calculated by subtracting the load data at time t from the previous load value at time t-1.

$$x'_t = x_t - x_{t-1} \tag{6}$$

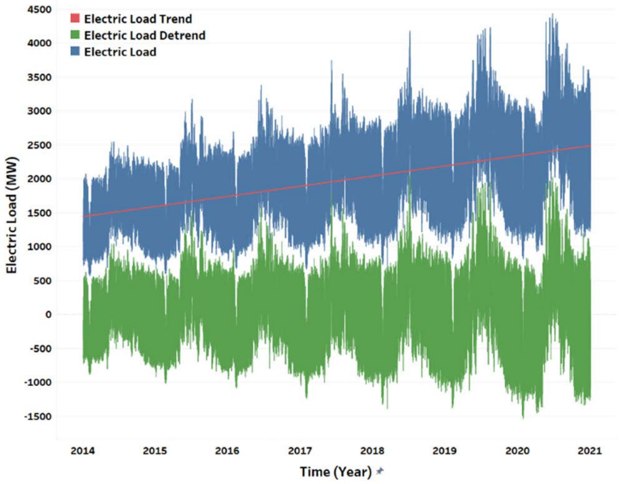

FIGURE 5. Electricity load with and without trend.

There are differences between training and forecasting phase. The training process is illustrated in Fig. 6. At the training phase, the data is prepared and trained at two different models: linear regression and temporal fusion transformer. The trend prediction model is used to calculate the data when removing the trend and then put it into the seasonal forecasting model.

The forecasting process is illustrated in Fig. 7. At the forecasting phase, the trend of 192-hour load data will be forecasted by a time-based linear regression model which includes 168-hour in the past and 24-hour in the future. After that, the 168-hour load data in the pass will be detrended by linear regression model, and then combined with the weather and time characteristics as input for the TFT model. The final forecast results is then combined by two forecasting outputs: the detrend load in the next 24-hour from the TFT model and the 24-hour trend of the load from linear regression model.

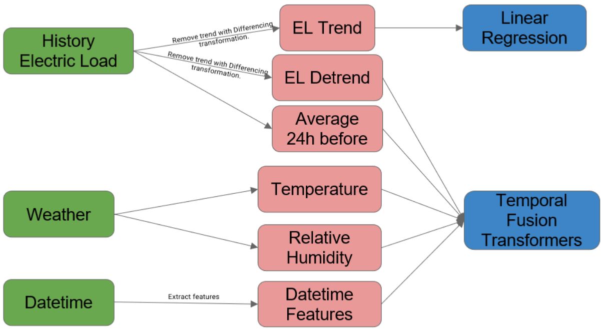

FIGURE 6. Training phase model.

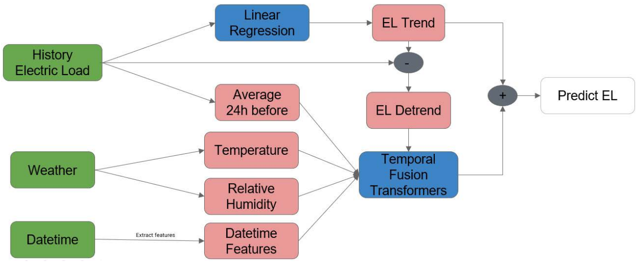

FIGURE 7. Forecasting phase model.

The load trend, after being separated from the original data, is a straight line that increases linearly over time. Therefore, the forecast model used to predict the load trend is the linear regression algorithm.

The linear regression (LR) model can be represented as:

$$\begin{aligned} y &\approx f(\boldsymbol{x}) \\ &\Leftrightarrow y \approx w_0 + x_1w_1 + x_2w_2 + \cdots + x_nw_n \\ &\Leftrightarrow y \approx \bar{\boldsymbol{x}}\boldsymbol{w} = \hat{y} \end{aligned} \tag{7}$$

where: y is the label of the data

$\bar{x} = [1, x_1, x_2, \ldots, x_n]$ are the input data vector.

$w = [w_0, w_1, \ldots, w_n]^T$ are the parameters learned during training.

$\hat{y}$ is the predicted value of the model.

The prediction error is defined as the square of the difference between the predicted value from the model and the label of the data. This is also considered as the objective function model.

$$L(w) = \frac{1}{2}\sum\nolimits_{i=1}^{N} (y_i - \bar{\boldsymbol{x}}\boldsymbol{w})^2 \tag{8}$$

The value of the weight vector w can be found by optimizing this objective function.

After removing the trend part, the formula in (5) becomes:

$$y_t - T_t \quad = S_t + C_t + I_t \tag{9}$$

Then the task of the TFT model is only to forecast seasonal, period, and random. The TFT model is illustrated in Fig. 8.

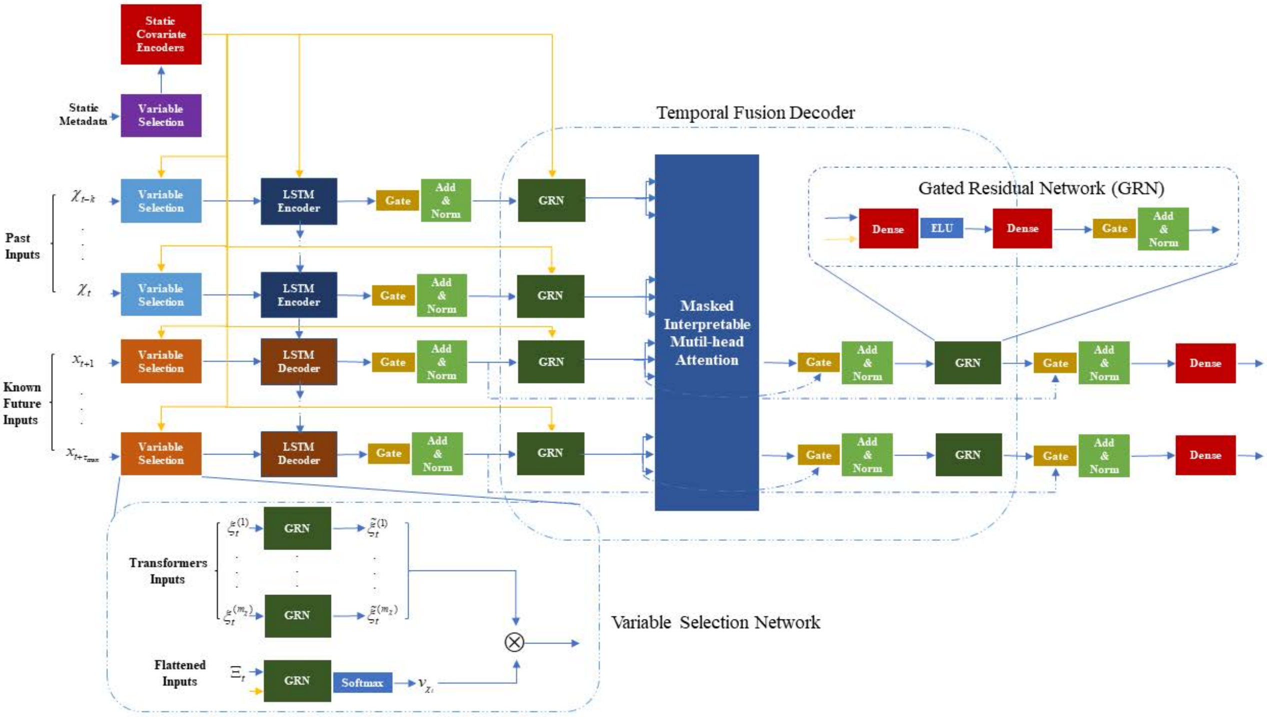

FIGURE 8. The temporal fusion transformer model.

In this model, known inputs are efficiently observed and learned by specifically designed components. The model uses a gating mechanism that allows unnecessary parts of the network to be omitted to provide complexity and depth of network architecture suitable for different data sets and scenarios. This mechanism is the "Gated Residual Network" (GRN) in TFT's architecture. Variable selection networks in TFT can choose which input variables are appropriate at each time step. TFT also uses a static input encoder to allow the model to learn the relationship between the static inputs in both the past and the future. In addition, to process both long-term and short-term temporal relationships from known and observed material inputs, the TFT model uses a sequence-to-sequence layer for local processing, while long-term dependencies are learned using an interpretable multi-head attention block.

TFT models have shown to be more advance than traditional RNN models in term of time series data processing and predicting. In addition, the model can learn the relationship and the importance of the input features. Therefore, these features are no longer considered "black box", compared to other deep neural networks [20].

The process consists of the following main steps:

- Step one: Perform data collection including load data, weather data and other related data.

- Step two: Perform data processing, feature extraction and data transformation.

- Step three: Build and train the model.

- Step four: Analyze and evaluate the newly trained model. if the model does not meet the accuracy, it will return to step two or three depending on the evaluation analysis. Continue training until the requirements are met.

IV. RESULTS

Electrical load data of Hanoi city is collected in 7 years from 2014 to the end of 2020 and provided by Hanoi load dispatch center (HLDC). The weather data is supported by Visual Crossing Corporation, which provide measured data from hydro-meteorological monitoring stations located in Hoan Kiem District, Hanoi City.

After collection, the raw data is processed and normalized before feature extraction. In the time series data such as electricity load, date and time contain important information commonly used in machine learning models. In previous forecast models, the historical load data is often used as the key feature. In this research, beside the historical load data, we also use several other features to improve the accuracy of the model. These features are weather, time of the year and holiday. In Vietnam and some other Asia countries, lunar holidays such as lunar new year, mid-autumn day are traditional events and electricity load pattern is very different in such events. Therefore, we propose to use those below features to improve the accuracy of the load forecasting:

- Calendar month.

- Lunar month.

- Day of the week.
- Hour of the day.
- Public holidays of the year.

Since time features are non-numerical data, it is necessary to convert this data to the numeric one. A simple technique for doing this is one-hot encryption. In this encoding, a ''dictionary'' needs to be constructed containing all possible values of each item data. Then each item value will be encoded with a binary vector with all elements equal to 0 minus a partial word equal to 1 corresponding to the position of that item value in the dictionary.

In this research, not only the instant load value but also average value of the load data is used as the input features for the forecasting model. Deep learning methods can be categorized into iterated approaches using autoregressive models. Therefore, this feature is considered as a moving average that smooths the data in a time frame, tracks the short-term fluctuations and identifies the long-term trend of the time series data. The value at time t of the series is determined by the average of the series over a time frame of 24 hours.

$$x(t) = \frac{\sum_{j=t-24}^{t} x(j)}{24} \tag{10}$$

In this study, we use an input feature set consisting of electric load, temperature, relative humidity, the average electric load of 24 hours before, calendar month, lunar month, day of the week, the hour of the day, and public holidays for the proposed method.

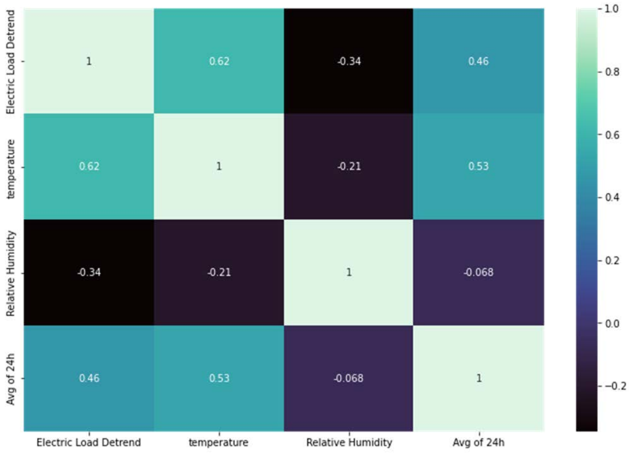

FIGURE 9. **Correlation matrix between continuous features.**

In the correlation matrix shown in Fig. 9, the temperature and average load of the previous 24 hours are directly proportional to the detrend electrical load with correlation coefficients of 0.62 and 0.46, respectively. In contrast, relative humidity is inversely proportional to the detrend electrical load with a correlation coefficient of −0.34.

The data set after normalization is divided into 2 main sets, the training data set and the test data set, these data sets are isolated from each other during training and testing. model evaluation. The initial dataset consists of 7 years from 2014 to the end of 2020. We divide the training set with 6 years of data from 2014 to the end of 2019 and the 2020 data is used for the test set.

In this study, we use two common measures in time series forecasting problems: mean absolute error and mean absolute percentage error.

Mean absolute error (MSE) is defined as:

$$MAE = \frac{1}{N}\sum\nolimits_{i=1}^{N} \left|y_i - \hat{y}_i\right| \tag{11}$$

where: N is the total number of forecast data points.

$y_i, \hat{y}_i$ are the ith label value and the ith forecast value of the model, respectively.

Mean Absolute Percentage Error (MAPE) is defined as:

$$MAPE = \frac{1}{N}\sum\nolimits_{i=1}^{N} \frac{\left|y_i - \hat{y}_i\right|}{y_i} \times 100 \tag{12}$$

where: N is the total number of forecast data points.

$y_i, \hat{y}_i$ are the ith label value and the ith forecast value of the model, respectively.

The training has been evaluated and tested. The number of hidden layers is 60, the dropout rate is 0.25, the concentration factor is 3 and the learning rate is 0.003. After training the models and evaluating the accuracy, we obtain the accuracy of the proposed method with mean absolute error (MAE) of 77.51 MW and mean absolute percentage error (MAPE) is 3.38%.

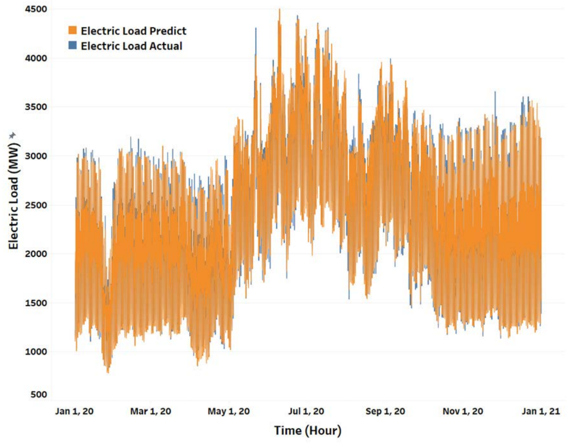

FIGURE 10. **Electricity load prediction in the testing dataset.**

The STLF and MAPE results are illustrated in Fig. 10 and 11, respectively. In Fig. 10, the predicted load (orange curve) is compared with the actual load (blue curve) in MW. This comparison is performed in the testing dataset of one year from Jan 1st 2020 to Jan 1st 2021. Fig. 11 shows the MAPE of the proposed model. The average MAPE in 2020 is 3.38%. According to the result, the MAPE The proposed model provides good forecast results, the predicted value is close to the actual value.

Fig. 12 shows the 24-hour forecast result in the day with maximum load power in 2020 and Fig. 13 shows the MAPE

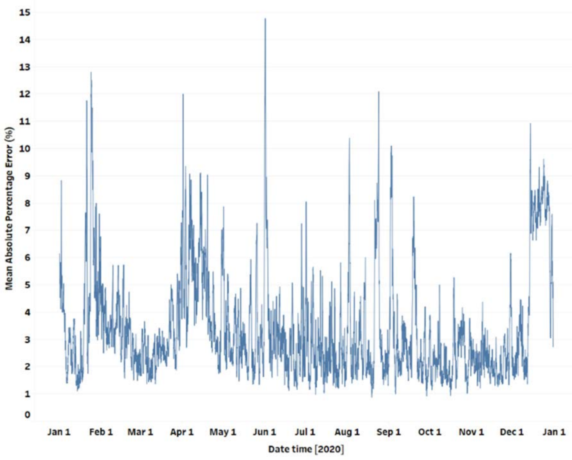

FIGURE 11. Mean absolute percentage error in the testing dataset.

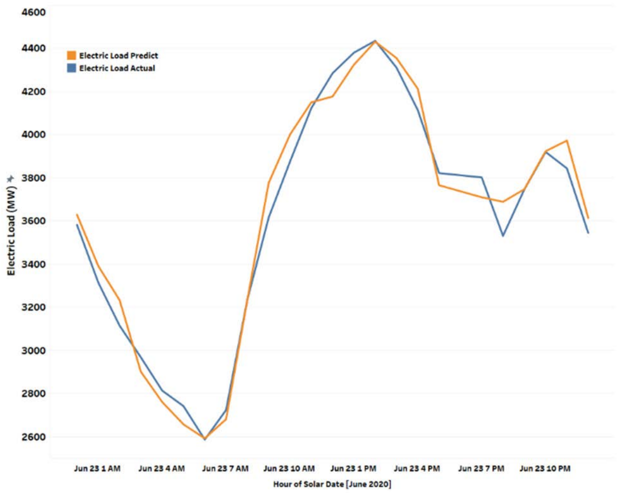

FIGURE 12. Electricity load prediction in the day with maximum load power (June 23, 2020).

comparison between normal days and special holidays. It can be observed that the prediction error seems to increase on special days such as Lunar New Year, Catholic New Year, Hung King's death anniversary, Southern Liberation Day, International Labor Day, and Labor Day. In addition, the error also varies in April and December due to blockade events and outbreaks caused by the Covid-19 epidemic, leading to a big change in the electricity load compared to the past years. The STLF is more interested in the peak days of highest power consumption and special days that cause the load to have a sudden change compared to the neighboring days. In the peak months of summer such as June, July and August, the average MAPE in the month ranges only from 2.6% to 3.2% while it can increase up to 9% in special holidays.

In order to compare the performance of the proposed method with previous methods, we also trained and evaluated in different models such as ARIMA, sequence-to-sequence (Seq2Seq) and prophet.

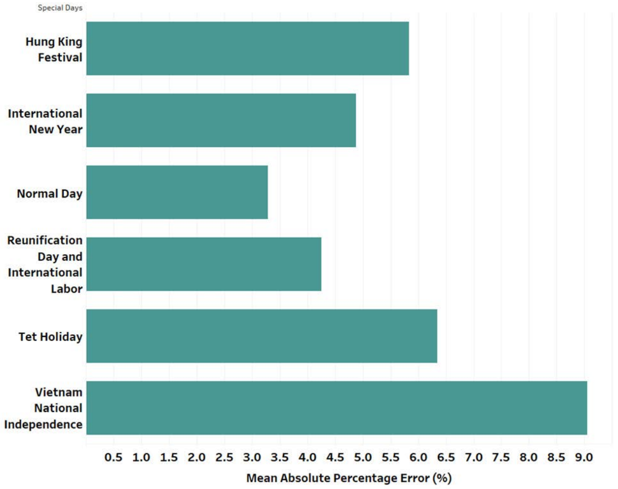

FIGURE 13. MAPE in normal days and special holidays.

The ARIMA models are a general class of models used for forecasting time series data. The model uses historical input data series for prediction. Such data series include: auto regression series (AR) and moving average series (MA). In case of non-stationary series, the model will convert to stationary series by difference. Then the feature parameter of the model will have additional order components: (p, d, q), where p is the order of autoregressive model, d is the degree of differencing, and q is the order of moving-average model. The ARIMA model as a univariate predictive statistical model, therefore, it can only use the past load data to predict the future. In addition, the (p, d, q) parameters are sometime difficult to determine.

Sequence to Sequence (often abbreviated to seq2seq) models is a special class of RNN architectures that usually used to solve complex problems like machine translation, text summarization and time series data. Normally, the input to the seq2seq model is strings of words, character, time series data and the output is another string. The seq2seq model usually consists of two the following main components: encoder code and decoder code. Encoder is responsible for encoding the input sentence into a vector by a recurrent neural network and the decoder will decode the vector into the first sentence based on another recurrent neural network.

The prophet model is introduced by Facebook in 2018. This model is used to forecast univariate time series by decomposing the time series into pieces, which is similar to exponential smoothing [22]. In prophet model, the time series data is split into three categorize: growth or trend, season and holiday. Basically, this model is a curve-fitting approach and the forecasting is done by extend the curve in the future.

Long-Short-Term-Memory (LSTM) based Recurrent Neural Network (RNN) is commonly used in electric load forecasting [23]. In this study, we also build an LSTM model, including 2 LTSM layers with 128 hidden states for the first layer and 64 hidden states for the next layer, and one fully connected layer with 24 hidden nodes.

The accuracy comparison of four models is shown in table 1. The results show that the accuracy of the proposed TFT model is much better than traditional statistical forecasting models as well as recent RNN models.

TABLE 1. Comparison between traditional and proposed model.

Model	MAE (MW)	MAPE (%)
ARIMA	307.75	13.42
Prophet	284.47	13.14
Seq2Seq	138.03	6.02
Linear Regression + LSTM	106,59	4,64
Linear Regression + TFT **(Proposed method)**	**77.51**	**3.38**

V. CONCLUSION

In this research, we proposed a novel RNN model named Temporal Fusion Transformer to forecast the short-term load power of Hanoi city. Different from traditional statistical forecast model, model based on deep learning takes into consideration of several features which greatly affect the electricity load usage such as air temperature, wind, humidity, and specially the holidays of the year. Moreover, TFT is an attention-based deep neural network optimized for great performance and interpretability so TFT gives predictive results that far exceed the results of previous methods. The results show that the model based on deep learning have much better performance than statistical-based models, the proposed model, LSTM model and seq2seq model have prediction error of 3.38%, 4.64%, 6.02%, respectively, while Prophet and ARIMA have prediction error of 13.14%, 13.42%. In addition, we also propose to use weather and time features to improve the accuracy of the STLF problem. In future works, we will focus on improving the accuracy of the model in the holidays and un-wanted events such covid-19.

REFERENCES

[1] J. Hinman and E. Hickey, "Modeling and forecasting short-term electricity load using regression analysis," Illinois State Univ., Normal, IL, USA, Tech. Rep., 2009.

[2] K. A. Al-zahra, K. Moosa, and B. H. Jasim, "A comparative study of forecasting the electrical demand in basra city using box-jenkins and modern intelligent techniques," *Iraqi J. Electr. Electron. Eng.*, vol. 11, no. 1, pp. 110–123, Jun. 2015.

[3] G. N. Shilpa and G. N. Sheshadri, "ARIMAX model for short-term electrical load forecasting," *Int. J. Recent Technol. Eng.*, vol. 11, no. 4, pp. 2786–2790, Nov. 2019.

[4] Q. Lou, Q. Lyu, Z. Na, D. Ma, and X. Ma, "Short-term electric power demand forecastingusing a hybrid model of SARIMA and SVR," *IOP Conf. Earth Environ. Sci.*, vol. 619, Aug. 2020, Art. no. 012035.

[5] K. Chapagain, S. Kittipiyakul, and P. Kulthanavit, "Short-term electricity demand forecasting: Impact analysis of temperature for Thailand," *Energies*, vol. 13, no. 10, p. 2498, May 2020.

[6] B. Farsi, M. Amayri, N. Bouguila, and U. Eicker, "On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach," *IEEE Access*, vol. 9, pp. 31191–31212, 2021.

[7] A. Wahab, M. A. Tahir, N. Iqbal, A. Ul-Hasan, F. Shafait, and S. M. R. Kazmi, "A novel technique for short-term load forecasting using sequential models and feature engineering," *IEEE Access*, vol. 9, pp. 96221–96232, 2021.

[8] S. V. Oprea and A. Bara, "Machine learning algorithms for short-term load forecast in residential buildings using smart meters, sensors and big data solutions," *IEEE Access*, vol. 7, pp. 177874–177889, 2019.

[9] Y. Gao, Y. Fang, H. Dong, and Y. Kong, "A multifactorial framework for short-term load forecasting system as well as the Jinan's case study," *IEEE Access*, vol. 8, pp. 203086–203096, 2020.

[10] N. Giamarelos, E. N. Zois, M. Papadimitrakis, M. Stogiannos, N.-A.-I. Livanos, and A. Alexandridis, "Short-term electric load forecasting with sparse coding methods," *IEEE Access*, vol. 9, pp. 102847–102861, 2021.

[11] D. Syed, H. Abu-Rub, A. Ghrayeb, S. S. Refaat, M. Houchati, O. Bouhali, and S. Banales, "Deep learning-based short-term load forecasting approach in smart grid with clustering and consumption pattern recognition," *IEEE Access*, vol. 9, pp. 54992–55008, 2021.

[12] B. Chen and Y. Wang, "Short-term electric load forecasting of integrated energy system considering nonlinear synergy between different loads," *IEEE Access*, vol. 9, pp. 43562–43573, 2021.

[13] M. Madhukumar, A. Sebastian, X. Liang, M. Jamil, and M. N. S. K. Shabbir, "Regression model-based short-term load forecasting for university campus load," *IEEE Access*, vol. 10, pp. 8891–8905, 2022.

[14] P. Lv, S. Liu, W. Yu, S. Zheng, and J. Lv, "EGA-STLF: A hybrid short-term load forecasting model," *IEEE Access*, vol. 8, pp. 31742–31752, 2020.

[15] Y. Hong, Y. Zhou, Q. Li, W. Xu, and X. Zheng, "A deep learning method for short-term residential load forecasting in smart grid," *IEEE Access*, vol. 8, pp. 55785–55797, 2020.

[16] A. Géron, *Hands-on Machine Learning With Scikit-Learn and TensorFlow*. Sebastopol, CA, USA: O'Reilly Media, Mar. 2017.

[17] A. Krenker, J. Bešter, and A. Kos, "Introduction to the artificial neural networks," in *Artificial Neural Networks—Methodological Advances and Biomedical Applications*. London, U.K.: IntechOpen, 2011.

[18] B. Denny. (2015). *Recurrent Neural Networks Tutorial, Part 1—Introduction to RNNs*. [Online]. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns

[19] O. Christopher. (2015). *Understanding LSTM Networks*. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs

[20] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," 2019, *arXiv:1912.09363*.

[21] A. Vaswani, "Attention is all you need," in *Computation and Language: Machine Learning*, 2017, pp. 1–15.

[22] T. Bashir, C. Haoyong, M. F. Tahir, and Z. Liqiang, "Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN," *Energy Rep.*, vol. 8, pp. 1678–1686, Nov. 2022.

[23] J. Zheng, C. Xu, Z. Zhang, and X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6, doi: 10.1109/CISS.2017.7926112.

[24] S. Arslan, "A hybrid forecasting model using LSTM and prophet for energy consumption with decomposition of time series data," *PeerJ Comput. Sci.*, vol. 8, p. e1001, Jun. 2022, doi: 10.7717/peerj-cs.1001.

PHAM CANH HUY was born in Hanoi, Vietnam, in 1971. He received the Engineering and M.Sc. degrees in energy economics and the Ph.D. degree in economics from the Hanoi University of Science and Technology, in 1993, 1995, and 2002, respectively. From 2002 to 2003, he was a Lecturer with the Department of Energy Economics, Faculty of Economics and Management, Hanoi University of Science and Technology. From 2003 to 2016, he was the Deputy Head of the Energy Economics Department. From 2016 to 2018, he was the Head of the Department of Industrial Economics and since 2018, he has been the Deputy Director of the Institute of Economics and Management, Hanoi University of Science and Technology. He has successfully guided over 50 master's and three Ph.D. students. He is the author of four books and textbooks, including one monograph, and published about 20 articles in specialized journals, domestic and international conferences, and also participated in seven scientific research projects at all levels, of which chaired two scientific research projects at ministerial level. His main research interests include econometric models, forecasting, and energy economics.

NGUYEN QUOC MINH (Member, IEEE) received the bachelor's and master's degrees in electrical engineering from the Hanoi University of Science and Technology, in 2007 and 2009, respectively, and the Ph.D. degree in electrical engineering from the University of Texas at Arlington, TX, USA, in 2016. Since 2016, he has been a Faculty Member with the Department of Electrical Engineering, Hanoi University of Science and Technology. Since 2022, he becomes the Deputy Head of the Department of Electrical Engineering. His research interests include power system protection and control, smart grid, integration of renewable energy, and wireless power transfer.

NGUYEN DANG TIEN is currently pursuing the degree in automation and control engineering with the Hanoi University of Science and Technology. His research interests include machine learning, image processing, and load forecasting. During his study, he has participated in several projects, such as brain tumor segmentation on MRI image and fault detection in PV systems based on machine learning.

TAO THI QUYNH ANH is currently pursuing the degree in energy economic with the Hanoi University of Science and Technology. She is currently doing her internship at Power Engineering Consulting Joint Stock Company 1 (PECC1). Her research interests include machine learning, renewable energy, energy policy, and load forecasting.