Received December 28, 2021, accepted January 13, 2022, date of publication January 25, 2022, date of current version February 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3145951

Recurrent Neural Network-Augmented Locally Adaptive Interpretable Regression for Multivariate Time-Series Forecasting

LKHAGVADORJ MUNKHDALAI[1], (Student Member, IEEE), TSENDSUREN MUNKHDALAI[2], VAN-HUY PHAM[3], MEIJING LI[4], KEUN HO RYU[3,5,6], (Life Member, IEEE), AND NIPON THEERA-UMPON[6,7], (Senior Member, IEEE)

[1]Database and Bioinformatics Laboratory, School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea
[2]Google, Mountain View, CA 94043, USA
[3]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
[4]College of Information Engineering, Shanghai Maritime University, Shanghai 200136, China
[5]School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea
[6]Biomedical Engineering Institute, Chiang Mai University, Chiang Mai 50200, Thailand
[7]Department of Electrical Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand

Corresponding authors: Keun Ho Ryu (khryu@tdtu.edu.vn; khryu@chungbuk.ac.kr) and Nipon Theera-Umpon (nipon.t@cmu.ac.th)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning, South Korea, under Grant 2019K2A9A2A06020672 and Grant 2020R1A2B5B02001717; and in part by the National Natural Science Foundation of China, China, under Grant 61702324 and Grant 61911540482.

ABSTRACT Explaining dynamic relationships between input and output variables is one of the most important issues in time dependent domains such as economic, finance and so on. In this work, we propose a novel locally adaptive interpretable deep learning architecture that is augmented by recurrent neural networks to provide model explainability and high predictive accuracy for time-series data. The proposed model relies on two key aspects. First, the base model should be a simple interpretable model. In this step, we obtain our base model using a simple linear regression and statistical test. Second, we use recurrent neural networks to re-parameterize our base model to make the regression coefficients adaptable for each time step. Our experimental results on public benchmark datasets showed that our model not only achieves better predictive performance than the state-of-the-art baselines, but also discovers the dynamic relationship between input and output variables.

INDEX TERMS Explainable AI, linear regression, recurrent neural network, time-series forecasting.

I. INTRODUCTION

Time-series forecasting is a crucial in many time dependent domains including inventory control, customer management and distribution to finance and marketing [1]. In addition, it is also important to explain the dynamic relationship between input and output variables rather than performing accurate time-series forecasting for some application areas. For example, in a business, it is important to determine how much the price goes up when the supply goes down [2]. Therefore, classical statistical time-series forecasting approaches are still more widely used than deep learning and machine learning methods [3], [4].

Recent advances in artificial intelligence (AI) have achieved superhuman performance in many computer vision applications, including image classification, speech recognition and machine translation. However, this improved predictive performance often increases the complexity of the model, turning such systems into "black box" models. As a result, it becomes hard to understand the decisions of the model and to interpret their predictions. This ambiguity has made it difficult to use the black box models for time-series data. Although very few interesting deep learning-based interpretable architectures have been proposed in recent years to achieve high predictive performance on time-series tabular data [1], [5]–[8], these models have not addressed the identifying relationship between explanatory variables and output.

The associate editor coordinating the review of this manuscript and approving it for publication was Dost Muhammad Khan.

In order to address this issue, we propose a novel adaptive interpretable deep learning architecture that can identify the relationship between input and output variables for multivariate time-series data. Our proposed model consists of two key aspects: an interpretable base-learner and a meta-learner. A simple linear regression is chosen as our base-learner. Linear regression is the most explainable model because it explains the linear relationship between input and output variables [9]. The sign of a regression coefficient determines whether there is a positive or negative correlation between each input variable and the target variable [10]. A positive (negative) coefficient indicates that as the input variable's value increases, the predicted mean value of the target variable also tends to increase (decrease). In other words, by determining the impact of input variables on the target variable, we can explain the behavior of models by capturing the relationship between input variables and their direction. However, the major drawback of linear regression is linearity as well. The linear relationships of input and output variables are hardly restricted and they usually oversimplify how complex reality is; therefore, the predictive ability of linear regression is often not good. Therefore, we augment our base-learner by a meta-learner to improve its predictive performance. We use long short-term memory (LSTM) network as our meta-learner. Our meta-learner re-parameterize a base-learner at each time step. In other words, we re-parameterize our base-learner using a meta-learner to determine a local linear function that "best fits" data at each time period. Once we find a local linear function for each observation in data, it is easy to explain the relationship between input and output by measuring the impact of each input variable on the target variable.

The overall framework of our proposed architecture is shown in Figure. 1. We first perform the OLS (base-learner) to obtain regression coefficients and their standard errors. Second, we apply LSTM neural networks (meta-learner) to predict the probabilities for finding the Gaussian critical value to update each regression coefficient for each observation. We use two inputs, which are non-normalized and normalized inputs. Our base-learner receives non-normalized inputs to explain the logical and global relationship between input and output variables. Our meta-learner takes normalized input to adapt regression coefficients locally, and these adapted regression coefficients demonstrate the local relationship between input and output by measuring the impact on target variable for each observation. Based on the predicted probabilities and the standard error of regression coefficients, we calculate the local regression coefficients using the confidence interval formula. Finally, we rebuild the linear regression equation based on the local regression coefficients for each observation.

We first train our proposed model on the public electricity and traffic datasets used for time series forecasting benchmark to compare to the state-of-the-art time-series models [11]. Our model achieved better predictive performance than other state-of-the-art baseline time-series forecasting models.

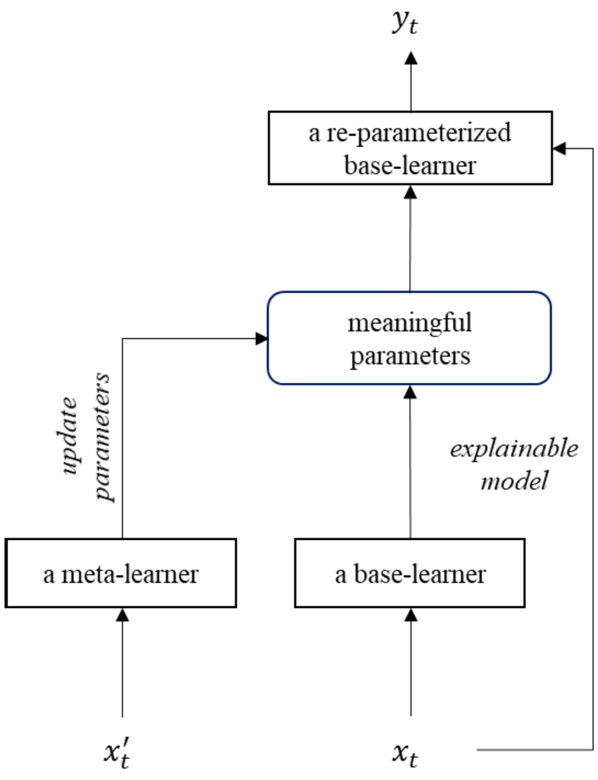

FIGURE 1. Overview of our proposed model, where x_t is input variables, x'_t is normalized input variables and y_t is the target variable at t time period. We first perform a linear regression to obtain our base-learner. In this case, data should not be normalized to obtain meaningful parameters to explain the relationship between input and output variables. We then train our meta-learner using normalized input to update the parameters of our base-learner.

In addition, we extensively studied the predictive performance and the model interpretability on several time-series datasets coming from different domains. As a result, our model also showed greatly higher performance than machine learning and regression baselines. These experiments on the benchmark datasets proved that our proposed deep learning-based explainable architecture can be one of best predictive methods for time-series data.

The main contributions of this work are summarized as follows:

1) We proposed a novel interpretable deep learning architecture to achieve both high predictive accuracy and model interpretability for time series data.
2) We proposed a novel augmentation method to improve the predictive performance of linear regression with keeping its interpretability.
3) We used recurrent neural networks to augment the linear regression to parameterize the family of linear functions.
4) We make the linear regression coefficients adaptable within their confidence intervals; therefore, our model

cannot be overfitted nor misrepresent the relationship between input and output variables.

5) Our proposed model can determine the dynamic relationship between input and target variables by measuring local impact of the input variables on the target variable.
6) We provided evaluation of the proposed model on benchmark datasets in terms of predictive accuracy and model explainability.

This paper is organized as follows: Section 2 discusses related work. Then, Section 3 presents concept of the proposed model. Section 4 demonstrates datasets and experimental results. Finally, Section 5 summarizes the general findings from this study and discusses possible future research areas.

II. RELATED WORK

Time-series forecasting methods have been developed into a significant and active research area [12], in which interpretable deep learning based models have also been widely studied [1], [5], [13]–[16]. Initially, attention-based neural network models were proposed to identify noticeable portions of input variables for each time step using the magnitude of attention weights for time series data with interpretability motivations [6], [18].

In addition, authors have been applied post-hoc explanation methods on pre-trained deep learning models to obtain the model understandability and increase humans' trust [19], [20]. Ribeiro *et al.* [19] proposed the LIME technique, short for Local Interpretable Model-agnostic Explanations, in an attempt to explain any decision process performed by a black-box model. Another popular method for explaining black-box models is SHapley Additive eXplanations (SHAP) [20], [21]; SHAP are Shapley values representing the variable importance measure for a local prediction. They are calculated by combining insights from 6 local feature attribution methods. Unfortunately, these post-explainable models can make misinterpretations on unseen data because these models are usually based on the permutations, which are randomly sampled from the marginal distribution. According to Rudin [22], the best explanation should be provided by the model itself. In other words, the explainability should be incorporated into the architecture to allow the model to make the correct predictions with the logical correlations. Although a number of interpretable modeling approaches have been proposed for time-series forecasting, these studies have only focused on explaining the variable contributions [5] or on decomposition for analysing univariate time series [1] and they have not explain relationship between input and output variables.

In conclusion, deep learning-based model that explains the logical relationship between input and output variables for time series data has not been proposed. On the other hand, the linear regression models are considered to be relatively explainable [23], especially when the regression coefficient has a particular meaningful value. Therefore, we propose a novel explainable deep learning architecture by improving linear regression predictability based on the recurrent neural networks.

Another related line of work focuses on meta-learning. Utilizing one neural network to produce parameters for another neural network has been studied earlier in meta-learning field [24]–[27]. Our proposed model is built on this method of establishing a meta-learning model. We train meta-model (neural network) to explain its underlying base-model (linear regression) parameters. Recently, Munkhdalai and Hong [28] proposed Meta Networks (MetaNet) that learns to fast parameterize underlying neural networks for rapid generalizations. Our method is based on the idea of the MetaNet that uses fast-weights, which has successfully been used on images, text, and audio data [29]–[32]. In order to successfully apply this approach for time-series data, we use meta-learner to estimate fast probabilities for finding the Gaussian critical value for each regression coefficient. Generally, we attempt to encourage predictive ability of interpretable model using neural networks. We will present our methodology in detail in the next section.

III. THE PROPOSED MODEL

Our proposed model consists of two main phases – linear regression (interpretable base-learner) and recurrent neural networks (meta-learner). We first perform simple linear regression on training set to obtain unbiased regression coefficients and their standard errors (see Figure. 2). Second, we train recurrent neural networks as a meta-learner on normalized training set to predict the probability for locating percentile of Gaussian distribution for each regression coefficient. Finally, we reconstruct the linear regression equation by using the updated local regression coefficients at each time step. In addition, there are two kinds of input; non-normalized for base-learner and normalized inputs for meta-learner. Neural networks usually take normalized input for faster convergence and higher predictive performance [33], [34]. However, data scaling leads to misinterpret the relationship between the input and output variables. Therefore, our base-learner takes non-normalized inputs to explain the logical and global relationship between input and output variables, and our meta-learner takes normalized input to predict the probabilities for finding the Gaussian critical value to update each regression coefficient locally. These adapted regression coefficients demonstrate the local relationship between input and output by measuring the impact on target variable for each observation.

A. BASE-LEARNER

Given a set of dataset $(x_1, y_1), \dots (x_t, y_t)$ of t timestamps, a linear regression model estimates the β coefficients that provide the best linear fits between the dependent variable (y_i) and n independent variables $(x_{i1}, \dots x_{in})$. The model for linear regression is [35]:

$$y_t = \beta_0 + \sum_{j=1}^{n} \beta_j x_{tj} + \varepsilon_t \tag{1}$$

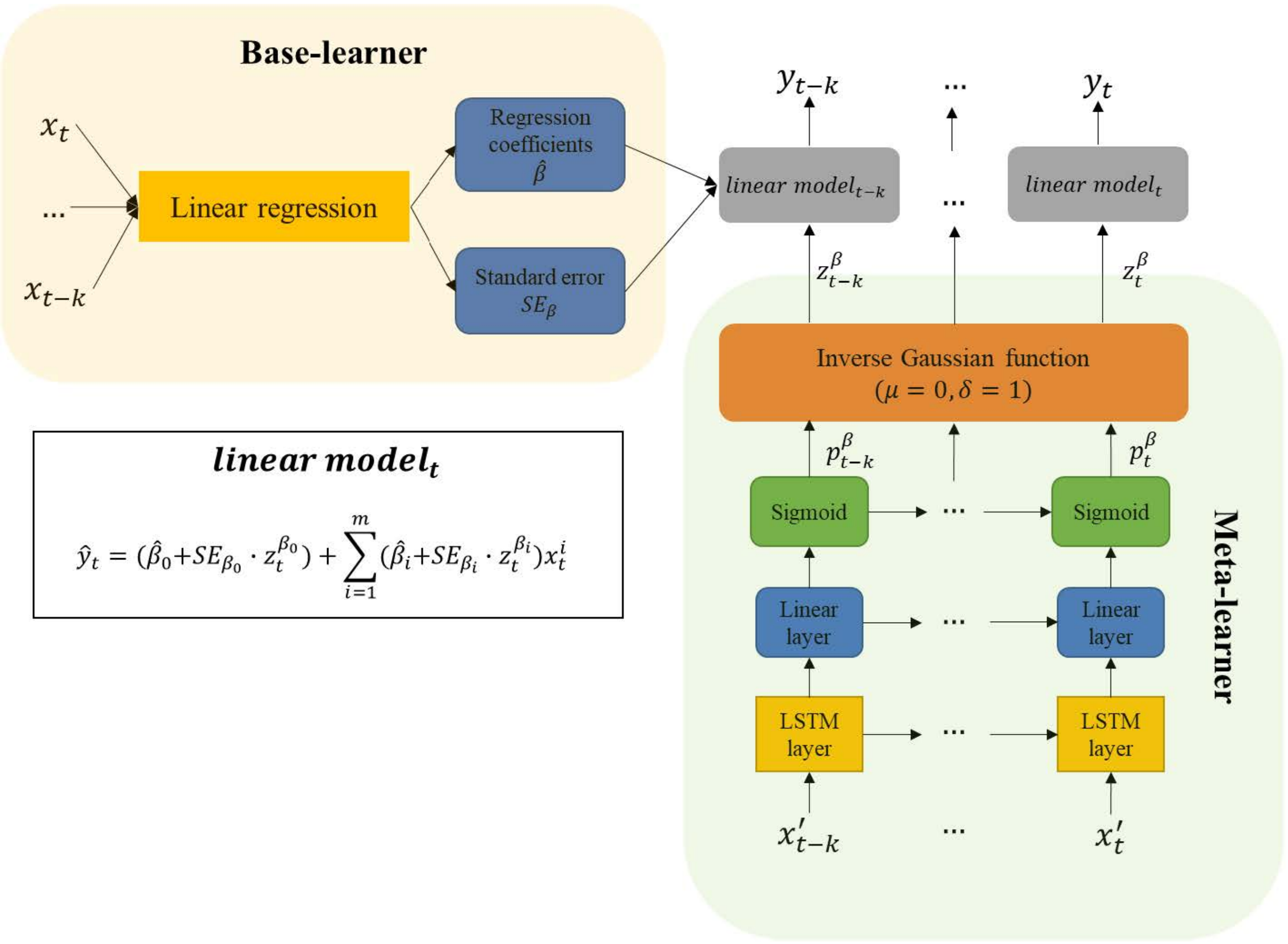

FIGURE 2. The proposed architecture. Where x_t is input variables, x'_t is normalized input variables and y is the target variable at t time period. The p_t^{β} is the predicted probabilities at t time period, we use these probabilities to obtain the gaussian critical values - z_t^{β} needed to build a local linear model for each time step.

where ε_t is independent, identically distributed ($i.i.d.$) random variables with $\mathbb{E}\{\varepsilon_t\} = 0, \mathbb{E}\left\{\varepsilon_t^2\right\} = \sigma^2$ and bounded third moment.

The regression coefficients can simply be computed by using the OLS estimator:

$$\hat{\beta}_{OLS} = \left(X_t^T X^T\right)^{-1} X_t^T y_t \tag{2}$$

where $X_t = [x_1^\top, \ldots x_n^\top] \in \mathbb{R}^{n \times t}$ is the design matrix and $y_t = [y_1, \ldots y_t] \in \mathbb{R}^t$. The regression coefficients estimated from data are subject to sampling uncertainty. In other words, the true value of the regression coefficient can never be estimated from the sample data. Instead, we could construct confidence interval for each regression coefficient:

$$CI_{\alpha/2}^{\beta_j} = \left[\hat{\beta}_j - IG_{\alpha/2} \cdot se\left(\hat{\beta}_j\right), \hat{\beta}_j + IG_{\alpha/2} \cdot se(\hat{\beta}_j)\right] \tag{3}$$

where α is the significance level, IG is the inverse Gaussian distribution and $se\left(\hat{\beta}_j\right)$ is the standard error of the regression coefficient $\hat{\beta}_j$. The main idea of our proposed model is to show a better performance on point forecasting by adapting the linear regression coefficients within their confidence interval for each time step. In order to find the "best fits" local linear function for each time step, we must determine the appropriate value in the confidence interval for each of the linear regression coefficients. In order to achieve better performance, we design meta-learner model to find the appropriate regression coefficients for each time-step based on the formula for calculating confidence interval. Our meta-learner model predicts the appropriate significance level for each regression coefficient to make it adaptable.

B. META-LEARNER

We use Long short-term memory (LSTM) neural network architecture as a meta-learner model. LSTM network is an extension of recurrent neural networks. It was proposed by Hochreiter & Schmidhuber [36] as a solution to the vanishing gradient problem. LSTM helps to solve long-term dependencies by extending their memory cells and utilizing a gating mechanism to control information flow. The memory

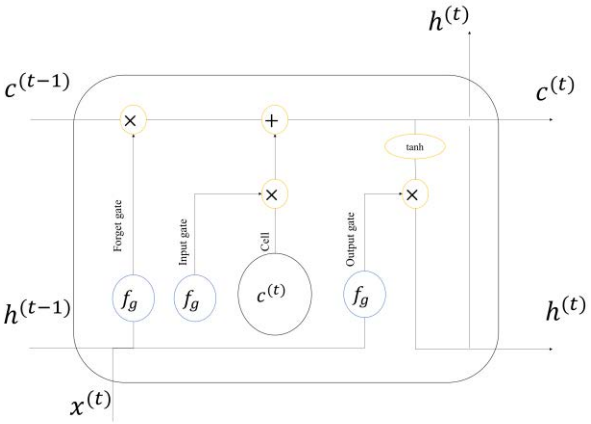

FIGURE 3. **LSTM architecture.**

cell consists of three gates – input, forget and output gate as shown in Figure. 3.

These gates decide whether or not to add new input in (input gate), erase the unnecessary information (forget gate) or to add it impact the output at the current time step (output gate). Theoretically, these gates are represented as:

$$\begin{aligned} f^{(t)} &= f_g\left(\omega_f x^{(t)} + u_f h^{(t-1)} + b_f\right) \\ i^{(t)} &= f_g\left(\omega_i x^{(t)} + u_i h^{(t-1)} + b_i\right) \\ o^{(t)} &= f_g\left(\omega_o x^{(t)} + u_o h^{(t-1)} + b_o\right) \\ c^{(t)} &= f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ f_c(\omega_c x^{(t)} + u_c h^{(t-1)} + b_c) \\ h^{(t)} &= o^{(t)} \circ f_h\left(c^{(t)}\right) \end{aligned} \tag{4}$$

where $f^{(t)}$ is the forget gate, $i^{(t)}$ is the input gate, $o^{(t)}$ is the output gate, $c^{(t)}, c^{(t-1)}$ are the cell state vectors, $\omega_f, \omega_i, \omega_o, \omega_c$ denote the weights of the hidden and output layers, b_f, b_i, b_o, b_c are the bias vectors, $x^{(t)}$ is the vector of inputs, $f_g(*)$, $f_c(*)$, $f_h(*)$ are the activation functions, $h^{(t)}$, $h^{(t-1)}$ denote the output of hidden layer neurons at time t and t-1, and u_f, u_i, u_o, u_c denote the weights that connect the hidden layer neurons to the recurrent layer and output, respectively.

C. RECURRENT NEURAL NETWORK-AUGMENTED LOCALLY ADAPTIVE INTERPRETABLE REGRESSION

As described above, input of our meta-learner can be normalized n independent variables $(x'_{ij}) \in R \times R^n$, $i = 1, \ldots, T$ and $j = 1, 2, \ldots, n$, and output should be the predicted probability $(prob_i)$ corresponding to the Gaussian critical value. Since we predict probability for finding the critical value of Gaussian distribution, the activation function of output layer can be sigmoid (σ) because it produces value between 0 and 1. Thus:

$$prob = \sigma\left(FC(LSTM\left(x'; \theta_{lstm}; b_{lstm}\right); \theta_{fc}; b_{fc}\right) \tag{5}$$

where x' is normalized input, $prob$ denotes the predicted probability, LSTM is a LSTM layer, FC is a fully connected linear layer and θ_{lstm}, θ_{fc}, b_{lstm}, and b_{fc} denote the weight parameters of our meta-learner model.

We also make additional smoothing parameters on the output of sigmoid function to control significance level of confidence interval.

$$prob = \frac{\sigma\left(FC(LSTM\left(x'; \theta_{lstm}; b_{lstm}\right); \theta_{fc}; b_{fc}\right) + \epsilon}{1 + \tau} \tag{6}$$

where $\epsilon, \tau(\tau > \epsilon)$ are smoothing parameters and these parameters should be close to 0. We can set upper and lower confidence intervals for the regression coefficients by adjusting these smoothing parameters. For example: at $\epsilon = 0.005, \tau = 0.006$, the significance level of the lower confidence interval is equal to 0.005 and the upper is equal to 0.999.

Recall that we pick the estimated regression coefficients and their standard errors as numerical input after performing linear regression. So we can easily reconstruct the original regression equation during the learning process of the meta-learner:

$$\begin{aligned} \hat{y} = &\left(\hat{\beta}_0 + IG\left(prob_0\right) \cdot se(\hat{\beta}_0)\right) \\ &+ \sum_{i=1}^{n} \left(\hat{\beta}_i + IG\left(prob_i\right) \cdot se(\hat{\beta}_i)\right) x_i \end{aligned} \tag{7}$$

where x_i is i-th independent variable (not normalized).

D. MODEL TRAINING

As explained in Section III.A, we first perform OLS estimator to obtain unbiased regression coefficients and their standard error. Then we apply our meta-learner to improve the predictive power of base-learner by adapting their regression parameters locally. If we do a little modification on Eq. 7, we can obtain a simple model that can be trained by using stochastic gradient descent (SGD) optimization algorithm.

$$\begin{aligned} \hat{y} &= \left(\hat{\beta}_0 + IG\left(prob_0\right) \cdot se(\hat{\beta}_0)\right) \\ &\quad + \sum_{i=1}^{n} \left(\hat{\beta}_i + IG\left(prob_i\right) \cdot se(\hat{\beta}_i)\right) x_i \\ &= \left(\hat{\beta}_0 + \sum_{i=1}^{n} \hat{\beta}_i x_i\right) + \left(IG\left(prob_0\right) \cdot se(\hat{\beta}_0)\right. \\ &\quad \left. + \sum_{i=1}^{n} IG\left(prob_i\right) \cdot se(\hat{\beta}_i) x\right) \end{aligned} \tag{8}$$

From Eq. 8, we can compute the prediction performed by the OLS estimator. If we subtract the prediction performed by the OLS estimator from the actual value, the error value (e) remains on the left-hand side of the equation, which should be predicted by our meta-learner. Then it can be as follows:

$$\begin{aligned} \hat{y} - \left(\hat{\beta}_0 + \sum_{i=1}^{n} \hat{\beta}_i x_i\right) = &\left(IG\left(prob_0\right) \cdot se(\hat{\beta}_0)\right. \\ &\left. + \sum_{i=1}^{n} IG\left(prob_i\right) \cdot se(\hat{\beta}_i) x\right) \end{aligned} \tag{9}$$

Algorithm: Model Training Algorithm

```
Input: Training set:      non-normalized $\{x_i y_i\}_{i=1}^{N}$
                          normalized input $\{x'_i\}_{i=1}^{N}$
       Validation set:    non-normalized $\{xx_i yy_i\}_{i=1}^{M}$
                          normalized input $\{xx'_i\}_{i=1}^{M}$
       Hyperparameter:    epoch number-$epoch$
                          patience number-$pn$
                          learning rate $\alpha$
Output: Meta-learner: $RNN(*)$
        Coefficients: $\hat{\beta}$
        Standard errors: $se(\hat{\beta})$
Procedure: {
perform OLS estimator to obtain $\left\{\hat{\beta}; se(\hat{\beta})\right\}; pred_{OLS}; pred_{OLS}^{val}$
$L_{val} = inf; e = y\text{-}pred_{OLS}; ee = yy\text{-}pred_{OLS}^{val}$
for $i = 0$ to $epoch$ do
    for j = 1 to N do
        $prob = RNN(\theta, b; x'_i)$// train meta-learner
        $\hat{e}_j = \left(IG(prob_0) \cdot se(\hat{\beta}_0)\right) + \sum_{l=1}^{K} \left(IG(prob_l) \cdot se(\hat{\beta}_l)\right) x_{lj}$
        $L_j \leftarrow loss_{train}(\hat{e}_j, e_j)$
        $\theta_j^* \leftarrow \theta_{j-1}^* - \alpha \nabla_\theta L_j$// gradient descent
        $b_j^* \leftarrow b_{j-1}^* - \alpha \nabla_b L_j$// gradient descent
    end for
    $prob\_val = RNN(\theta_j^*, b_j^*; xx'_i)$
    $\hat{ee}_j = \left(IG(prob\_val_0) \cdot se(\hat{\beta}_0)\right) + \sum_{l=1}^{K} \left(IG(prob\_val_l) \cdot se(\hat{\beta}_l)\right) xx_l$
    $L_j^{val} \leftarrow loss_{val}(\hat{ee}_j, ee)$
    if $L_j^{val} < L_{val}$ do
        $L_{val} = L_j^{val}$
        $patience = 0$
        return $m(\theta_j^*, b_j^*); \left\{\hat{\beta}; se(\hat{\beta})\right\}$
    else do
        $patience = patience + 1$
    if $patience > pn$ do
        break
end for
}
```

$$e = \left(IG\left(prob_0\right) \cdot se(\hat{\beta}_0) + \sum_{i=1}^{n} IG\left(prob_i\right) \cdot se(\hat{\beta}_i) x \right) \quad (10)$$

We can consider Eq.10 as a nonlinear part of our model, which is similar with these studies [37], [38]. However, the main advantage of our model is that the prediction performance is improved by updating the parameters of base-learner without compromising its interpretability.

We now can design our loss function as follows:

$$\mathcal{L}(\theta, b) = loss_{MSE}\left(m\left(x, x', \hat{\beta}, se\left(\hat{\beta}\right); \theta, b\right), e\right) \quad (11)$$

where $loss_{MSE}$ is the mean squared error (MSE) and $m(*)$ is our proposed model with parameters θ and b, $\hat{\beta}$, and $se\left(\hat{\beta}\right)$ are the estimated regression coefficients by OLS, and e is error values.

In addition, the output of meta-learner should be equal to the number of independent variables, and our architecture can now easily be trained with SGD optimization with the backpropagation algorithm. The model training algorithm for our proposed architecture is as shown below:

TABLE 1. Summary benchmark dataset for predictive comparison.

Dataset details	Electricity	Traffic
Target type	$\mathbb{R}$	[0,1]
Number of entities	370	963
Number of samples	2.1 million	4 million
Number of continues variables	2	2
Number of categorical variables	4	2

TABLE 2. The variables and description of financial dataset.

№	Variables	Description	Units
1	Nasdaq	Nasdaq stock market index	Index
2	DTWEXB	Trade weighted U.S. dollar index	Index
3	DGS5	5-year treasury constant maturity rate	%
4	T5YIFR	5-year forward inflation expectation rate	%
5	BAA10Y	Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity	%
6	EPUI	Economic policy uncertainty index for U.S.	Index
7	WILLREIT PR	Wilshire U.S. real estate investment trust price index	Index
8	WTI	Crude oil prices	Dollars per barrel

TABLE 3. The variables and description of air quality dataset.

№	Variables	Description	Units
1	CO(GT)	True hourly averaged concentration CO	mg/m^3
2	PT08.S1(CO)	Hourly averaged sensor response	tin oxide
3	NMHC(GT)	True hourly averaged overall Non Metanic Hydro Carbons concentration	microg/m^3
4	C6H6(GT)	True hourly averaged Benzene concentration	microg/m^3
5	PT08.S2(NMHC)	Hourly averaged sensor response	titania
6	NOx(GT)	True hourly averaged NOx concentration	ppb
7	PT08.S3(NOx)	Hourly averaged sensor response	tungsten oxide
8	NO2(GT)	True hourly averaged NO2 concentration	microg/m^3
9	PT08.S4(NO2)	Hourly averaged sensor response	tungsten oxide
10	PT08.S5(O3)	Hourly averaged sensor response	indium oxide
11	T	Temperature	◦C
12	RH	Relative Humidity	%

IV. EXPERIMENTAL RESULTS

A. DATASETS

We used 3 public benchmark time-series datasets (see Table 1) to compare the predictive performance of our proposed model to the other state-of-the-art time-series baselines [11]. We also chose additional 3 multivariate time-series datasets (see Table 2 and 3) to demonstrate the interpretability of our proposed model [6].

Electricity Set: this dataset contains the electricity of consumption of 370 customers and aggregated on an hourly level [11].

Traffic Set: traffic dataset describes hourly occupancy rate, between 0 and 1, of 963 car lanes of San Francisco bay area freeways [11].

The electricity, traffic, energy use of appliances and air quality datasets from UCI Machine Learning Repository [39].

TABLE 4. The variables and description of the energy use of appliances dataset.

№	Variables	Description	Units
1	Appliances	Appliances energy consumption	Wh
2	lights	Light energy consumption	Wh
3	T1	Temperature in kitchen area	◦C
4	RH1	Humidity in kitchen area	%
5	T2	Temperature in living room area	◦C
6	RH2	Humidity in living room area	%
7	T3	Temperature in laundry room area	◦C
8	RH3	Humidity in laundry room area	%
9	T4	Temperature in office room	◦C
10	RH4	Humidity in office room	%
11	T5	Temperature in bathroom	◦C
12	RH5	Humidity in bathroom	%
13	T6	Temperature outside the building	◦C
14	RH6	Humidity outside the building	%
15	T7	Temperature in ironing room	◦C
16	RH7	Humidity in ironing room	%
17	T8	Temperature in teenager room 2	◦C
18	RH8	Humidity in teenager room 2	%
19	T9	Temperature in parents room	◦C
20	RH9	Humidity in parents room	%
21	T_out	Temperature outside	◦C
22	Press_mm_hg	Pressure	mm Hg
23	RH_out	Humidity outside	%
24	Windspeed	Windspeed	m/s
25	Visibility	Visibility	km
26	Tdewpoint	Tdewpoint	◦C
27	rv1	Random Variable 1	

The financial dataset was downloaded from The Federal Reserve Bank of St. Louis, https://fred.stlouisfed.org.

Financial Set: the financial market dataset in Table 2 contains Nasdaq stock market price, trade weighted U.S. dollar index (DTWEXB) and other 6 time series, which are correlated to financial market such as 5-year treasury constant maturity rate (DGS5), 5-year forward inflation expectation rate (T5YIFR), Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity (BAA10Y), Economic policy uncertainty index for U.S. (EPUI), Wilshire U.S. real estate investment trust price index (WILLREITPR) and Crude oil prices (WTI). This dataset is daily time series that covers the period between 01/02/2003 and 11/19/2018.

Air Quality Set: De Vito *et al.*, [40] published this dataset. The dataset contains 12 variables as summarized in Table 3 We considered CO(GT) and NO2(GT) variables as target variables to predict them the same as De Vito *et al.*, [40]. Data length consists of 9,357 hourly data points that recorded between 3/10/2004 18:00 and 4/04/2005 14:00.

Energy Set: the energy use of appliances dataset was introduced by [41]. They investigated data-driven predictive models for this dataset. The dataset consists of 27 time series variables including appliances energy consumption as shown in Table 4. The data were recorded every 10 minutes and 19735 data points were generated between 1/11/2016 17:00 and 5/27/2016 18:00. In accordance with the focus variable of Candanedo *et al.*, [41] the appliances was chosen as a target variable.

B. BASELINE MODELS AND HYPERPARAMETER

For the state-of-the-art time-series baselines, we directly used the predictive performances from Yu, Rao & Dhillon [11]. They presented a temporal regularized matrix factorization (TRMF) framework that supports data-driven temporal learning and forecasting. We also directly compared our results to DeepAR model introduced in [7]. According to our benchmark the state-of-the-art models:

○ TRMF-AR: Temporal Regularized Matrix Factorization model [11]

○ SVD-AR(1): Explained in [11]

○ TCF: Matrix factorization with the simple temporal regularizer proposed in [42]

○ AR(1): n-dimensional AR(1) model [43]

○ DLM: the code provided in [44]

○ DeepAR [7]

For other 3 datasets, we aimed to demonstrate how our model can be interpreted on these models. In addition, we presented how our model improves the predictive ability of linear regression models such as linear, lasso, ridge and Bayesian regressions. We also compared the predictive performance of our proposed model to deep learning baselines, introduced in this work [6] such as AIS-RNN, AIS-GRU and AIS-LSTM. In addition, TabNet model [45], which is the most popular and high-performance deep learning-based model for tabular data, is used as baseline model.

In our proposed model, we need to define neural network architecture and other hyper-parameters for meta-learner. LSTM neural network was chosen as our meta-learner for time-series data. For Electricity and Traffic datasets, our meta-learner consists of two hidden layers, each layer consisting of 64 neurons since these datasets contain large number of entities. For other 3 datasets, we chose a larger meta-learner architecture, which consists of 2 hidden layers and each layer has 512 neurons.

We set the learning rate to 0.001 and the maximum epoch number for training to 1000. In addition, we can optimize the hyper-parameters of our meta-learner using additional experiments to improve the predictive performance, but the chosen hyperparameters render desirable performance in this study. Therefore, we did not make any additional experiments for hyperparameter optimization to save the computation time.

In addition, an Early Stopping algorithm was used for selecting the best model based on given hyperparameters. The smoothing parameters are chosen as $\epsilon = 1\text{e-}06$ and $\tau = 1\text{e-}05$. We configured the same model settings for all datasets, and datasets are partitioned into three parts; i.e., training (the first 70% of total data), validation (the next 10% of total data) and test (the last 20% of total data) sets, based on the time sequence.

C. EVALUATION METRICS

For regression task, the difference between the predicted and the actual values are usually considered as evaluation metrics [46]. We used the root mean square error (RMSE), mean absolute error (MAE) and Mean Absolute Percentage Error (MAPE) to evaluate model performances.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \quad (12)$$

$$MAE = \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{n} \quad (13)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (14)$$

where $\hat{y}_i$ denotes the *i-th* predicted value, y_i denotes the *i-th* actual value and n is the number of observations.

In addition, in order to make the same settings to directly compare the forecasting performance to these studies; [7], [11], [42], and [44]. Normalized Deviation (ND) and Normalized root mean square error (NRMSE) metrics were used to evaluate the predictive performance.

$$ND = \frac{\sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{n}}{\sum_{i=1}^{n} \frac{|y_i|}{n}} \quad (15)$$

$$NRMSE = \frac{\sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}}{\sum_{i=1}^{n} \frac{|y_i|}{n}} \quad (16)$$

where $\hat{y}_i$ denotes the *i-th* predicted value, y_i denotes the *i-th* actual value, and n is the number of observations.

D. PREDICTIVE PERFORMANCE

We first evaluated the predictive performance of our proposed method to compare other state-of-the-art baselines. We made the same settings to directly compare the forecasting performance with these studies [7], [11]. ND and NRMSE metrics were used to compare the predictive performances.

As shown in Table 1, these two datasets are very large and consist of numerous number of entities. In order to apply our model to these datasets, we trained our model on each entity; 370 models were trained on electricity set and 963 models were trained on traffic set. We also need to set maximum sequence (maximum lag) length for our meta-learner (input variables for base-learner).

For electricity dataset, maximum sequence is equal to 2 and input variables includes one and two lags of 'power_usage', 'hours_from_start' and 'hour', 'day', 'day_of_week', 'month' variables. The 'hour', 'day', 'day_of_week', and 'month' were used as categorical variable and we used one-hot encoding for categorical variables.

Regarding traffic dataset, we chose maximum sequence is equal to 3 and one and two lags of 'values' and 'hours_from_start' are used as input continuous variables and 'time_on_day' and 'day_of_week' are used as input categorical variables. We also used one-hot encoding for categorical variables.

TABLE 5. The predictive performance comparison between the state-of-the-art baselines and our model.

Models	Electricity		Traffic	
	ND	NRMSE	ND	NRMSE
TRMF-AR	0.255	1.397	0.187	0.423
SVD-AR(1)	0.257	1.865	0.555	1.194
TCF	0.349	1.838	0.624	0.931
AR(1)	0.219	1.439	0.275	0.536
DLM	0.435	2.753	0.639	0.951
DeepAR	0.07	1.000	**0.17**	0.42
Ours	0.078	**0.791**	0.204	**0.407**

Our proposed model outperformed the state-of-the-art baselines on electricity and traffic datasets in terms of NRMSE metric, and showed the comparable results for ND evaluation metric (see Table 5).

We reported the predictive performance comparison between the state-of-the-art baselines and our proposed model in Table 5. Our model outperformed the state-of-the-art baselines by achieving 0.791 NRMSE on electricity dataset and by achieving 0.407 NRMSE on traffic dataset. For ND evaluation metric, DeepAR model showed the best performance by achieving 0.07 and 0.17 ND on Electricity and Traffic datasets, respectively.

Second, we performed the additional experiments to compare our proposed model with deep learning and regression baselines. In order to prove the effectiveness and robustness of our proposed model for time-series data, 'Nasdaq' from the finance dataset, 'CO(GT)' from the air quality dataset and 'Appliances' from the energy use of appliances dataset, were chosen as target variable. In addition, the maximum sequence is equal to 5 for all datasets and all independent variables were chosen as an input for meta-learner for experimental analysis.

For regression baselines, we chose lags (from 1 to 5) of independent and target variables as input to make the same settings with our proposed model.

Regarding deep learning baselines, we directly compared our predictive performance to our previous work [6]. Munkhdalai *et al.*, [6] proposed an end-to-end recurrent neural network architecture equipped with an adaptive input selection mechanism, named AIS-RNN, to improve the prediction performance for multivariate time series forecasting. The proposed AIS-RNN model outperformed the baselines including Elman RNN, Gated Recurrent Unit (GRU), LSTM, Support Vector Machine, Random Forest, AdaBoost and Decision tree models by up to 38% on these 3 benchmark datasets.

To obtain unbiased regression coefficients, we selected input variables based on t-test and the variance inflation factor (VIF) for our base-learner. We selected the variables whose p-value for t-test is less than 0.10 and VIF value is less than 10. The VIF value of selected variables for Finance, Air quality and Energy datasets were displayed in Figure 4-6, respectively.

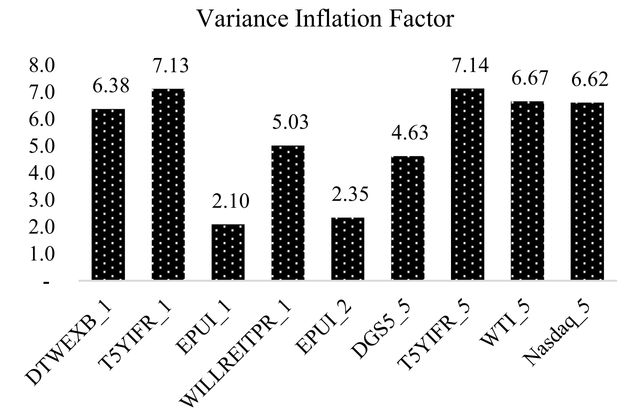

FIGURE 4. Variance inflation factor for finance dataset.

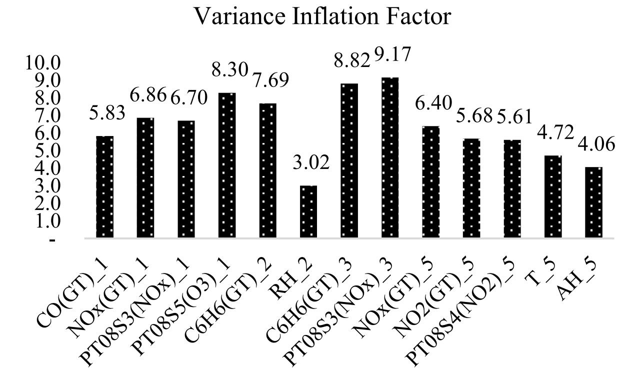

FIGURE 5. Variance inflation factor for air quality dataset.

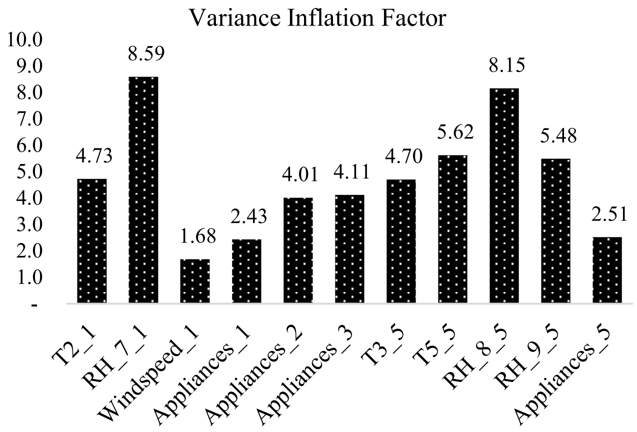

FIGURE 6. Variance Inflation factor for energy dataset.

TABLE 6. RMSE of baselines and our proposed model on three datasets.

	Methods	Finance	Air quality	Energy
Regression	OLS	142.12	0.69	60.22
	Lasso	135.32	0.69	60.17
	Ridge	143.19	0.69	60.19
	Bayesian	143.35	0.69	60.16
Deep learning	TabNet	647.31	0.64	60.80
	AIS-LSTM	109.38	0.64	**59.81**
	AIS-GRU	**60.70**	0.64	61.50
	AIS-RNN	79.74	0.64	61.50
	Ours	115.03	**0.62**	59.94

TABLE 7. MAE of baselines and our proposed model on three datasets.

	Methods	Finance	Air quality	Energy
Regression	OLS	142.12	0.69	60.22
	Lasso	135.32	0.69	60.17
	Ridge	143.19	0.69	60.19
	Bayesian	143.35	0.69	60.16
Deep learning	Tabnet	549.24	0.42	28.01
	AIS-LSTM	78.07	0.42	**23.42**
	AIS-GRU	**42.20**	0.42	27.21
	AIS-RNN	55.96	0.42	27.20
	Ours	79.79	**0.41**	25.73

TABLE 8. MAPE of baselines and our proposed model on three datasets.

	Methods	Finance	Air quality	Energy
Regression	OLS	1.75	35.34	27.36
	Lasso	1.68	36.68	25.85
	Ridge	1.77	36.75	26.31
	Bayesian	1.77	36.79	25.92
Deep learning	Tabnet	8.47	31.71	25.84
	AIS-LSTM	1.30	33.61	**18.89**
	AIS-GRU	**0.70**	33.76	26.08
	AIS-RNN	0.93	33.59	26.08
	Ours	1.30	**30.71**	22.63

Based on the selected base-learner models, we trained our meta-learner to improve their predictive performance. Table 6 reported the predictive performance for all models in terms of RMSE. We observed that our proposed model outperformed the baseline models including deep learning and regression baselines on the air quality dataset by achieving 0.62 RMSE, and showed the comparable results on energy dataset by performing 59.94 RMSE. While AIS-GRU performed the best on Finance dataset by achieving 60.70 RMSE, AIS-LSTM architecture achieved the lowest 59.81 RMSE on Energy dataset. Our proposed model for time-series data outperformed the OLS model on all datasets.

For MAE and MAPE evaluation metrics, our proposed model outperformed the baseline models on air quality datasets by achieving 0.41 MAE and 30.71 MAPE, as shown in Table 7 and 8. We also displayed the actual and predicted target variables on Finance, Air quality and Energy test datasets in Figure 7, 8 and 9, respectively.

Typically, the predictive accuracy of linear regression is weaker than the state-of-the-art models, but after the augmenting by neural network, its predictive performance is

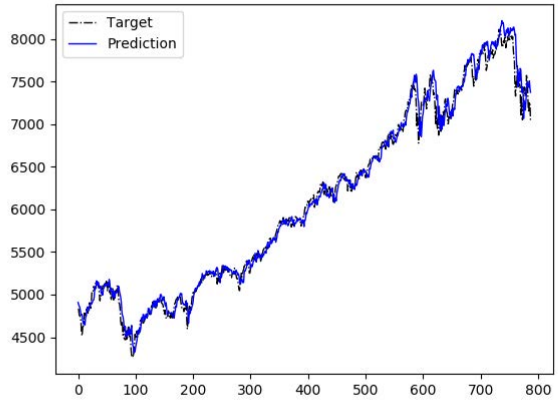

FIGURE 7. Actual and predicted Nasdaq price for finance test set.

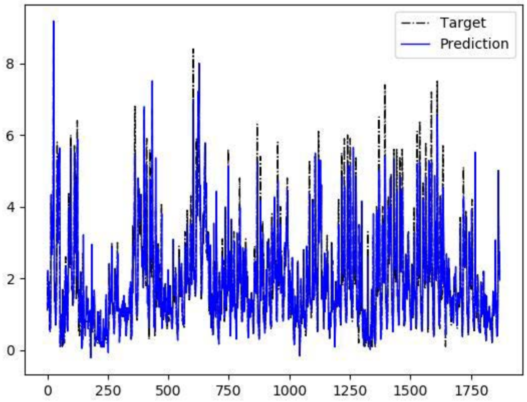

FIGURE 8. Actual and predicted CO(GT) for air quality test set.

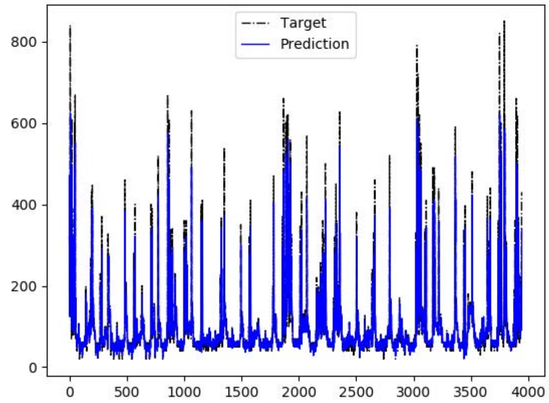

FIGURE 9. Actual and predicted appliances for energy test set.

improved dramatically. The most significant contribution of this work is that we improved the predictive power of linear regression without compromising its interpretability for time-series data.

The experiment results proved that our proposed model can achieve high predictive performance on time-series data, and it can be one of deep learning-based interpretable architecture for time-series forecasting problem. The next part of the experiments will show the interpretability of our proposed model.

TABLE 9. Estimated coefficients of base-learner on finance dataset. ***significantly different from zero at 0.01 level.

Variables	Coefficients	SE error	p-value
DTWEXB_1	0.58***	0.10	0.00
T5YIFR_1	119.03***	10.17	0.00
EPUI_1	-0.10***	0.023	0.00
WILLREITPR_1	0.49***	0.05	0.00
EPUI_2	-0.05**	0.02	0.03
DGS5_5	-20.17***	1.69	0.00
T5YIFR_5	-113.63***	10.26	0.00
WTI_5	-0.22***	0.07	0.00
Nasdaq_5	0.97***	0.003	0.00

E. MODEL INTERPRETABILITY

In this section, we consider 3 real-world datasets, which are the finance dataset [6], the air quality time-series data [40] and the energy use of appliances dataset [41]. We aim to explore the dynamic relationship between the target variable and input variables on these datasets using our proposed model.

1) FINANCE DATASET

We select Nasdaq (stock index) variable as target variable and maximum sequence length (max lag) is equal to 5. The result of our base-learner is reported in Table 9. We can now easily interpret this result; for example: we can say that if the DTWEXB_1 (1 day lag of trade weighted U.S. dollar index) increases by 1 point the previous day, the Nasdaq index will fall by an average of −0.58 points. In addition, figure 10 showed the dynamic relationship between input and output variables on test set. We can see how the impact of the variable DTWEXB_1 on Nasdaq index has changed over time. The impact of DTWEXB_1 was highly volatile in 2015-2016, but since 2017, the impact has increased to 0.65. In addition, the coefficients of input variables are moved depending on the change of the target variable over time. We can also explain other input variables same as DTWEXB_1.

2) AIR QUALITY DATASET

CO(GT) (True hourly averaged concentration CO) variable was chosen as target variable for air quality dataset. The result, as shown in Table 10, we can see the impact of each variable on the CO(GT). For example, CO (GT) is positively correlated with its lag of one hour, and if CO (GT) increases by 1 point an hour earlier, it will be increased 0.82 points an hour later. We also displayed the dynamic relationship between input variables and CO(GT) for 24 hours in Figure 11. We now can see that the impact of CO(GT)'s lag of one hour is equal to 0.85 between 12:00AM and 5:00AM and less than 0.80 between 5:00AM and 10:00AM. Furthermore, the impact of CO(GT)'s lag of one hour is increased from 10:00AM to 5:00PM and started to decrease after 5:00 PM. We can make similar conclusions for other input variables.

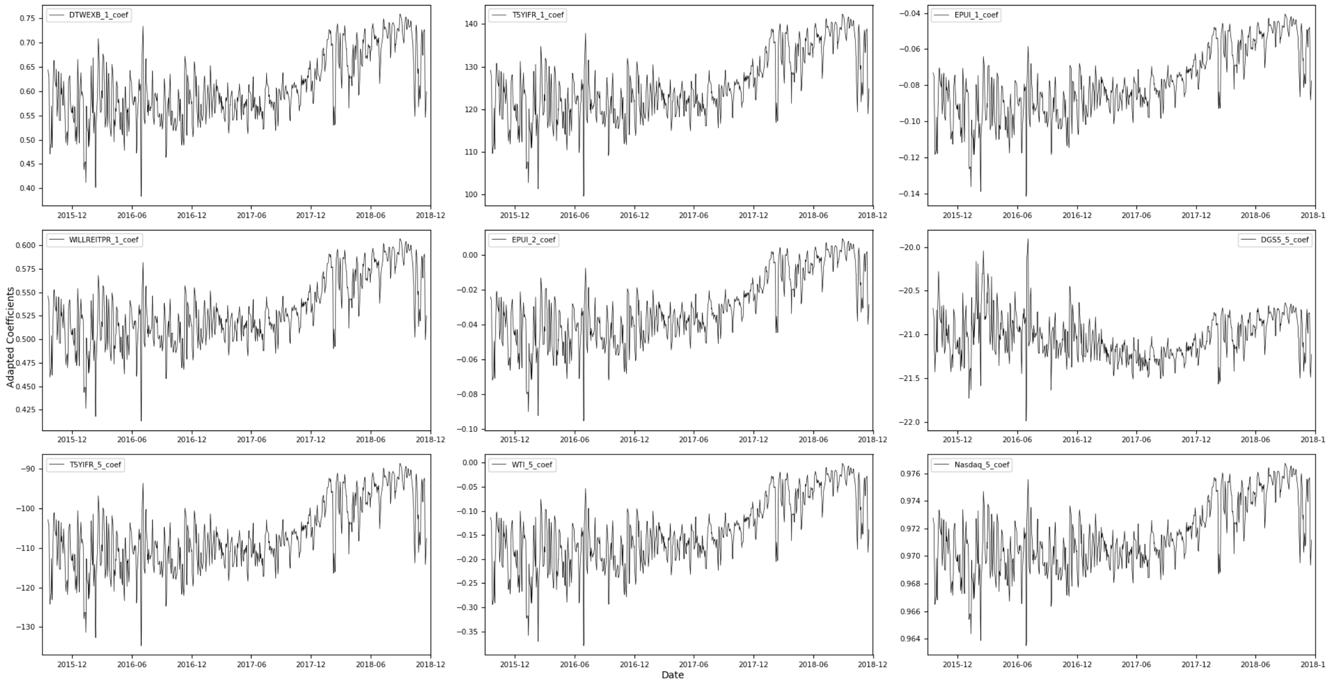

FIGURE 10. The daily local impact of input variables on nasdaq index over time for finance test set. Y-axis represents locally adapted regression coefficients for each input variable.

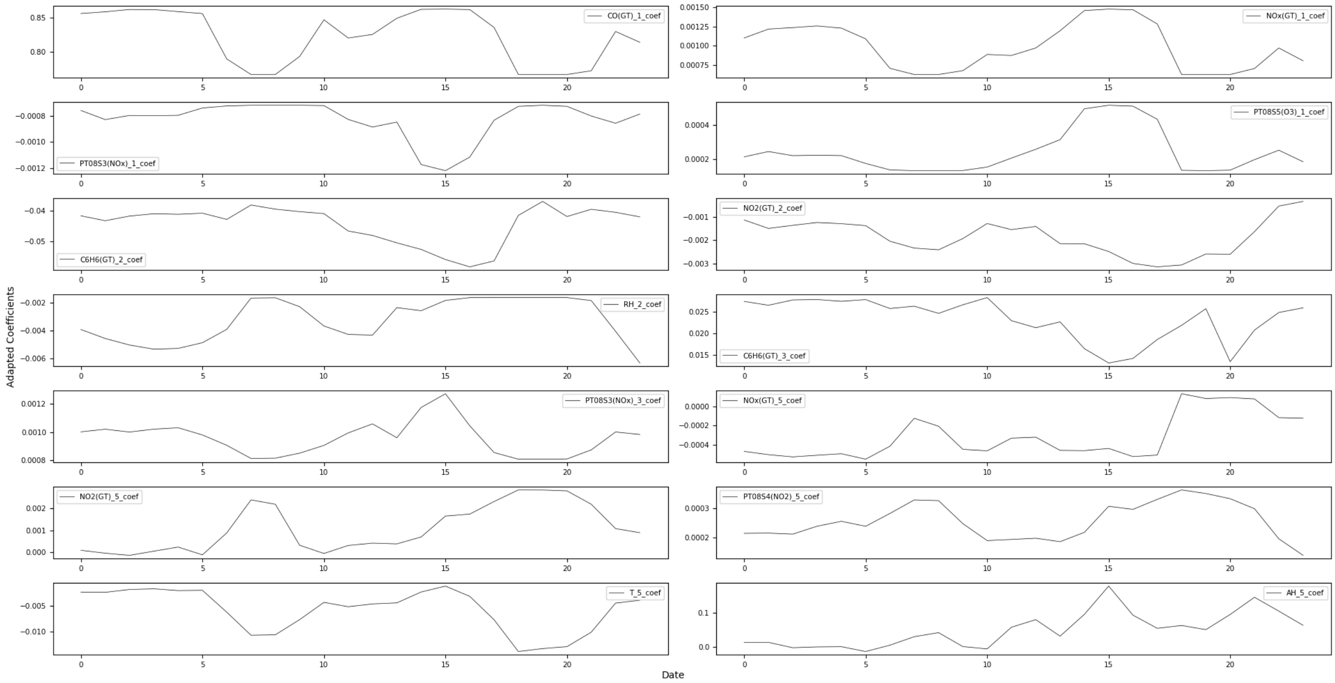

FIGURE 11. The hourly local impact of input variables on CO(GT) for 24 hours for air quality dataset. Y-axis represents locally adapted regression coefficients for each input variable.

In addition, we displayed the hourly local impact of input variables on CO(GT) over time for Air quality dataset in Figure 13.

3) ENERGY DATASET

Appliances energy consumption was chosen as target variable. Table 11 presented the result of linear regression. From the result, we can conclude that if 10 minutes lag of T2 (Temperature in living room area) is increased by 1 celsius, appliances energy consumption will be increased by 1.85Wh. This may be due to the use of air conditioners or fans. In addition, Figure 12 showed the dynamic relationship between input variables and appliances energy consumption for 24 hours. We can see that the impact of 10 minutes lag

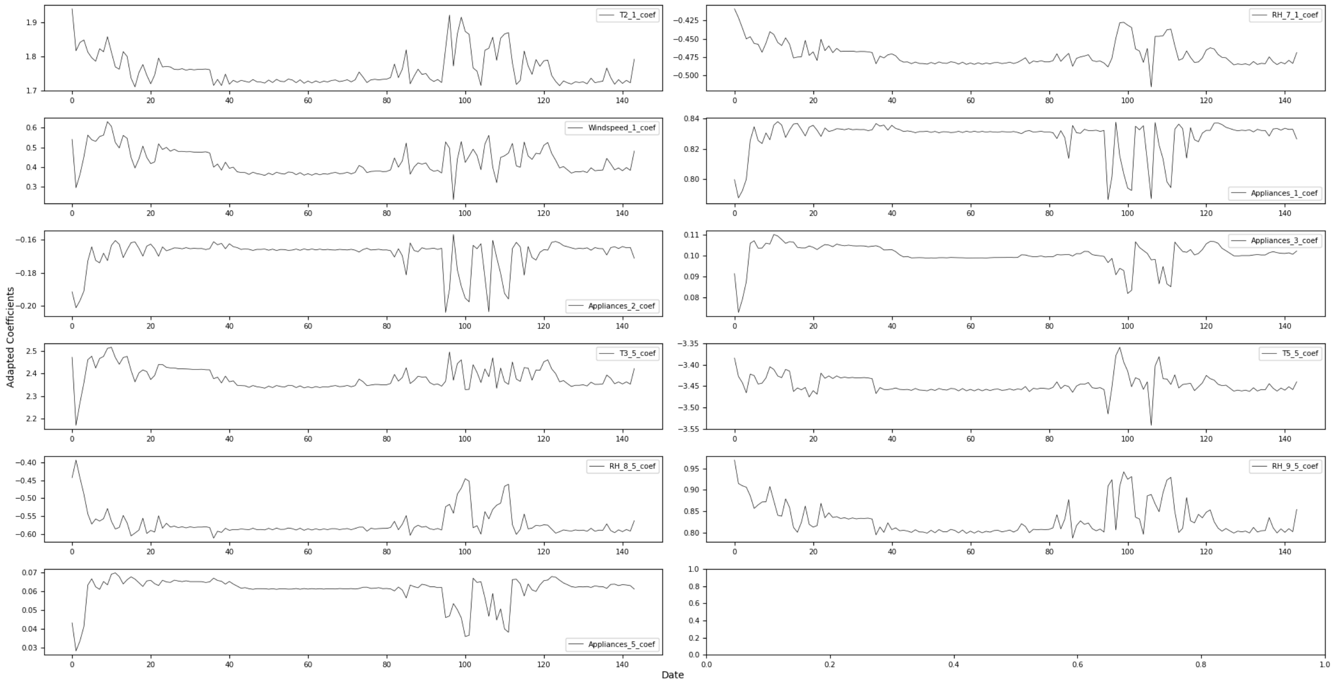

FIGURE 12. **The 10 minutely local impact of input variables on appliances for 24 hours for Energy dataset. Y-axis represents locally adapted regression coefficients for each input variable.**

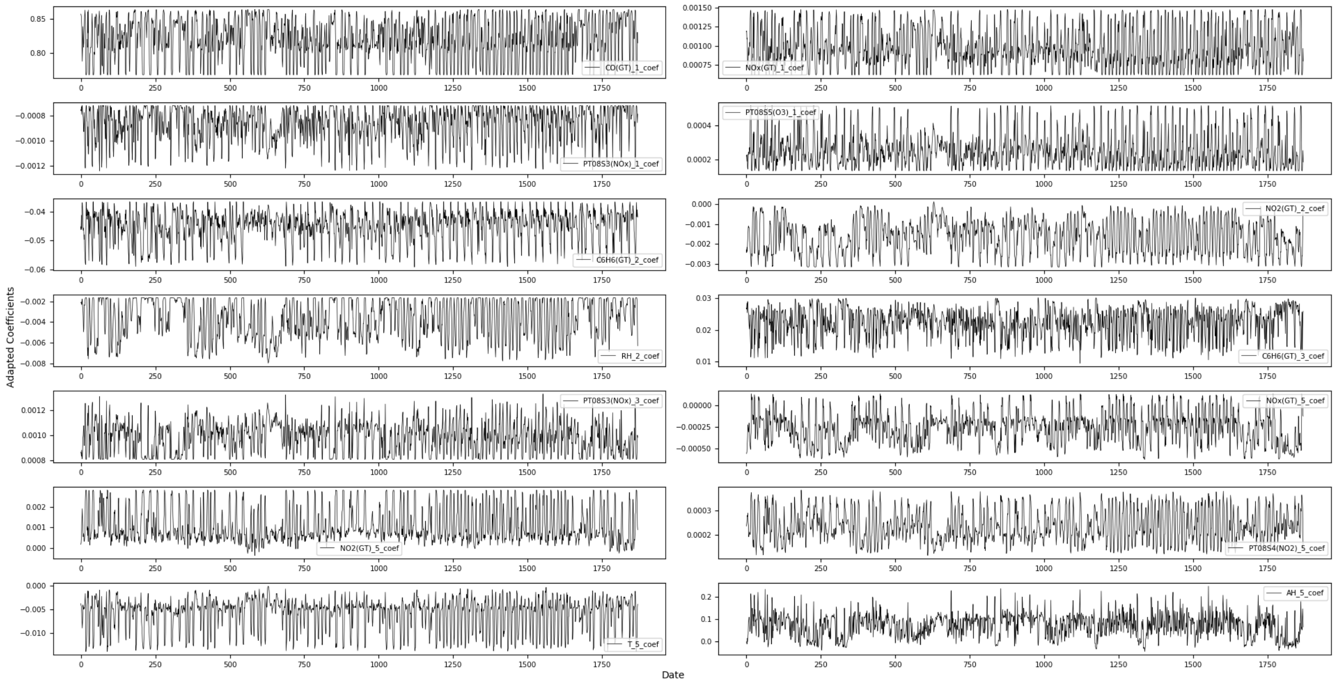

FIGURE 13. **The hourly local impact of input variables on CO(GT) over time for air quality dataset. Y-axis represents locally adapted regression coefficients for each input variable.**

of T2 is constant at night, while during the day its impact is highly changeable. We also displayed the 10 minutely local impact of input variables on Appliances over time for Energy test dataset in Figure 13.

In the end, our experimental results showed that our proposed model can suggest a promising direction for interpretable machine learning that can combine the linear regression and neural networks.

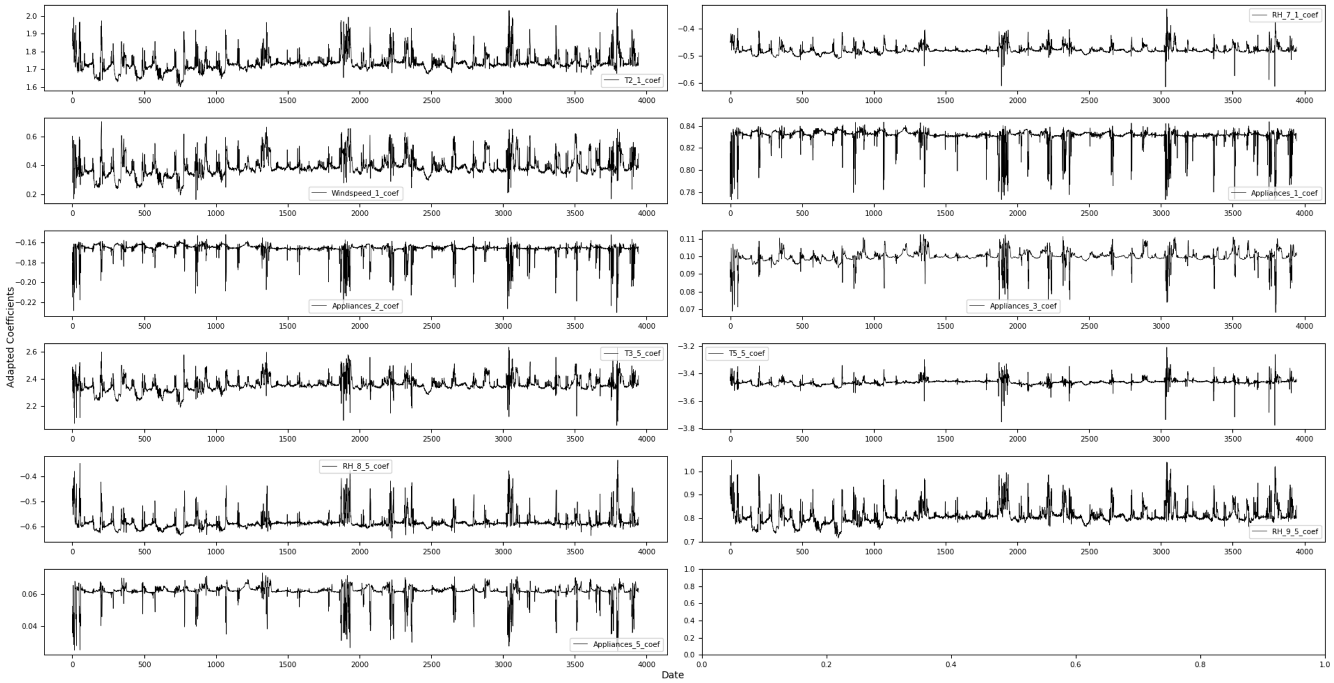

FIGURE 14. The 10 minutely local impact of input variables on appliances over time for energy dataset. Y-axis represents locally adapted regression coefficients for each input variable.

TABLE 10. Estimated coefficients of base-learner on air quality dataset. ***significantly different from zero at 0.01 level.

Variables	Coefficients	SE error	p-value
CO(GT)_1	0.82***	0.01	0.00
NOx(GT)_1	0.001***	0.00	0.00
PT08S3(NOx)_1	-0.001***	0.00	0.00
PT08S5(O3)_1	0.000***	0.00	0.00
C6H6(GT)_2	-0.05***	0.00	0.00
NO2(GT)_2	-0.001***	0.00	0.00
RH_2	-0.005***	0.00	0.00
C6H6(GT)_3	0.02***	0.00	0.00
PT08S3(NOx)_3	0.001***	0.00	0.00
NOx(GT)_5	0.000***	0.00	0.01
NO2(GT)_5	0.001***	0.00	0.01
PT08S4(NO2)_5	0.000***	0.00	0.00
T_5	-0.01***	0.00	0.00
AH_5	0.09***	0.04	0.03

TABLE 11. Estimated coefficients of base-learner on energy dataset. ***significantly different from zero at 0.01 level.

Variables	Coefficients	SE error	p-value
T2_1	1.85***	0.46	0.00
RH_7_1	-0.484**	0.24	0.05
Windspeed_1	0.563**	0.23	0.02
Appliances_1	0.811***	0.01	0.00
Appliances_2	-0.18***	0.01	0.00
Appliances_3	0.100***	0.01	0.00
T3_5	2.443***	0.65	0.00
T5_5	-3.48***	0.74	0.00
RH_8_5	-0.552**	0.24	0.02
RH_9_5	0.885***	0.27	0.00
Appliances_5	0.054***	0.01	0.00

V. CONCLUSION

In this work, we introduced a novel locally adaptive interpretable regression for time series data. We augmented a linear regression by recurrent neural networks that predicts percentile of Gaussian distribution for each regression coefficient to make them adaptable. We conducted an extensive set of experiments to show the interpretability and predictive power of our proposed model. Our model significantly improved the predictive performance of linear regression without comprising its interpretability, and demonstrated the good predictive performance to compare with the state-of-the-art time series models and regression baselines. We also applied our model to finance, air quality and energy time-series data to explain the dynamic relationship between input and output variables. As a result, we displayed how the input variables affects the target variable over time.

A more general AI-based solution to the interpretable issue is to train another model to learn to explain the main predictive model. Our proposed architecture is the first attempt to design interpretable model that has high predictive performance and interpretability for time-series data. We believe that it opens an exciting venue for future work.

REFERENCES

[1] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," 2019, *arXiv:1905.10437*.

[2] A. Koutsoyiannis, *Modern Microeconomics*, 2nd ed. London, U.K.: Springer, 1979.

[3] S. Makridakis and M. Hibon, "The M3-competition: Results, conclusions and implications," *Int. J. Forecasting*, vol. 16, no. 4, pp. 451–476, 2000.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLoS ONE*, vol. 13, no. 3, 2018, Art. no. 0194889.

[5] T. Guo, T. Lin, and N. Antulov-Fantulin, "Exploring interpretable LSTM neural networks over multi-variable data," in *Proc. ICML*, Long Beach, CA, USA, 2019, pp. 2494–2504.

[6] L. Munkhdalai, T. Munkhdalai, K. H. Park, T. Amarbayasgalan, E. Erdenebaatar, H. W. Park, and K. H. Ryu, "An end-to-end adaptive input selection with dynamic weights for forecasting multivariate time series," *IEEE Access*, vol. 7, pp. 99099–99114, 2019.

[7] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020.

[8] G. Dudek, P. Pelka, and S. Smyl, "A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 8, 2021, doi: 10.1109/TNNLS.2020.3046629.

[9] D. F. Andrews, "A robust method for multiple linear regression," *Technometrics*, vol. 16, no. 4, pp. 523–531, 1974.

[10] A. F. Hayes, C. J. Glynn, and M. E. Huge, "Cautions regarding the interpretation of regression coefficients and hypothesis tests in linear models with interactions," *Commun. Methods Measures*, vol. 6, no. 1, pp. 1–11, Jan. 2012.

[11] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 847–855.

[12] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philos Trans. R, Soc. London A*, vol. 379, no. 2194. 2021, Art. no. 20200209.

[13] J. Wang, Z. Peng, X. Wang, C. Li, and J. Wu, "Deep fuzzy cognitive maps for interpretable multivariate time series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 9, pp. 2647–2660, Sep. 2021.

[14] B. Lim, S. Ö. Arák, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021.

[15] L. Li, J. Yan, X. Yang, and Y. Jin, "Learning interpretable deep state space model for probabilistic time series forecasting," 2021, *arXiv:2102.00397*.

[16] A. Ramchandani, C. Fan, and A. Mostafavi, "Deepcovidnet: An interpretable deep learning model for predictive surveillance of COVID-19 using heterogeneous features and their interactions," *IEEE Access*, vol. 8, pp. 159915–159930, 2020.

[17] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. AAAI*, New Orleans, LA, USA, 2018, pp. 4091–4098.

[18] E. Choi, M. Bahadori, J. Kulas, and A. Schuetz, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 3512–3520.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, San Francisco, CA, USA, 2016, pp. 1135–1144.

[20] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. NIPS*, Long Beach, CA, USA, 2017, pp. 4768–4777.

[21] K. Davagdorj, J.-W. Bae, V.-H. Pham, N. Theera-Umpon, and K. H. Ryu, "Explainable artificial intelligence based framework for non-communicable diseases prediction," *IEEE Access*, vol. 9, pp. 123672–123688, 2021.

[22] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.

[23] P. Bracke, A. Datta, C. Jung, and S. Sen. (2019). *Machine Learning Explainability in Finance: An Application to Default Risk Analysis*. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3435104

[24] G. E. Hinton and D. C. Plaut, "Using fast weights to deblur old memories," in *Proc. CogSci*, Hillsdale, NY, USA, 1987, pp. 177–186.

[25] J. Schmidhuber, "A neural network that embeds its own meta-levels," in *Proc. IJCNN*, San Francisco, CA, USA, 1993, pp. 407–412.

[26] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Comput.*, vol. 4, no. 1, pp. 131–139, 1992.

[27] P. Greengard, "The neurobiology of slow synaptic transmission," *Science*, vol. 294, no. 5544, pp. 1024–1030, Nov. 2001.

[28] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. ICML*, Sydney, NSW, Australia, 2017, pp. 2554–2563.

[29] T. Munkhdalai and A. Trischler, "Metalearning with Hebbian fast weights," 2018, *arXiv:1807.05076*.

[30] T. Munkhdalai, X. Yuan, and S. Mehri, "Rapid adaptation with conditionally shifted neurons," in *Proc. ICML*, Stockholm, Sweden, 2018, pp. 3664–3673.

[31] T. Munkhdalai, A. Sordoni, and T. Wang, "Metalearned neural memory," in *Proc. NIPS*, Vancouver, BC, Canada, 2019, pp. 13310–13321.

[32] D. Ha, "Hypernetworks," in *Proc. ICLR*, Toulon, France, 2017, pp. 1–18. [Online]. Available: https://openreview.net/forum?id=rkpACe1lx

[33] L. Munkhdalai, T. Munkhdalai, and K. H. Ryu, "GEV-NN: A deep neural network architecture for class imbalance problem in binary classification," *Knowl Based Syst.*, vol. 194, pp. 105534–105552, Apr. 2020.

[34] L. Munkhdalai, T. Munkhdalai, K. Park, H. Lee, M. Li, and K. Ryu, "Mixture of activation functions with extended min-max normalization for Forex market prediction," *IEEE Access*, vol. 7, pp. 183680–183691, 2019.

[35] A. S. Goldberger, "Best linear unbiased prediction in the generalized linear regression model," *J. Amer. Stat. Assoc.*, vol. 57, no. 298, pp. 369–375, Jun. 1962.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] U. Yolcu, E. Egrioglu, E. Bas, O. C. Yolcu, and A. Z. Dalar, "Probabilistic forecasting, linearity and nonlinearity hypothesis tests with bootstrapped linear and nonlinear artificial neural network," *J. Exp. Theor. Artif. Intell.*, vol. 33, no. 3, pp. 383–404. 2021.

[38] R. Fildes, "A new bootstrapped hybrid artificial neural network approach for time series forecasting," *Comput. Econ.*, vol. 4, pp. 1–29, Nov. 2020.

[39] D. Dheeru and K. T. Efi. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[40] S. De Vito, M. Piga, L. Martinotto, and G. Di Francia, "CO, NO_2 and NOx urban pollution monitoring with on-field calibrated electronic nose by automatic Bayesian regularization," *Sens. Actuators B, Chem.*, vol. 143, no. 1, pp. 182–191, Dec. 2009.

[41] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy Buildings*, vol. 140, pp. 81–97, Apr. 2017.

[42] L. Xiong, X. Chen, T. K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. ICDM*, Sydney, NSW, Australia, 2010, pp. 211–222.

[43] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting With Exponential Smoothing: The State Space Approach*. Berlin, Germany: Springer-Verlag, 2008.

[44] L. Li and B. A. Prakash, "Time series clustering: Complex is simpler!" in *Proc. ICML*, Washington, DC, USA, 2011, pp. 185–192.

[45] S. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," in *Proc. AAAI, Held Virtually*, 2021, vol. 35, no. 8, pp. 6679–6687.

[46] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, p. 623, Oct. 2021.

LKHAGVADORJ MUNKHDALAI (Student Member, IEEE) received the B.Sc. (Econ.) degree from the National University of Mongolia, and the M.Sc. degree from Chungbuk National University, South Korea, where he is currently pursuing the Ph.D. degree in computer science. He has been involved in many research projects about deep learning-based time series forecasting, such as infectious disease forecasting, short-term export & import forecasting, and demand forecasting for postal delivery service. His research interests include the development of an adaptive deep learning model for forecasting time series, adaptive regression models, deep learning along with their applications in domains, such as financial, medical informatics, and public health informatics.

TSENDSUREN MUNKHDALAI received the bachelor's degree from the National University of Mongolia, Ulaanbaatar, Mongolia, and the M.Sc. and Ph.D. degrees from the Department of Computer Science, Chungbuk National University, South Korea. He is currently a Research Scientist at Google, USA. Before Google, he was a Researcher at Microsoft Research Montreal. He spent two years with the University of Massachusetts as a Postdoctoral Researcher sitting right next to the Neurology Department. His research interests include meta-learning, memory and attention systems, rapid and temporal adaptations and feedbacks in artificial neural nets for language, and text understanding.

VAN-HUY PHAM received the M.S. degree in computer science from the University of Sciences, Ho Chi Minh City, Vietnam, in 2007, and the Ph.D. degree in computer science from Ulsan University, South Korea, in 2015. Since 2015, he has been a Lecturer and a Researcher with the Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City. His research interests include artificial intelligence, image processing, computer vision, and deep learning applications.

MEIJING LI received the B.S. degree in computer science from Dalian University, Dalian, China, in 2007, and the M.S. degree in bio information technology and the Ph.D. degree in computer science from Chungbuk National University, Cheongju, South Korea, in 2010 and 2015, respectively. She worked with Chungbuk National University as a Postdoctoral Researcher. She is currently an Assistant Professor with the College of Information Engineering, Shanghai Maritime University, Shanghai, China. Her research interests include data mining, information retrieval, database systems, bioinformatics, and biomedicine.

KEUN HO RYU (Life Member, IEEE) received the Ph.D. degree in computer science and engineering from Yonsei University, South Korea, in 1988. He has worked with the Reserve Officers' Training Corps (ROTC), Korean Army. He was with The University of Arizona, Tucson, AZ, USA, as a Postdoctoral Researcher and a Research Scientist. He was also with the Electronics and Telecommunications Research Institute, South Korea, as a Senior Researcher. He is currently a Professor with the Faculty of Information Technology, Ton Duc Thang University, Vietnam, and an Emeritus and the Endowed Chair Researcher with Chungbuk National University, South Korea, and also an Adjunct Professor with Chiang Mai University, Thailand. He is also an Honorary Doctorate of the National University of Mongolia. He has been not only the Director of the Database and Bioinformatics Laboratory, South Korea, since 1986, but also the Director of the Data Science Laboratory, Research Group, Vietnam, since March 2019. He is the Former Vice-President of the Personalized Tumor Engineering Research Center. He has published over 1000 refereed technical articles in various journals and international conferences, in addition to authoring a number of books. His research interests include databases, spatiotemporal databases, big data analysis, data mining, deep learning, biomedical informatics, and bioinformatics. He has been a member of the ACM, since 1983. He has served on numerous program committees, including roles as the Demonstration Co-Chair of the VLDB, as the Panel and Tutorial Co-Chair of the APWeb, and as the FITAT General Co-Chair. In 2008, he founded the FITAT International Group for providing a professional community the opportunities for publications, knowledge exchange, teaming, and cooperation.

NIPON THEERA-UMPON (Senior Member, IEEE) received the B.Eng. degree (Hons.) in electrical engineering from Chiang Mai University, the M.S. degree in electrical engineering from the University of Southern California, and the Ph.D. degree in electrical engineering from the University of Missouri, Columbia. He has served as an editor, a reviewer, the general chair, the technical chair, and a committee member for several journals and conferences. He has been bestowed several royal decorations and won several awards. He was the Associate Dean of Engineering and the Chairperson of graduate study in electrical engineering and graduate study in biomedical engineering. Since 1993, he has been with the Department of Electrical Engineering, Chiang Mai University, where he is currently working as the Director of the Biomedical Engineering Institute. He has served as the Vice President of the Thai Engineering in Medicine and Biology Society, and the Vice President of the Korea Convergence Society. He has published more than 190 full research papers in international refereed publications. His research interests include pattern recognition, machine learning, artificial intelligence, digital image processing, neural networks, fuzzy sets and systems, big data analysis, data mining, medical signal, and image processing. He is a member of Thai Robotics Society, Biomedical Engineering Society of Thailand, Council of Engineers in Thailand. He is also a member of IEEE-IES Technical Committee on Human Factors.