IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 5, 2020, accepted October 10, 2020, date of publication October 14, 2020, date of current version October 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3030938

Deep Learning-Based Demand Forecasting for Korean Postal Delivery Service

LKHAGVADORJ MUNKHDALAI[1], (Student Member, IEEE), KWANG HO PARK[1], (Graduate Student Member, IEEE), ERDENEBILEG BATBAATAR[1], NIPON THEERA-UMPON[2,3], (Senior Member, IEEE), AND KEUN HO RYU[2,3,4], (Life Member, IEEE)

[1]Database/Bionformatics Laboratory, School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, South Korea
[2]Biomedical Engineering Institute, Chiang Mai University, Chiang Mai 50200, Thailand
[3]Department of Electrical Engineering, Faculty of Engineering, Chiang Mai University, Chiang Mai 50200, Thailand
[4]School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, South Korea

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

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, and Future Planning under Grant 2019K2A9A2A06020672 and Grant 2020R1A2B5B02001717. The work of Lkhagvadorj Munkhdalai and Kwang Ho Park was supported in part by the Biomedical Engineering Institute, Chiang Mai University.

ABSTRACT Proper demand forecasting for postal delivery service can be used for optimal logistic management, staff scheduling and topology planning. More especially, during special holidays, such as the Lunar New Year and the Chuseok (Mid-autumn day), the demand for delivery service increases sharply in South Korea. It makes a challenge to forecast demand to provide a normal delivery schedule for the Korean mail center. To address this problem, we propose a novel deep learning model equipped with selection and update layers (MLP-SUL) to achieve high predictive performance. The proposed model consists of three main parts: the first part of the model learns to generate context-dependent weights to decide which input feed to the next layer; the second part updates the weighted input to prepare encoded input, and the third part is a prediction layer that consists of a linear layer. A linear layer takes encoded input for forecasting demand. We also introduce a special data preprocessing step for our task that requires long-term forecasting. The experimental results show that our proposed deep learning model outperforms state-of-the-art baselines on Korean mail center datasets.

INDEX TERMS Time series forecasting, deep learning, postal delivery service.

I. INTRODUCTION

Accurate postal demand forecasting is a key role in deciding how to allocate resources efficiently for its distribution. In particular, during special holidays, such as the Lunar New Year and the Chuseok (Mid-autumn day) in South Korea, the demand for delivery service extremely increases. At this point, in order to provide the normal operation of the postal delivery service, one of the most important issues is to accurately predict the demand [1]. By accurately forecasting long-term demand could not only ensure the normal operation of the Korean Mail center by preparing resources in advance for the special holidays but also reduce operating costs.

On the other hand, this is a time series forecasting problem that extracts the behavior of time-dependent data and predicts future values depending on the history of observed values [2]. There have been many approaches proposed in the field of time series forecasting such as exponential smoothing [3], autoregressive integrated moving average – ARIMA [4], autoregressive conditionally heteroscedastic – ARCH [5] and generalized ARCH – GARCH [6]. Recently, deep learning-based models for time series forecasting have become more popular and demonstrate better performance [7]–[12]. This work proposes a deep learning model to forecast the demand for postal delivery service to obtain better predictive accuracy. However, our task is slightly different from the traditional time series forecasting, which is to forecast the long-term demand for postal delivery service during special holidays. In other words, our work deals with a certain issue that predicting the sudden change in volume occurring in a particular period rather than predicting the usual volume. Thus, our forecasted values could be used as input again during the

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

forecasting process. This has led to a tendency to significantly increase the error (cumulative error) of future predictive values, because the demand for delivery service extremely increases during special holidays, so we introduce an additional data preprocessing step. We prepare input using only historical data to predict future values, therefore, our model should capture the long-term dependencies in time series to reduce multi-step forecasting error. To address this issue, we introduce selection and update layers (MLP-SUL) similar to the gates of the Gated Recurrent Unit – GRU network to capture the long-term dependencies of input variables [13]. Our proposed model consists of three main components: the first component generates the weighted input variables to capture the long-term dependencies and the second component updates the weighted input variables to feed the third component for forecasting target value. We apply a multilayer perceptron – MLP architecture to construct our deep learning model. The generated weights in our proposed model can be seen as fast-weights, which is to utilize one neural network to produce the parameters of another neural network [14]. Neurobiologically, synapses are dynamic at different time scales, and fast weights play this role in the artificial neural network [15]. By using fast-weights, neural networks might promote from variables that change slower than activities but much faster than the slow weights [16]. These weights have been used for rapid learning [17] and implementation of recurrent neural nets [18]. Recently, T. Munkhdalai & H. Yu [19] proposed a meta-learning model named MetaNet using fast weights for rapid generalization across tasks. In addition, in our previous work, we proposed a novel model equipped with an input selection mechanism by using a fast-weights generation neural network that consists of a time distributed layer with a softmax activation function for time series forecasting task [2]. Unlike previous work, in this work, we apply two neural networks with totally different roles that select and update the input variables to obtain efficient encoded input for the final prediction. We assume that the additional neural networks increase the generalization capacity of our model as well as improve the predictive accuracy [20]. On the contrary, increasing capacity of the model allows for overfitting [21]. Surprisingly, Neyshabur *et al.* [22] notices that increasing the capacity of the model only helps in improving the predictive accuracy, even when the neural networks are trained without any explicit regularization. We also demonstrate that additional neural networks improve predictive accuracy by performing an ablation experiment.

For forecasting mail demand, multivariate linear regression has been widely utilized [23], [24]. Although machine learning models, especially recurrent neural networks (RNNs), have been widely used in time series forecasting [9]–[12], [25], deep learning models, to the best of our knowledge, have not been applied to predict the demand for delivery services. However, multivariate time series prediction has been developed into a significant and active research area, in which forecasting approaches have been widely studied. For example, recent studies have also introduced hybrid approaches by classical time series forecasting methods such as autoregression – AR and exponential smoothing with RNNs [26]–[28].

Purpose of our work is also related to extreme event forecasting which another line is of related research is. Extreme event prediction is an important topic for estimating peak electricity demand, traffic jam severity, and surge pricing for ride-sharing and other applications [29]. To address the extreme event forecasting problem, univariate time-series and deep learning approaches have been proposed [29]–[32]. More recently, UBER, which is a peer-to-peer ridesharing company in the USA, proposed a Long-short term memory – LSTM based extreme event prediction model, named Uber-LSTM model [32]. Because UBER has a problem similar problem with the Korean mail center, which is during some special events such as Christmas, thanksgiving days, etc. demand for UBER drivers is extremely increased. This model has achieved an average 14.09% improvement over the multi-layer LSTM.

To demonstrate the superiority of our proposed model, we choose standard multivariate linear regression, MLP, LSTM and UBER-LSTM models as a benchmark. In the experimental part, the selected deep learning models are evaluated on 24 mail centers' dataset. The mean absolute error – MAE, mean absolute percentage error – MAPE and the symmetric mean absolute percentage error – sMAPE metrics are utilized to measure model performances. The results show that our proposed model MLP-SUL demonstrates promising results. In summary, the main contributions of this work are as follows:

- The problem we have considered in this work is different from the usual time series forecasting problem. Based on deep learning approach, we attempt to perform the long-term demand forecasting the mail volume, which is dramatically increased during special holidays.
- We propose MLP-SUL, a novel deep learning-based time series forecasting model consisting of the selection and updating layers. Experimental results show that our model achieves state-of-the-art performance.
- We extensively evaluate MLP-SUL on 24 mail centers' datasets. The experimental results show that MLP-SUL achieves state-of-the-art performance in reducing the forecasting error, against four baselines.

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

II. MATERIALS AND METHODS

A. DATA

1) KOREAN MAIL CENTER DATASET

Korea Post is the domestic postal service of South Korea, under the authority of the Ministry of Science and ICT. Korea Post is in charge of postal service (registered mail, customer pickup, P.O. Box, sales of local products by mail

TABLE 1. **The summary of Korean mail center datasets.**

Mail center	#Observation	Mean value	Min value	Max value
Mail Center-1	1251	74,179.2	0.0	187,868.0
Mail Center-2	1265	13,852.6	1.0	50,998.0
Mail Center-3	1066	6,013.0	2.0	26,404.0
Mail Center-4	1248	39,955.4	3.0	181,361.0
Mail Center-5	1211	20,705.6	0.0	60,750.0
Mail Center-6	1030	13,784.1	15.0	38,514.0
Mail Center-7	1228	11,448.2	2.0	58,782.0
Mail Center-8	1238	19,408.6	1.0	73,858.0
Mail Center-9	1241	25,373.5	1.0	78,890.0
Mail Center-10	1244	31,154.9	2.0	104,446.0
Mail Center-11	1243	25,427.6	5.0	100,991.0
Mail Center-12	1220	34,153.3	1.0	121,732.0
Mail Center-13	1018	15,586.6	40.0	68,093.0
Mail Center-14	1021	10,439.8	0.0	39,546.0
Mail Center-15	1133	25,041.0	82.0	77,634.0
Mail Center-16	1214	47,528.7	8.0	154,460.0
Mail Center-17	1040	10,651.1	1.0	35,732.0
Mail Center-18	1013	11,139.9	18.0	48,775.0
Mail Center-19	1044	16,350.3	145.0	45,859.0
Mail Center-20	1221	11,022.0	1.0	71,591.0
Mail Center-21	1233	42,028.1	24.0	139,314.0
Mail Center-22	1020	11,642.1	2.0	60,337.0
Mail Center-23	1084	11,092.8	20.0	58,183.0
Mail Center-24	1438	133,436.6	2582.0	445,392.0

order and postal errand service), postal banking (handling and delivering mail and parcels), and insurance services. A total of 24 mail centers' datasets from different cities are used in this work as summarized in Table 1. Postal time series data are generated from the 1st of September in 2015 to the 17th of September in 2019, and the number of observations varies depending on the mail center.

Figure 1 shows the postal time series data of Mail Center-24 from the 1st of January to the 31st of December in 2018 as an example. We can observe that the demand for postal services has been extremely increased during the Lunar New Year and Chuseok. Our task is to forecast the demand for postal services during a period of extreme events.

2) DATA PREPROCESSING

In order to perform accurate forecasting, we extract some additional features [33] such as calendar and historical variables as described in Table 2. Because the target period we predict is to cover a public holiday in Korea, we have considered that including some holidays as an explanatory variable would improve forecasting performance. In addition, since our prediction period is a special case, we have included the number of packages of the previous Lunar New Year and the Chuseok periods in the explanatory variables. For calendar data, we generate dummy variables for special holidays, one and two days before and after holidays, beginning, middle, and end of the month, and weekend. The holiday (special holidays) variable takes values of 1 and 0, where a value of 1 when the day is a public holiday and a value of 0 when the day is not a public holiday. Moreover, the demand for postal delivery service tends to increase before and after the holiday in South Korea, therefore we take 1 and 2 days before and after the holiday as dummy variables. If the day is one or two days before or after the public holiday, the value of variable is 1, in other cases, the value of variable is equal to 0. We have also added seasonal variables such as weekdays, beginning, middle, and end of the month, and weekend. For the beginning, middle, and end of the month, we divided the month into three parts: up to 10th, from 10th to 20th, and after the 20th. If the date is within the 10th of the month, the value of the beginning of the month variable is equal to 1, in other cases 0. The other two variables, the middle and end of the month variables, are created by the same procedure of the beginning of the month variable.

We also generate some numerical variables such as number of days in a holiday, number of working days in a holiday. For the number of days in a holiday, we have counted the number of days during the public holiday. In addition, we have created a variable by counting the number of working days during the public holiday because sometimes the public holidays coincided with weekends.

One more dummy variable is created where a value of 1 when the day is both a public holiday and weekend.

For historical data, we take a lag of 7 to 20 weeks, the moving average values of the previous weeks between 7 and 9, 10 and 12, 13 and 15, and 16 and 18, the values of the previous Lunar New Year and Chuseok. For lag variables, we have taken the number of packages from 7 to 20 weeks ago. For moving average variables, we have created variables by calculating the average number of packages from the previous 7 to 9, from 10 to 12, from 13 to 15 and from 16 to 18 weeks.

Furthermore, for our target forecasting period only, we have generated additional variables that the number of packages of previous lunar new year and Chuseok.

As shown in Table 2, the minimum lag of the input variable is equal to 7 weeks, therefore, we can easily predict 7 weeks in the future. Using this data preprocessing helps to avoid re-usage of the predicted value as input and prevents the cumulative errors. We show the example of input and output for neural networks using historical values in Figure 2. According to our data preprocessing, the output-1 is Monday, so we took the number of packages of each Monday from the previous 7 to 20 weeks as input. Similarly, since output-49 is Wednesday, we used the values of each Wednesday from the previous 7 to 20 weeks as input.

B. MULTILAYER PERCEPTRON WITH SELECTION AND UPDATE LAYERS

Figure 3 demonstrates an overview of the proposed deep learning model named MLP-SUL that consists of three main MLPs. The first component named Selection MLP generates weights that can decide which input variable to be fed into the prediction MLP to obtain a predicted target value. The second Update MLP produces additional vectors that update weighted input vectors. We use a skip connection with the multiplication layer between initial input and generated weights from Selection MLP. We also add produced vectors from Update MLP to weighted input to obtain encoded input vectors to feed the prediction MLP for forecasting target value. The task of additional neural networks is to take the same input and encode it into vectors with different roles.

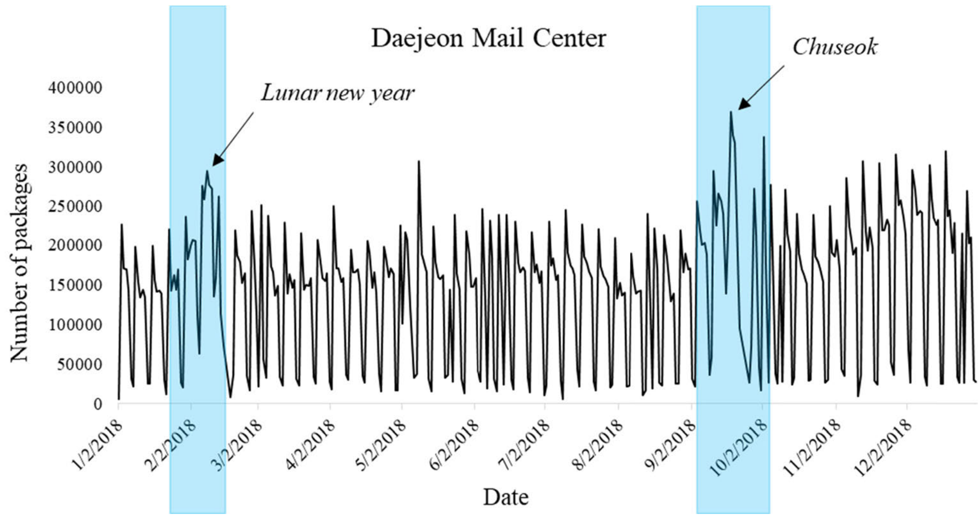

FIGURE 1. Example of mail center dataset (Mail Center-24, from 1st of January to 31st of December in 2018).

TABLE 2. Description of input variables.

Group	Input variables	Description
Historical variables	lag_7_weeks	Number of packages 49 days ago
	lag_8_weeks	Number of packages 56 days ago
	lag_9_weeks	Number of packages 63 days ago
	lag_10_weeks	Number of packages 70 days ago
	lag_11_weeks	Number of packages 77 days ago
	lag_12_weeks	Number of packages 84 days ago
	lag_13_weeks	Number of packages 91 days ago
	lag_14_weeks	Number of packages 98 days ago
	lag_15_weeks	Number of packages 105 days ago
	lag_16_weeks	Number of packages 112 days ago
	lag_17_weeks	Number of packages 119 days ago
	lag_18_weeks	Number of packages 126 days ago
	lag_19_weeks	Number of packages 133 days ago
	lag_20_weeks	Number of packages 140 days ago
Moving average	rmean_week_7_9	An average number of packages between the previous 49 and 63 days.
	rmean_week_10_12	An average number of packages between the previous 63 and 84 days.
	rmean_week_13_15	An average number of packages between the previous 84 and 105 days.
	rmean_week_16_18	An average number of packages between the previous 105 and 136 days.
Events	prev_lunar	The number of packages during the previous lunar new year. (only for our target period)
	prev_chuseok	The number of packages during the previous Chuseok. (only for our target period)
Calendar variables	holiday	Special holiday (If a day is holiday)
	prev_day_holiday	A day before holiday (If the day is a day before holiday)
	prev_2_days_holiday	Two days before holiday (If the day is 2 days before holiday)
	after_day_holiday	A day after holiday (If the day is a day after holiday)
	after_2_days_holiday	Two days after holiday (If the day is 2 days after holiday)
	End_of_month	End of month (If date is after the 20th of month)
	Middle_of_month	Middle of month (If date is between the 10 and 20th of month)
	Begin_of_month	Beginning of month (If date is within the 10th of month)
	Lunar_new_year	Lunar new year's dummy (If the day is a day during Lunar new year)
	Chuseok	Chuseok's dummy (If the day is a day during Chuseok)
	Holiday_and_weekend	Both holiday and weekend (If the day is both a public holiday and weekend)
	Weekend	Weekend's dummy (If the day is Saturday or Sunday)
	Holiday_period	Number of days in a holiday (The number of days during the public holiday)
	Number_of_working_days	Number of working days in a week (The number of working days during the public holiday)

Therefore, we assume that it is more capable to train them using separated neural networks. Because the role of selection neural network is similar to that of the attention layer, while the role of the update neural network is to process additional useful information from the input.

The Selection MLP: the goal of Selection MLP is to produce the weights that govern the influence of each input variable to capture long-term dependencies. Since we only prepare historical values as input in the data preprocessing step, it is crucial to determine important variables

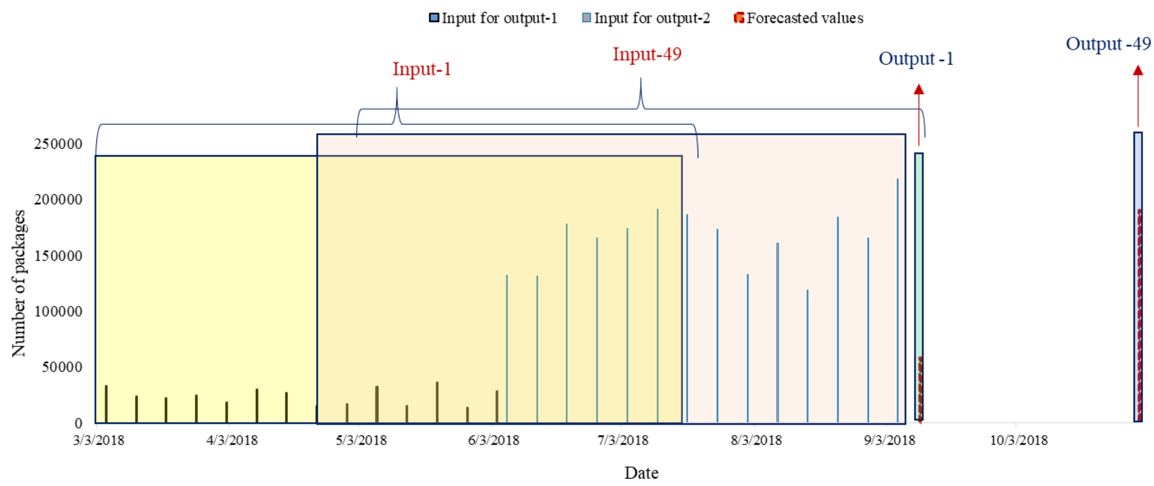

FIGURE 2. **Example of input and output for neural networks using historical values.**

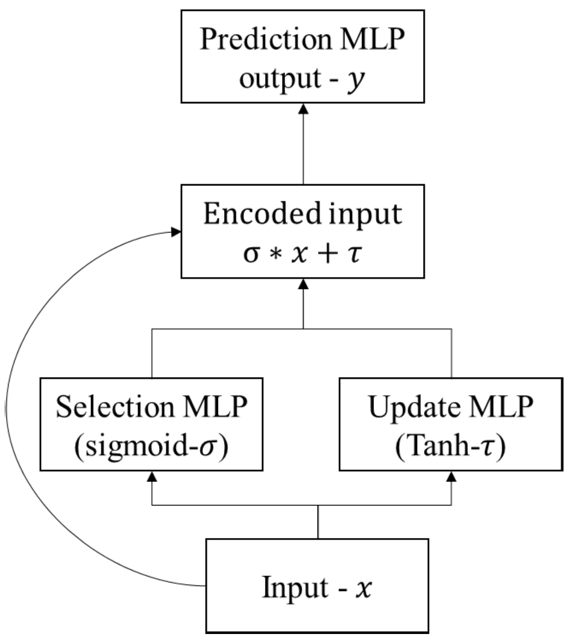

FIGURE 3. **Overview of the proposed deep learning architecture. The output of the selection network (σ) is the sigmoid function that generates importance weights between 0 and 1. Therefore, those weights can decide the input variables. The output of Update MLP is a hyperbolic tangent function that produces the vectors between -1 and 1, so it can update the input variables. We then add the update vectors to the initial input multiplied by the sigmoid output using a skip connection to obtain the encoded input.**

as well as their lags. The selection MLP can solve this issue.

Let $Y_t = (y_1, y_2, \ldots y_T)$ denote a target variable. Assuming we use n different variables such as calendar, lags and moving average variables as shown in Table 2, the i-th input of T timestamps can be represented by a sequence of $T^{(i)}$ tuples $(x_j^{(i)}) \in R^n, i = 1, \ldots, T$ and $j = 1, 2, \ldots, n$. We use MLP to find weights (ν) in the selection network as follows:

$$\sigma = sigmoid(MLP_{selection}(x; \theta)) \tag{1}$$

where θ denotes the weight parameters for selection network. Output of this layer is a sigmoid activation. Therefore, the generated vectors from selection network is between 0 and 1, which are context-dependent importance weights for variables. If the weight that corresponds to one of the input variables is close to 0, the variable or lag multiplied by the weight cannot affect the predicted output. In other words, our selection MLP network similar to the "forget gate layer" in LSTM, it decides which information from the input data to forget [34]. In addition, the significance of using adaptive weights is that the importance of some variables may change over time in a changing environment, at which point the adaptive weights could help to rapidly readjust input variables [35].

The update MLP: Although we retrieved important information from the input data using the selection MLP network, we did not produce additional new information. In order to generate new information, we add an update MLP network into our proposed architecture. We set a hyperbolic tangent activation function as an output of the update MLP. This layer produces the vectors between -1 and 1 that can encode the input variables between -1 and 2 because our data is normalized within the range of 0 and 1.

$$\tau = \tanh(MLP_{update}(x; \omega)) \tag{2}$$

where ω denotes the weight parameters for update MLP.

The update MLP network is similar to the "update gate layer" in GRU network, it generates new information to update the weighted input [13].

The encoding layer: this layer takes the importance weights generated by selection network, update vectors from update network and input variables. We use a skip connection to encode our input variables as follows:

$$x' = \sigma \odot x + \tau \tag{3}$$

where $\odot$ denotes an element-wise multiplication, $\boldsymbol{\sigma}[\mathbf{0}, \mathbf{1}]$ denotes importance weights and $\boldsymbol{\tau} \in [-\mathbf{1}, \mathbf{1}]$ denotes update weights.

The prediction network: this network consists of a linear layer and takes encoded input $\boldsymbol{x}' \in \boldsymbol{R}^{\boldsymbol{n}}$ to predict the target variable $\boldsymbol{y}$.

$$y = MLP_{prediction}x'; v \tag{4}$$

where v denotes the weight parameters of prediction network. In order to implement an end-to-end learning framework, our encoding and prediction networks are jointly optimized to determine their weight parameters. We use the mean squared error – MSE function that measures the average of the squares of difference between actual and predicted values. Once we implement end-to-end learning, the output weights or probabilities (σ, τ) are adaptively adjusted depending on the prediction results at each time step. Because these weights are generated by the selection and update neural networks for every input and these neural networks learn how to intelligently encode the input data for the final prediction.

III. EXPERIMENTAL RESULT

A. BASELINE MODELS

The baseline models for our task include:

Regression refers a linear regression, which has been the most widely used method for regression task [36]. The regression model estimates coefficients that provide the best linear fits between the dependent and independent variables.

MLP - A standard MLP is a core architecture of neural networks, which has been similarly developed to the human brain [37]. There are three layers with different roles named input, hidden, and output. Each layer consists of a given number of neurons with the activation function and neurons in the neighbor layer are connected by weight parameters. The weight parameters of MLP will be estimated by optimizing loss function using a stochastic gradient descent method with a back-propagation algorithm.

LSTM - Long short-term memory network was proposed as a solution to the vanishing gradient problem [34]. This network has internal mechanisms named gates that can control the flow of information. In other words, LSTMs help to solve long-term dependencies by extending their memory cells using gates to control information flow. The memory cell consists of three gates – input, forget and output gate. Those 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.

UBER-LSTM- UBER is that peer-to-peer ridesharing company in the US. UBER faces similar problem with Korean mail center, which is during special events such as Christmas, Thankgiving Days, etc. demand of UBER driver is extremely increased. Therefore, they proposed UBER-LSTM model to forecast extreme demands [32]. We directly applied UBER-LSTM model to our task. UBER-LSTM model consists of two main components. The first component is LSTM auto-encoder that helps to generate meaningful new features for extreme events as well as it decreases uncertainty of input variables. The second component is LSTM forecaster that predicts the target value using concatenated new features and initial inputs. The main advantage of UBER-LSTM model is an auto-encoder that has automatic meaningful feature extraction abilities.

TABLE 3. Hyperparameter space.

Neural net	Hyperparameter space
MLP	Number of hidden layers = [1, 2] Number of nodes = {[16], [64], [256], [512], [64, 16], [256, 64], [512, 256]} Activation function = {'linear', 'sigmoid', 'tanh'} Batch size = {8, 16} Learning rate = {0.01, 0.001}
LSTM	Number of hidden layers = [1, 2] Number of nodes = {[16], [32], [64], [64, 32], [32, 16]} Activation function = {'linear', 'sigmoid', 'tanh'} Batch size = {8, 16} Learning rate = {0.01, 0.001}

B. HYPERPARAMETER OPTIMIZATION

The hyperparameters of neural networks are determined by using a grid search algorithm based on the parameter space as shown in Table 3. We set most parameters as the same for MLP and LSTM networks but we chose a smaller number of neurons for LSTM because of computation time.

According to the combination of hyperparameters, 48 models were trained for the MLP network and 60 models for the LSTM network based on total number of packages for all mail centers and the combination of hyperparameters with the best performance was selected as shown in Figure 4 and 5. We have shown the results of the first 10 models with the best performance.

For MLP architecture, batch size to 16, learning rate to 0.001, two hidden layers with 256 and 64 neurons, and a hyperbolic tangent activation function were selected as shown in Figure 6. In the case of the LSTM architecture, a single hidden layer with 16 neurons showed the best performance, which is much smaller than in the MLP architecture.

We set the maximum epoch number of 3000 and apply an early stopping algorithm for finding the optimal epoch number.

For evaluation metrics, MAE, MAPE and sMAPE are used to evaluate models performances.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{5}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{6}$$

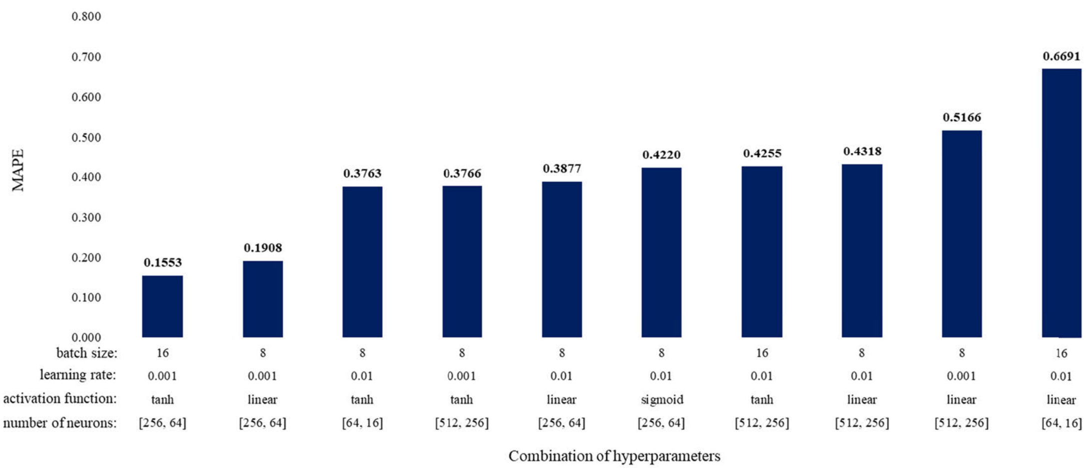

FIGURE 4. Hyperparameter optimization result on MLP architecture.

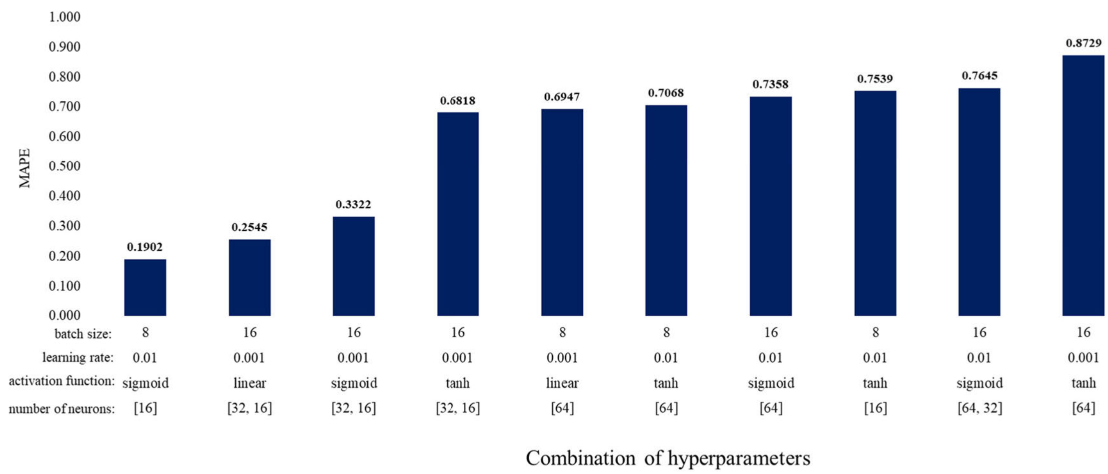

FIGURE 5. Hyperparameter optimization result on LSTM architecture.

$$sMAPE = = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{(\hat{y}_i + y_i)} \tag{7}$$

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

C. FORECASTING RESULT

To obtain the best model for each mail center data, we compare three deep learning-based baselines and linear regression to our proposed model.

In this work, our goal is to build a model in which the MAPE value is less than 0.15.

We choose training set period from the 1st of September in 2015 to the 31st of July in 2019 and test set period from the 1st of August in 2019 to the 17th of September in 2019 (7 weeks), which is the period of Chuseok in 2019.

Table 4, 5 and 6 show the forecasting performances of the deep learning models for each mail center. The forecasting results show that our proposed MLP-SUL model outperforms other benchmark models on most of the mail centers' datasets.

The UBER-LSTM and LSTM models did not achieve good performances on most datasets, which may be due to data preprocessing. Because when we prepared the input data, we use the historical data as input for each time step and set the time step of the RNN models by 1.

Although we traditionally prepared time series data and performed long-term forecasting for the UBER-LSTM and LSTM models, using the predicted value as an input again increases the error. Therefore, we choose MLP neural network architecture for our task in this work. In order to improve the predictive performance, we then propose a novel deep learning model based on MLP architecture consisting of selection and update layers similar to gate mechanisms in GRU. Our proposed MLP-SUL model outperformed

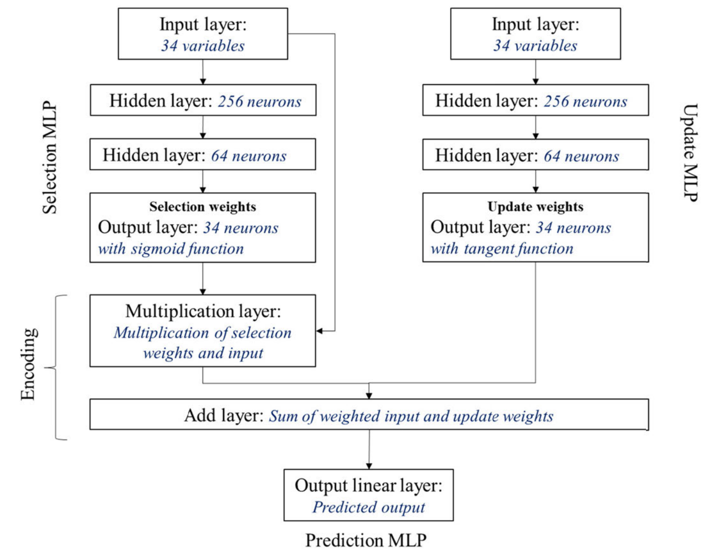

FIGURE 6. Visualization of MLP-SUL model.

TABLE 4. MAE of benchmark models and MLP-SUL on mail centers' datasets.

Mail centers	MLP	UBER	LSTM	Regression	Ours
Mail Center-1	111,360.3	172,102.3	97,862.1	44,506.9	**33,745.6**
Mail Center-2	17,369.7	**694.8**	16,675.0	33,532.0	9,727.1
Mail Center-3	6,246.6	5,309.6	8,120.6	4,222.4	**3,748.0**
Mail Center-4	26,397.0	29,330.0	**4,399.5**	25,837.4	23,464.0
Mail Center-5	32,187.1	64,374.2	46,181.5	20,426.0	**6,997.2**
Mail Center-6	**418.2**	22,166.7	1,254.7	6,112.7	5,855.3
Mail Center-7	17,746.7	26,915.8	26,324.3	**8,045.4**	9,169.1
Mail Center-8	52,642.0	66,998.9	62,213.3	21,639.5	**17,547.3**
Mail Center-9	11,863.9	**5,536.5**	15,818.6	39,767.7	9,491.1
Mail Center-10	33,602.7	40,555.0	39,396.2	19,387.4	**17,380.7**
Mail Center-11	40,883.4	48,745.5	58,180.2	37,308.8	**7,862.2**
Mail Center-12	29,190.6	42,812.9	36,001.8	18,429.7	**14,595.3**
Mail Center-13	17,562.9	15,664.2	15,189.5	16,642.1	**11,866.8**
Mail Center-14	**5,622.6**	8,835.6	11,646.9	6,843.3	6,827.5
Mail Center-15	33,349.5	35,255.2	31,443.9	15,127.1	**10,481.3**
Mail Center-16	9,410.2	14,543.1	26,519.7	**9,123.9**	11,976.6
Mail Center-17	2,850.1	**2,216.8**	6,333.7	3,550.9	4,116.9
Mail Center-18	**419.5**	6,712.6	10,068.9	6,008.2	5,454.0
Mail Center-19	25,579.7	27,804.0	16,682.4	20,509.4	**11,121.6**
Mail Center-20	**4,119.1**	15,299.7	13,534.3	20,667.7	8,826.7
Mail Center-21	14,114.0	18,034.5	36,853.1	**10,703.0**	15,682.2
Mail Center-22	3,094.9	7,516.1	**884.2**	6,855.2	8,400.4
Mail Center-23	**7,210.2**	11,536.3	11,055.6	7,874.0	7,690.8
Mail Center-24	132,160.9	115,640.8	**12,390.1**	53,863.8	49,560.3
Average	26,475.1	33,525.0	25,209.6	19,041.0	**12,982.8**

TABLE 5. MAPE of benchmark models and MLP-SUL on mail centers' datasets.

Mail centers	MLP	UBER	LSTM	Regression	Ours
Mail Center-1	0.33	0.51	0.29	0.13	**0.10**
Mail Center-2	0.25	**0.01**	0.24	0.48	0.14
Mail Center-3	0.20	0.17	0.26	0.14	**0.12**
Mail Center-4	0.18	0.20	**0.03**	0.18	0.16
Mail Center-5	0.23	0.46	0.33	0.15	**0.05**
Mail Center-6	**0.01**	0.53	0.03	0.15	0.14
Mail Center-7	0.60	0.91	0.89	**0.27**	0.31
Mail Center-8	0.33	0.42	0.39	0.14	**0.11**
Mail Center-9	0.15	**0.07**	0.20	0.50	0.12
Mail Center-10	0.29	0.35	0.34	0.17	**0.15**
Mail Center-11	0.26	0.31	0.37	0.24	**0.05**
Mail Center-12	0.30	0.44	0.37	0.19	**0.15**
Mail Center-13	0.37	0.33	0.32	0.35	**0.25**
Mail Center-14	**0.14**	0.22	0.29	0.17	0.17
Mail Center-15	0.35	0.37	0.33	0.16	**0.11**
Mail Center-16	0.22	0.34	0.62	**0.21**	0.28
Mail Center-17	0.09	**0.07**	0.20	0.11	0.13
Mail Center-18	**0.01**	0.16	0.24	0.14	0.13
Mail Center-19	0.23	0.25	0.15	0.18	**0.10**
Mail Center-20	**0.07**	0.26	0.23	0.35	0.15
Mail Center-21	0.18	0.23	0.47	**0.14**	0.20
Mail Center-22	0.07	0.17	**0.02**	0.16	0.19
Mail Center-23	**0.15**	0.24	0.23	0.16	0.16
Mail Center-24	0.32	0.28	**0.03**	0.13	0.12
Average	0.22	0.30	0.29	0.21	**0.15**

benchmark models by around 7% MAPE and 3% sMAPE (see Figure 7).

Using these results, it can be used as a model for estimating the volume of mail available on holidays such as Lunar New Year's Day and Chuseok when the volume of mail in Korea is at its peak. In addition, this prediction will help to carry out mail delivery by efficiently calculating the costs required for delivery.

These findings can also be applied to the problem of detecting events for various four-day or more holidays that have been activated as a five-day workweek and alternative holidays, as well as important holidays such as New Year's Day or Chuseok.

In order to show the superior of our data preprocessing technique, we also compare our method to traditional data preprocessing technique. For traditional data preprocessing, we select a maximum lag length equal to 7 and concatenate time series data with calendar data. Since the maximum lag is equal to 7, when performing long-term forecasting, the pre-dicted values are used as a re-input. As shown in Figure 8, the traditional data preprocessing technique demonstrates better performance on 7 out of 24 mail centers compared with our data preprocessing method. We also compare average

TABLE 6. sMAPE of benchmark models and MLP-SUL on mail centers' datasets.

Mail centers	MLP	UBER	LSTM	Regression	Ours
Mail Center-1	0.21	0.26	0.37	0.18	**0.15**
Mail Center-2	0.16	**0.16**	0.93	0.32	0.20
Mail Center-3	0.16	0.16	0.15	0.14	**0.13**
Mail Center-4	**0.23**	0.29	0.32	0.27	0.24
Mail Center-5	0.20	0.43	0.76	0.30	**0.17**
Mail Center-6	0.23	0.27	0.32	0.23	**0.17**
Mail Center-7	0.48	0.72	0.72	**0.23**	0.32
Mail Center-8	0.22	0.28	0.36	**0.15**	0.17
Mail Center-9	**0.19**	0.35	0.25	0.25	0.19
Mail Center-10	0.25	0.30	0.29	0.25	**0.23**
Mail Center-11	0.23	0.41	0.30	0.24	**0.22**
Mail Center-12	**0.22**	0.28	0.33	0.28	0.23
Mail Center-13	0.50	0.37	0.22	0.20	**0.21**
Mail Center-14	0.21	0.23	0.49	0.22	**0.21**
Mail Center-15	0.26	0.26	0.24	0.25	**0.13**
Mail Center-16	**0.23**	0.31	0.47	0.46	0.86
Mail Center-17	0.17	0.19	0.23	0.19	**0.13**
Mail Center-18	0.21	0.17	0.19	0.18	**0.14**
Mail Center-19	**0.12**	0.14	0.21	0.34	0.13
Mail Center-20	0.42	**0.31**	0.31	0.43	0.32
Mail Center-21	0.28	0.35	0.42	0.24	**0.22**
Mail Center-22	0.22	0.23	0.28	0.22	**0.20**
Mail Center-23	0.27	0.29	**0.25**	**0.29**	0.29
Mail Center-24	0.28	0.26	0.20	0.21	**0.15**
Average	0.25	0.29	0.36	0.25	**0.22**

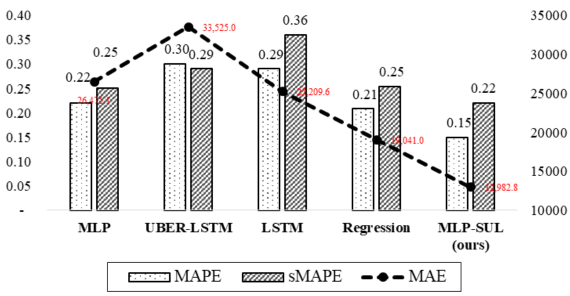

FIGURE 7. Average performance on the mail centers' datasets.

performances of these two preprocessing techniques (see Figure 9). For MAPE and MAE metrics, it is observed that our data preprocessing method is certainly better than the traditional data preprocessing technique.

In addition, we compare the predictive performances by training MLP-SUL model on total number of packages for all mail centers to show the impact of additional variables, as shown in Figure 10. Total variables are categorized into 4 groups as shown in Table 2. (1) Lags – we used only lag variable to train the model; (2) Lags +MA – moving average variables are added into lag variables; (3) Lags +MA + Events – we added the number of packages during the previous Chuseok and Lunar new year into the dataset; (4) Lags +MA + Events + Cal – Calendar variables are added into the dataset. This ablation experiment clearly shows that each group of generated variables play an important role in forecasting demand for Korean delivery service. First, we observe that moving average variables significantly decreases forecasting error. This is because moving average variables have the ability to reduce the variation of current data. Second, event and calendar variables also significantly contribute to

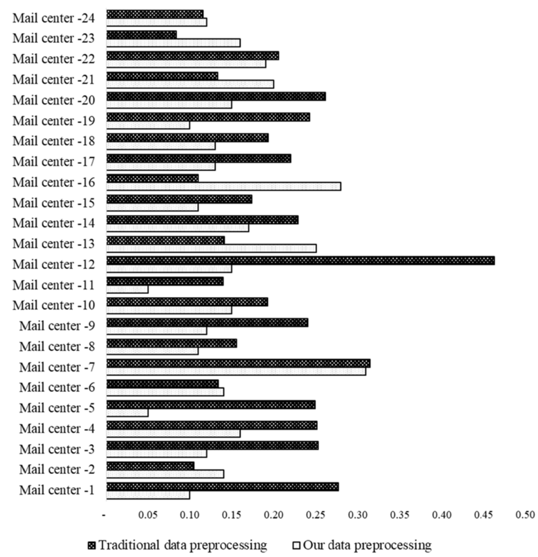

FIGURE 8. MAPE of traditional and our data preprocessing techniques on mail centers' datasets.

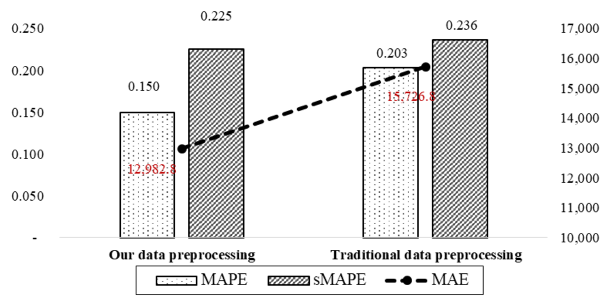

FIGURE 9. Comparison of average performance of traditional and our data preprocessing techniques.

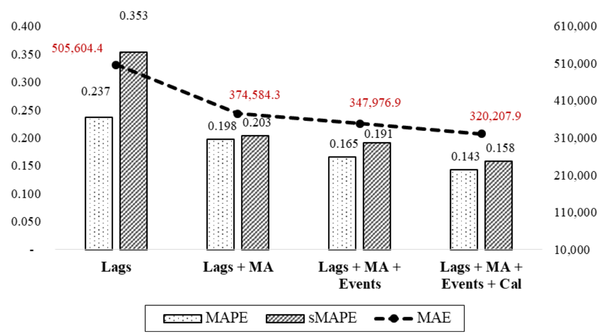

FIGURE 10. Impact of additional variables.

reduce the error. Event variables could explain the increased number of packages for extreme periods and calendar variables handle seasonality because our data is strongly seasonal with a cycle of week.

Figure 11 shows a comparison between alternative architectures and our proposed MLP-SUL model. The simple MLP model represents a simple feed-forward neural network without selection and update MLPs. The shared MLP model

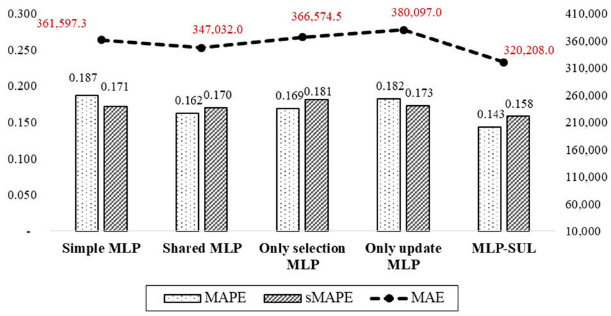

FIGURE 11. **Comparison between alternative model architectures and MLP-SUL.**

consists of a single feed-forward neural network with two output layers (selection and update outputs). Only selection MLP model consists of a single feed-forward neural network with sigmoid output to weigh input variables for forecasting target variable. Similarly, only update MLP model has a single feed-forward neural network with tangent output. We can now observe that merging selection and update MLPs into a single MLP with two output layers decreases the predictive accuracy. In addition, the use of either selection or update neural networks drastically reduces the predictive accuracy of the model. Then it is observed that these two neural networks work together to improve predictive accuracy.

IV. CONCLUSION

In this work, we attempted to perform a multi-step demand forecasting for Korean delivery service using a deep learning model. We proposed a novel deep learning model named MLP-SUL to improve predictive performance. MLP-SUL model consists of three main components: the first part of the model learns to generate weights to decide which input feed to the next layer; the second part updates the weighted input to obtain encoded input, and the third part is a prediction layer that consists of a linear layer. A linear layer takes encoded input for forecasting demand. In addition, we introduced a data preprocessing step to reduce the multi-step forecasting error. We also compared standard MLP, LSTM, and UBER-LSTM models to MLP-SUL.

For the experimental part, 24 Korean mail centers' datasets were used to evaluate selected models. As a result, our proposed MLP-SUL model outperformed benchmark models on most of the mail centers' datasets.

Our findings can be applied to the problem of detecting events for various four-day or more holidays that have been activated as a five-day workweek and alternative holidays, as well as important holidays such as New Year's Day or Chuseok. In addition, it has the practicality to be fully applied to other problems due to long-term holidays, such as traffic control or demand forecasting based on various events.

ACKNOWLEDGMENT

(Lkhagvadorj Munkhdalai and Kwang Ho Park contributed equally to this work.)

REFERENCES

[1] U. Trinkner and G. Martin, "Forecasting Swiss mail demand," in *Progress Toward Liberalization of the Postal and Delivery Sector*. Boston, MA, USA: Springer, 2006, pp. 267–280.
[2] 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.
[3] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *Int. J. Forecasting*, vol. 20, no. 1, pp. 5–10, Jan. 2004.
[4] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Holdan-Day, 1970.
[5] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, Jul. 1982.
[6] S. J. Taylor, "Forecasting the volatility of currency exchange rates," *Int. J. Forecasting*, vol. 3, no. 1, pp. 159–170, Jan. 1987.
[7] T. Guo, Z. Xu, X. Yao, H. Chen, K. Aberer, and K. Funaya, "Robust online time series prediction with recurrent neural networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Montreal, QC, Canada, Oct. 2016, pp. 816–825.
[8] T. Lin, T. Guo, and K. Aberer, "Hybrid neural networks for learning the trend in time series," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Vienna, Austria, Aug. 2017, pp. 2273–2279.
[9] T. Guo and T. Lin, "Multi-variable LSTM neural network for autoregressive exogenous model," 2018, *arXiv:1806.06384*. [Online]. Available: http://arxiv.org/abs/1806.06384
[10] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 3504–3512.
[11] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," 2017, *arXiv:1704.02971*. [Online]. Available: http://arxiv.org/abs/1704.02971
[12] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, A. Aït-Bachir, and V. Strijov, "Position-based content attention for time series forecasting with sequence-to-sequence RNNs," in *Proc. ICONIP*, Guangzhou, China, 2017, pp. 533–544.
[13] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259
[14] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, "Mixture of activation functions with extended min-max normalization for Forex market prediction," *IEEE Access*, vol. 7, pp. 183680–183691, 2019.
[15] P. Greengard, "The neurobiology of slow synaptic transmission," *Science*, vol. 294, no. 5544, pp. 1024–1030, Nov. 2001.
[16] J. Ba, G. E. Hinton, V. Mnih, J. Z. Leibo, and C. Ionescu, "Using fast weights to attend to the recent past," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 4331–4339.
[17] G. E. Hinton and D. C. Plaut, "Using fast weights to deblur old memories," in *Proc. CogSci*, 1987, pp. 177–186.
[18] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Comput.*, vol. 4, no. 1, pp. 131–139, Jan. 1992.
[19] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. ICML*, Jul. 2017, pp. 2554–2563.
[20] B. Neyshabur, R. Tomioka, and N. Srebro, "In search of the real inductive bias: On the role of implicit regularization in deep learning," in *Proc. ICLR (Workshop)*, Jan. 2015, pp. 1–9.
[21] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: An empirical study," in *Proc ICLR*, Feb. 2018, pp. 1–21.
[22] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "The role of over-parametrization in generalization of neural networks," in *Proc ICLR*, Sep. 2018, pp. 1–20.
[23] N. W. Chlosta and L. Froberg, "Forecasting mail volume for production planning in a mail center," *Comput. Oper. Res.*, vol 4, no. 4, pp. 247–255, 1977.
[24] F. Fève, J. P. Floren, F. Rodriguez, and S. Soteri, "Forecasting mail volumes in an evolving market environment," in *Heightening Competition in the Postal and Delivery Sector*. Northampton, MA, USA: Edward Elgar, 2010, pp. 116–134.

[25] G. Petneházi, "Recurrent neural networks for time series forecasting," 2019, *arXiv:1901.00069*. [Online]. Available: http://arxiv.org/abs/1901.00069
[26] 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.
[27] K. Bandara, C. Bergmeir, and H. Hewamalage, "LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 21, 2020, doi: 10.1109/TNNLS.2020.2985720.
[28] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *Int. J. Forecasting*, vol. 36, no. 1, pp. 75–85, Jan. 2020.
[29] P. Friederichs and T. L. Thorarinsdottir, "Forecast verification for extreme value distributions with an application to probabilistic peak wind prediction," *Environmetrics*, vol. 23, no. 7, pp. 579–594, Nov. 2012.
[30] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 947–956.
[31] T. Opitz, "Modeling asymptotically independent spatial extremes based on laplace random fields," *Spatial Statist.*, vol. 16, pp. 1–18, May 2016.
[32] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," in *Proc. ICML*, vol. 34, 2017, pp. 1–5.
[33] E. H. Kim and H. Jung, *Development of Prediction Technology for Parcel Acceptance and Sorting on Traditional Holidays*, document PLICT-2019-R2-3, ETRI, 2019.
[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[35] L. Munkhdalai, T. Munkhdalai, and K. Ho Ryu, "A locally adaptive interpretable regression," 2020, *arXiv:2005.03350*. [Online]. Available: http://arxiv.org/abs/2005.03350
[36] D. F. Andrews, "A robust method for multiple linear regression," *Technometrics*, vol. 16, no. 4, pp. 523–531, Nov. 1974.
[37] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, p. 386, 1958.

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 and Import Forecasting, and Demand Forecasting for Postal Delivery Service. His current research interests include the development of an adaptive deep learning model for forecasting time series as well as adaptive regression models. His research interests include deep learning along with their applications in domains, such as financial, medical informatics, and public health informatics.

KWANG HO PARK (Graduate Student Member, IEEE) received the B.S. degree in biochemistry from Chungbuk National University, Cheongju, South Korea, in 2015, and the M.Sc. degree from the Database and Bioinformatics Laboratory, in 2017, where he is currently pursuing the Ph.D. degree. His main research interests are data mining, bioinformatics, and medical informatics.

ERDENEBILEG BATBAATAR received the M.S. and Ph.D. degrees in data mining, medical informatics, and computer science from the Database and Bioinformatics Laboratory, Chungbuk National University, South Korea. He is currently the Postdoctoral Researcher of Bioinformatics and Computer Science with Chungbuk National University. His research interests include software engineering, data mining, big data analysis, bioinformatics, machine learning, deep learning, and their applications.

NIPON THEERA-UMPON (Senior Member, IEEE) received the B.Eng. degree (Hons.) from Chiang Mai University, the M.S. degree from the University of Southern California, and the Ph.D. degree from the University of Missouri-Columbia, all in electrical engineering.

He has been with the Department of Electrical Engineering, Chiang Mai University, since 1993, where he is currently serving as the Director of the Biomedical Engineering Institute. He has published more than 190 full research articles 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 IEEE-IES Technical Committee on Human Factors. He is a member of the Thai Robotics Society, the Biomedical Engineering Society of Thailand, and the Council of Engineers in Thailand. 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 Chairman of graduate study in electrical engineering and graduate study in biomedical engineering. 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.

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 served at Reserve Officers' Training Corps (ROTC) of the Korean Army. He was with The University of Arizona, Tucson, AZ, USA, as a Postdoctoral Researcher and a Research Scientist, and 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, as well as an Emeritus and 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 Leader of the Database and Bioinformatics Laboratory, South Korea, since 1986, but also the co-leader of the Research Group, Data Science Laboratory, Vietnam, since March 2019. He is also the former Vice-President of the Personalized Tumor Engineering Research Center. He has published more than 1000 referred 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.