**Donghwan SONG, PhD Candidate**
**Department of System Design and Control Engineering**
**Ulsan National Institute of Science and Technology**
**Moise BUSOGI, PhD**
**Department of Electrical and Computer Engineering**
**Carnegie Mellon University Africa**
**Adrian M. CHUNG BAEK, PhD Candidate**
**Department of System Design and Control Engineering**
**Ulsan National Institute of Science and Technology**
**Associate Professor Namhun KIM, PhD (Corresponding author)**
**E-mail: nhkim@unist.ac.kr**
**Department of System Design and Control Engineering**
**Ulsan National Institute of Science and Technology**

# FORECASTING STOCK MARKET INDEX BASED ON PATTERN-DRIVEN LONG SHORT-TERM MEMORY

***Abstract.*** *Stock trend prediction is an important area of study for researchers and practitioners. In recent years, along with traditional statistical prediction models, machine learning and deep learning techniques have been increasingly adopted in various financial studies. Long Short-Term Memory (LSTM) is one of the deep learning models for predicting time-series data. In the case of vanilla LSTM, shared weights are learned based on all available data; hence, it is difficult to accurately learn patterns and predict the future value from a subset of data. In this paper, a pattern-driven hybrid model that combines an LSTM with an unsupervised learning algorithm is proposed for precise prediction of stock prices. The performance of the hybrid model is evaluated using Korea stock index data. The results demonstrate that the proposed model outperforms traditional recurrent neural network (RNN) and LSTM models.*

***Keywords:*** *Long Short-Term Memory, Forecasting, Pattern Clustering, Stock Index, Time-series Analysis.*

**JEL Classification:C45, E37**

## 1. Introduction

Forecasting has been a subject of interest for a long time, and this interest has led to a great number of studies in diverse fields, especially finance. However, financial time-series forecasting, particularly stock price forecasting, has been a



**DOI: 10.24818/18423264/54.3.20.02**

challenging problem for researchers and investors due to financial sectors having high volatility and time-varying environments. Thus, various stock market prediction approaches have been proposed. These methods include simple statistical linear regression models to state-of-the-art deep learning-based models.

For decades, traditional statistical models, such as the autoregressive moving average and autoregressive integrated moving average models, have been widely used in economics for time-series prediction, primarily due to their ability to model linear relationships between the input and output (Box et al., 2015; Cheong, 2013; Idrees et al., 2019). However, linear assumptions often lack sufficient evidence (Enkeand Thawornwong, 2005); thus, adopting artificial neural network algorithms in the prediction of financial data has attracted increasing attention in industry and academia. Neural networks have rapidly gained preference because they demonstrate great suitability in handling incomplete, missing, or noisy data without requiring assumptions about the data distribution (Vellido et al., 1999).

Numerous applications of ANNs have been studied, some of which have successfully predicted a wide range of financial data indicators (Refenes et al., 1997; Vanstone & Finnie, 2009; Zhang et al., 1998). Enhanced neural networks have also been introduced as forecasting methods. For example, Maknickienė et al. (2011)applied a recurrent neural network (RNN) to investigate financial market predictions. Besides, a long short-term memory (LSTM) network, which is a variation of an RNN to solve vanishing gradient problems, has been implemented extensively in recent studies (Moon and Kim, 2019; Nelson et al., 2017). The remarkable performance of such methods demonstrates the superiority of LSTM on time-series data. However, a limitation exists, e.g., LSTMs tend to prefer recent data, which affect shared parameters significantly (Fu et al., 2018). As financial time-series data involve time-dependent relationships, LSTM often fails to capture such dynamics (Oh et al., 2019).

Despite the high performance of LSTM, even the slightest improvement is desirable because this has a positive influence on financial investments. Therefore, improving the efficiency of forecast models remains a primary concern. Accordingly, hybrid models have been proposed. For example, Bildirici and Ersin (2009), Kristjanpoller et al. (2014), Wang and Que (2018), and Yang and Lin (2015) demonstrated hybrid models that combine different artificial intelligence techniques with traditional forecasting approaches. In addition, Pulido et al. (2014) suggested a combination of neural network and fuzzy aggregation based on particle swarm optimization to predict the Mexican stock exchange. Baek and Kim (2018) developed a modular architecture comprising two LSTM networks to refine the resulting deterioration in prediction accuracy. Ramos-Pérezet al. (2019) proposed a financial forecasting system using machine learning techniques with neural


**DOI: 10.24818/18423264/54.3.20.02**

networks. Overall, hybridized models outperform conventional single models by yielding lower test errors.

This paper presents a hybrid pattern-driven LSTM (P-LSTM) model, which combines unsupervised and supervised learning algorithms, to solve the limitation of LSTM in which patterns in data are not learned precisely due to weights being shared between cells. The proposed P-LSTM model clusters financial time-series data based on subsequences of the same pattern and trains them separately for stable learning of weights.

The remainder of this paper is organized as follows. Section 2 summarizes the two methodologies adopted to develop the proposed hybrid model. Section 3 discusses the framework of the proposed P- LSTM model. Section 4 provides an in-depth discussion of experimental setups and processes, and, in Section 5, the experimental results are provided with detailed analysis. Finally, conclusions are given in Section 6.

## 2. Background

This section describes the architectures of the two key models that adopt the pattern-driven hybrid model, i.e., the balanced iterative reducing and clustering using hierarchies (BIRCH) clustering and LSTM algorithms.

### 2.1 BIRCH clustering algorithm

BIRCH, developed by Zhang et al. (1996), is a typical integrated hierarchical clustering algorithm applicable to very large datasets and is known to achieve high-quality clustering with only one traversal. Its high efficiency is a result of the clever use of available memory to derive the best quality sub-clusters while reducing processing costs. In BIRCH, not every data point is equally crucial for clustering purposes. Consequently, the BIRCH clustering method provides a robust solution for managing noise (Thilagavathi et al. 2013). Moreover, when a Euclidean or Manhattan distance measure is used, BIRCH is preferred because the data attributes are continuous (Chiu et al. 2001).Such aspects make BIRCH algorithm a primary candidate for clustering.

BIRCH is mainly composed of two phases. It first performs pre-clustering in which dense regions of data points are represented in summary statistics called clustering feature (CF). Then, hierarchical clustering is conducted to cluster the set of summaries. The CF represents the characteristics of a cluster and is denoted $(n, LS, SS)$, where n is the number of data in a sub-cluster, LS is the linear sum of the points, and SS is the sum of the squared of the points. New CF can be obtained by adding the CFs of two disjoint clusters as follows.



DOI: 10.24818/18423264/54.3.20.02

$$\mathrm{CF}_1 + \mathrm{CF}_2 = (n_1 + n_2, LS_1 + LS_2, SS_1 + SS_2)$$

The information in the CF is adequate to achieve high-quality clustering. Thus, the BIRCH algorithm is memory-efficient because it does not require the whole dataset.

The CF comprises the nodes in a CF tree that is an in-memory data structure acting as the core of the BIRCH clustering algorithm. In a CF tree, leaf nodes form the non-leaf node, and non-leaf nodes form the root. The parameters of the CF tree are branching factor $B$, threshold T, and the number of entries in a leaf node L. Branching factor B establishes the limiting number of sub-clusters for each non-leaf node, and threshold T is the maximum radius of a cluster in a leaf node. If threshold Tis too small, significant space could be taken; thus, it is recommended to begin from 0 and increment the threshold slowly (Bing et al. 2004).

The steps of the BIRCH clustering algorithm are summarized as follows.

1. Build a CF tree to load data to the memory. Reconstruct the tree if memor y is exhausted.
2. Resize the dataset by removing outliers to adjust the size of the CF tree.
3. Use the existing clustering algorithm on CF entries.
4. (Optional) Refine the cluster by fixing incorrectly assigned data points.

### 2.2 Long Short-term Memory

Recurrent neural networks (RNN) are remarkably powerful models that process sequential patterns through internal loops; however, it is frequently difficult to train RNNs when handling long time lags due to back propagated error blowing up or decaying exponentially, also known as exploding and vanishing gradients, respectively. To address the vanishing gradients problem, the LSTM architecture was developed by Hochreiter and Schmidhuber (1997). LSTM introduces an innovative concept called a memory block, which is essentially an accumulator of state information.


DOI: 10.24818/18423264/54.3.20.02

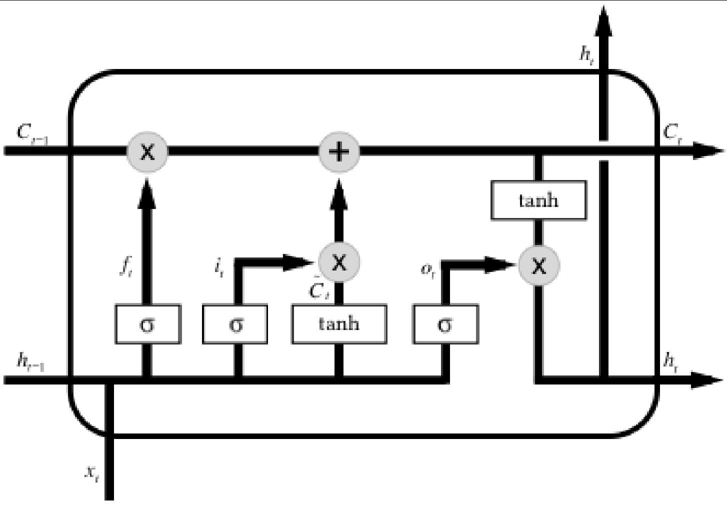



**Figure 1. Cell diagram of LSTM**

As shown in Fig. 1, the memory block comprises a memory cell and three main gates. The memory cell $c^t$ has a recurrently self-connected linear unit, an input gate $i^t$ that allows an incoming signal to alter (or block) the state of the memory cell, a forget gate $f^t$ that modulates the memory cell's connection to keep only the relevant information from its previous state, and an output gate $o^t$ allowing (or preventing) the state of the memory cell to influence other neurons. These components control the information flow and allow a gradient to be trapped in the cell, which prevents the gradient from becoming increasingly unstable.

Mathematical expressions for the most common LSTM architecture, i.e., with forget gates, are given as follows.

Input gate: $i^t = \sigma(W_i x^t + U_i h^{t-1} + b_i)$ (1)
Forget gate: $f^t = \sigma(W_f x^t + U_f h^{t-1} + b_f)$ (2)
Memory cell: $c^t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1}$ (3)
Input modulate gate: $\tilde{c}^t = \tanh(W_c x^t + U_c h^{t-1} + b_c)$ (4)
Output gate: $o^t = \sigma(W_o x^t + U_o h^{t-1} + b_o)$ (5)
Block output: $h^t = o^t \odot \tanh(c^t)$ (6)

Here, $\tilde{c}^t$ determines the amount of new information received in the cell state, $x^t$ is the input vector at time t, $h^t$ is the hidden layer vector, W denotes input weighted matrices, U denotes recurrent weight matrices, and b denotes the bias vectors of LSTM. $\sigma(\cdot)$ is a pointwise nonlinear activation function, i.e., logistic sigmoid $\left(\frac{1}{1+e^{-x}}\right)$ in this case. Note that elementwise multiplication is denoted $\odot$.



DOI: 10.24818/18423264/54.3.20.02

### 3. Proposed Pattern-driven LSTM

The proposed P-LSTM framework has three primary stages, as shown in Fig. 2. The first stage involves pre-processing the input time-series data. The time-series data are partitioned to be the same size as the timestep in the LSTM network. Then, each vector set is paired with a Y value, i.e., the closing price of the following day. These paired sets are divided into training and testing data. Each X value is min/max scaled from 0 to 1 based on their X vector set for the clustering process and Y values are also scale during the min and max of the corresponding X vector set.

Patterns are defined in the second stage. Here, M clusters are classified by applying the BIRCH clustering algorithm to data with essentialinput features. These clusters depict all possible patterns in the provided data. Subsequently, all the batches in the training data are then distributedappropriately to the clusters, thereby generating a new form of training data. Additionally, batches of testing data are fed into the trained BIRCH model for assignment to a proper cluster.

In the last stage, a LSTM network is built for each cluster. Here, the LSTM network is trained using the training data collected in the corresponding cluster. Finally, the testing data from the same cluster is fed into the trained LSTM neural network to perform the prediction.

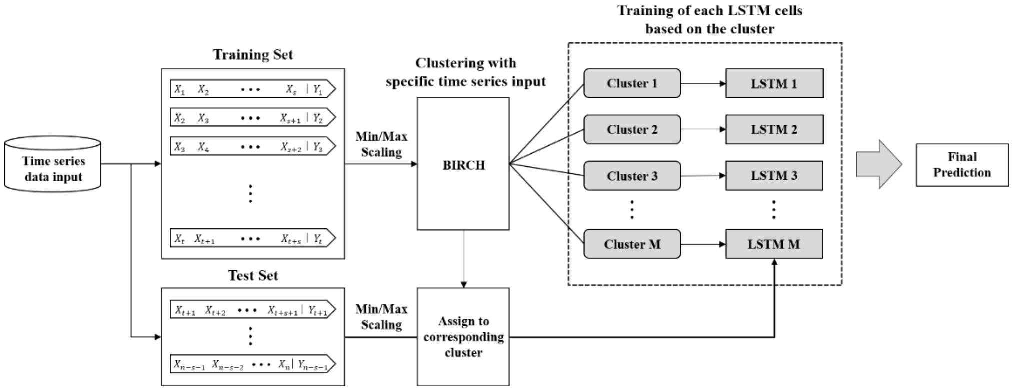



**Figure 2. Architecture of proposed P-LSTM hybrid model**

### 4. Experimental Setup

#### 4.1 Data and pre-processing

Daily data from the Korean stock market indexes, i.e., Korea Composite Stock Price Index (KOSPI) and Korea Securities Dealers Automated Quotation (KOSDAQ) were used to evaluate the performance of the prediction models. As seen in Table 1 and Figure 3, the data comprise 4593 daily records from January 2, 2001 to July 31, 2019. The data were split into training data (80%) and testing data


DOI: 10.24818/18423264/54.3.20.02

(20%) after batch pre-processing. Here, the training data were used to train the LSTM models, and the testing data were used to validate the proposed model.

Each data point is defined as $X_t$, and comprises five indicators: open price (stock price at the start of each trading day), high price (highest price of each trading day), low price (lowest price of each trading day), close price (stock price at the end of each trading day), and volume (number of shares traded in a security during each trading day).

Note that the input variables were pre-processed to maximize the performance of the proposed P-LSTM model. First, the indicators of each day were bundled into a vector $X_t$,. Then, several of these vectors were put together to form the batch $\tilde{X}$by considering the size of the timestep (n) of the corresponding LSTM cell. As the magnitude of data varies, which prevents accurate clustering, each time-series vector set comprising consecutive days was min/max scaled as follows.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (7)$$

Here, x is the original value, $x'$is the scaled value, and $x_{max}$and $x_{min}$are the maximum and minimum values in each batch $\tilde{X} = \{X_1, X_2, \cdots X_n\}$,respectively.

The Y value in each batch are also min/max scaled based on the minimum and maximum values of the corresponding $\tilde{X}$ vector set.In this study, the data dimension was set 20 days because, in this given period, the features are most effective as a moving average in the stock market.

**Table 1. Summary of training data and test dataset after batch preprocessing**

| Dataset | Train period (sample) | Test period (sample) |
|---|---|---|
| KOSPI | 2001/01/02 ~ 2015/11/05 (3658) | 2015/11/06 ~ 2019/07/31 (915) |
| KOSDAQ | 2001/01/02 ~ 2015/11/05 (3658) | 2015/11/06 ~ 2019/07/31 (915) |

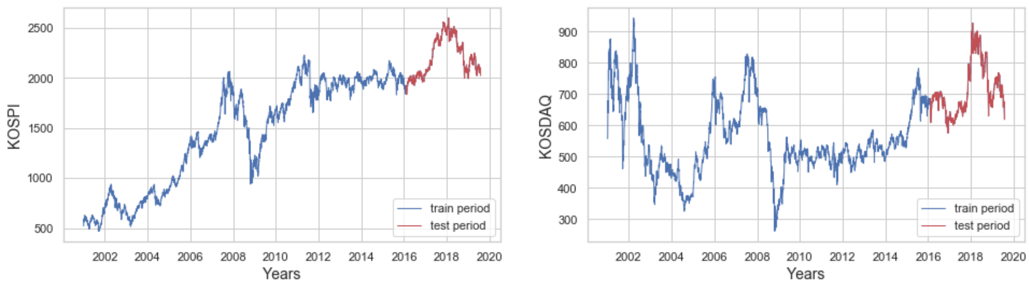



**Figure 3. KOSPI (left) and KOSDAQ (right) index**



DOI: 10.24818/18423264/54.3.20.02

### 4.2 Pattern Clustering

Pattern clusters were classified with the 3658 pre-processed training data from the previous step based on the BIRCH clustering algorithm. For clustering, min/max scaled values between 0 and 1 were used. Then, the 915 pre-processed test data were assigned to the corresponding clusters based on the trained model.

Here, the BIRCH hyperparameters were set to appropriate values through preliminary experiments based. First, the number of clusters was set to 8to prevent the number of data in each cluster from becoming too small relative to the limited number of data, although better patterns can be classified using a greater number of clusters. Second, the threshold and branching factor were set to 0.2 and 20, respectively, to distribute data evenly among clusters without destroying the pattern. As a result, eight clusters were obtained. Figures 4 and 5 show the number of allocated training and testing samples from KOSPI and KOSDAQ in each cluster.

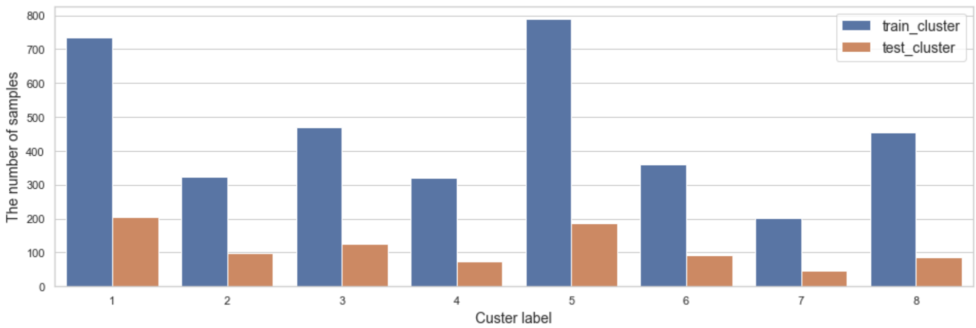



**Figure4. Number of training and test samples in each cluster for KOSPI**

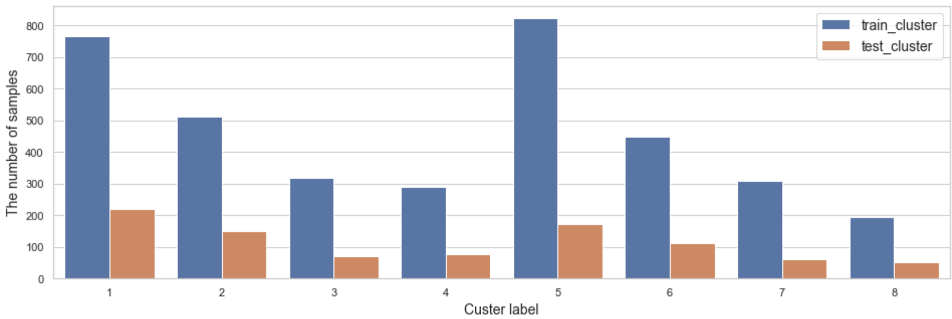



**Figure 5. Number of training and test samples in each cluster for KOSDAQ**


DOI: 10.24818/18423264/54.3.20.02

#### 4.3 Model Hyperparameter

An LSTM neural network was applied to each cluster. Thus, the number of networks was equal to the number of clusters. For the LSTM hyperparameters, optimum values were found by conducting inductive experiments. To prevent the model from overfitting, early-stopping was employed to stop training as required. Except for the epochs, the other hyperparameters were kept constant to maintain the universality of the experiment. The hyperparameters were set as follows: timestep = 20, learning rate = 0.001, input dimension = 5, hidden dimension = 1, and output dimension = 1 (the output dimension is the dimension of the closing price of the next day). Finally, the Adam optimizer and mean squared error were used as the optimization algorithm and loss function, respectively.

#### 4.4 Evaluation Metrics

The experimental results were analysed according to the mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) metrics.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - f_i| \tag{8}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2} \tag{9}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - f_i}{y_i}\right| \tag{10}$$

Here, $y_i$ and $f_i$ are the actual and forecast values, respectively, and n is the number of forecasted values. The evaluation indexes of the M clusters were averaged using a simple arithmetic average method to facilitate the evaluation of the forecast results output by the proposed P-LSTM model.

#### 5. Results

The KOSPI and KOSDAQ datasets were used to evaluate the effectiveness of the proposed P-LSTM model in forecasting actual financial time-series data. The prediction accuracy was compared to that of conventional RNN- and LSTM-based deep learning models.

Figures 6 and 8 show the patterns obtained when BIRCH clustering was applied to the KOSPI and KOSDAQ training data, and Figs. 7 and 9 show the clustered KOSPI and KOSDAQ testing data. The lines in the figures are the connections of $X^4$ values, i.e., the closing price, from time (t-20) to time t. The x-axis represents the length of the time step (20 days), and the y-axis represents the min/max scaled closing price. These figures show that the line density depends on the number of samples, all eight clusters demonstrate different patterns, and the pattern of the testing clusters perfectly matches that of the trained cluster.



DOI: 10.24818/18423264/54.3.20.02

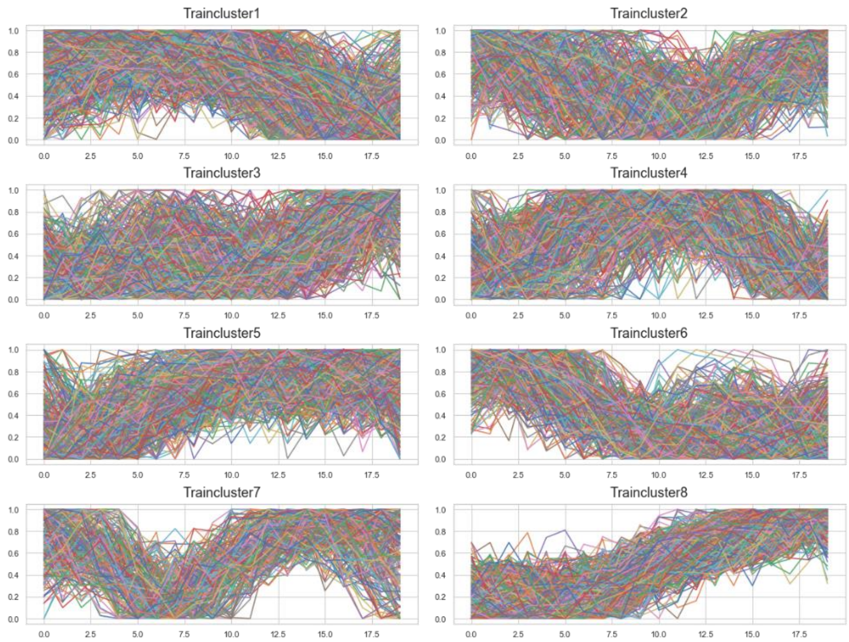



**Figure 6. Visualization of KOSPI patterns for each training cluster classified using BIRCH algorithm**

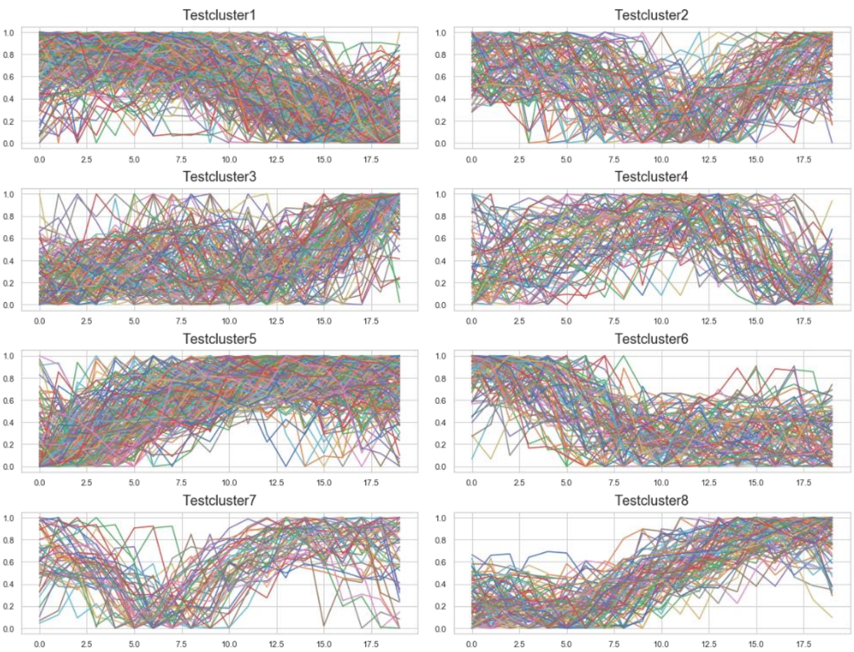



**Figure 7. Visualization of KOSPI patterns for each testing cluster classified using BIRCH algorithm**


**DOI: 10.24818/18423264/54.3.20.02**

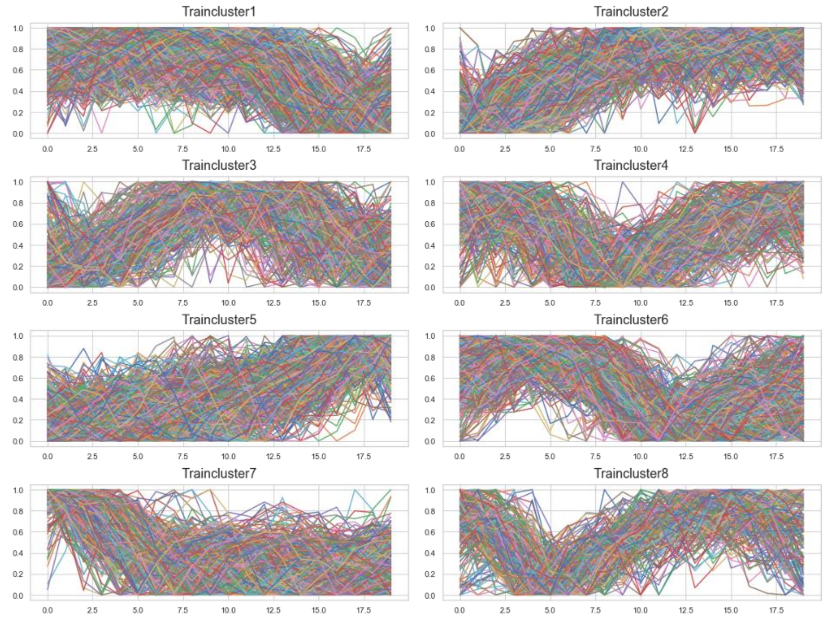



**Figure 8. Visualization of KOSDAQ patterns for each training cluster classified using BIRCH algorithm**

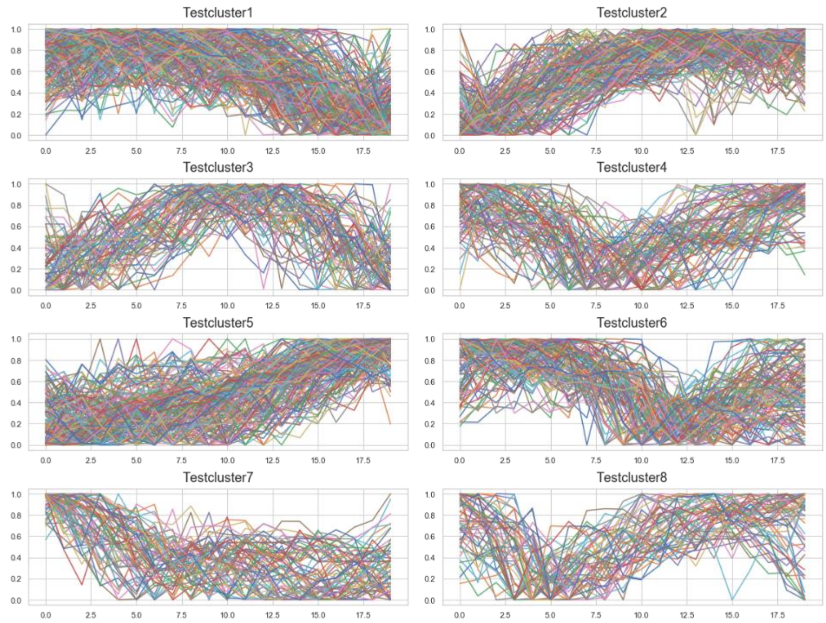



**Figure 9. Visualization of KOSDAQ patterns for each testing cluster classified using BIRCH algorithm**



**DOI: 10.24818/18423264/54.3.20.02**

**Table 2.Number of Sample data and early-stopping epoch for each model**

| | | KOSPI | | | KOSDAQ | | |
|---|---|---|---|---|---|---|---|
| | | Train | Test | Early Stopping epoch | Training | Test | Early Stopping epoch |
| P-LSTM | Cluster 1 | 736 | 204 | 21 | 766 | 220 | 48 |
| | Cluster 2 | 323 | 97 | 87 | 511 | 149 | 38 |
| | Cluster 3 | 471 | 126 | 42 | 319 | 71 | 51 |
| | Cluster 4 | 322 | 75 | 40 | 289 | 78 | 83 |
| | Cluster 5 | 788 | 187 | 28 | 821 | 171 | 39 |
| | Cluster 6 | 359 | 92 | 35 | 447 | 112 | 39 |
| | Cluster 7 | 203 | 48 | 34 | 309 | 62 | 35 |
| | Cluster 8 | 456 | 86 | 14 | 196 | 52 | 115 |
| LSTM | | 3658 | 915 | 35 | 3658 | 915 | 49 |
| RNN | | 3658 | 915 | 22 | 3658 | 915 | 34 |

The number of sample data and epoch for each model are summarized in Table 2. For the proposed P-LSTM, the numbers of distributed samples according to the patterns are noted. The evaluation results of the testing data fed into the corresponding neural network are listed in Table 3. The average results obtained using the KOSPI data are 13.285 (MAE), 18.475 (RMSE), and 0.610 (MAPE), and the average results obtained using the KOSDAQ data are 6.887 (MAE), 9.413 (RMSE), and 0.948 (MAPE). The obtained results suggest that the values of cluster 7 for the KOSPI data and cluster 3 for the KOSDAQ data are optimum.

**Table 3. Summary of P-LSTM forecasting evaluation results for each cluster**

| | | KOSPI | | | KOSDAQ | | |
|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| P-LSTM | Cluster 1 | 15.110 | 19.517 | 0.704 | 8.184 | 10.969 | 1.169 |
| | Cluster 2 | 13.530 | 17.650 | 0.635 | 6.505 | 9.156 | 0.878 |
| | Cluster 3 | 9.754 | 13.722 | 0.437 | 8.798 | 11.537 | 1.188 |
| | Cluster 4 | 12.697 | 19.384 | 0.587 | 5.368 | 7.851 | 0.740 |
| | Cluster 5 | 11.783 | 15.392 | 0.534 | 5.329 | 7.246 | 0.718 |
| | Cluster 6 | 15.213 | 20.513 | 0.692 | 6.391 | 8.366 | 0.884 |
| | Cluster 7 | 14.151 | 20.632 | 0.674 | 7.930 | 10.241 | 1.124 |
| | Cluster 8 | 14.042 | 20.987 | 0.616 | 6.592 | 9.940 | 0.886 |
| | Mean | 13.285 | 18.475 | 0.610 | 6.887 | 9.413 | 0.948 |
| | Best | 9.754 | 13.722 | 0.437 | 5.329 | 7.246 | 0.718 |


DOI: 10.24818/18423264/54.3.20.02

Figures 10 and 11 compare the actual values of the KOSPI and KOSDAQ indexes to the min/max scaled values returned to the original exponential scale. Here, the x-axis represents the independent data samples within each cluster rather than continuous time-series data.

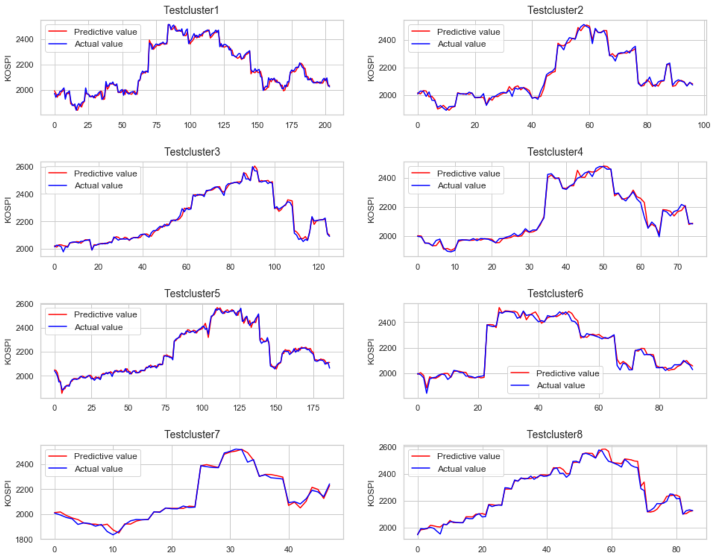



**Fig. 10. Comparison of predicted data from P-LSTM and actual data for KOSPI in each cluster**

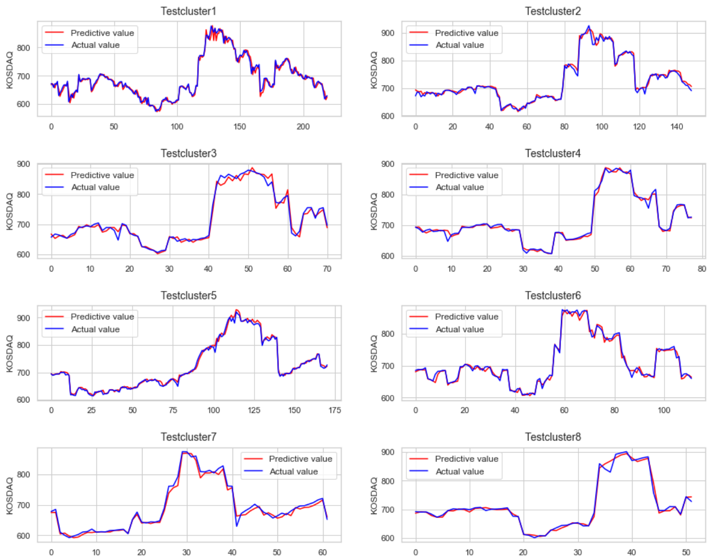



**Fig. 11. Comparison of predicted data from P-LSTM and actual data for KOSDAQ in each cluster**

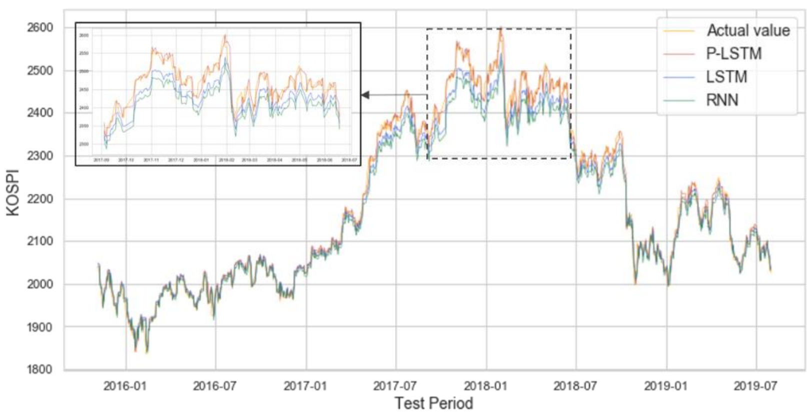


**DOI: 10.24818/18423264/54.3.20.02**

The testing data were also restored to the original index data to facilitate comparison with the actual indexes or the other models. The inverse scaled results compared to the results from the LSTM and RNN methods and the actual KOSPI and KOSDAQ values are shown in Fig. 12. As shown in Fig. 12 and Table 4, the pattern-based LSTM model captures the movement of the actual indexes as demonstrated by the lowest error compared to LSTM only or RNN.

Table 4 summarizes the test errors of the compared models. As can be seen, the proposed model demonstrates improvements over the RNN model of 49.07% (MAE), 45.80% (RMSE), and 46.51% (MAPE) for the KOSPI data, and 10.49% (MAE), 8.59% (RMSE), and 11.15% (MAPE) for the KOSDAQ data. Compared to the LSTM model, the proposed model demonstrated approximately 30.53% (MAE), 27.13% (RMSE), and 27.80% (MAPE) differences in performance for the KOSPI data, and 3.07% (MAE), 2.69% (RMSE), and 3.36% (MAPE) differences for the KOSDAQ data.

(a) KOSPI test set




(b) KOSDAQ test set

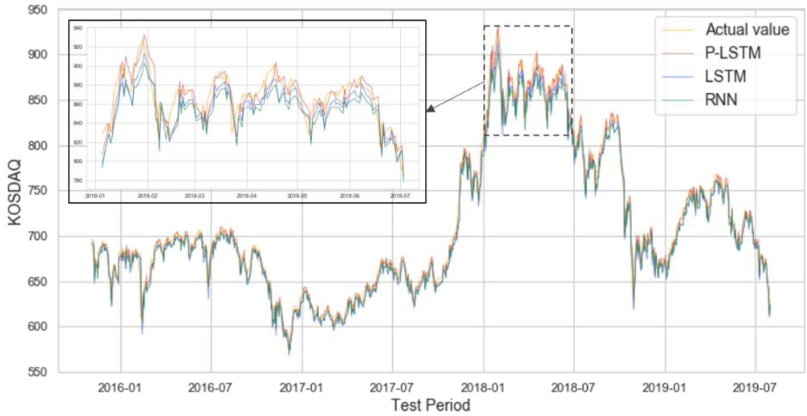



**Fig. 12. Comparison of predicted and actual values for (a) KOSPI data and (b) KOSDAQ data**


DOI: 10.24818/18423264/54.3.20.02

**Table 4. Comparison of test errors**

| | KOSPI | | | KOSDAQ | | |
|---|---|---|---|---|---|---|
| | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| P-LSTM | 13.187 | 18.142 | 0.605 | 6.857 | 9.448 | 0.948 |
| LSTM | 18.983 | 24.898 | 0.838 | 7.074 | 9.709 | 0.981 |
| RNN | 25.891 | 33.473 | 1.131 | 7.661 | 10.336 | 1.067 |

## 6. Conclusion

With technological advancements, academics have adopted deep learning-based methods to conduct meaningful studies into solving diverse time-series problems. In particular, RNN and LSTM models are commonly applied to such problems, and their architectures are frequently modified to improve predictive performance. However, due to their tendency to prefer recent inputs, they demonstrate limited efficacy when handling data over a period, e.g., financial time-series data. Therefore, in this paper, we have proposed the P-LSTM model, which integrates LSTM into a classical unsupervised learning model, i.e., the BIRCH clustering algorithm, and works with an emphasis on a divide and conquer strategy. The proposed P-LSTM model (1) pre-processes time-series data by partitioning the data into a specific size and pairing to the closing price of the next day and then min/max scales in the range 0-1; (2) classifies patterns using the BIRCH clustering algorithm and assigns test data to proper clusters; and (3) builds individual LSTM neural networks for each cluster and feeds the test data for evaluation.

The predictive power of the proposed P-LSTM model was validated by applying it to the KOSPI and KOSDAQ indexes. According to the results, the MAE, RMSE, and MAPE values for the KOSPI data decreased to 30.53% (49.07%), 27.13% (45.80%), and 27.80% (46.51%), respectively, of the corresponding KOSPI forecasting errors of LSTM (RNN). Besides, the test MAE, RMSE, and MAPE values for the KOSDAQ data decreased to 3.07% (10.49%), 2.69% (8.59%), and 3.36% (11.15%), respectively, of the corresponding KOSDAQ forecasting errors of LSTM (RNN). These results demonstrate that the proposed P-LSTM model outperforms conventional RNN and LSTM models and is more suitable for financial time-series prediction problems than conventional networks because their unique structures make it possible to learn patterns contained in input sequences.

This study has significance because suggests a new approach to pre-processing and forecasting financial time-series data by combining learning techniques. Besides, the proposed P-LSTM model is not restricted to financial data, i.e., it can be applied to time-series data from other fields. Finally, the proposed network can extend research into hybrid frameworks.



**DOI: 10.24818/18423264/54.3.20.02**

Note that the effectiveness of the proposed P-LSTM model improves as the number of clusters increases, i.e., as more patterns are classified, more sophisticated predictions can be achieved. However, this increase of clusters undoubtedly leads to a reduced number of samples in each cluster, which results in the model prone to overfitting. Thus, in the future, solutions for insufficient data will be explored, such as data augmentation that reassembles time-series data with different properties based on patterns or using time-series data in minutes.

## REFERENCES

[1]**Baek, Y. & Kim, H. Y. (2018),** ***ModAugNet: A New Forecasting Framework for Stock Market Index Value with an Overfitting Prevention LSTM Module and a Prediction LSTM Module.*** *Expert Systems with Applications*, 113, 457–480;

[2] **Bildirici, M. &Ersin, O. (2009),** ***Improving Forecasts of GARCH Family Models with the Artificial Neural Networks: An Application to the Daily Returns in Istanbul Stock Exchange.*** *Expert Systems with Applications*, 36(4), 7355–7362;

[3] **Bing, Z., Junyi, S. &Qinke, P. (2004),** ***An Adjustable Algorithm for Color Quantization.*** *Pattern Recognition Letters*, 25(16), 1787-1797;

[4] **Box, G. E, Jenkins, G.M., Reinsel, G. C. &Ljung, G. M. (2015),** ***Time Series Analysis: Forecasting and Control.*** *John Wiley & Sons*;

[5] **Cheong, C. W. (2013),** ***The Computational of Stock Market Volatility from the Perspective of Heterogeneous Market Hypothesis.*** *Economic Computation and Economic Cybernetics Studies and Research; ASE Publishing; 47*(2), 247-260;

[6] **Chiu, T., Fang, D., Chen, J., Wang, Y. &Jeris, C. (2001, August),** ***A Robust and Scalable Clustering Algorithm for Mixed Type Attributes in Large Database Environment.*** *In Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining*, 263-268;

[7] **Enke, D. & Thawornwong, S. (2005),** ***The Use of Data Mining and Neural Networks for Forecasting Stock Market Returns***. *Expert Systems with applications*, 29(4), 927–940;

[8] **Fu, X., Yang, J., Li, J., Fang, M. & Wang, H. (2018),** ***Lexicon-enhanced LSTM with Attention for General Sentiment Analysis.*** *IEEE Access*, 6, 71884–71891;

[9]**Hochreiter, S. &Schmidhuber, J. (1997),** ***Long Short-term Memory.*** *Neural computation*, 9(8), 1735-1780;

[10] **Idrees, S. M., Alam, M. A. & Agarwal, P. (2019),** ***A Prediction Approach for Stock Market Volatility Based on Time Series Data.*** *IEEE Access*, 7, 17287–17298;

[11] **Kristjanpoller, W., Fadic, A. & Minutolo, M. C. (2014),** ***Volatility Forecast Using Hybrid Neural Network Models .****Expert Systems with Applications*, 41(5), 2437–2442;



**DOI: 10.24818/18423264/54.3.20.02**

[12]**Maknickienė, N., Rutkauskas, A. V. &Maknickas, A. (2011),** ***Investigation of Financial Market Prediction by Recurrent Neural Network.*** *Innovative Technologies for Science, Business and Education*, 2(11), 3–8;
[13] **Moon, K. & Kim, H. (2019),** ***Performance of Deep Learning in Prediction of Stock Market Volatility.*** *Economic Computation and Economic Cybernetics Studies and Research; ASE Publishing;* 53(2);
[14] **Nelson, D. M., Pereira, A. C. & de Oliveira, R. A. (2017),** ***Stock Market's Price Movement Prediction with LSTM Neural Networks.*** *In Proceedings of the international joint conference on neural networks (IJCNN)* (pp. 1419–1426). IEEE;
[15] **Oh, J., Wang, J., Tang, S., Sjoding, M. & Wiens, J. (2019),** ***Relaxed Weight Sharing: Effectively Modeling Time-varying Relationships in Clinical Time-series.*** *arXiv preprint arXiv*: 1906.02898;
[16] **Pulido, M., Melin, P. & Castillo, O. (2014),** ***Particle Swarm Optimization of Ensemble Neural Networks with Fuzzy Aggregation for Time Series Prediction of the Mexican Stock Exchange.*** *Information Sciences*, 280, 188–204;
[17] **Ramos-Pérez, E., Alonso-González, P. J. &Núñez-Velázquez, J. J. (2019),** ***Forecasting Volatility with a Stacked Model Based on a Hybridized Artificial Neural Network.*** *Expert Systems with Applications*, 129, 1-9;
[18] **Refenes, A. P., Burgess, A. N. & Bentz, Y. (1997),** ***Neural Networks in Financial Engineering: A Study in Methodology.*** *IEEE transactions on Neural networks*, 8(6), 1222–1267;
[19] **Thilagavathi, G., Srivaishnavi, D. & Aparna, N. (2013),** ***A Survey on Efficient Hierarchical Algorithm Used in Clustering.*** *International Journal of Engineering*, 2(9), 165-176;
[20] **Vanstone, B. & Finnie, G. (2009),** ***An Empirical Methodology for Developing Stockmarket Trading Systems Using Artificial Neural Networks .****Expert Systems with Applications*, 36(3), 6668–6680;
[21] **Vellido, A., Lisboa, P.J. & Vaughan, J. (1999),** ***Neural Networks in Business: A Survey of Applications (1992-1998).****Expert Systems with Applications*, 17(1), 51–70;
[22] **Wang, J. & Que, D. (2018),** ***An Experimental Investigation of Two Hybrid Frameworks for Stock Index Prediction Using Neural Network and Support Vector Regression.*** *Economic Computation and Economic Cybernetics Studies and Research*; *ASE Publishing,* 52(4);
[23] **Yang, H. L. & Lin, H. C. (2015),** ***Integrating EMD, Chaos-based Neural Network and PSO for Financial Time Series Forecasting.*** *Economic Computation and Economic Cybernetics Studies and Research*; *ASE Publishing;* 49(1);
[24] **Zhang, G., Patuwo, B. E. & Hu, M. Y. (1998),** ***Forecasting with Artificial Neural Networks: The State of the Art.*** *International journal of forecasting*, 14(1), 35–62;
[25] **Zhang, T., Ramakrishnan, R. &Livny, M. (1996, June),** ***BIRCH: An Efficient Data Clustering Method for Very Large databases.*** *In ACM Sigmod Record* (Vol. 25, No. 2, pp. 103-114). ACM.



**DOI: 10.24818/18423264/54.3.20.02**