applied sciences

MDPI

Article

A Hybrid Framework Using PCA, EMD and LSTM Methods for Stock Market Price Prediction with Sentiment Analysis

Krittakom Srijiranon, Yoskorn Lertratanakham and Tanatorn Tanantong *

Thammasat Research Unit in Data Innovation and Artificial Intelligence, Department of Computer Science, Faculty of Science and Technology, Thammasat University, Pathum Thani 12121, Thailand
* Correspondence: tanatorn@sci.tu.ac.th

Abstract: The aim of investors is to obtain the maximum return when buying or selling stocks in the market. However, stock price shows non-linearity and non-stationarity and is difficult to accurately predict. To address this issue, a hybrid prediction model was formulated combining principal component analysis (PCA), empirical mode decomposition (EMD) and long short-term memory (LSTM) called PCA-EMD-LSTM to predict one step ahead of the closing price of the stock market in Thailand. In this research, news sentiment analysis was also applied to improve the performance of the proposed framework, based on financial and economic news using FinBERT. Experiments with stock market price in Thailand collected from 2018–2022 were examined and various statistical indicators were used as evaluation criteria. The obtained results showed that the proposed framework yielded the best performance compared to baseline methods for predicting stock market price. In addition, an adoption of news sentiment analysis can help to enhance performance of the original LSTM model.

Keywords: hybrid framework; stock market price; principal component analysis; long short-term memory; empirical mode decomposition; sentiment analysis

check for updates

Citation: Srijiranon, K.; Lertratanakham, Y.; Tanantong, T. A Hybrid Framework Using PCA, EMD and LSTM Methods for Stock Market Price Prediction with Sentiment Analysis. *Appl. Sci.* **2022**, *12*, 10823. https://doi.org/10.3390/app122110823

Academic Editors: Yujin Lim and Hideyuki Takahashi

Received: 30 September 2022
Accepted: 22 October 2022
Published: 25 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Predicting stock price behavior is an investor's goal in order to make the correct decision. A stock trader is a type of investor who always attempts to profit from the purchase and sale of stock. Therefore, this sort of investor must predict stock price changes to make the right decision on whether to sell or hold the stock they currently own. To earn money, stock traders must purchase stocks whose prices are expected to grow over the predicted period and sell stocks whose prices are dropping. If stock traders predict trends in stock prices correctly, they will have the potential to make a profit. Thus, predicting stock price trends is very important for stock traders' decision-making. However, the stock market shows highly complex trends. It is influenced by a wide range of economic factors such as Market Capitalization (MAC), general economic conditions, sentiment indices of social media and financial news [1,2]. Therefore, stock market prediction is known as one of the most challenging issues in time series prediction due to noise and volatility characteristics [3].

Previous research on predicting stock prices with effective machine learning models has largely been divided into two main approaches. The first approach aims to propose a prediction model using only historical stock data as input features and the second approach aims to apply related features to create models, including external indicators (e.g., news sentiment and social sentiment) and technical indicators.

For the first approach, there are a vast number of methodologies used to create predicting models. The common techniques include Artificial Neural Networks (ANN), Support Vector Machine (SVM), Auto Regressive Integrated Moving Average (ARIMA), etc. In addition, ANN has various structures for each data type such as Recurrent Neural

Networks (RNN) for time series data and Convolutional Neural networks (CNN) for image and video data [4,5]. Recently, Long Short-Term Memory (LSTM), a type of RNN, has attracted great attention as a result of the rapid growth of machine learning in the field of time series prediction, due to its ability to solve long-term dependence [6,7].

For the second approach, financial prediction based on machine learning techniques frequently adopts technical analysis to construct input features. Over 20% of financial market prediction models utilize technical indicators as input features [8]. Therefore, many researchers have tried to demonstrate that media sentiment affects stock prices and use it as an input feature to create a prediction model. A research article presented in [9] proposed models using financial news and technical indicators to predict intraday directional movements in the stock price of Chevron Corporation (CVX). A research article presented in [10] proposed a method for predicting stock market future patterns by using news and social media.

However, an original single machine learning method is not effective enough to predict stock prices. Additional processes and methods are required to increase model performance. The Empirical Mode Decomposition (EMD) proposed by [11] is a popular method to apply in order to transform input features before creating a prediction model. The EMD is a method to decompose the signal into physically meaningful components, called Intrinsic Mode Functions (IMFs). The EMD can analyze non-linear and non-stationary time series data by decomposing them into different resolutions of components. The trends of data are extracted and non-linear and non-stationary eliminated. The prediction model which applied EMD to raw data in data preprocessing outperformed the prediction model which did not [12,13]. Therefore, many researchers have proposed hybrid models based on the EMD method to improve their prediction models. A research article presented in [14] proposed a multistep-ahead predicting methodology that combines EMD and Support Vector Regression (SVR) for the prediction of the S&P 500. In addition, a hybrid model that combines EMD and BiLSTM to enhance $PM_{2.5}$ concentration prediction performance was proposed by [15].

The EMD method has proved to be useful in decomposing the components from non-linear and non-stationary signals. However, EMD retains the problem of mode splitting and mode mixing [16]. To address this, advanced versions of EMD have been proposed. Ensemble Empirical Mode Decomposition (EEMD) for a noise-assisted method was proposed by [17]. The idea of the EEMD consists of adding different series of white Gaussian noise into the original signal. However, EEMD still has numerically negligible errors and, when different types of white Gaussian noise are added, the number of IMFs can alter [18]. To resolve this problem, Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) was proposed by [19]. In this method, at each stage of the decomposition, a different noise is added to the residue of the current stage instead of white Gaussian noise [20]. The advanced versions of EMD can be used in various applications such as a predicting approaching wind speed based on EEMD and LSTM [21], a predicting approach for crude oil prices based on CEEMDAN, and LSTM with news sentiment index developed by [22].

In this research, a hybrid framework based on the combination of Principal Component Analysis (PCA), EMD and LSTM is proposed to predict one step ahead the closing price of the stock market in Thailand. The proposed framework is divided into two parts: the features engineering part and the prediction model part, with a total of five processes. Concretely, first, the news sentiment index score is created by using Financial Sentiment Analysis with Bidirectional Encoder Representations from Transformers (FinBERT). After that, PCA is used to reduce the dimension of the technical indicator and as an input feature for the prediction model. Next, the closing price of the stock market is decomposed into several IMFs via EMD. LSTM is applied to predict each IMF along with the news sentiment score and principal components from PCA. Finally, the prediction values of each IMF are composed to obtain the final closing price of the stock market.

2. Background Theories

This section describes related theories used in this research. It divides into seven sub-sections. The first two sub-sections deal with processes to create input features including feature transformation and sentiment analysis. The next two sub-sections cover processes to create a prediction model including Empirical Mode Decomposition and Long Short-Term Memory. Finally, the last three subsections look at statistical methods to check model performance including time series cross-validation, performance metrics and the Augmented Dickey-Fuller Test.

2.1. Feature Transformation

The Curse of Dimensionality basically means that the error increases along with the number of features. In other words, increasing the number of features does not always improve accuracy. Nowadays, this concept is applied in the fields of machine learning. In theory, increasing the dimensions can add more information to the dataset and improve its quality. Nevertheless, it rarely helps improve model performance in practice because real-world data contains more noise and redundancy [23].

The model is likely to underfit when a dataset does not have enough features. On the other hand, it is likely to overfit when the dataset has too many features. Thus, many dimensionality reduction methods have been proposed to overcome this limitation. Dimensionality reduction is a method to eliminate some features of the dataset and create a restricted set of features that contains all the data needed to predict more efficiently and accurately. There are two methods of dimensionality reduction including feature selection and feature transformation. The key difference between them is that feature selection keeps a subset of the original features, whereas feature transformation creates a new feature that catches most of the important data.

Principal Component Analysis (PCA) is one of the most well-known techniques for depletion reduction [24]. PCA is a feature transformation method used to reduce the dimension of massive data sets by transforming many variables to fewer, while retaining most of the information in the large set. This technique saves resources for running models and increases accuracy [25].

In the field of stock prediction, since technical indicators depend on trend, volatility, volume, momentum and daily returns, they can generalize to various scenarios. PCA can consider a high number of technical indicators as input features without encountering the curse of dimensionality [26]. The advantages of PCA can be applied in various data sources and applications such as tourist behavior analysis [27] and offshore wind turbines selection [28]. In addition, some research indicates that combining machine learning and PCA results in significant model improvement, particularly in comparison to mature dimensionality reduction techniques [29]. The basic steps of PCA are as follows:

- The first step is normalization of the original data to ensure that each set contributes equally to the analysis. Mathematically, the normalization equation is represented as (1) where x_{min} and x_{max} denote the minimum and maximum value of a feature, x denotes an original value and $x_{normailzed}$ denotes a new value.

$$x_{normailzed} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (1)$$

- The second step is establishing a covariance matrix according to the normalized data matrix. Since the dataset is n-dimensional, this will result in an $n \times n$ covariance matrix represented as matrix A.
- The third step is to calculate the eigenvectors and eigenvalues of the covariance matrix to identify the principal components. The eigenvalues (λ) of matrix A are found by solving (2), where I denotes the same dimensional identity matrix as A, which is an

essential requirement for matrix subtraction. For each λ, a corresponding eigenvector (v) can be found by solving (3).

$$det(\lambda I - A) = 0 \quad (2)$$

$$(\lambda I - A)v = 0 \quad (3)$$

- The last step is decreasing the original matrix by sorting eigenvectors with corresponding eigenvalues from largest to smallest. The eigenvector with the highest eigenvalue becomes the principal component of the data. After that, first p eigenvalues are chosen to reduce the dimensions and then principal components are received.

2.2. Sentiment Analysis

Sentiment Analysis is a method for defining whether data are positive, negative, or neutral by using Natural Language Processing (NLP). Sentiment analysis is commonly used on textual data to examine the attentions, feelings, behaviors, decisions and emotions of persons who are either the speaker or writer concerning the target topics. The basic task in sentiment analysis is grouping texts in sentences or documents. The grouping of texts are determined by the opinions of people which are either positive, negative, or neutral.

Sentiment analysis techniques can be categorized into three approaches: lexicon-based approaches, machine learning-based approaches and hybrid approaches. First, lexicon-based approaches are a method of using a lexicon to perform sentiment classification by calculating the weighting of labeled words and counting. Second, machine learning-based approaches are a method of using machine learning techniques, for example, Naive Bayesian and Support Vector Machine, which are considered as standard machine learning techniques. The input of the model includes lexical features, sentiment lexicon-based features and parts of speech [30]. Last, hybrid approaches are methods that use the aggregation of both lexicon-based and machine learning techniques [31]. In addition, sentiment analysis can generate profits for investors because it can help to make decisions [32].

Financial Sentiment Analysis with Bidirectional Encoder Representations from Transformers (FinBERT) proposed by [33] is a language model based on Bidirectional Encoder Representations from Transformers (BERT) for financial NLP tasks. The FinBERT model includes two phases: pre-training and fine-tuning. During the pre-training phase, the FinBERT model constructs a large variety of pre-training objectives to help better capture language knowledge and semantic information. This phase trains the BERT language model in the finance domain, using a large financial corpus and a general corpus. During the fine-tuning phase, datasets for financial sentiment classification are labeled. The main sentiment analysis dataset is Financial PhraseBank. Researchers extracted 4845 sentences from the dataset with financial terms. Then, 16 experts and master students with finance backgrounds labeled the data with sentiments including positive, negative and neutral. The FinBERT model will provide a polarity score for a given text and SoftMax outputs for one of three labels: positive, negative, or neutral.

2.3. Empirical Mode Decomposition (EMD)

EMD proposed by [11] is used to divide a signal without leaving the time domain. It can be equated to other analysis methods such as Fourier Transformation and Wavelet Decomposition. The EMD is beneficial for analyzing natural signals and it often applies to non-linear and non-stationary situations.

The EMD distinguishes the complexity of the original signal into a series of Intrinsic Mode Functions (IMF) with amplitude and a residual difference. IMFs satisfy the following two conditions:

1. The IMFs have only one extreme between zero crossings. In another word, the difference in number of maxima and minima is at most 1.
2. The mean of the wave of IMF is zero.

The EMD decomposes the signal into IMFs through a sifting process. As shown in Figure 1, the sifting method can be explained using the following algorithm. Decompose a data set $x(t)$ into IMFs $x_n(t)$ and a residual $r(t)$, as a result of which the signal can be described by (4)

$$x(t) = \sum_n x_n(t) + r(t) \tag{4}$$

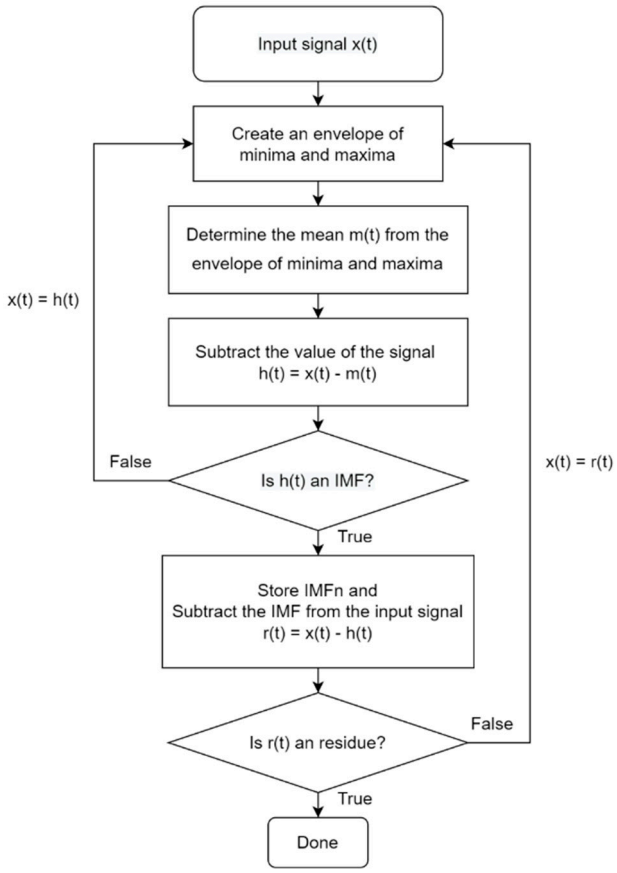


Figure 1. A flowchart of sifting processes.

2.4. Long Short-Term Memory (LSTM)

Deep learning is a type of machine learning that simulates the process of the human brain in terms of data and pattern formation for making decisions. The number of architectures and algorithms used in deep learning is wide and various [34]. Countless deep learning architectures such as Recurrent Neural Network (RNN) have been applied to NLP [35]. RNN is a variant of the Artificial Neural Network (ANN) which is designed to handle tasks with sequence data. The idea of RNNs is to make use of the output from the previous state as an input to the next state. This allows the model to recognize the pattern of the input sequence. RNN has the benefit of using past data to predict future events. As a result, everything that has occurred in the past will have an influence on the future. However, RNN is ineffective for very long-term dependencies. This is due to the exponentially decreasing gradients and the decay of information for long-term dependencies. This problem is called the vanishing gradient problem.

LSTM proposed by [36] is an improved version of RNN, avoiding the encountering of problems. LSTM is specifically modeled to manage tasks involving long-term dependencies information because it has a capacity to forget irrelevant information or store information for longer periods of time with memory cell support. The LSTM has a chain-like structure consisting of several subunits joined together. The unit of the LSTM architecture is a block memory with memory cells. These memory cells have three structures to control

information flow: forget gate layer, input gate layer and output gate layer. The forget gate layer determines what information from the previous cell is fed onto the current cell. The input gate layer determines the relevant information to update the cell state. The output gate layer determines the output value for the next hidden state based on the input and memory of the block [37].

Furthermore, LSTM is appropriate for time series prediction because it can learn and remember long-term memory topics such as market movement [38]. Advanced versions of LSTM can be used for various applications such as energy consumption [39], gas field production [40], chatbot messages classification [41] and rice export price prediction [42].

2.5. K-Fold Cross-Validation with Time Series Data

Cross-validation is a data resampling method for estimating the actual prediction performance of models and tuning hyper-parameters. In order to overcome the problem of overfitting, cross-validation is used to check overall model performance to detect this problem. In addition, it is used to adjust appropriate hyper-parameters, such as the appropriate batch size and epoch in ANN model.

K-fold cross-validation is one of the methods. The procedure begins by randomly splitting the dataset into folds of equal size. The model is trained by using k-1 folds that represent the training set. Then, the trained model is applied to the remaining fold, which represents the testing set and the performance of the model is evaluated. This procedure is repeated until every fold is used as a testing set. The final metrics are the average of the errors obtained in each fold [43].

However, K-fold cross-validation cannot be utilized in the case of time series due to randomly splitting the dataset, because it is irreconcilable in the real world to use values from the future to forecast values from the past. The K-fold Cross-validation with Time Series Data has a different procedure. The idea is that each observation is the first used as a testing set and then added to the training set of the model [44]. The procedure begins by splitting the dataset into k folds of equal size. In the initial iteration, only the first k folds are used as a training set and the next folds are used as a testing set. In the next iteration, the old training set and testing set are merged and used as a training set. This procedure continues until the last fold is tested. The comparable training set only contains observations that occurred before the testing set observation. Hence, no future observations are used to make the prediction [45].

2.6. Performance Metrics

In this research, the performance metrics are separated into two main parts. The first part evaluates the performance of the financial news sentiment analysis model and the second part evaluates the performance of the stock price prediction model.

In the first part, confusion metrics are used to validate the financial news sentiment analysis model, in order to compare performance with other models using precision, recall, $F1$-score and accuracy from (5)–(8), respectively where TP denotes true positive, TN denote true negatives, FP denotes false positives, FN denotes false negatives and n denotes the number of observations.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 - score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{7}$$

$$Accuracy = \frac{TP + TN}{n} \tag{8}$$

In the second part, to evaluate the performance of the stock price prediction model, the Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean

Absolute Error (MAE) and coefficient of determination (R^2) statistics are used to compare performance with the other models (9)–(11), respectively, where y_i denotes actual value, $\hat{y}_i$ denotes predicted value and $\overline{y}_i$ denotes the mean of y value.

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

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \times 100 \tag{10}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{11}$$

$$R^2 = 1 - \frac{\sum(\hat{y}_i - \overline{y}_i)}{\sum(y_i - \overline{y}_i)} \tag{12}$$

2.7. Augmented Dickey-Fuller (ADF) Test

An ADF test is a fundamentally statistical significance test for determining whether time series is stationary or non-stationary. The ADF test is suitable for testing the stationarity of a time series because it belongs to a category of tests called the Unit Root Test. It exists in a time series of ρ values calculated by (13) where Y_t denotes the value of the time series at time t, X'_t denotes exogenous variables ϵ_t denotes a white noise, and ρ and δ denotes estimated parameters.

$$Y_t = \rho Y_{t-1} + X'_t \delta + \epsilon_t \tag{13}$$

If $|\rho| \geq 1$, Y is a non-stationary series while $|\rho| < 1$, Y is a stationary series, as a result, the stationarity hypothesis can be determined by determining if the total value of ρ is strictly smaller than 1.

The ADF test expands the Dickey-Fuller test (DF) equation to include high order regressive process in the model. The DF test is a unit root test that tests the null hypothesis. The standard DF test is carried out by subtracting from both sides of the Unit Root Test from (14) where α denotes a constant equal ρ, β denotes a coefficient.

$$\Delta Y_t = \alpha Y_{t-1} + X'_t \delta + \epsilon_t 1 \tag{14}$$

The null and alternative hypotheses are evaluated using the conventional t-ratio for α. The ADF equation, which is a DF equation but includes a high-order regressive process in the model, can be calculated as (15),

$$\Delta Y_t = \alpha Y_{t-1} + X'_t \delta + \beta_1 \Delta Y_{t-1} + \beta_2 \Delta Y_{t-2} + \ldots + \beta_p \Delta Y_{t-p} + v_t \tag{15}$$

The t-ratio is then used to test the same null hypothesis as the DF test. Assuming the null hypothesis involves the presence of unit root, that is $\alpha = \rho - 1$, the ρ-value derived from the equation (13) should be greater than the significance level and the statistical test value be greater than the critical value in order to reject the null hypothesis. As a result, the series is inferred to be non-stationary [46,47].

3. Materials and Methods

This section describes the proposed hybrid framework for predicting stock market price in Thailand. It divides into two sub-sections, data collection and system architecture.

3.1. Data Collection

This research uses two types of data to create the prediction model, financial news data as text data and historical data as numerical data. The details of each type are described.

3.1.1. Financial News Data

In this research, the news was collected from a total of six news agencies. The news content mainly focuses on the fields of finance and economics. The news texts were crawled from the source websites using the web scraping method, resulting in a total of 12,667 articles from 21 February 2013, to 24 February 2022. Due to the insufficiency of news data in some time periods, the news data from 24 February 2018 to 24 February 2022 was chosen to guarantee data continuity. Lastly, after removing this part of the news, the dataset had a total of 11,386 news texts and an average of 8 news samples per day. In addition, this research used 1500 labeled pieces of financial news in Thailand during the fine-tuned phase in order to improve the FinBERT performance according to Thai Financial news sentiment analysis.

3.1.2. Historical Data

This research used a one-step-ahead prediction to testify to the prediction preciseness of the proposed model on the closing price of the stock market in Thailand. The dataset obtained from investing.com included close, open, high, low and volume. The range of closing price of the stock market is from 24 February 2018 to 24 February 2022. Only the data from trading days was used for research. The value of data in the selected period is visualized in Figure 2.

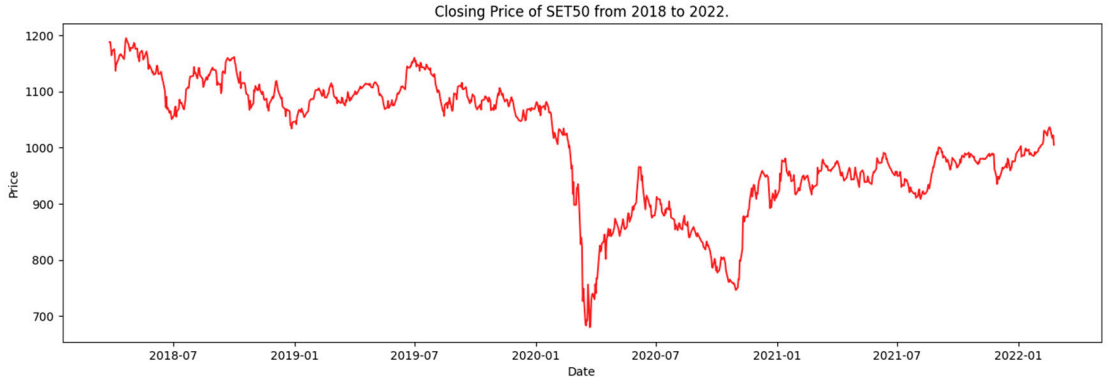


Figure 2. The daily closing price of the stock market in Thailand.

The statistical analysis of the closing price of the stock market, including the amount of data contained in the closing index and the minimum, maximum, mean, standard deviation and ρ-value of the ADF test, is shown in Table 1. To calculate the ADF test, Python module Statsmodels is used in this research [48]. There was a significant difference between the maximum and minimum values; furthermore, the closing prices are extremely volatile due to the high standard deviation.

Table 1. Descriptive statistics of the closing price.

Statistics Indicators	Value
Count	953
Average	1006.859
Minimum	1194.870
Maximum	680.070
Standard deviation	107.833
ADF test	0.469

In this proposed framework, the ADF test is used to check the stationary or non-stationary nature of time series data. If the ρ-value of the ADF test result (as presented in Table 1) was greater than a threshold of 0.05, which fails to reject the null hypothesis, it indicates that the dataset was highly volatile and non-stationary. Apparently, this dataset

was suitable to use with the EMD method because it was an effective method for analyzing non-linear and non-stationary time series.

In addition, this research selected other input features related to the closing price of the stock market, called technical indicators. A total of nine technical indicators are selected. Categories and names are shown in Table 2.

Table 2. Categories and names of technical indicators.

Categories	Technical Indicators
Trend	Detrended Price Oscillator
Volume	Negative Volume Index
Volatility	Bollinger Bands Width
Trend	Simple Moving Average
	Detrended Price Oscillator
Momentum	The Stochastic RSI
	Price Rate of Change
	Percentage Volume Oscillator–Histogram
	Williams %R

3.2. System Architecture

The purpose of this research is to propose a hybrid framework for the closing price of the stock market in Thailand using a combination of PCA, EMD and LSTM. The overall architecture of the proposed system is shown in Figure 3. The system was divided into two parts: the feature engineering part and the prediction model part.

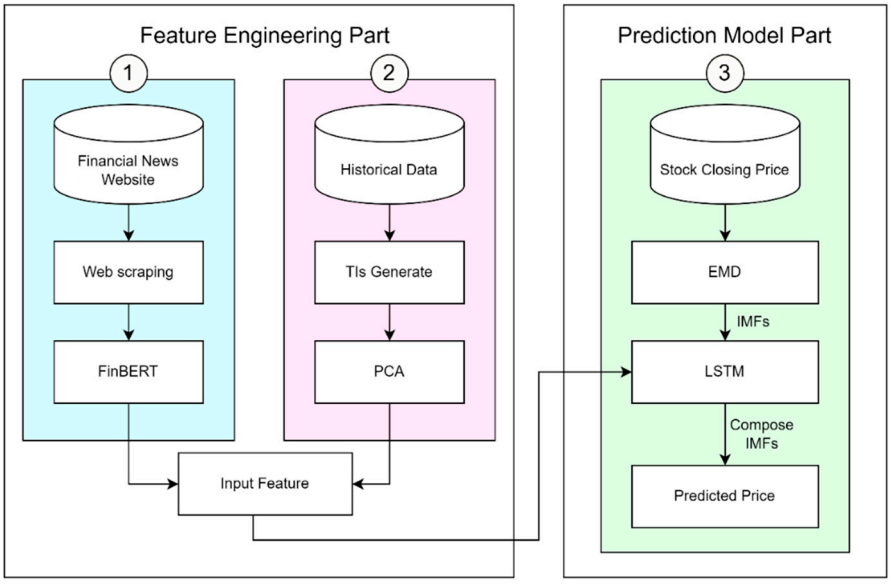


Figure 3. An overall view of the proposed hybrid framework.

3.2.1. Feature Engineering

The process of developing input features for prediction models is described in this section. In this research, the input feature was a combination of technical indicator components and news sentiment score to create a better predictive model.

- News Sentiment Score: In news sentiment analysis, FinBERT was used to generate the sentiment score. In order to efficiently use FinBERT for Thai news analysis, FinBERT with Thai news fine-tuning was implemented. This method trained an original FinBERT with extra data, which is Thai news in this research. According to Figure 4, there were six steps in the FinBERT with Thai news fine-tuning modeling process. The first step was to collect headlines from news sources. In this research, news sources

in the financial and economic fields were collected from a total of six news agencies by using the web scraping method. After that, the acquired dataset was cleaned and text preprocessed, including text to lowercase, removal of punctuations and removing extra spaces. The next step is to manually label the news into three categories, negative and neutral. Then, the dataset is randomly divided into a training set with 80% and a testing set with 20%. The training set was used to add FinBERT supervised fine-tuning for sentiment analysis to fit a particular task in its training stage. Finally, the model was tested with the testing data set and performance measured with an F1-score and accuracy.

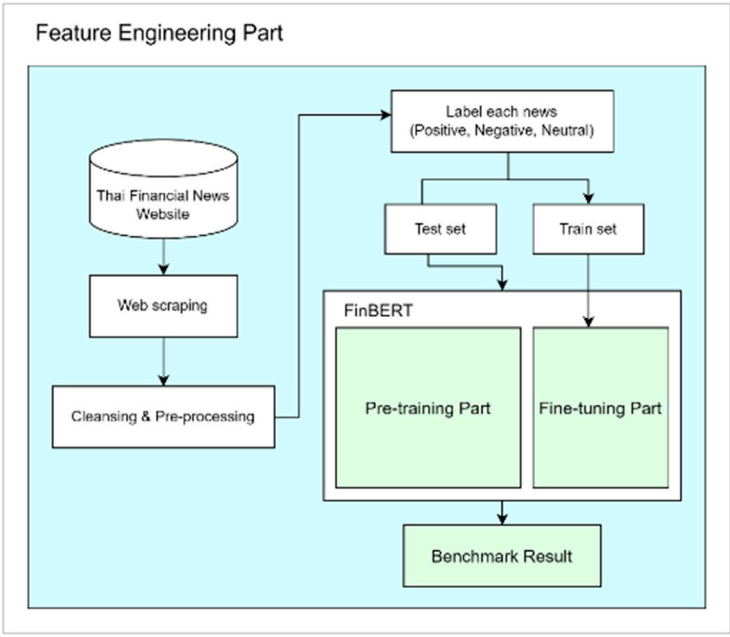


Figure 4. A sentiment analysis modeling process.

- Technical Indicator Component: In this research, nine technical indicators are selected and used as input features of the proposed model. In the simple terms of the curse of dimensionality, the more features there are, the higher the risk of overfitting. To solve this issue, PCA is adopted to decrease the feature space along with consideration of a set of principal features. In order to create a principal component from PCA, there are steps to follow as shown in Figure 5. Firstly, the historical data was obtained from investing.com including open, high, low, close and volume data. After that, the "ta" package from [49] was used to generate technical indicators. Then, the technical indicators were normalized before reducing the dimensions of the data by using PCA. The result of PCA was the principal component, which in this research is done by starting from the first principal component until the sum of the explained variance ratio is greater than 80%. Therefore, this indicates that 80% of the information in the technical indicator can be explained by the n-principal component.

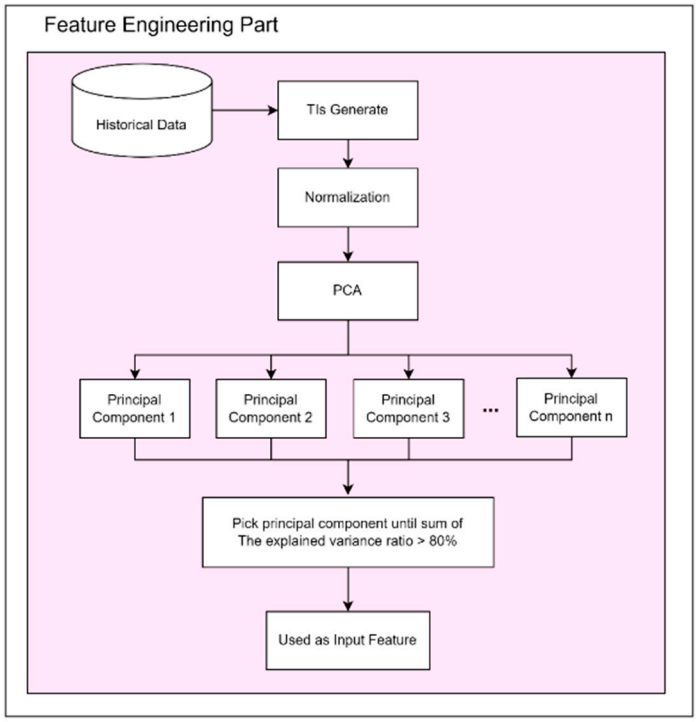


Figure 5. Technical Indicator Component Creation.

3.2.2. Prediction Model

An integrated prediction model based on the combination of EMD and LSTM is proposed to maximize the prediction effectiveness and minimize the complexity of calculations. The proposed model is shown in Figure 6 consisting of the following four steps:

- Firstly, the EMD algorithm was applied to decompose the original stock closing price time series into several independent IMF components and one residual component.
- Secondly, the news sentiment score and the principal component from the feature engineering part were included as input features to the model.
- Thirdly, the LSTM model was used as the prediction tool for each IMF component. Consequently, the corresponding components acquired the prediction values. The LSTM was trained individually by each IMF; thus, the network parameters, epoch and batch size are specially tuned for each IMF. This is the significant difference that makes a hybrid EMD-LSTM model better than a single LSTM model.
- Finally, each predicted IMF was combined using (4) to get the final predicted stock closing price after obtaining the predicted results of the IMFs. Then, the results were compared with other models using performance metrics.

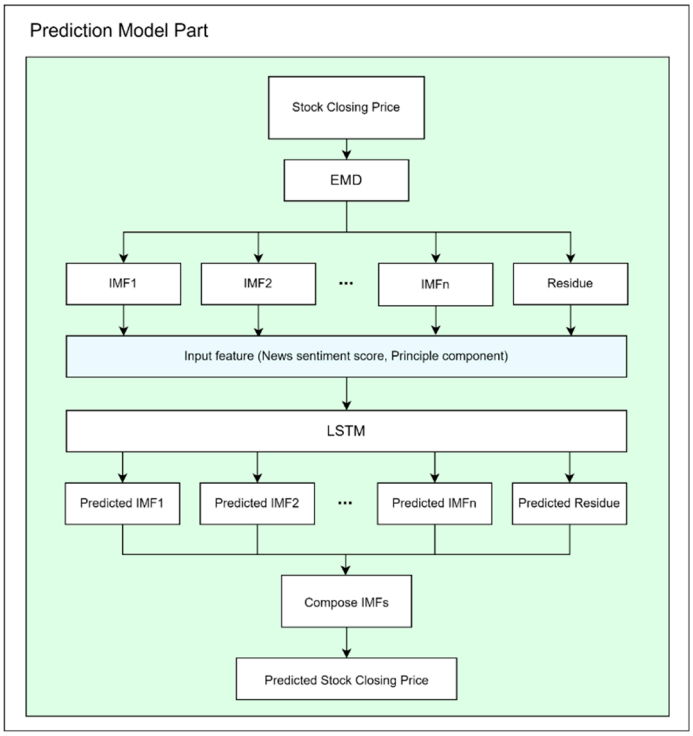


Figure 6. The process of the prediction model.

4. Experimental Methods and Results

In this section, there are four sub-sections. The decomposition results of EMD re discussed in the first sub-section. The comparison results between the FinBERT with the Thai news fine-tuning model, an original FinBERT and other sentiment analysis models are presented to verify the effectiveness of the FinBERT in the second sub-section. The different combinations of the PCA, EMD, LSTM and news components are presented to validate the proposed model from several perspectives in the third sub-section. The comparison results between different advanced versions of EMD are presented to verify the best version of EMD in the last sub-section.

4.1. Experimental Methods and Results of the Decomposition Component by EMD

For construction of the prediction model, the closing price of stock data as historical data was transformed into the new data using EMD. As shown in Figure 7, this experiment demonstrates decomposing results to create IMFs using EMD. The seven IMFs were decomposed from the original closing price sequence and the results of the IMFs' scale from high to low frequency. However, the number of IMFs are different depending on the raw data. The processes of EMD are repeated until there are only one global maxima and minima value showing on IMF 7 in Figure 7. The number of IMFs will be changed if the raw data is changed. On the other hand, the number of IMFs will still be of the same value when applying EMD to the same data.

The result shows that it can be divided into three groups. The first group are high-frequency components in the original data. This group was represented by the first few IMFs with a lot of noise. The second group are middle-frequency components. represented by the center IMFs with a medium noise. The last group are low-frequency components. This group was represented by the last few IMFs with little noise. Moreover, the last IMF is comparable to the trend of a stock. It is common to hypothesize that the LSTM can accurately predict low-frequency IMFs, but it will struggle with high-frequency IMFs. To maximize the prediction efficiency, the LSTM is trained individually by each IMF. Thus, the hyper-parameter, the number of hidden layers and weights are different for each IMF. This is the significant difference making a hybrid EMD-LSTM model perform better than a

single LSTM model, which is applied directly to the original closing price time series, with characteristics of noise and volatility.

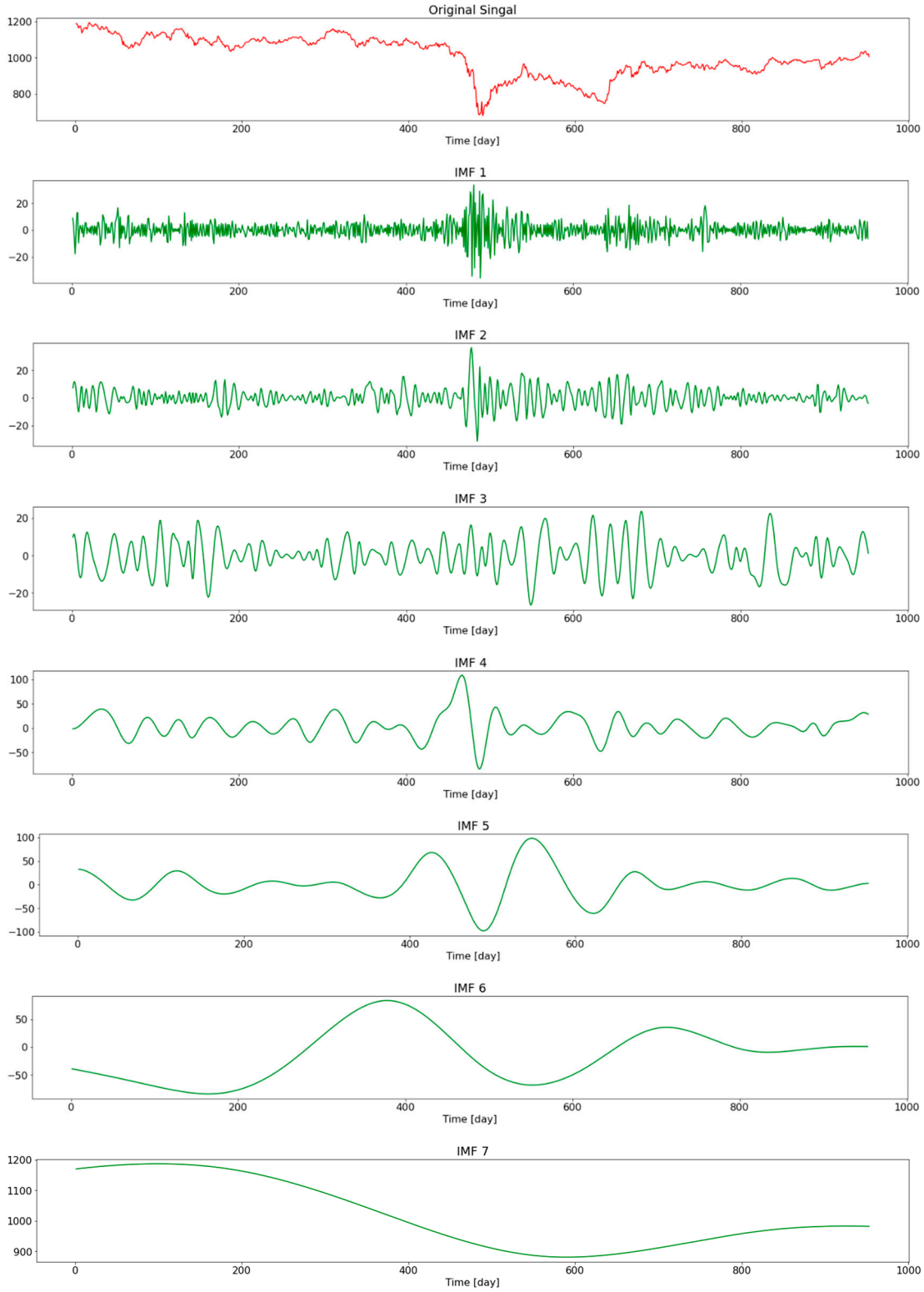


Figure 7. The original closing price and decomposed IMFs.

IMFs were obtained by subtracting from the original closing price, so the summation of all IMFs is totally identical to the original. For this reason, the summation of the prediction results of all IMFs can be considered as the prediction result for the original closing price.

4.2. Experimental Methods and Results of News Sentiment Analysis

FinBERT with Thai news fine-tuning was used in the financial news sentiment analysis model in the feature engineering part of the proposed model. In order to assess the efficacy of Thai news analysis, FinBERT with Thai news fine-tuning was compared to the original FinBERT, which is FinBERT with default, and other popular sentiment analysis models, such as Vader [50] and Text-blob [51].

This research manually annotated the financial news dataset. The annotated dataset was random from news data to label with three classes of sentiments: positive, negative and neutral, totaling 1500 articles. The annotated dataset was used for training, FinBERT supervised fine-tuning and model performance testing. The top 80% of the data is used as the training dataset for supervised fine-tuning training and the remaining 20% was used as the testing dataset to evaluate the model performance. Moreover, each class has an equal number of examples in the testing set. Other models were used, similar to both training and testing datasets and to FinBERT with Thai news fine-tuning.

From Table 3, the result shows that the FinBERT with Thai news fine-tuning has the highest average accuracy and average F1-score of the compared models. When considering the F1-score in each class, the FinBERT with Thai news fine-tuning has the highest value. Both FinBERT models perform similarly well when it comes to categorizing news as Class Negative. In classes Positive and Neutral, the FinBERT with Thai news fine-tuning has a moderate F1-score value. However, both Vader and Textblob have very low model performance for this dataset.

Table 3. Comparison results between the FinBERT with the Thai news model and other models for sentiment analysis.

Model	Model Performance Indicators				
	F1-Score				Accuracy (%)
	Positive	Neutral	Negative	Average	
Textblob	0.43716	0.48763	0.28358	0.40279	42.67
Vader	0.51777	0.49402	0.47368	0.49515	49.67
Original FinBERT	0.76724	0.73256	0.83673	0.77884	78.00
FinBERT with Thai news fine-tuning	0.80198	0.81319	0.84259	0.81925	82.00

4.3. Experimental Methods and Results of the Proposed Framework

The proposed framework was used for the closing price of stock market prediction. This framework contained many processes in both the feature engineering and prediction model part. Therefore, this sub-section was used to verify the efficacy of the proposed model in each process. A set of sensitivity experiments was established using various combinations of the EMD, LSTM, PCA and financial news components validating the proposed model from several perspectives. The experimental design can be seen in Table 4 and the output data for all experiments are the closing price of the stocks market in Thailand.

- Experiment 1 was a comparison result between the EMD-LSTM model and other prediction models. The purpose of this experiment is to apply the models to the original closing price directly without using additional input features. The comparison results between the proposed model and other models evaluate whether the EMD-LSTM model effectively improves the outcomes of prediction over state-of-the-art models in stock price time series modeling.
- Experiment 2 is a comparison of the effects of adding principal components and technical indicators to EMD-LSTM. This experiment applies an additional input feature to the proposed model, which is the original technical indicator and the principal component of PCA. The experiment aims to show whether the principal component from PCA can improve the prediction of EMD-LSTM. The comparison results comparing

using the principal components as input features and using the original technical indicators as input features examines whether the model, when applying PCA effectively, improves the outcomes of prediction due to the curse of dimension.

- Experiment 3 is a comparison of the effects of adding news sentiment score to prediction models. This experiment applied an additional input feature from FinBERT to the proposed model. The experiments are evaluated to identify whether applying a news sentiment score improves model performance.

Table 4. Details of the model and input features in each experiment.

Experiment	Model Name	Input Features
1	LSTM	Closing price
	ARIMA	Closing price
	EMD-LSTM	Closing price
2	TIs-EMD-LSTM [1]	Closing price + technical indicators
	PCA-EMD-LSTM	Closing price + PC [2]
3	LSTM with News	Closing price + News
	PCA-EMD-LSTM	Closing price + PC [2]
	PCA-EMD-LSTM with News	Closing price + PC [2] + News

[1] TIs is technical indicators, [2] PC is principal components from PCA.

4.3.1. Comparisons Result between the EMD-LSTM Model and Other Models

In this experiment, machine learning methods and the original closing price are applied to estimate the prediction performance. The EMD-LSTM, LSTM and ARIMA are used for comparison. Table 5 shows that the EMD method has great advantages in the closing price of stock market prediction, with MAE dropping by 56.13% when compared to LSTM and 85.67% when compared to the ARIMA model. Moreover, LSTM and ARIMA had a close model performance. This implies that a single model cannot impressively solve data patterns and make brilliant predictions. In addition, Figure 8 shows the predictive results of the LSTM and EMD-LSTM, revealing that the predicted values of the EMD-LSTM series visibly deviate from the original data.

Table 5. Results of stock price prediction from three prediction models.

Model	MAE	RMSE	MAPE	R^2
LSTM	7.2746	8.9451	0.7488	0.8925
ARIMA	7.9062	10.1950	0.7095	0.9328
EMD-LSTM	3.1047	3.7832	0.3197	0.9808

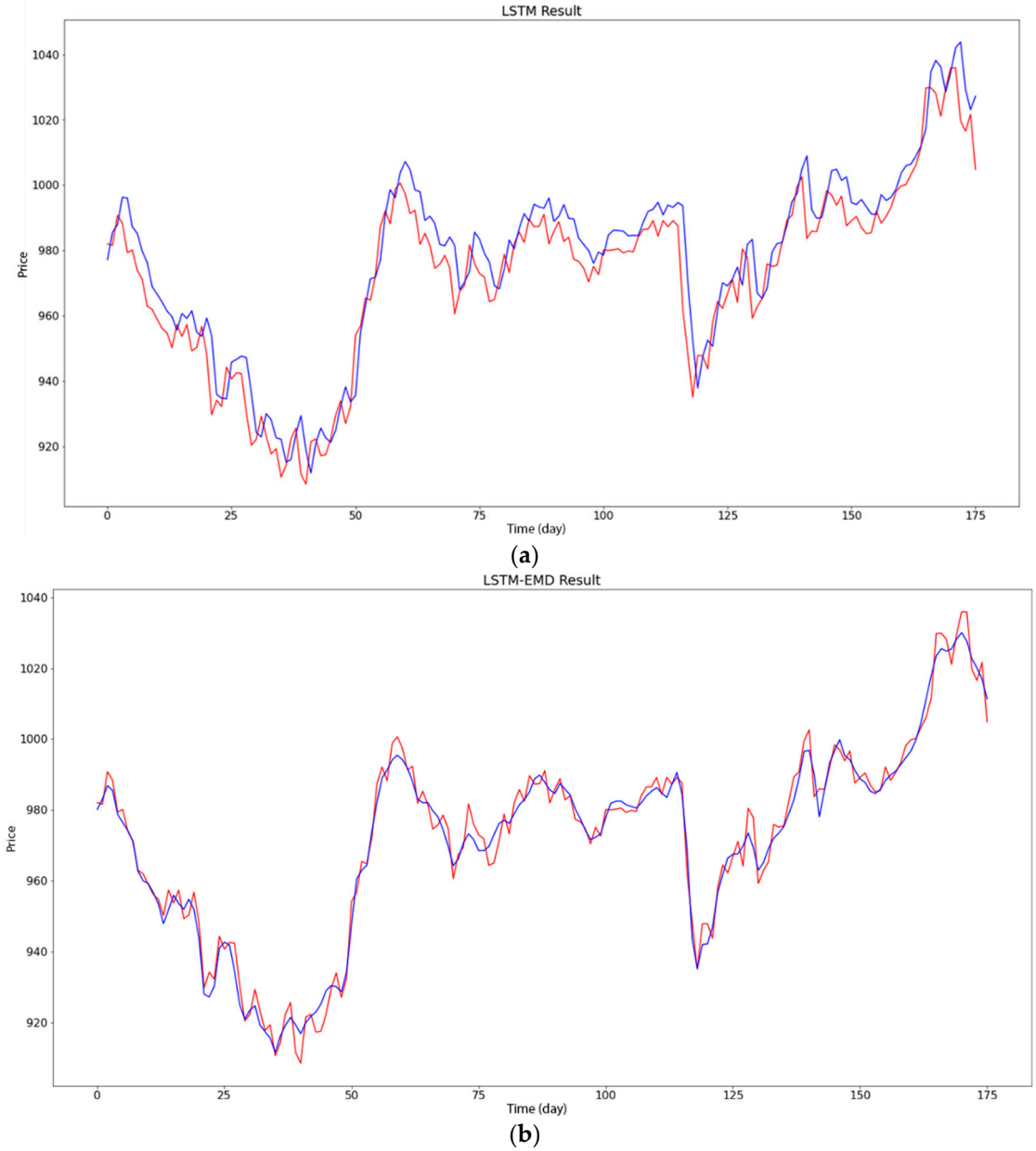


Figure 8. Results of stock price prediction from LSTM (**a**) and LSTM-EMD (**b**).

4.3.2. Comparisons of the Effects of Adding Principal Component and Technical Indicator to EMD-LSTM

From the previous experiment, the EMD-LSTM model outperforms when compared with the other prediction models. This experiment applied an additional input feature to the proposed model, which is the original technical indicator and the principal components from PCA. In order to make predictions with the EMD-LSTM model, individual IMFs were predicted with LSTM and the additional input feature. After tuning of the LSTM model, the optimal hyper-parameters were obtained to achieve the prime prediction results for the IMFs, as shown in Table 6. The batch size was between 8 and 32 while the epoch was between 150 and 300. In addition, the other settings of the LSTM model included hyperbolic tangent as activation function, ADAM as optimizer, mean squared error as loss function and learning with 0.001.

Table 6. Hyper-parameters of LSTM.

IMFs	Batch Size	Epoch
1	8	300
2	8	250
3	32	200
4	16	180
5	16	200
6	16	150
7	32	150

The experimental results of adding the input feature of each IMF are shown in Table 7. The results show that the principal components can improve the efficiency of the model and outperform the IMF and technical indicators in the first four IMFs. Nevertheless, after the fifth IMF, the model that uses only the IMF value outperforms the other models using additional input features, as can be seen in Figure 9. Meanwhile, models with a technical indicator as an input feature perform the worst across all IMFs. In addition, Figure 10 shows the IMF of closing price testing set prediction results. Due to the high frequency of the components, the prediction values of the first several IMF components explicitly diverge from the original data, but the prediction values of the last IMF nearly matched the original data.

Next, the prediction results of each IMF are assembled in order to compare the final closing price prediction results. In addition, the PCA-EMD-LSTM model used principal components as an input feature to predict IMF 1 to 4 but uses only the IMF value for IMF 5 to 7. From Table 8, the result shows that PCA-EMD-LSTM achieves the best prediction result, followed by a close second to the model that uses only the IMF value, whereas models using technical indicators as the input feature have the worst prediction results.

Table 7. Results of IMFs prediction from EMD-LSTM with additional input feature.

IMFs	Input features	MAE	RMSE	MAPE	R^2
1	IMF	3.0886	3.7415	1.4874	0.0186
	IMF + TIs	3.8096	4.7048	2.7588	−0.5518
	IMF + PCA	2.8581	3.5379	1.3411	0.1225
2	IMF	1.3146	1.5134	2.8782	0.7544
	IMF + TIs	1.9184	2.2518	2.8495	0.4562
	IMF + PCA	0.7432	1.0220	1.3655	0.8880
3	IMF	0.8446	1.0072	0.6026	0.9867
	IMF + TIs	0.8597	1.0726	0.5518	0.9849
	IMF + PCA	0.5220	0.6367	0.3536	0.9947
4	IMF	0.5653	0.8389	0.2236	0.9960
	IMF + TIs	1.1520	1.5367	0.1784	0.9865
	IMF + PCA	0.3113	0.4243	0.0622	0.9990
5	IMF	0.4080	0.4232	0.3565	0.9972
	IMF + TIs	1.1124	1.4344	1.1847	0.9679
	IMF + PCA	0.6061	0.6862	0.6483	0.9927
6	IMF	0.2432	0.3083	0.0914	0.9948
	IMF + TIs	0.7971	0.9648	2.1934	0.9489
	IMF + PCA	0.9819	1.0147	3.9208	0.9435
7	IMF	0.0275	0.0309	0.0000	1.0000
	IMF + TIs	2.4216	2.4945	0.0025	0.9414
	IMF + PCA	0.7128	0.8066	0.0007	0.9939

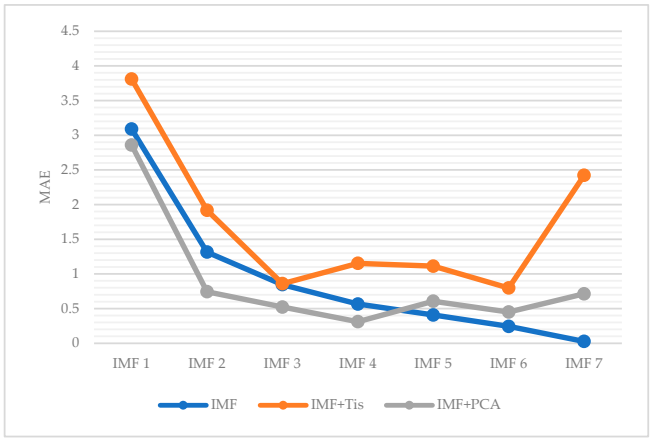
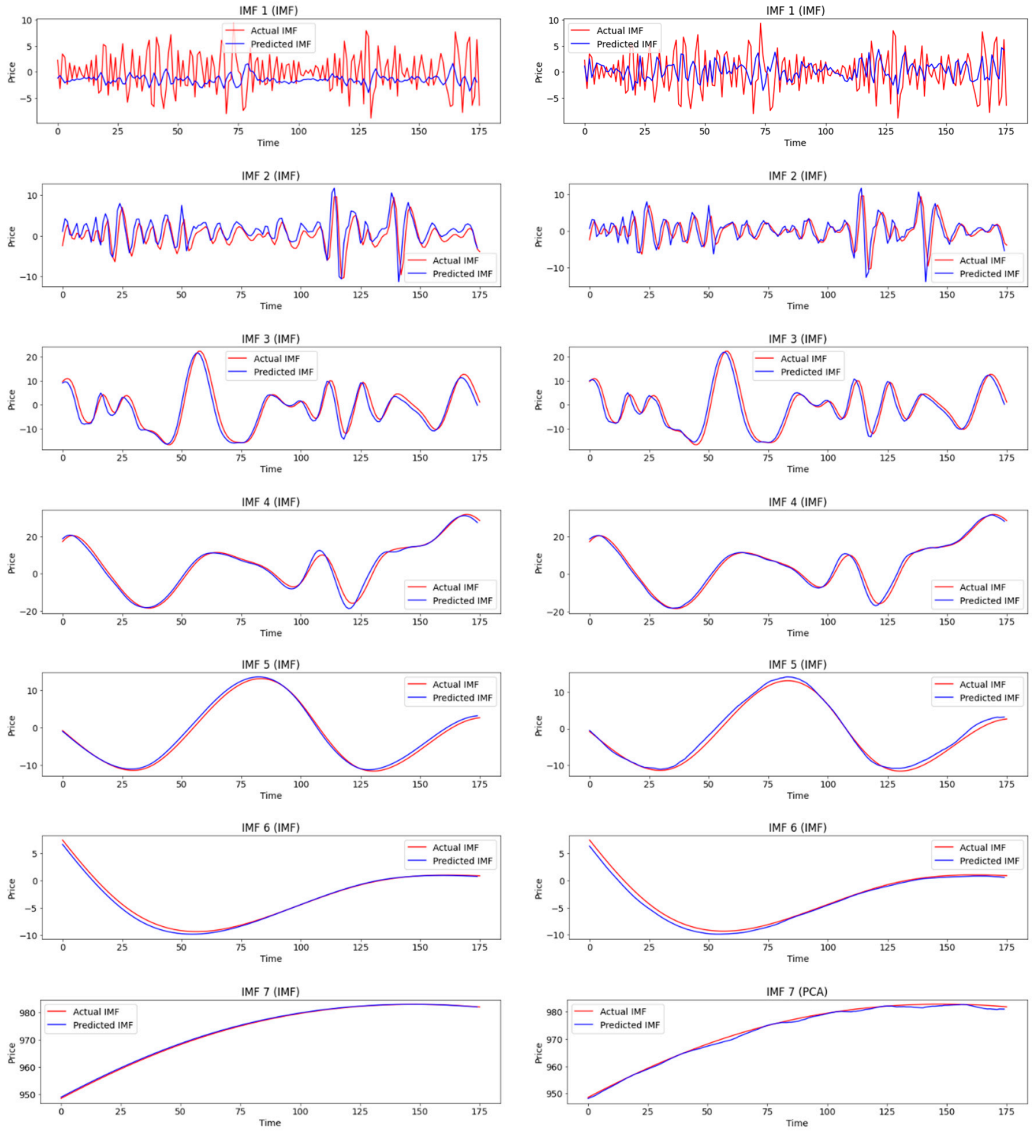


Figure 9. The MAE result applies an additional input feature to each IMF of the EMD-LSTM.



(**a**) (**b**)

Figure 10. Results of IMF prediction using only IMF value (**a**) and principal component (**b**).

Table 8. Results of EMD-LSTM with additional input features.

Model	MAE	RMSE	MAPE	R^2
EMD-LSTM	3.1047	3.7832	0.3197	0.9808
TIs-EMD-LSTM [1]	4.5122	5.6629	0.4636	0.9569
PCA-EMD-LSTM	3.0086	3.6896	0.3102	0.9817

[1] TIs is technical indicators.

4.3.3. Comparisons of the Effects of Adding News Sentiment score to Prediction Models

The preliminary model in the previous experiment used only input features from historical data. In this experiment, the news sentiment score was applied to a prediction model to identify whether applying a news sentiment score improves the prediction model.

From Table 9, the result shows that the news sentiment score has great advantages as an input feature in stock price prediction, with MAE dropping by 20.82% when compared to a single LSTM. On the other hand, the prediction results become worse when the news sentiment score of PCA-EMD-LSTM is included compared with only PCA-EMD-LSTM. Evidently, it is better to ignore the news sentiments component part of the proposed model. However, the news sentiment score part can improve the model performance of the original model.

Table 9. Results of the prediction models using news sentiment.

Model	MAE	RMSE	MAPE	R^2
LSTM	9.0636	7.0307	0.7255	0.8896
LSTM with News	7.6459	5.7050	0.5842	0.8127
PCA-EMD-LSTM	3.0086	3.6896	0.3102	0.9817
PCA-EMD-LSTM with News	5.0747	4.1863	0.4310	0.9654

4.4. Comparison Results between Difference Advanced Versions of EMD

From the previous three experiments, the best architecture of the proposed model was PCA-EMD-LSTM. However, there are advanced versions of EMD such as EEMD and CEEMDAN. Therefore, this experiment changed the EMD part from the proposed model to both EEMD and CEEMDAN, called PCA-EEMD-LSTM and PCA-CEEMDAN-LSTM, respectively. To create the prediction model, closing price of the stock market in Thailand and principal components from PCA are used as input features.

The experiment result shows in Table 10 that the PCA-EMD-LSTM had the lowest model performance and PCA-EEMD-LSTM had a moderate model performance. On the other hand, using the EMD as a decomposition method is the most effective for prediction with PCA-LSTM. Finally, the PCA-EMD-LSTM architecture had the highest model performance and Figure 3 can exclude the news sentiment score part with blue background.

Table 10. Results of the prediction models with different advanced versions of EMD.

Model	Input features	MAE	RMSE	MAPE	R^2
PCA-EMD-LSTM	Closing price + PC [1]	3.0086	3.6896	0.3102	0.9817
PCA-EEMD-LSTM	Closing price + PC [1]	3.3064	4.0338	0.3410	0.9782
PCA-CEEMDAN-LSTM	Closing price + PC [1]	3.4806	4.2967	0.3574	0.9752

[1] PC is principal components from PCA.

5. Discussion

In order to verify the effectiveness of the proposed hybrid framework, several experiments on various factors were examined. The observation results are as follows:

- The EMD-LSTM model outperforms state-of-the-art benchmark models indicating that decomposition methods with EMD decrease the complexity of sequences and develop prediction performance. Moreover, EMD decomposed the original signal into

minor components based on their frequencies. In order to maximize the prediction effectiveness, the LSTM is trained individually by each component; therefore, the hyper-parameters and weights are different for each component. This is the significant difference that makes a hybrid EMD-LSTM model perform better than a single LSTM.

- The prediction result shows that PCA can help to reduce prediction errors in the first few IMFs when applying the principal components from PCA to the EMD-LSTM. This indicates that the PCA method creates useful features from technical indicators for improving IMF with high-frequency prediction. From the obtained results in Table 10, the PCA-EMD-LSTM achieves the highest prediction performance for the closing price of the stock market. However, based on the experimental results in Table 5, the MAPE value of the EMD-LSTM model is slightly higher than the obtained MAPE result of the PCA-EMD-LSTM. Therefore, further experiments on different datasets are required to verify the performance improvement of using PCA in the EMD-LSTM model.
- Applying the news sentiment score to the EMD-LSTM does not improve prediction results in every IMF. On the other hand, adding the news sentiment score can improve the original LSTM performance. This means that news sentiment can be used to predict the closing price of the stock market while it cannot be used to predict the decomposed component of a closing price of the stock market.

To increase the efficiency of this proposed framework, there are a number of gaps that need further development. For example, IMFs may be adaptively predicted by various traditional or hybrid machine learning models. Recently, a novel approach [52] to select effective machine learning model combination for time series forecasting was proposed. Based on the machine learning combination approach, it can be applied to this proposed framework for improving the prediction results of each IMF and the prediction of the closing price of the stock market. In addition, based on a recently published re-search study [53], an interesting decomposition method, i.e., a hybrid time series decom-position strategy (HTD), can be applied instead of EMD for further improvement of this proposed framework.

6. Conclusions

In this research, a hybrid framework based on the combination of PCA, EMD and LSTM is proposed to predict one step ahead of the closing price of the stock market. Moreover, the proposed model is capable of combining both historical and textual data as the input features. The overall design of the proposed system is separated into two parts: the feature engineering part and the prediction model part. The feature engineering part is used to create input features for the prediction model. There are two main processes: the finance and economics news sentiment score using FinBERT with Thai news fine-tuning and the principal components from technical indicators using PCA. The prediction model part is used to predict the closing price of the stock market. Historical data were decomposed into several IMFs via EMD. Next, LSTM was utilized to predict each IMF along with input features from the previous part. Finally, the prediction values of each IMF were used together to produce the final stock price prediction. Based on the results of the experiments, the proposed framework using PCA, EMD and LSTM had the best prediction performance for the closing price of the stock market. Moreover, based on the obtained experimental results in the LTSM model, the performance of the original LSTM is improved when applying news sentiment analysis.

Future research can be conducted in order to optimize the model's efficiency. For example, different machine learning algorithms can be adaptively selected for the different IMFs after decomposing the original data. This process may improve the prediction results of each IMF and the prediction of the closing price of the stock market. In addition, the effect of different sets of technical indicators can be explored to find the best set for IMF prediction.

Author Contributions: K.S., Y.L. and T.T. did comprehensive searching for research background. K.S. and Y.L. collected the experimental data. T.T. developed the research framework. K.S. and T.T. designed the research methodology. Y.L. did programming and implementation. All authors evaluated the research framework. K.S. and T.T. performed data analysis and wrote the manuscript. K.S. submitted the manuscript for publication and communicated with the journal editor. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data of this work are available from investing.com and six news agencies in Thailand upon request.

Acknowledgments: This work was supported by Thammasat Research Unit in Data Innovation and Artificial Intelligence.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pongsupatt, T.; Pongsupatt, A. Factors affecting stock price: The case of Thailand stock exchange SET100 index. In Proceedings of the 51st International Academic Conference, Vienna, Austria, 17 September 2019; pp. 110–122. [CrossRef]
2. Han, X.; Li, Y. Can investor sentiment be a momentum time-series predictor? evidence from China. *J. Empir. Financ.* **2017**, *42*, 212–239. [CrossRef]
3. Wang, B.; Huang, H.; Wang, X. A novel text mining approach to Financial Time Series forecasting. *Neurocomputing* **2012**, *83*, 136–145. [CrossRef]
4. Tanantong, T.; Yongwattana, P. A convolutional neural network framework for classifying inappropriate online video contents. *Int. J. Artif. Intell.* **2023**, *12*, 124–136. [CrossRef]
5. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
6. Kumar, D.; Sarangi, P.K.; Verma, R. A systematic review of stock market prediction using machine learning and statistical techniques. *Mater. Today Proc.* **2022**, *49*, 3187–3191. [CrossRef]
7. Zhao, K.; Zhang, J.; Liu, Q. Dual-hybrid modeling for option pricing of CSI 300ETF. *Information* **2022**, *13*, 36. [CrossRef]
8. Atsalakis, G.S.; Valavanis, K.P. Surveying stock market forecasting techniques—part II: Soft computing methods. *Expert Syst. Appl.* **2009**, *36*, 5932–5941. [CrossRef]
9. Vargas, M.R.; dos Anjos, C.E.; Bichara, G.L.; Evsukoff, A.G. Deep Leaming for stock market prediction using technical indicators and financial news articles. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8. [CrossRef]
10. Khan, W.; Ghazanfar, M.A.; Azam, M.A.; Karami, A.; Alyoubi, K.H.; Alfakeeh, A.S. Stock market prediction using machine learning classifiers and social media, news. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *13*, 3433–3456. [CrossRef]
11. Chen, G.; Liu, S.; Jiang, F. Daily Weather Forecasting Based on Deep Learning Model: A Case Study of Shenzhen City, China. *Atmosphere* **2022**, *13*, 1208. [CrossRef]
12. Wu, C.; Huang, L.; Wang, W. De-noising Method of Joint Empirical Mode Decomposition and Principal Component Analysis. In Proceedings of the IEEE International Conference on Power, Intelligent Computing and Systems, Virtual Conference, 28–30 July 2020; pp. 193–195. [CrossRef]
13. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.-C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [CrossRef]
14. Nava, N.; Matteo, T.; Aste, T. Financial time series forecasting using empirical mode decomposition and support vector regression. *Risks* **2018**, *6*, 7. [CrossRef]
15. Teng, M.; Li, S.; Xing, J.; Song, G.; Yang, J.; Dong, J.; Zeng, X.; Qin, Y. 24-hour prediction of PM2.5 concentrations by combining empirical mode decomposition and bidirectional long short-term memory neural network. *Sci. Total Environ.* **2022**, *821*, 153276. [CrossRef] [PubMed]
16. Stallone, A.; Cicone, A.; Materassi, M. New insights and best practices for the successful use of empirical mode decomposition, iterative filtering and derived algorithms. *Sci. Rep.* **2020**, *10*, 15161. [CrossRef] [PubMed]
17. Wu, Z.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41. [CrossRef]
18. Lei, Y.; Liu, Z.; Ouazri, J.; Lin, J. A fault diagnosis method of rolling element bearings based on CEEMDAN. *J. Mech. Eng. Sci.* **2017**, *231*, 1804–1815. [CrossRef]

19. Torres, M.E.; Colominas, M.A.; Schlotthauer, G.; Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 4144–4147. [CrossRef]
20. Liu, T.; Luo, Z.; Huang, J.; Yan, S. A Comparative Study of Four Kinds of Adaptive Decomposition Algorithms and Their Applications. *Sensors* **2018**, *18*, 2120. [CrossRef]
21. Yan, Y.; Wang, X.; Ren, F.; Shao, Z.; Tian, C. Wind speed prediction using a hybrid model of EEMD and LSTM considering seasonal features. *Energy Rep.* **2022**, *8*, 8965–8980. [CrossRef]
22. Hu, Z. Crude oil price prediction using CEEMDAN and LSTM-attention with news sentiment index. *Oil Gas Sci. Technol.—Rev. D'ifp Energ. Nouv.* **2021**, *76*, 28. [CrossRef]
23. Curse of Dimensionality-A "Curse" to Machine Learning. Available online: https://towardsdatascience.com/curse-of-dimensionality-a-curse-to-machine-learning-c122ee33bfeb (accessed on 1 May 2022).
24. Smallman, L.; Artemiou, A. A literature review of (sparse) exponential family PCA. *J. Stat. Theory Pract.* **2022**, *16*, 14. [CrossRef]
25. 7.1: Eigenvalues and Eigenvectors of a Matrix. Mathematics LibreTexts. Available online: https://math.libretexts.org/Bookshelves/Linear_Algebra/A_First_Course_in_Linear_Algebra_(Kuttler)/07%3A_Spectral_Theory/7.01%3A_Eigenvalues_and_Eigenvectors_of_a_Matrix (accessed on 1 May 2022).
26. Joshi, C.; Panda, S. PCA-LSTM: Deep Learning Approach for the Indian Large-Caps. In Proceedings of the 7th International conference for Convergence in Technology, Pune, India, 7–9 April 2022; pp. 1–6. [CrossRef]
27. Wang, L.; Wang, S.; Yuan, Z.; Peng, L. Analyzing potential tourist behavior using PCA and modified affinity propagation clustering based on baidu index: Taking beijing city as an example. *Data Sci. Manag.* **2021**, *2*, 12–19. [CrossRef]
28. Xu, L.; Wang, J.; Ou, Y.; Fu, Y.; Bian, X. A novel decision-making system for selecting offshore wind turbines with PCA and D numbers. *Energy* **2022**, *258*, 124818. [CrossRef]
29. Zhong, X.; Enke, D. Forecasting daily stock market return using dimensionality reduction. *Expert Syst. Appl.* **2017**, *67*, 126–139. [CrossRef]
30. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics* **2020**, *9*, 483. [CrossRef]
31. Bhavitha, B.K.; Rodrigues, A.P.; Chiplunkar, N.N. Comparative Study of Machine Learning Techniques in Sentimental Analysis. In Proceedings of the International Conference on Inventive Communication and Computational Technologies, Tamilnadu, India, 10–11 March 2017; pp. 216–221. [CrossRef]
32. Bartov, E.; Faurel, L.; Mohanram, P.S. Can twitter help predict firm-level earnings and stock returns? *SSRN Electron. J.* **2016**, *2631421*, 1–66. [CrossRef]
33. Liu, Z.; Huang, D.; Huang, K.; Li, Z.; Zhao, J. Finbert: A pre-trained financial language representation model for financial text mining. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 7 – 15 January 2021; pp. 4513–4519. [CrossRef]
34. Deep Learning Architectures. Available online: https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures (accessed on 5 May 2022).
35. Benuwa, B.B.; Zhan, Y.Z.; Ghansah, B.; Wornyo, D.K.; Banaseka Kataka, F. A review of Deep Machine Learning. *Int. J. Eng. Res. Afr.* **2016**, *24*, 124–136. [CrossRef]
36. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural. Comput.* **1997**, *9*, 1735–1780. [CrossRef]
37. Colah's Blog, Understanding LSTM Networks. Available online: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 15 October 2022).
38. Budiharto, W. Data Science Approach to stock prices forecasting in Indonesia during COVID-19 using long short-term memory (LSTM). *J. Big Data* **2021**, *8*, 47. [CrossRef]
39. Chou, S.-Y.; Dewabharata, A.; Zulvia, F.E.; Fadil, M. Forecasting Building Energy Consumption Using Ensemble Empirical Mode Decomposition, Wavelet Transformation, and Long Short-Term Memory Algorithms. *Energies* **2022**, *15*, 1035. [CrossRef]
40. Zha, W.; Liu, Y.; Wan, Y.; Luo, R.; Li, D.; Yang, S.; Xu, Y. Forecasting monthly gas field production based on the CNN-LSTM model. *Energy* **2022**, *260*, 124889. [CrossRef]
41. Lhasiw, N.; Sanglerdsinlapachai, N.; Tanantong, T. A Bidirectional LSTM Model for Classifying Chatbot Messages. In Proceedings of the 16th International Joint Symposium on Artificial Intelligence and Natural Language Processing, Virtual Conference, 21–23 December 2021; pp. 1–6. [CrossRef]
42. Mahawan, A.; Jaiteang, S.; Srijiranon, K.; Eiamkanitchat, N. Hybrid ARIMAX and LSTM Model to Predict Rice Export Price in Thailand. In Proceedings of the International Conference on Cybernetics and Innovations, Ratchaburi, Thailand, 28 February–2 March 2022; pp. 1–6. [CrossRef]
43. Berrar, D. Cross-validation. In *Encyclopedia of Bioinformatics and Computational Biology*; Ranganathan, S., Gribskov, M., Nakai, K., Schönbach, C., Eds.; Academic Press: Cambridge, MA, USA, 2019; Volume 1, pp. 542–545. [CrossRef]
44. Cerqueira, V.; Torgo, L.; Mozetič, I. Evaluating time series forecasting models: An empirical study on performance estimation methods. *Mach. Learn.* **2020**, *109*, 1997–2028. [CrossRef]
45. Syed, A.R. A Review of Cross Validation and Adaptive Model Selection. Master's Thesis, Georgia State University, Atlanta, GA, USA, 2011.

46. Unit Root Testing. Available online: http://www.eviews.com/help/helpintro.html#page/content/advtimeser-Unit_Root_Testing.html (accessed on 5 May 2022).
47. Augmented Dickey-Fuller (ADF) Test—Must Read Guide. Available online: https://www.machinelearningplus.com/timeseries/augmented-dickey-fuller-test (accessed on 5 May 2022).
48. Statsmodels, Statistical Models, Hypothesis Tests, and Data Exploration. Available online: https://www.statsmodels.org/stable/index.html (accessed on 5 May 2022).
49. GitHub Repository, Technical Analysis Library in Python. Available online: https://github.com/bukosabino/ta (accessed on 5 May 2022).
50. GitHub Repository, VADER-Sentiment-Analysis. Available online: https://github.com/cjhutto/vaderSentiment (accessed on 5 May 2022).
51. GitHub Repository, TextBlob: Simplified Text Processing. Available online: https://github.com/sloria/textblob (accessed on 5 May 2022).
52. Lv, S.-H.; Peng, L.; Hu, H.; Wang, L. Effective machine learning model combination based on selective ensemble strategy for time series forecasting. *Inf. Sci.* **2022**, *612*, 994–1023. [CrossRef]
53. Lv, S.-H.; Wang, L. Deep learning combined wind speed forecasting with hybrid time series decomposition and multi-objective parameter optimization. *Appl. Energy* **2022**, *311*, 118674. [CrossRef]