Stock Price Prediction Using XCEEMDAN-Bidirectional LSTM-Spline

[1]Kelvin, [2]Ronsen Purba, [3]Arwin Halim
[1,2]Magister Information Technology, Mikroskil University
[3]Informatics Engineering, Mikroskil University
Email: [1]kelvin.chen@mikroskil.ac.id, [2]ronsen@mikroskil.ac.id, [3]arwin@mikroskil.ac.id

Article Info

Article history:
Received Feb 16th, 2022
Revised Mar 8th, 2022
Accepted Apr 8th, 2022

Keyword:
Bidirectional LSTM
Exogenous Features
Spline
Stock Price Prediction
XCEEMDAN

ABSTRACT

Bidirectional Long Short Term Memory (Bidirectional LSTM) is a machine learning technique with the ability to capture data context by traversing backward data to forward data and vice versa. However, the characteristics of stock data with large fluctuations, high dimensions and non-linearity become a challenge in obtaining high stock price prediction accuracy values. The purpose of this study is to provide a solution to the problem of stock data characteristics with large fluctuations, high dimensions and non-linearity by combining the Complete Ensemble Empirical Mode Decomposition With Adaptive Noise method for exogenous features (XCEEMDAN), Bidirectional Long Short Term Memory (LSTM), and Splines. The predicted data will go through normalization and preprocessing using XCEEMDAN then the XCEEMDAN decomposition results are divided into high and low frequency signals. The bidirectional LSTM handles high frequency signals and the Spline model handles low frequency signals. The test is carried out by comparing the proposed XCEEMDAN-Bidirectional LSTM-Spline model with the XCEEMDAN-LSTM-Spline model using the same parameters and changing the noise seed randomly 50 times. The test results show that the proposed model has the smallest RMSE average value of 0.787213833 while model which is compared only has the smallest RMSE average value of 0.807393567.

Corresponding Author:
Kelvin,
Magister Information Technology,
Mikroskil University,
Email: kelvin.chen996@gmail.com

DOI: http://dx.doi.org/10.24014/ijaidm.v5i1.14424

1. INTRODUCTION

Stock price prediction is an act of trying to determine the future value of a company's shares or other financial instruments traded on the stock market. The success of predicting stock prices can maximize investor profits [1]. The stock market has data characteristics with large fluctuations, high dimensions, and non-linearity, making it difficult to predict [2]. Changes in stock prices can be influenced by various factors such as politics, economy, markets, technology, and investor behavior [3]. The latest stock price prediction model with better accuracy is a combination of three elements, namely signal processing techniques, machine learning models and traditional stochastic models. Signal processing techniques serve to control irregularity (noise) data and machine learning models for studying time series patterns with high complexity, while traditional stochastic models for studying time series patterns with low complexity. The combination of these three elements is able to build a prospective model with respect to backward and data data forward [4].

P. Flandrin, E. Torres, and MA Colominas [5] introduces a signal processing technique using Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN). CEEMDAN is able to analyze and process non-linear and non-stationary signals with a faster processing time than the Ensemble Empirical Mode Decomposition (EEMD) method. M. Roondiwala, H. Patel, and S. Varma [6] predict stock prices using the Long Short Term Memory (LSTM) model. However, the LSTM model has the limitation of using only the dependence on forward data and does not efficiently consider the dependence on backward data to obtain useful information [7]. Cao et al. [8] proposed the CEEMDAN-LSTM model which is able to produce stock price predictions with a low error rate compared to the LSTM, SVM, CEEMDAN-SVM, CEEMDAN-MLP, and EMD-LSTM models.

Y. Xuan, Y. Yu, and K. Wu [9] proposed the Empirical Mode Decomposition - Long Short Term Memory - Cubic Spline Interpolation (EMD-LSTM-CSI) model which has faster processing time and strong stability in predicting stock prices. EMD is used to parse the stock price series to be stable and extract local feature information from the original series. The LSTM model serves as a multi-variable input and a CSI model to avoid overfitting problems. The EMD-LSTM-CSI model provides better predictive values than the Support Vector Machine (SVM), LSTM, Attention based LSTM (LSTM-ATTE), EMD-LSTM, and EMD-LSTM-ATTE models. Teja, Tiwari, and Mohanty (2020) compared the EMD, EEMD and CEEMDAN signal processing methods with the result that the CEEMDAN method obtained higher statistical values than other models.

R. De Luca Avila and G. De Bona [4] proposed the XCEEMDAN-LSTM-Spline model which is able to increase the accuracy of stock price predictions from the CEEMDAN-LSTM model. However, researchR. De Luca Avila and G. De Bona still have not tried to see the limitations of the LSTM model as in the study of Cui et al [7]. To resolve the issue, Siami-Namini et al. [10] and Jia et al. [2] proposed a Bidirectional LSTM model that can provide better predictions than the ARIMA and LSTM models. Bidirectional LSTM can capture the data context by traversing the input data twice, namely from data backward to data forward and vice versa. The results showed that the use of an additional layer that traverses the input data on the Bidirectional LSTM helps in increasing accuracy because it can capture additional information in the form of information related to backward data to forward data and vice versa.

From the problems and literature review, a model using XCEEMDAN-Bidirectional LSTM-Spline is proposed. Prediction data will go through normalization and XCEEMDAN preprocessing to overcome chaotic data characteristics by controlling noise. After that, an additional layer of Bidirectional LSTM is applied to capture additional information as well as a Spline model to avoid overfitting problems that support stock price prediction results.

2. RESEARCH METHOD

2.1. Stock Price Prediction

The stock market is a place where shares can be transferred, traded and circulated [3]. The stock market has data characteristics with large fluctuations, high dimensions, and non-linearity, making it difficult to predict [2]. Stock price fluctuations will be determined by the forces of supply and demand. If the number of offers is greater than the number of requests, in general the stock price will fall. Conversely, if the number of requests is greater than the number of offers, the stock price tends to rise [11].

Stock price prediction is an act of trying to determine the value of future shares of a company or other financial instrument traded on the stock market. Stock price predictions have been in focus for years because they can yield significant profits[1]. The technique of predicting stock prices by analyzing trends over several years can be proven to determine stock market movements that are dynamic, difficult to predict, and non-linear. Traditionally, there have been two main approaches to predicting an organization's stock price[12]:

1. Technical analysis uses historical stock prices such as close, open, traded volume, adjacent close, and more to predict future stock prices.
2. Qualitative analysis based on external factors, such as company profile, market situation, political and economic factors, textual information in the form of new financial articles, social media or blogs by economic analysts.
3. Machine learning techniques that have special capabilities for stock market analysis. This technique is useful for very large and non-linear data sizes. Machine learning techniques can identify hidden patterns and complex relationships in large data sets and have been shown to increase efficiency by 60-86 percent compared to previous approaches.

Stock price analysis and prediction has several approaches over time. The traditional model follows a stochastic probabilistic approach, while the newer model is based on machine learning methods. There are various stochastic probabilistic approach models such as the Autoregressive (AR), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA) and Cubic Spline Interpolation (CSI) models. focuses on studying the behavior of a series from data without explicit assumptions such as linearity or stationarity. This model is a combination of three elements, namely signal processing techniques, machine learning models and traditional stochastic models. The signal processing technique is responsible for parsing or filtering the time series prior to model fitting. One of the signal processing techniques is CEEMDAN which is able to analyze and process non-linear and non-stationary signals. Traditional machine learning and stochastic models are responsible as models for predicting stock prices[4]. The machine learning model that will be used in this research is Bidirectional LSTM while the traditional stochastic model that will be used is CSI.

2.2. XCEEMDAN

Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) is a signal decomposition method based on Fourier transform for adaptively processing nonlinear and non-stationary signals [8]. The beginning NE Huang, Z. Shen, and SR Long [13] introduced Empirical Mode Decomposition (EMD) for analyzing nonlinear data. EMD can decompose nonlinear and nonstationary data into multiple signals Intrinsic Mode Functions (IMF) and 1 residual signal. Signal Chaotic nonlinear data is a source of uncertainty and irregularity in the observed data. Irregularity (noise) in the measurement of data arises from the process that is side by side with the data being studied or due to inaccuracies in the process of measuring the data.[14].The main stages of EMD, namely:

Step 1: Determine all maximum values of each original data $x(t), t = 1,2,3, \ldots, T$;

Step 2: Connects the minimum and maximum values locally to generate the undercover $x_{low}(t)$ and top cover $x_{up}(t)$

Step 3: Perform local average calculations with the formula:

$$m(t) = \frac{x_{up}(t)+x_{low}(t)}{2} \tag{1}$$

Step 4: Get the value of IMF 1 and residual with the formula:

$$IMF_1(t) = x(t) - m(t) \; and \; R_1(t) = m(t) \tag{2}$$

Step 5: for all parts, if more than two maximum values of are found $for\ i = 1,2,3, \ldots, n, R_i(t)$ then go back to step 2 and calculate $IMF_i + 1(t)\ dan\ R_i + 1(t)$.

EMD has the advantage of handling non stationary and nonlinear signals, but it still has the problem of “Mixing mode”. Mode mixing refers to the presence of very similar oscillations in different modes or very different amplitudes in a mode. Ensemble Empirical Mode Decomposition (EEMD) can largely eliminate mode mixing in the EMD algorithm [15]. By adding Gaussian white noise to the signal, with the formula:

$$x^i(t) = x(t) + w^i(t) \tag{3}$$

However, the EEMD algorithm cannot completely remove the Gaussian white noise after signal reconstruction, it causes reconstruction errors. To solve this problem, CEEMDAN is proposed as an upgraded version of EEMD[5]. CEEMDAN is able to eliminate mode mixing more effectively, reconstruction errors are almost zero, and the cost of complexity is low[8].

CEEMDAN will be applied to exogenous features (X). Exogenous features (X) are features that affect the price series to be predicted. Suppose the features in the data that will be used to train and test the model consist of stock prices open, high, low, close, and daily financial volume. The close price is a price series that will be predicted while other prices such as open, high, low, and volume are exogenous features (X). The application of CEEMDAN to exogenous features (X) is abbreviated to XCEEMDAN. Here are the steps of the XCEEMDAN algorithm:

Step 1: Decomposition of each xi(n) = using EMD to get the first IMF mode with calculations The formula used is as follows: $x(n) + \varepsilon_0 w^i(n)$

$$\widetilde{IMF_1} = (n) = \frac{1}{I} \sum_{i=1}^{I} IMF_1^i(n) = \overline{IMF_1}\,(n) \tag{4}$$

Step 2: Calculate the first residue with the following formula:

$$r_1(n) = x(n) - \widetilde{IMF_1}(n) \tag{5}$$

Step 3: Calculate the decomposition of the first residue where is the standard deviation of White Gaussian Noise. can be broken down as follows: $r_1(n) + \varepsilon_1 E_1\left(w^i(n)\right), dengan\ i = 1, 2, \dots, I\ \ \varepsilon_1\ \widetilde{IMF}_2(n)$

$$\widetilde{IMF}_2(n) = \frac{1}{I}\sum_{i=1}^{I} E_1\left(r_1(n) + \varepsilon_0 E_1\left(W^i(n)\right)\right) \tag{6}$$

Step 4: Calculate the residual for IMF 2 with a value of K showing the resulting total k through the following calculation:

$$r_k(n) = r_{k-1}(n) - \widetilde{IMF}_k(n), k = 2, 3, \dots, K \tag{7}$$

Step 5: Calculate the decomposition and then calculate the IMF can be calculated in the following way: $r_k(n) + \varepsilon_k E_k\left(w^i(n)\right)$, with $i = 1, 2, \dots, I(k+1)^{th}$

$$IMF_{k+1}(n) = \frac{1}{I}\sum_{i=1}^{I} E_1\ \left(r_k(n) + \varepsilon_0 E_k\left(W^i(n)\right)\right) \tag{8}$$

Step 6: Repeat steps 4 to 6 until the residue becomes a fixed function, so that the IMF cannot be extracted anymore. Assume k and represent the total number of values in the final mode and residual. The input x(n) can be expressed as follows: $r_k(n)$

$$x(n) = \sum_{k=1}^{k} \widetilde{IMF}_k\ (n) + r_k(n) \tag{9}$$

2.3. Bidirectitonal Long Short Term Memory (Bidirectional LSTM)

Bidirectional LSTM is a variant Recurrent Neural Network, which solves the long-term dependency of RNN and LSTM. Bidirectional LSTM combines LSTM in two different directions and extracts information data forward and backward at the same time making it easier to collect potentially unused data. Bidirectional LSTM using data pre-process as input, passes through the LSTM neural network layer forward layer and back layer, then go to full connection layer, and output the prediction result, as shown in Figure 1. In the forward layer, the calculation is carried out from the start time to the end time, and the output from the hidden layer of the forward layer is always obtained and stored. In the backward layer, the calculation is reversed over the end time to the start time to get and store the output of the hidden layer backward layer every time. In Figure 2, it can be seen that each of the six unique weights is used repeatedly each time. Put the hidden layer forward layer and backward layer (w1, w3), the hidden layer to the hidden layer itself (w2, w5), the hidden layer forward layer and the backward layer to the output layer (w4, w6). Output layer is the result of merging forward layer and backward layer [2].

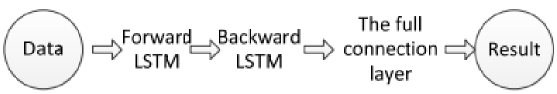


Figure 1. Structure Diagram Bidirectional LSTM [2]

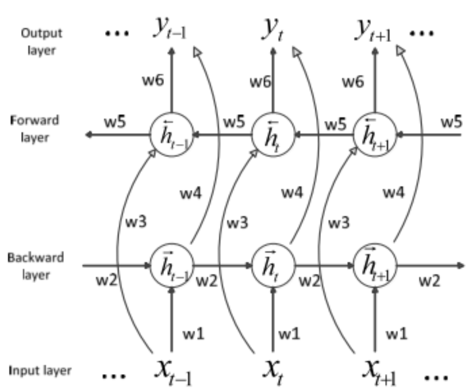


Figure 2. Bidirectional Neural Network LSTM [2]

Algorithm steps *Bidirectional* LSTM as follows [2]:

Step 1: Calculate input to layer hidden forward layer with the following formula:

$$\overrightarrow{h_t} = f(w_1 x_t + w_2 h_{t-1}) \tag{10}$$

Step 2: Count layers hidden backward layer with the following formula:

$$\overleftarrow{h_t} = f(w_3 x_t + w_5 h_{t+1}) \tag{11}$$

Step 3: Count layers output by combining the outputs of the forward and backward layers as in the following formula:

$$o_t = g(w_4 \overrightarrow{h_t} + w_6 \overleftarrow{h_t}) \tag{12}$$

where f is the activation function and g is the optimizer used.

2.4. Cubic Spline Interpolation

One of the interpolation techniques that can produce smooth merged curves is cubic spline interpolation. Each part of the point formed in the interpolation of the cubic spline is integrated with each other and is represented by a cubic curve (degree 3 polynomial). Cubic Spline Interpolation Algorithm steps as follows [16]:

Step 1: Determine the paired point set (x, y)

Step 2: Determine the coefficient a_i using Equation:

$$S_i(x_i) = y_i \text{ , } i = 1, .., n-1, \text{ where } a_i = y_i \tag{13}$$

Step 3: Calculate the lower diagonal element and the upper diagonal element of a tridiagonal matrix using the equation$l_i\, u_i$ $u_i = l_i = x_{i+1} - x_i$, where $u_0 = l_n = 0$

Step 4: Calculate the main diagonal elements (m_i) use Equality

$$m_i = 2(x_{i+1} - x_i + x_i - x_{i-1}) \text{ where } d_0 = d_n = 1 \tag{14}$$

Step 5: Calculate vector elements using Equation V

$$v_i = 3\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \text{ where } v_0 = v_n = 0 \tag{15}$$

Step 6: Arrange the elements into a tridiagonal matrix $l_i\, u_i$, m_i A

Step 7: Define and solve the system of linear equations in the equations so that a vector is obtained which is a set of coefficients $AC = V$

Step 8: Calculate coefficient using Equation:

$$b = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{x_{i+1} - x_i}{3}(2c_i - c_{i+1}) \tag{16}$$

Step 9: Calculate coefficient using Equation:

$$d_i = \frac{c_{i+1} - c_i}{3(x_{i+1} - x_i)} \tag{17}$$

Step 10: Form a series of polynomial equations using the calculated elements and coefficients $a, b, c\ d$

Step 11: To extrapolate with observation points outside the known point range, use the equation that is at the end closest to the x value for which you want to find the y value as shown in the **Figure 3**.

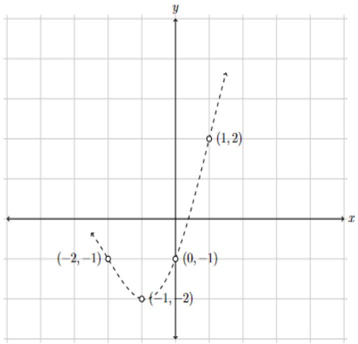


Figure 3. Four-point cubic spline interpolation [17]

In this research method there are 8 focus stages to be carried out. The first stage starts from reading the dataset, determining the target features, exogenous features, and the IMF threshold. The second stage is dataset normalization to convert the data into a range between 0 and 1. The third stage is preprocessing the data using XCEEMDAN to generate IMF signals. The fourth stage groups the IMF signal frequency based on the specified IMF threshold value. The fifth stage predicts low-level IMF signals with high frequencies using the Bidirectional LSTM model. High level IMF signal with low frequency will be predicted using spline model. The results of the IMF signal prediction are then combined to obtain the predicted value of the target feature. After that, testing is carried out to produce target feature predictions as shown in the flowchart of Figure 4.

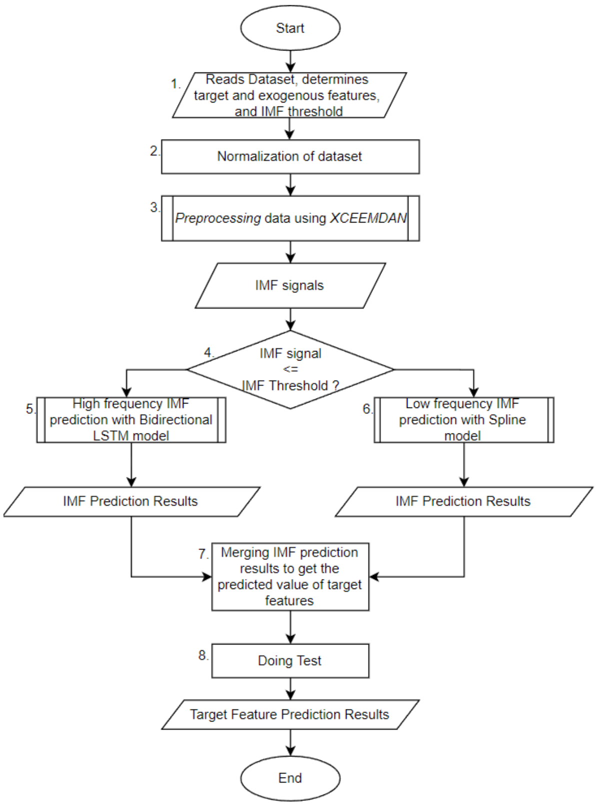


Figure 4. Research Method Flowchart

The stock price adj close prediction is obtained by combining the overall prediction results of the IMF (recomposing) the Bidirectional LSTM - Spline model. The result of adj close price prediction will be compared with the real price of adj close, namely comparing the value of train_predicted with train_real, validation_predicted with validation_real, and test_predicted with test_real. The graph of the comparison between the predicted value of adj close and the real value of PETR4 equity with the IMF 2 threshold value can be seen in Figure 5. The y-axis shows the predicted value of adj close while the x-axis shows the time period. The RMSE generated in the experiment can be seen in Table 1.

Figure 5. Comparison graph of the predicted value of adj close with the real value of PETR4 equity with the IMF threshold value of 2

Table 1. RMSE value on each equity

Ekuitas	RMSE
PETR4	0,678
VALE3	1,27
BOVA11	1,858
ITUB4	0,586
BBDC4	0,631
B3SA3	0,405
BBAS3	0,937
ABEV3	0,321
MGLU3	0,561
VVAR3	0,574
Average	0,7821

Problem solving framework can be shown in Figure 6. The initial conditions begin by applying preprocessing data using a signal processing technique, namely CEEMDAN to exogenous features to process input data (XCEEMDAN). After that, the results of the XCEEMDAN output are used as input for the LSTM-Spline model to predict stock prices. XCEEMDAN-LSTM-SPLINE was able to provide a performance increase of 65.233% for the MSE average and 27.234% for the MAPE average against the CEEMDAN-LSTM model [4]. The research regarding the Bidirectional LSTM model used to predict stock prices with an average accuracy increase of 37.78% against the LSTM model [10]. Based on these initial conditions, the research to be carried out is to use XCEEMDAN for data preprocessing. The results of the XCEEMDAN output are used as input to the Bidirectional LSTM-Spline to predict stock prices. The combination of these three methods has not been tested in predicting stock prices with the ability to remember backward and forward data and otherwise to capture additional information that supports prediction results. The data used in this study is secondary data (dataset) taken from Yahoo Finance, namely historical data on Brazilian equity PETR4, VALE3, BOVA11, ITUB4, BBDC4, B3SA3, BBAS3, ABEV3, MGLU3, VVAR3 from 01 January 2010 to 31 December 2020, the Dataset is publicly available and can be accessed at the link https://finance.yahoo.com/ by removing the day gap caused by weekends and holidays. The attributes that will be used in this study are the same as R. De Luca Avila and G. De Bona [4] namely open, high, low, adj close, and volume because these 5 attributes are more correlated with the output results adj close price.

In this research, we solve the following (figure 6) problems:

1. Comparison between the XCEEMDAN-LSTM, XCEEMDAN-Bidirectional LSTM, XCEEMDAN-LSTM-Spline models with XCEEMDAN-Bidirectional LSTM-Spline (the proposed model) at the IMF 2 and IMF 3 threshold values. We prove that the performance of the proposed model has better than other models compared.
2. We ensure that the proposed model outperforms the compared model by carrying out further steps, namely changing the value of the noise seed parameter 50 times randomly with each noise seed stage being carried out 3 times for a total of 150 trials.

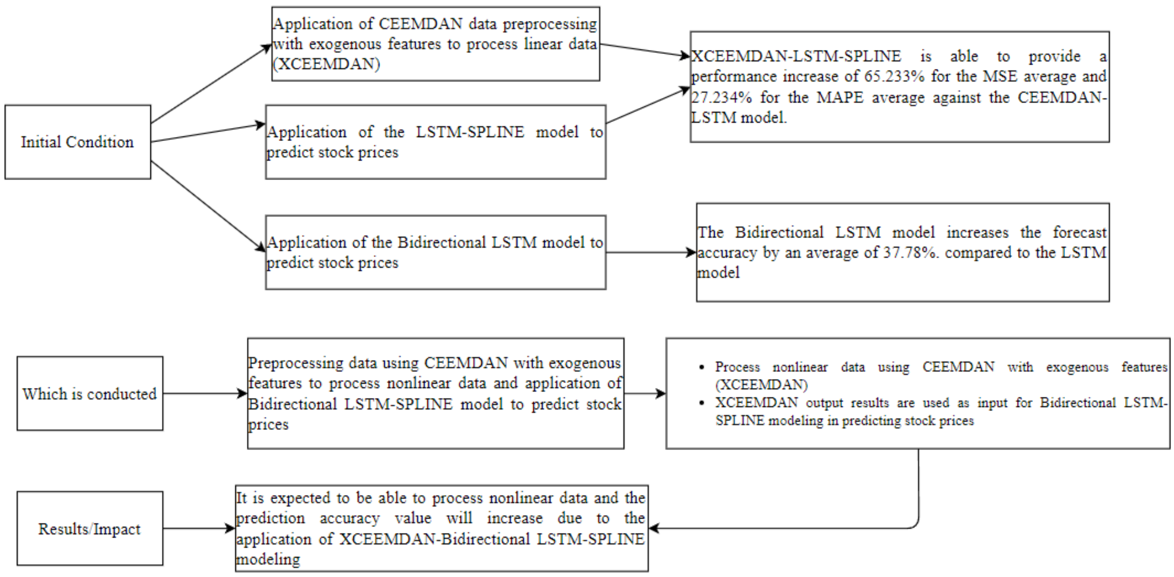


Figure 6. Problem Solving Framework

3. RESULTS AND ANALYSIS

3.1. Performance comparison of XCEEMDAN-Bidirectional LSTM- Spline model with other models

This test was conducted to compare which model has the lowest RMSE value in the PETR4, VALE3, BOVA11, ITUB4, BBDC4, B3SA3, BBAS3, ABEV3, MGLU3, VVAR3 datasets with the same parameters. The parameters used in testing this model can be seen in Table 2 and the results of the comparison of the performance of the tested models can be seen in Table 3. The XCEEMDAN-Bidirectional LSTM-Spline model has a better performance at the IMF 2 threshold with an RMSE value of 0.7929055 compared to the XCEEMDAN-LSTM-Spline model which has an RMSE value of 0.8082717. The difference in RMSE values in the XCEEMDAN-Bidirectional LSTM-Spline and XCEEMDAN-LSTM-Spline models is still not significant enough to produce a low RMSE value. However, the RMSE value of the XCEEMDAN-Bidirectional LSTM-Spline model is still low when compared to the XCEEMDAN-LSTM, XCEEMDAN-Bidirectional LSTM, and XCEEMDAN-LSTM-Spline models. To ensure that the XCEEMDAN-Bidirectional LSTM-Spline model outperforms the tested model, it is not coincidental, then the next test is carried out by changing the noise seed value randomly.

Table 2. Parameters used in model testing

Model	XCEEMDAN-LSTM	XCEEMDAN-Bi LSTM	Threshold at IMF 2		Threshold at IMF 3	
			XCEEMDAN-LSTM-Spline	XCEEMDAN-Bi LSTM-Spline	XCEEMDAN-LSTM-Spline	XCEEMDAN-Bi LSTM-Spline
Desc. Parameter	De Luca Avila and De Bona (2020)	Compared model	De Luca Avila and De Bona (2020)	Proposed Model	De Luca Avila and De Bona (2020)	Proposed Model
Neuron	128,64	128,64	128,64	128,64	128,64	128,64
Dense	16,4,1	16,4,1	16,4,1	16,4,1	16,4,1	16,4,1
Start Date	1/1/ 2010	1/1/ 2010	1/1/ 2010	1/1/ 2010	1/1/ 2010	1/1/ 2010
End Date	31/12/ 2020	31/12/ 2020	31/12/ 2020	31/12/ 2020	31/12/ 2020	31/12/ 2020
Noise scale	0.15	0.15	0.15	0.15	0.15	0.15
Noisee seed	12345	12345	12345	12345	12345	12345

Table 3. The results of the comparison of the performance of the tested model

Model	RMSE					
	XCEEMDAN-LSTM	XCEEMDAN-Bi LSTM	Threshold at IMF 2		Threshold at IMF 3	
			XCEEMDAN-LSTM-Spline	XCEEMDAN-Bi LSTM-Spline	XCEEMDAN-LSTM-Spline	XCEEMDAN-Bi LSTM-Spline
Desc. Parameter	De Luca Avila and De Bona (2020)	Compared model	De Luca Avila and De Bona (2020)	Proposed Model	De Luca Avila and De Bona (2020)	Proposed Model
PETR4	2.523431	2.804388	0.701492	0.655873	0.69452	0.811092

VALE3	9.990018	10.083954	1.251943	1.216536	1.22031	1.250733
BOVA11	11.132216	5.728592	1.975055	1.966339	1.8244	1.790849
ITUB4	1.085577	1.173333	0.639474	0.621454	0.689057	0.693081
BBDC4	1.414488	0.979345	0.645617	0.646884	0.601636	0.662083
B3SA3	0.969505	1.063933	0.412075	0.408277	0.496553	0.456976
BBAS3	1.273263	1.746552	0.974368	0.934958	0.998028	0.973233
ABEV3	0.819126	1.395172	0.334776	0.321272	0.370545	0.365649
MGLU3	2.509032	3.143169	0.560603	0.55937	0.697827	0.69751
VVAR3	1.077924	0.954684	0.587314	0.598092	0.628417	0.665221
AVERAGE	3.279458	2.9073122	0.8082717	**0.7929055**	0.8221293	0.8366427

3.2. Random noise seed changes

Changes in the noise seed value were carried out randomly as much as 50 times with 3 trials each to see the effect of the performance of the XCEEMDAN-Bidirectional LSTM-Spline model on the XCEEMDAN-LSTM-Spline model at the IMF 2 threshold and ensure that the XCEEMDAN-Bidirectional LSTM-Spline model is not coincidentally produces the lowest RMSE value. A recap of the RMSE results of random noise seed changes can be seen in Table 4. The detailed results regarding the RMSE of random noise seed changes can be seen in Appendix and the graph can be seen in Figure 7. The resulting RMSE is in the range of values from 0.787213833 to 0.820195467. The best noise seed value was obtained at the 42 nd random, with a noise seed value of 3877415934 which resulted in the lowest RMSE of 0.787213833 for the XCEEMDAN-Bidirectional LSTM-Spline model at the IMF 2 threshold. This proves that changes in the noise seed can also affect the price prediction results. The stock and additional layers of the XCEEMDAN-Bidirectional LSTM-Spline model proved to outperform the XCEEMDAN-LSTM-Spline model not by chance.

Table 4. Recap of RMSE result of noise seed change

No	Noise Seed	RMSE							
		Experiment 1		Experiment 2		Experiment 3		AVERAGE	
		Threshold IMF 2		Threshold IMF 2		Threshold IMF 2		Threshold IMF 2	
		XCEEMDAN - LSTM-Spline	XCEEMDAN -Bi LSTM-Spline	XCEEMDAN -LSTM-Spline	XCEEMDAN -Bi LSTM-Spline	XCEEMDAN -LSTM-Spline	XCEEMDAN -Bi LSTM-Spline	XCEEMDAN -LSTM-Spline	XCEEMDAN -Bi LSTM-Spline
1	12345	0.8083	0.7929	0.8069	0.7974	0.8037	0.7897	0.8063	0.7933
2	648	0.8138	0.8135	0.8138	0.8064	0.8184	0.8068	0.8153	0.8089
3	8754	0.8116	0.8083	0.8098	0.8018	0.8111	0.8183	0.8108	0.8095
4	10310	0.8181	0.7978	0.8121	0.8128	0.8084	0.7950	0.8128	0.8019
5	476114	0.8113	0.8095	0.8166	0.8106	0.8116	0.8011	0.8132	0.8071
6	847459393	0.8079	0.7972	0.8040	0.7994	0.8037	0.8034	0.8052	0.8000
7	3580345544	0.8134	0.8002	0.8097	0.7930	0.8092	0.8004	0.8107	0.7979
8	4022411438	0.8109	0.8066	0.8103	0.8018	0.8127	0.8172	0.8113	0.8085
9	3102976445	0.8132	0.7909	0.8057	0.8005	0.8111	0.8182	0.8100	0.8032
10	2397416255	0.8107	0.7919	0.8255	0.8068	0.8080	0.7962	0.8147	0.7983
11	877884397	0.8110	0.8042	0.8071	0.8296	0.8118	0.8002	0.8100	0.8114
12	502275570	0.8038	0.8070	0.8168	0.8022	0.8144	0.8141	0.8116	0.8078
13	2487001655	0.8077	0.8035	0.8112	0.7974	0.8057	0.8008	0.8082	0.8006
14	3575834324	0.8135	0.7972	0.8138	0.8180	0.8165	0.8035	0.8146	0.8062
15	398048047	0.8164	0.8024	0.8143	0.7979	0.8012	0.8114	0.8106	0.8039
16	2572482085	0.8205	0.8001	0.8123	0.8033	0.8167	0.7988	0.8165	0.8007
17	3698665810	0.8154	0.8063	0.8121	0.8099	0.8260	0.7934	0.8178	0.8032
18	759774810	0.8063	0.8029	0.8090	0.7977	0.8098	0.8016	0.8084	0.8008
19	7781442	0.8204	0.8071	0.8252	0.7962	0.8149	0.8046	0.8202	0.8026
20	998909803	0.8079	0.7997	0.8040	0.7946	0.8118	0.8056	0.8079	0.7999
21	2214797360	0.8080	0.8030	0.8128	0.7961	0.7993	0.8106	0.8067	0.8032
22	869201902	0.8097	0.7943	0.8118	0.8025	0.8080	0.8091	0.8098	0.8020
23	847055484	0.8125	0.7945	0.8123	0.8046	0.8078	0.8044	0.8109	0.8012
24	312333679	0.8069	0.8062	0.8156	0.8112	0.8099	0.8071	0.8108	0.8082
25	20301983	0.8038	0.8065	0.8013	0.8141	0.8122	0.8092	0.8058	0.8099
26	2049262079	0.8130	0.8045	0.8141	0.8153	0.8171	0.8219	0.8147	0.8139
27	1835658684	0.8023	0.8006	0.8063	0.7996	0.8058	0.8058	0.8048	0.8020
28	2214797360	0.8062	0.7973	0.8094	0.7905	0.8151	0.8024	0.8102	0.7967
29	3421847641	0.8153	0.7987	0.8125	0.8077	0.8167	0.8171	0.8149	0.8078
30	2602217941	0.8088	0.7985	0.8163	0.8115	0.8098	0.8061	0.8116	0.8054
31	75739528	0.8028	0.7916	0.8080	0.7984	0.8027	0.8032	0.8045	0.7977
32	2388411218	0.8042	0.7837	0.8041	0.7944	0.8057	0.7950	0.8047	0.7911
33	2315666833	0.8066	0.7946	0.8087	0.8077	0.8082	0.7972	0.8079	0.7998
34	123270304	0.8107	0.7956	0.8040	0.7990	0.8060	0.7986	0.8069	0.7977
35	2826796014	0.8104	0.7987	0.8099	0.8006	0.8102	0.8065	0.8102	0.8019
36	317668668	0.8001	0.7995	0.8121	0.8038	0.8103	0.8028	0.8075	0.8021
37	466618176	0.8171	0.7988	0.8157	0.8086	0.8227	0.8017	0.8185	0.8030

No	Noise Seed	RMSE Experiment 1 Threshold IMF 2 XCEEMDAN - LSTM-Spline	Experiment 1 Threshold IMF 2 XCEEMDAN -Bi LSTM-Spline	Experiment 2 Threshold IMF 2 XCEEMDAN -LSTM-Spline	Experiment 2 Threshold IMF 2 XCEEMDAN -Bi LSTM-Spline	Experiment 3 Threshold IMF 2 XCEEMDAN -LSTM-Spline	Experiment 3 Threshold IMF 2 XCEEMDAN -Bi LSTM-Spline	AVERAGE Threshold IMF 2 XCEEMDAN -LSTM-Spline	AVERAGE Threshold IMF 2 XCEEMDAN -Bi LSTM-Spline
38	3652025911	0.8094	0.7929	0.8075	0.7929	0.8097	0.7981	0.8089	0.7946
39	3923414613	0.8009	0.8011	0.8091	0.8043	0.8077	0.7937	0.8059	0.7997
40	3343038556	0.8086	0.7976	0.8071	0.7935	0.8388	0.7988	0.8182	0.7966
41	1570202524	0.8213	0.8163	0.8184	0.8129	0.8155	0.7963	0.8184	0.8085
42	3877415934	0.8027	0.7826	0.8070	0.7864	0.8126	0.7926	0.8074	0.7872
43	1240864836	0.8145	0.8105	0.8119	0.8036	0.8105	0.7984	0.8123	0.8041
44	1831814659	0.8064	0.7973	0.8061	0.7968	0.8061	0.8161	0.8062	0.8034
45	51429610	0.8172	0.8085	0.8101	0.8079	0.8156	0.8020	0.8143	0.8061
46	2678083308	0.8088	0.8099	0.8062	0.7980	0.8084	0.8012	0.8078	0.8030
47	2243003283	0.8157	0.7978	0.8095	0.8054	0.8131	0.8134	0.8128	0.8055
48	2083061937	0.8157	0.7970	0.8155	0.8039	0.8078	0.8115	0.8130	0.8041
49	2638726079	0.8127	0.7999	0.8097	0.8059	0.8084	0.7989	0.8103	0.8016
50	528508131	0.8149	0.8093	0.8155	0.7999	0.8102	0.8271	0.8135	0.8121

4. CONCLUSION

Based on the results of the tests and discussions that have been carried out, it can be drawn conclusion:

1. The test results using the same parameters on the XCEEMDAN-LSTM-Spline model show that the proposed model, namely XCEEMDAN-Bidirectional LSTM-Spline has better performance at the IMF 2 threshold with an RMSE value.of 0.7929055 compared to the XCEEMDAN-LSTM-Spline model which only has an RMSE value of 0.8082717. The higher the threshold value does not guarantee better model performance.
2. Testing with random noise seed changes 50 times with 3 trials each turns out to have an effect on the RMSE to be lower than the previous test. The best noise seed value was obtained at the 42 nd random, amounting to 3877415934 which resulted in average The lowest RMSE is 0.787213833 for the XCEEMDAN-Bidirectional LSTM-Spline model at the IMF 2 threshold compared with the XCEEMDAN-LSTM-Spline model which produces an average RMSE of 0.807393567.

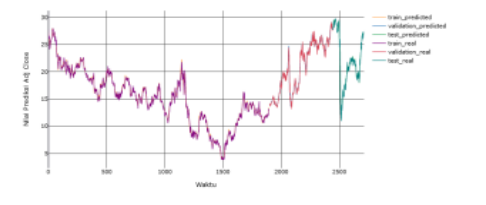
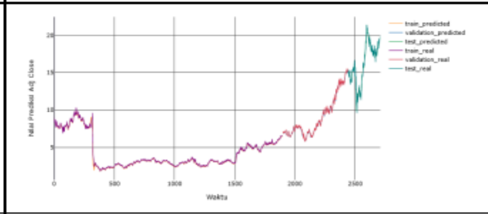
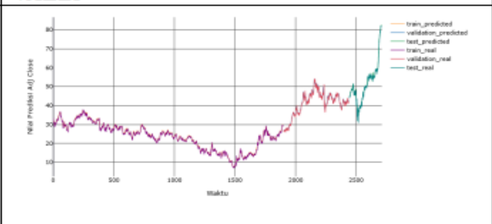
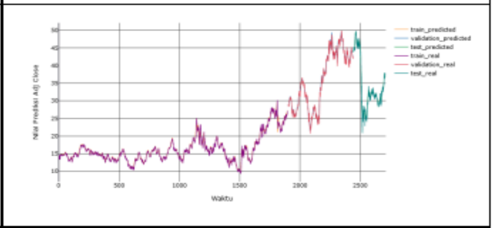
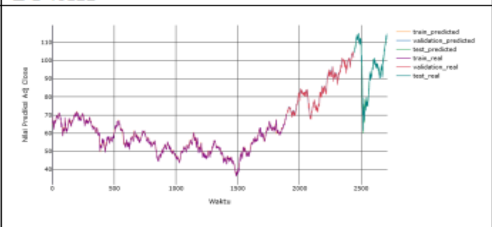
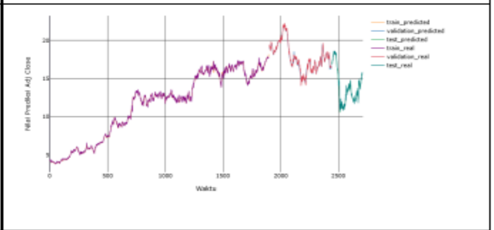
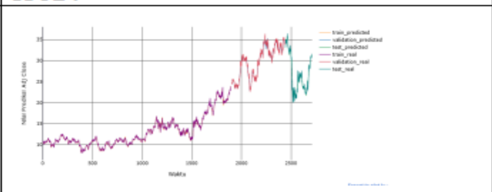
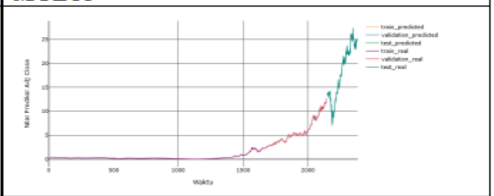
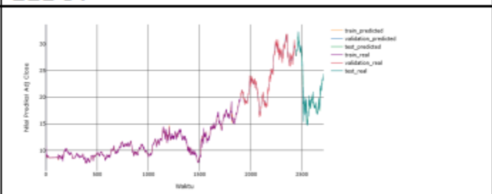
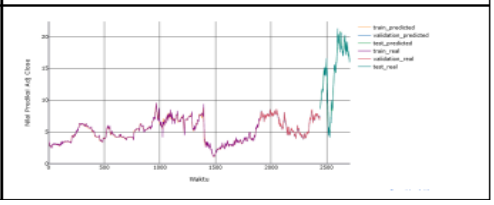
No	Equity	No	Equity
1	**PETR4**	6	**B3SA3**
			
2	**VALE3**	7	**BBAS3**
			
3	**BOVA11**	8	**ABEV3**
			
4	**ITUB4**	9	**MGLU3**
			
5	**BBDC4**	10	**VVAR3**
			

Figure 7. Test Result Graph

REFERENCES

[1] O. Hegazy, OS Soliman, and MA Salam, "A Machine Learning Model for Stock Market Prediction," Int. J. Comput. science. Telecommun., vol. 4, no. May 2014, pp. 17–23, 2014, [Online]. Available: http://arxiv.org/abs/1402.7351.

[2] M. Jia, J. Huang, L. Pang, and Q. Zhao, "Analysis and Research on Stock Price of LSTM and Bidirectional LSTM Neural Network," in International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019) Analysis , 2019, vol. 90, no. Iccia, pp. 467–473, doi:10.2991/iccia-19.2019.72.

[3] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," Neural Comput. Appl., vol. 32, no. 13, pp. 9713–9729, 2020, doi:10.1007/s00521-019-04504-2.

[4] R. De Luca Avila and G. De Bona, "Financial Time Series Forecasting via CEEMDAN-LSTM with Exogenous Features," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2020, vol. 12320 LNAI, pp. 558–572, doi:10.1007/978-3-030-61380-8_38.

[5] P. Flandrin, E. Torres, and MA Colominas, "A COMPLETE ENSEMBLE EMPIRICAL MODE DECOMPOSITION," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 4144–4147, doi:10.109/ICASSP.2011.5947265.

[6] M. Roondiwala, H. Patel, and S. Varma, "Predicting Stock Prices Using LSTM," Int. J. Sci. Res., vol. 6, no. 4, pp. 2319–7064, 2015, [Online]. Available: https://www.quandl.com/data/NSE.

[7] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for

network-wide traffic speed prediction," arXiv, pp. 1–11, 2018.
[8] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," Phys. A Stats. mech. its Appl., vol. 519, pp. 127–139, 2019, doi:10.1016/j.physa.2018.11.061.
[9] Y. Xuan, Y. Yu, and K. Wu, "Prediction of Short-term Stock Prices Based on EMD-LSTM-CSI Neural Network Method," IEEE Int. conf. Big Data Anal., pp. 135–139, 2020, doi:10.1109/ICBDA490404.2020.9101194.
[10] S. Siami-Namini, N. Tavakoli, and AS Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019, 2019, pp. 3285–3292, doi:10.109/BigData470990.2019.9005997.
[11] Zulfikar, Introduction to Capital Markets with a Statistical Approach. 2016.
[12] M. Vijh, D. Chandola, VA Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," Procedia Comput. Sci., vol. 167, no. 2019, pp. 599–606, 2020, doi:10.1016/j.procs.2020.03.326.
[13] NE Huang, Z. Shen, and SR Long, "A NEW VIEW OF NONLINEAR WATER WAVES : The Hilbert Spectrum 1," 1999.
[14] M. Camilleri, "Forecasting Using Non-Linear Techniques In Time Series Analysis: An Overview Of Techniques and Main Issues," Univ. Malta Comput. science. soo. res. Work., pp. 19–28, 2004.
[15] Z. Wu and NE Huang, "Ensemble Empirical Mode Decomposition : A Noise Assisted Data Analysis Method," no. August, 2005.
[16] M. Rosidi, Numerical Method Using R for Environmental Engineering. 2019.
[17] I. James P. Howard, Computational Methods for Numerical Analysis with R. Taylor & Francis Group, LLC, 2017.

BIBLIOGRAPHY OF AUTHORS

Kelvin, S.Kom., M.Kom.,
Graduate Informatics Engineering in Mikroskil University, Medan

Dr. Ronsen Purba, M.Sc.
Head of Magister Information Technology, Mikroskil University

Arwin Halim, S.Kom., M.Kom.
Head of Informatics Engineering, Mikroskil University