JOIV : Int. J. Inform. Visualization, 7(4) - December 2023 2270-2279

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model

Anggit Dwi Hartanto [a,*], Yanuar Nur Kholik [a], Yoga Pristyanto [a]

[a] *Department of System Information, Universitas Amikom Yogyakarta, Depok, Sleman, 55282, Indonesia*

Corresponding author: [*]anggit@amikom.ac.id

***Abstract*— In the world of stock investment, one of the things that commonly happens is stock price fluctuations or the ups and downs of stock prices. As a result of these fluctuations, many novice investors are afraid to play stocks. However, on the other hand, stocks are a type of investment that can be relied upon during disasters or economic turmoil, such as in 2019, namely the Covid-19 pandemic. For stock price fluctuations to be estimated by investors, it is necessary to carry out a forecasting activity. This study builds stock price forecasting using the Light Gradient Boosting Machine (LightGBM) algorithm, which has high accuracy and efficiency. To forecast stock price time series, the model used is the LightGBM ensemble. At the same time, they were optimizing the determination of hyperparameters using Grid Search Cross Validation (GSCV). This study will also compare the LGBM algorithm with other algorithms to see which model is optimal in forecasting price stock data. In this study, the test used the RMSE metric by comparing the original data (testing data) with the predicted results. The experimental results show that the LightGBM model can compete with and outperform boosting-based forecasting models like XGBoost, AdaBoost, and CatBoost. In comparing forecasting models, the same dataset is used so that the results are accurate, and the comparisons are equivalent. In future research, paying attention to the data during pre-processing is necessary because it has many outliers. In addition, it is necessary to include exogenous variables and external variables, which are determined to involve many parties.**

***Keywords*— Machine learning; prediction; forecasting; time series; LightGBM.**

Manuscript received 6 Apr. 2023; revised 12 Jun. 2023; accepted 26 Jul. 2023. Date of publication 31 Dec. 2023.

I. INTRODUCTION

At the end of 2019, the coronavirus disease 2019 (COVID-19) hit the world. At the beginning of the emergence of COVID-19, Indonesia experienced losses in various sectors. The Ministry of Finance of the Republic of Indonesia stated that COVID-19 caused Indonesia to experience a contraction in economic growth 2020 of -2.07 percent. Investment, according to KBBI, is the investment to gain profit. Investment is needed to face the future if unexpected events such as COVID-19 occur [1]–[3]. By investing, the community can minimize the problems caused, such as the COVID-19 pandemic. However, many people need help to start investing or make beginner mistakes when making decisions due to various psychological factors, namely overconfidence or vice versa [4]–[6].

One form of investment is stocks which have always been a hot topic of conversation both in the financial and technical fields. If we look at the last few decades, the development of computers has reached a stage where computers can forecast or forecast the future. Forecasting is a technique that can be used to predict future data trends based on existing and past data [7]–[10].

Stock predictions generally use data in the form of time series or time series data which is a collection of data arranged chronologically and measured over a certain period. Therefore, stock market predictions are critical when making decisions in the financial sector, making it a considerable challenge. The general and popular forecasting models used to make predictions on time series data include Autoregressive Integrated Moving Average (ARIMA) [11], [12], Seasonal Autoregressive Integrated Moving Average (SARIMA) [13], [14], Long Short Term Memory (LSTM) [15]–[17], and Gradient Boosting Decision Tree (GBDT) [18], [19], [20]. Forecasting techniques and models commonly used to predict data [21]–[25].

A study conducted by Ke et al. [25] developed the Light Gradient Boosting Machine (LightGBM) to overcome the shortcomings of GBDT when handling big data by speeding up the training process up to 20 times with nearly the same

accuracy. In the M5 Forecasting-Accuracy competition, LightGBM got first place and proved that LightGBM is superior to other techniques or models and can be used effectively to process many correlated and exogenous series and minimize the potential for forecasting errors [26]. In addition, one advantage of LightGBM is that LightGBM is compatible with several algorithms, such as Gradient Decision Trees (GDT), Gradient Boosting Decision Trees (GBDT), Gradient-based One-Side Sampling (GOSS), Dropouts meet Multiple Additive Regression Trees (DART), and Random Forests (RF). Another advantage of LightGBM is that it can perform sparse optimization, parallel training, multiple loss functions, regularization, bagging, and early stopping [25].

The difference between LightGBM and Extreme Gradient Boosting (XGBoost) lies in how it increases Gradient Boosting (GB) by using multiple cores from the Central Processing Unit (CPU) so that the learning process can be carried out in parallel, distributing calculations, cache optimization, and out-of-core processing. Whereas LightGBM has a leaf-wise growth structure, XGBoost has level-wise growth or levels [27], [28]. LightGBM was built using two novel techniques: GOSS and Exclusive Feature Bundling (EFB). GOSS is a sampling method for GDBT that can balance reducing data instances and maintaining accuracy in the decision tree that has been studied [29].

A similar study was also carried out by Pokhrel [30] by predicting dominant ocean waves using the Light Gradient Boosting Machine (LightGBM). The data used is The Coastal Data Information Program (CDIP) which has been filtered based on specific parameters to improve data quality. Features based on sea waves, such as wave height, period, kurtosis, and skewness, and then features related to the atmosphere, such as humidity, pressure, and temperature, are extracted from the data set. The model used is a decision tree-based model, Extra Trees (ET), which performs the bagging process, and LightGBM performs the booting process. Both processes use the ensemble method in making predictions or forecasting. Then the model is easy to use and has a high-efficiency level. The study's results by Pokhrel [30] showed that the two forecasting models proposed experienced a decrease in performance from zero to one day (one day range). However, the performance was relatively consistent in the 15-day and 30-day trials. The proposed forecasting model also outperforms comparison data originating from weather forecasting institutions or bodies concerned with marine research, such as the Fleet Numerical Meteorology and Oceanography Center (FNMOC), European Center for Medium-Range Weather Forecasts (ECMWF), National Centers for Environmental Prediction (NCEP) and others. So, the two proposed models show better performance when paired with data spanning 15 or 30 days compared to one day. Of the two proposed models, LightGBM has better results than ET and data from several marine research agencies or institutions.

Forecasting based on machine learning has also been widely used in economics to support decision-making processes. Examples are forecasting sales predictions, stock predictions, sentiment analysis, and others. Research by Husein and Harahap [31] reveals that forecasting product sales time series with the M5 Forecasting dataset. This study used the Cross Industry Standard Process for Data Mining (CRISP-DM) framework by comparing five different algorithms, namely Linear Regression (LR), Ridge Regression (RR), XGBoost Classifier, LightGBM, and LSTM, which Root Mean Squared Errors (RMSE) then evaluated. RMSE was chosen because it is more concerned with the most significant errors. The result is that LightGBM has the lowest error value compared to other alternative models.

Forecasting in the economic field is also carried out by Pokhrel [30], predicting stock returns from Nvidia. This study uses regression as the basis for forecasting calculations. Therefore, the data is sorted based on predetermined features. The same thing was done by Chlebus et al. [32] when preparing data before carrying out the training process. However, Pokhrel [30] adds several data sets as exogenous variables or variables that can affect fluctuations in data originating from outside. The data used is from competitor companies or business partners of Nvidia and public or investor sentiment towards Nvidia. In this study, the models whose performance was tested included ARIMA, ARIMAX, K-Nearest Neighbors (KNN), Support Vector Regression (SVR), XGBoost, LightGBM, and LSTM. The results show that SVR and LightGBM have nearly the same performance, but SVR has a better test score than other alternative forecasting methods based on stationary variables. In addition, machine learning forecasting models have better predictive performance than econometric models.

Similar research was also carried out by Li [33] applying LightGBM to predictions of monthly house rental prices. The result is that LightGBM has an RMSE value of 0.1429 and a goodness of fit (R2) value of 96.13 percent. In his research, Gan et al. [29] collected hourly water discharge data from the National Oceanic and Atmospheric Administration (NOAA) by searching data via the Internet. The research methodology conducted by Gan et al. [29] is the same as the studies above, using the Boosting model. The results show that the small learning rate parameters require iteration with high hyperparameter values to get the best RMSE; num_leaves and max_depth are selected by comparing each combination and displayed on a grid graph to calculate the NMSE. Evaluation is done by comparing LightGBM with physics-based models, such as non-stationary tidal harmonic analysis (NS_TIDE). The result is that LightGBM can outperform NS_TIDE on RMSE, MAE, non-dimensional skill score (SS), and correlation coefficient (CC) scores.

Most previous studies on forecasting used the Boosting model, but some of the Boosting models experienced overfitting. Therefore, this study proposes the LightGBM method with hyperparameter tuning for forecasting stock market prices. This is done because the LightGBM method has little risk of overfitting. Then the proposed method will be compared with several boosting methods in previous studies, including XGBoost, CatBoost, and AdaBoost. The contributions to this research include the following. First, the proposed method can handle the risk of overfitting in predicting stock market prices. Both results of the proposed method can be used as a technology model to predict stock market prices. The three proposed methods can be a reference for future research related to forecasting in the economic field, especially stock market prices.

II. Material and Methods

This research was conducted based on several stages illustrated in Figure 1 below.

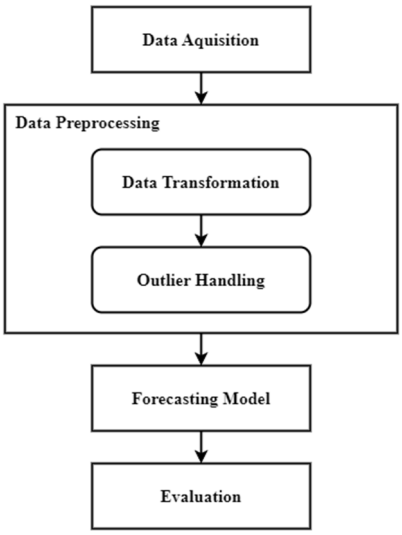

Fig. 1 Research Stages

The following subsections explain each step carried out in the research flow.

A. Dataset Acquisition

The data used in this study is stock data taken from Yahoo Finance using an application programming interface (API), namely finance. Time series data with the stock code AAPL was withdrawn from 01/01/1992 to 01/01/2022 with a total of 7558. Data pulled from finance was collected in July 2, 2022. The AAPL stock code dataset pulled from finance has six features: open, high, low, close, and volume. For research purposes, the data used is comparative column data because close is the market closing price. At the stock market's closing, the price will no longer change in recording stock transactions. This study divides the data into training data, data validation, and data testing. Data training starts from 02/01/1992 to 11/03/2011, data validation from 14/032011 to 30/12/2015, and data testing from 04/01/2016 to 31/12/2021.

B. Data Pre-processing

A critical element in forecasting besides data pre-processing refers to manipulating data so that the data becomes of higher quality or maintains the model's performance during training [32]. At this step, data manipulation does not mean altering data with bad intentions but processing data with specific techniques so that data becomes another form that data can display certain information or remove information that is not needed.

In this study, data pre-processing was carried out using data transformation techniques and outlier handling. The purpose of data transformation is to make data stationery. In short, stationary data can be described as data that has a constant mean and variance. Chlebus et al. [32] show that selecting models and variables included in the essential category in data processing and feature engineering or feature engineering is vital to get stationary data. They conducted several stock price forecasting experiments, one comparing forecasting using stationary and non-stationary data. The result is that stationary data have higher accuracy after differentiating data sets with the same reaction as the data sets used in the experiment.

Then outlier handling aims to eliminate outlier data. Outlier data or also known as anomalies in the data can cause forecasting bias. Outlier data is illustrated as data that does not follow the "flow" because it has a difference in value that is too high with the value of the previous index or the value of most indexes. This causes bias because the model learns data patterns to make predictions so that if the model encounters an outlier, it can affect the pattern studied. Therefore, data indicated as an outlier must be removed or replaced, generally replaced with the mean or median.

1) Data Transformation:

One of the elements that can affect the data is the unit root. The unit root is a stochastic (uncertain) attribute that can cause problems in drawing statistical conclusions, especially series data. To eliminate the unit root, data transformation can be performed, or it can also be called the de-trending process so that the data can be stationary before carrying out the data transformation. A unit root test, also known as the unit root process, is carried out to determine the trend of data based on mathematical calculations so that the data distribution can be known with certainty.

The Augmented Dickey-Fuller (ADF) [34], [35] and Kwiatkowski–Phillips–Schmidt–Shin (KPSS) [36], [37] tests were carried out. The ADF test was conducted to find out the unit root of the data, while the KPSS test was to find out the stationarity of the data. The parameters used in the ADF test are regression equal to 'ct,' which means constant and trend in ADF and KPSS, and the auto lag parameter is the same as 'Akaike's and Schwarz's Information Criteria' (AIC) in ADF. This is done so that the results obtained are more valid and can also determine the cause of the problem if there is an oddity in the data.

It can be seen in Table 1 that there are no signs of stationary data in the raw data column or raw data, namely data that has not gone through any treatment. The raw data has an ADF p-value of 1.0 and a KPSS p-value of 0.01; this means that the data has a trend or is not constant. The results of the ADF and KPSS tests can be interpreted by looking at Table 2, or it can be said that the raw data rejects the 0 KPSS hypothesis because the p-value is less than the alpha value (0.05) and the p-value of the raw data also accepts the 0 ADF hypothesis because the value is more than the alpha (0.05). From the two hypothesis tests, it can be concluded that the raw data is not stationary because it rejects the 0 KPSS hypothesis and accepts the 0 ADF hypothesis. The proof can be seen in Figure 2 above, and stock price data still has a trend in an exponential form that has risen suddenly in the last decade. Data that is not stationary has a mean and median that are not constant because they still have a trend, so a de-trending procedure must be carried out utilizing data transformation.

Data transformation can be done in various ways, including using square roots. Then the data that has undergone the procedure is tested again using KPSS and ADF. However, in Table 1, the sqrt column of the transformation results still does not change, so differentiation is made to replace the square root [32]. After differentiating unexpected things, it turns out that the ADF and KPSS tests have different conclusions on the data resulting from the differentiation

transformation; of course, this is remarkably interesting. This is because the differentiation transformation data no longer has a unit root, but the data is still not stationary but stationary differentiation. Data that has been trended out but still needs to be constant based on the KPSS test. Stationary data that should have a deterministic or definite mean becomes a stochastic or uncertain mean, and this is a particular case in this study.

Because square root data transformation and differentiation transformation cannot turn the data into static data, this study tries to combine the two techniques into differentiation roots. This experiment produced results; this can be seen in Figure 2. The image at the bottom of the graph shows stationary data located not far from the mean or with a deterministic mean. Then the ADF p-value is relatively tiny, but the data meets the stationary requirements, namely rejecting the 0 hypotheses because the value is less than the alpha value (0.05). Then the p-value of KPSS can also fulfill the stationary requirements because the value accepts the 0 hypotheses, or it can be said that the p-value is more than the alpha value (0.05).

TABLE I
ADF AND KPSS TEST

Metric	ADF p-value	KPSS p-value
Data	1.0	0.010
Square Root	1.0	0.010
Differentiation	0.000000000000000000003721	0.010
Square Root-Differentiation	0.0000000000000000000601 7	0.051

TABLE II
ADF AND KPSS TEST HYPOTHESIS

KPSS	ADF
H0: if the ***p-value > 0.05*** then the data is stationary.	**H0**: if the ***p-value > 0.05*** then the data is not stationary.
H1: if the ***p-value < 0.05*** then the data is not stationary.	**H1**: if the ***p-value < 0.05*** then the data is stationary.

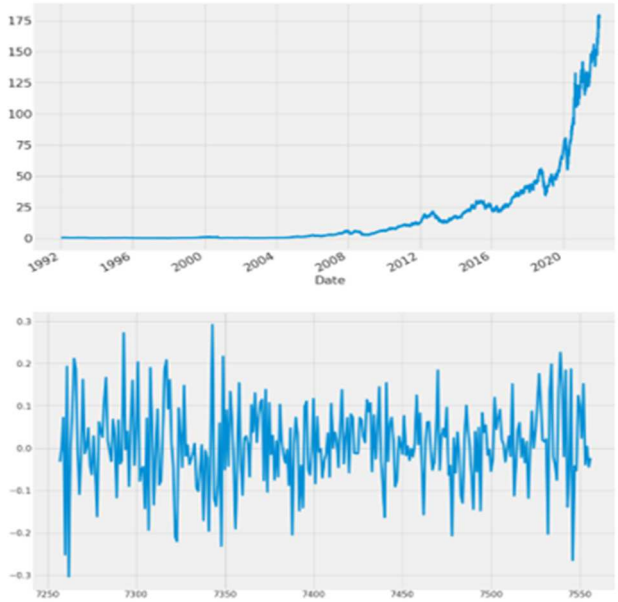

Fig. 2 Visualization of data conditions

2) Outlier Handling

One way to prevent biased forecasting results is to reduce or eliminate outlier data using specific techniques or methods. The most manageable technique to detect outliers is by visualizing data using graphs. Boxplots are the easiest way to display the distribution of data which is summarized into five indicators ("minimum," first quartile (Q1), median, third quartile (Q3), and "maximum") as shown in Figure 2. Based on the illustration in Figure 3, outlier data are points outside the box plot's minimum and maximum limits [38], [39].

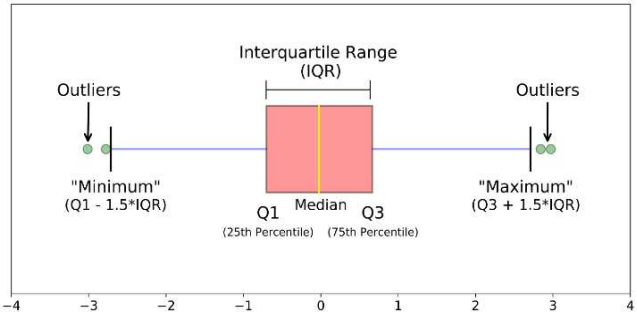

Fig. 3 Boxplot Visualization

Then in Figure 4, the top picture is a box plot visualization of the AAPL stock price data, which has many outliers, while good data is data with no outliers. This can also cause the data distribution to be abnormal, as in the distribution chart in Figure 4 below, which shows the frequency of abnormal data gathered in the median.

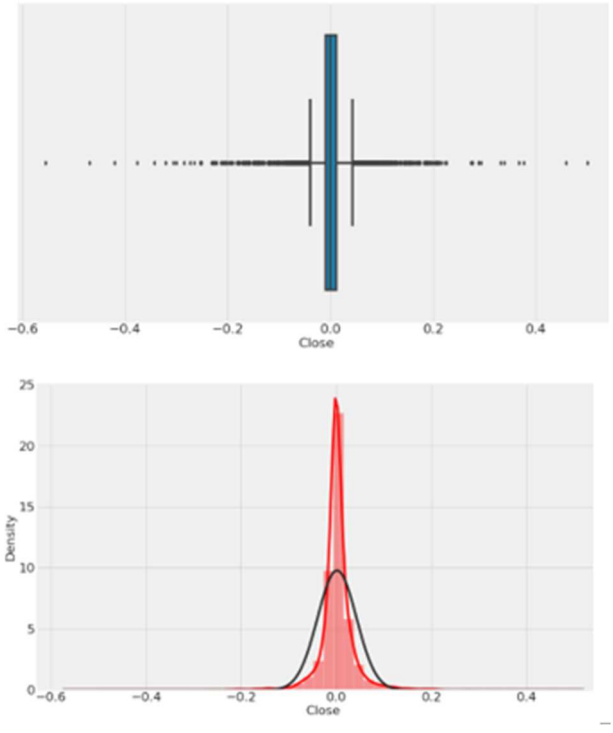

Fig. 4 Data Visualization Before Outlier Handling

The abnormal distribution of the data can also be proven by the high kurtosis and skewness values, as shown in Table 3. It turns out that the data distribution becomes abnormal after transforming the data. The kurtosis and skewness values are respectively 7,459 and 2,696, whereas normal kurtosis and skewness values are values greater than equal to negative

three and less than equal to three. For this reason, it is necessary to detect outliers and then take further action on them so that the distribution becomes normal.

TABLE III
KURTOSIS AND SKEWNESS VALUES BEFORE OUTLIERS HANDLING

	Kurtosis	Skewness
Before Transformation	7.459	2.696
After Transformation	26.668	-0.249

Tukey's method separates outlier probability (possibility) and probability (probability). The outlier probability value is between the inner (inside) and outer (outer) fence, while the outlier probability value is outside the outer fence. In contrast to the whisker, which is shaped like a horizontal line that connects the IQR box with the minimum and maximum limits, the inner fence and outer fence do not appear in the boxplot visualization. For this reason, the inner fence and outer fence are calculated by entering the IQR into equation (2), and to find the IQR values, it is done as in equation (1).

$$IQR = Q3 - Q1 \tag{1}$$

$$\begin{aligned} inner\ fence &= [Q1 - 1.5 * IQR, Q3 + 1.5 * IQR] \\ outer\ fence &= [Q1 - 3 * IQR, Q3 + 3 * IQR] \end{aligned} \tag{2}$$

ThymeBoost is a forecasting model like LightGBM, but in this study, ThymeBoost is used because of its ability to detect outliers in data by using the detect outlier function [39]. Then in, Table 4 shows the number of outliers netted by Tukey's Method and ThymeBoost. It can be seen that the Tukey method produces two outputs, as mentioned in the previous presentation, namely, the number of outlier probability values is 424, and the number of outlier probability values is 1038 then the number of ThymeBoost outlier values is 740.

TABLE IV
NUMBER OF OUTLIERS

Outlier	Tukey (prob)	Tukey (poss)	ThymeBoost
Amount	424	1038	740

Outlier handling can be done by removing the outliers based on the index of the captured data and then filling them back in with the mean or median value. Filling in the deleted values is known as imputed. Impute is done so that the information in the data is recovered. However, in Table 5, imputing the transformation data does not affect kurtosis.

Then experiment with a scaler or smooth the values so that the outliers are not too extreme. Smoothing the value using a scaler reduces a data value based on a specific range of values, generally zero to one. This process is called data normalization. In short, data normalization occurs when there is a significant difference between the smallest and largest values so that the most significant value is considered an outlier. Usually, the methods that are often used are the robust scaler, min-max scaler, standard scaler, and power scaler methods. However, the scaler method could more successfully overcome abnormal data distribution due to outliers. Then the next experiment removes outlier data based on data captured by Tukey's Method and ThymeBoost. It can be seen in Figure 5 that the distribution graph is normal with kurtosis and skewness values of 0.304 and 0.159, respectively, and also the boxplot shows that the data still has outliers, but these outliers are not too influential.

TABLE V
KURTOSIS AND SKEWNESS VALUES AFTER OUTLIERS HANDLING

	Kurtosis	Skewness
Imputer	28.324	-0.249
Scaler	26.667	-0.248
Delete (*outlier*)	0.304	0.159

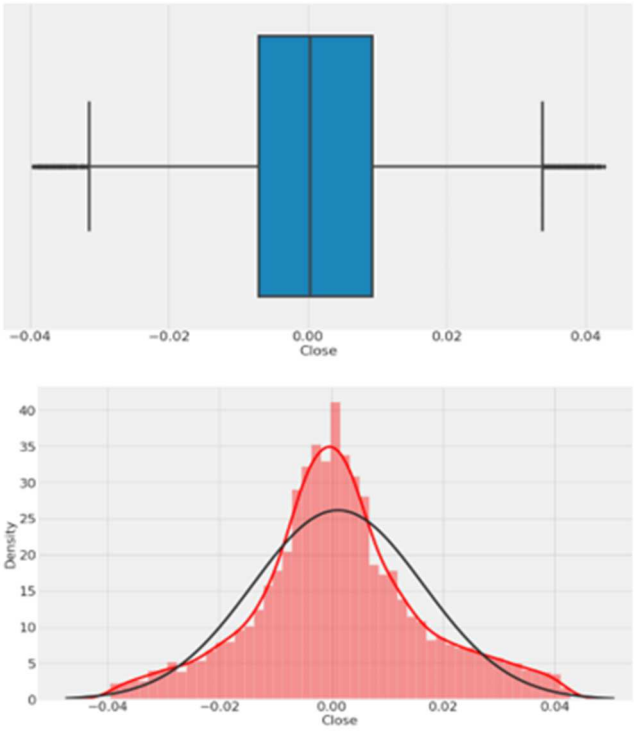

Fig. 5 Data Visualization After Outlier Handling

C. *Forecasting Model Using LGBM*

In this study, the model used is the Light Gradient Boosting Machine (LGBM). The LightGBM model has many hyperparameters, some of which can be adjusted to improve model performance, while others can shorten training time due to the large volume of data. In short, a hyperparameter is a parameter whose value controls the learning process and determines the value of a model parameter which is eventually learned by the learning algorithm. Therefore, the hyperparameter is one of the most critical variables and can determine the conclusion in machine learning [29], [30], [32], [33].

The technique used in determining hyperparameters is grid search cross-validation (GSCV). GSCV is a hyperparameter optimization standard commonly used in machine learning [40], [41], [42], [40]. GSCV has two methods: grid search (GS) and cross-validation (CV). GSCV looks for the best combination of parameter values by mapping the hyperparameters illustrated as the x-axis, then CV is illustrated as the y-axis so that cross-validation and parameter values intersect and when drawn vertical and horizontal straight lines will form a grid (grid) like Figure 6. Many of these iterations can be calculated by the number of parameters raised to the power of length from the list of each hyperparameter. The number of candidates hyperparameters is the number of CV folds multiplied by the number of

iterations, so there were 729 iterations and 3645 hyperparameter candidates in this study.

In Table 6, the 666th GSCV index has the best mean test score and mean training score because it is calculated based on the RMSE parameter in the model. The max_depth parameter is the depth of the model tree; max_depth can also overcome over-fitting if the training data set is small. Directly, overfitting means 'too fitting'; overfitting is a phenomenon where the training data used is 'too fitting' so that it can reduce accuracy if paired with other data. The max_depth parameter has an integer data type, meaning that the hyperparameter must be filled with integers. Then the training model uses the parameters bagging freq and bagging_fraction because the Random Forest (RF) algorithm requires both of these; however, GOSS cannot use 'bagging' at all. The value of bagging freq must be an integer greater than zero (n>0), and the bagging fraction value must be a decimal number greater than zero and less than one (1.0<n>0.0). "Bagging" means taking random samples without changing values during training. The bagging_freq parameter is equal to 20, and the bagging_fraction parameter is equal to 0.95. This means the model is told to resample without changing values every 20 iterations and sample 95 percent of the training data. However, the GOSS algorithm cannot use bagging freq and bagging fraction. Then n_estimator is an alias of num_iteration, and n_estimator must be assigned an integer value. The n_estimator parameter equals 100, which means the model must boost for 100 iterations. Then the learning_rate parameter or learning rate is a parameter that determines the step size of each iteration during the training process. The learning_rate value must be a decimal number that is greater than zero (n> 0). The smaller the learning_rate value, the higher the accuracy of the learning. The boosting_type parameter is filled with the boosting or algorithm compatible with LightGBM, the boosting type compatible with LGBM, namely GBDT, RF, GOSS, and DART. Then the last one is the objective, which has many parameters, some of which are regression, Huber, Poisson, and others. Then the objective parameter in the model is 'regression_l1', which means regularization or L1 regularization (regularization lasso). Regularization is a form of regression, and the model is given regularization to control over-fitting phenomena.

TABLE VI
SAMPLE OF GRID SEARCH CROSS VALIDATION RESULTS

Rank	Index	Params	Mean Test Score	Mean Train Score
1	666	{'bagging_fraction': 0.95, 'bagging_freq': 20, 'boosting_type': 'dart', 'learning_rate': 0.001, 'max_depth': 3, 'metric': 'rmse', 'n_estimators': 100}	0.03069857617	0.03079684814
364	242	{'bagging_fraction': 0.5, 'bagging_freq': 20, 'boosting_type': 'rf', 'learning_rate': 0.001, 'max_depth': 10, 'metric': 'rmse', 'n_estimators': 1000}	0.5276550848	0.5278574062
729	692	{'bagging_fraction': 0.95, 'bagging_freq': 20, 'boosting_type': 'gbdt', 'learning_rate': 0.01, 'max_depth': 10, 'metric': 'rmse', 'n_estimators': 1000}	0.9998572055	0.9998936874

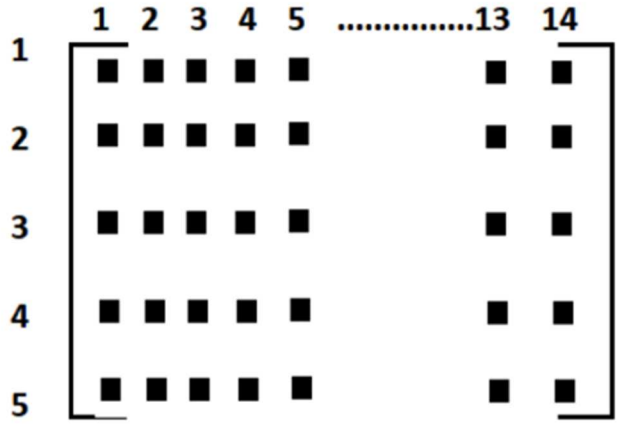

Fig. 6 Grid Search Cross Validation (GSCV) Illustration

D. Evaluation

In this study, testing was carried out based on the scenario that had been made. The scenarios tested vary according to needs. For example, researchers want to test algorithms compatible with LightGBM for the best results. This study will compare these algorithms with balanced hyperparameter settings in the sense that the parameters of the hyperparameters can work or are compatible with the algorithm. Then the best model is selected, then the model is compared with other alternative models.

In this study, testing uses metrics that are based on the course of research. The metric used is RMSE. RMSE compares the original data (testing data) with the predicted data. RMSE can be interpreted as "how concentrated the data is around the regression line" so the smaller the RMSE value, the better. However, RMSE is quite sensitive to outliers, so researchers use MAE and MedAE because they resist outliers.

MAE measures the average magnitude of error in the data, while MedAE measures the median value of the error. MAE and MedAE measure data error regardless of the direction of the data (positive/negative).

III. Results and Discussion

A. Experimental

Given the many challenges in the pre-processing circuit, experiments need to be carried out to get the best results. This step also provides insight or description of data that has undergone previous processes. In the outlier handling step, it is indicated that the data has many outliers, so the data distribution becomes abnormal. This can be seen in Table 3. This is also supported by the large number of outliers netted in Table 4. Because there are many outliers in the data, the data is treated so that it can remove outliers. In Table 4, the outlier detection methods, namely Tukey's Method and ThymeBoost, each provide two outputs and one output. Because each output has different conclusions, a step is needed to determine which output is the most ideal as a basis for conducting model training.

Initially, the stock price data that has been transformed is deleted based on the index data, which is indicated as an outlier. Then the data is entered into the training model with default hyperparameter settings. Default settings provide more "natural" results because the model is still not optimal so the data quality will be more visible. Table 7 and Figure 7 shows that the data subject to reduced probability outlier data from Tukey's Method has better RMSE, MAE, and MedAE scores than others. While the data subjected to reduction of Tukey's Method, outlier probability data has the worst score. From this, it can be concluded that the more outlier data that is captured, the data will have a better score.

TABLE VII
COMPARISON OF RESULTS BASED ON THE OUTLIER HANDLING METHOD

Metric	Tukey (poss)	Tukey (prob)	ThymeBoost
RMSE	**0.0248335694**	0.0402051601	0.0300122047
MAE	**0.0198679087**	0.0318043206	0.0244017374
MedAE	**0.0167887351**	0.0265773916	0.0217605065

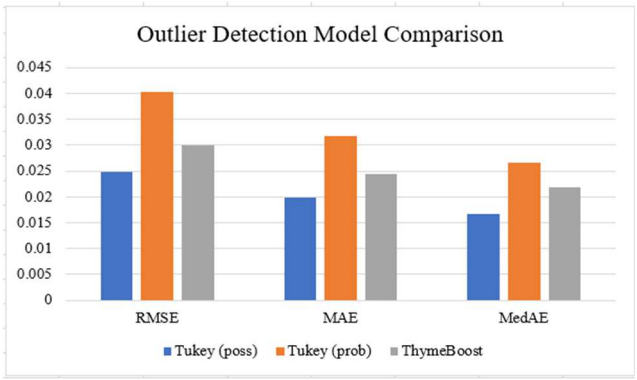

Fig. 7 Outlier Detection Model Comparison

After finding out a better data set, the next thing that needs to be proven is the effect of hyperparameters and the results of learning each LightGBM algorithm, such as DART, GOSS, RF, and GBDT. Then the hyperparameters are adjusted according to the GSCV results, and the max_bin parameters are added to develop training and validation data sets. The max_bin parameter means the data set has a unique maximum value per feature based on the max_bin value.

Table 8 and Figure 8 show that DART has the best RMSE, MAE, and MedAE scores. So, in the next stage, the algorithm used in the LightGBM model is DART because it is proven to have the best score compared to other algorithms.

TABLE VIII
COMPARISON OF LGBM

Metric	RMSE	MAE	MedAE
Default	0.02483357	0.01986791	0.0167887
GBDT	0.02180185	0.01790202	0.0161667
GOSS	0.02179386	0.01789748	0.0161734
RF	0.02169517	0.01780414	0.0160431
DART	**0.02145566**	**0.01765761**	**0.0160092**

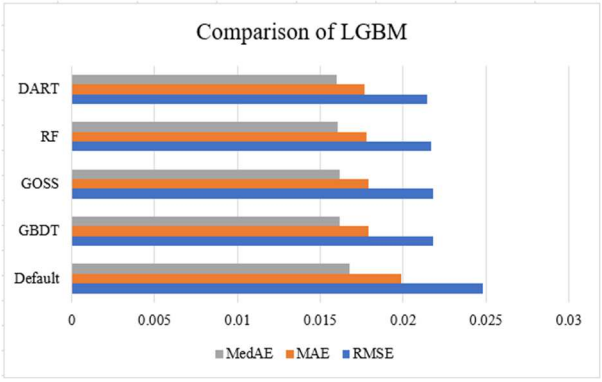

Fig. 8 Comparison of LGBM Model

B. Evaluation

In the model evaluation phase, alternative forecasting models such as XGBoost, Adaptive Boosting (AdaBoost), and CatBoost are used. In testing, the hyperparameter model is adjusted to GSCV, and the training and validation data sets are installed with the max_bin parameter. Table 9 and Figure 9 show that LightGBM has the lowest RMSE, MAE, and MedAE scores of the other alternative forecasting models. In short, LightGBM has the slightest error difference compared to other models. Then CatBoost took second place with a score difference just a short distance from LightGBM. Then the third rank is XGBoost, and the fourth rank is AdaBoost.

TABLE IX
LGBM VERSUS ANOTHER FORECASTING MODEL

Metric	LGBM	CatBoost	XGBoost	AdaBoost
RMSE	**0.021456**	0.021992	0.023581	0.024612
MAE	**0.017658**	0.017988	0.019075	0.019712
MedAE	**0.016009**	0.015980	0.016094	0.016481

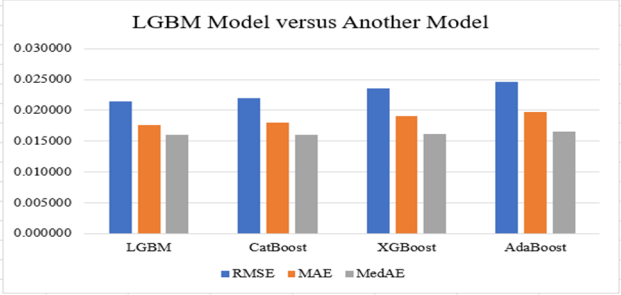

Fig. 9 LGBM Model versus Another Forecasting Model

To find out a model's ability and the data's validity, the researchers tested it by forecasting using data with different timescales. In practice, the hyperparameters are adjusted to the results of the GSCV. The alternative model used is the same as the previous evaluation: CatBoost, XGBoost, and AdaBoost. Based on the tests conducted, LightGBM occupies the top position despite using a different timeframe. In Table 10, LightGBM and other alternative models will give better results using more data.

TABLE X
FORECASTING RESULTS WITH DIFFERENT TIME RANGES

Range	Model	RMSE	MAE	MedAE
30 years (1992-2022)	LightGBM	**0.021456**	**0.017658**	0.016009
	CatBoost	0.021992	0.017988	**0.015980**
	XGBoost	0.023581	0.019075	0.016094
	AdaBoost	0.024612	0.019712	0.016481
20 years (2002-2022)	LightGBM	**0.031023**	**0.025378**	**0.021841**
	CatBoost	0.033460	0.027367	0.024269
	XGBoost	0.036289	0.029398	0.025747
	AdaBoost	0.037814	0.030789	0.026934
15 years (2007-2022)	LightGBM	**0.042737**	**0.035547**	**0.031844**
	CatBoost	0.046121	0.038053	0.033140
	XGBoost	0.051026	0.041805	0.035897
	AdaBoost	0.052323	0.042610	0.036483
10 years (2012-2022)	LightGBM	**0.052208**	**0.042680**	**0.038333**
	CatBoost	0.056176	0.046187	0.040557
	XGBoost	0.061559	0.050350	0.042521
	AdaBoost	0.063670	0.052129	0.043877
5 years (2017-2022)	LightGBM	**0.063721**	**0.051706**	**0.048172**
	CatBoost	0.067825	0.055354	0.050920
	XGBoost	0.076624	0.061588	0.056169
	AdaBoost	0.079039	0.062733	0.052944
1 year (2021-2022)	LightGBM	**0.095871**	**0.072390**	**0.046005**
	CatBoost	0.101959	0.077168	0.050629
	XGBoost	0.118092	0.094113	0.073922
	AdaBoost	0.128051	0.102729	0.085067

IV. CONCLUSION

This study uses stock price time series data derived from the Yahoo Finance and LightGBM ensemble model as a forecasting model. LightGBM's ability to use the DART algorithm is proven superior in obtaining RMSE, MAE, and MedAE scores compared to other alternative forecasting models. In comparing forecasting models, the same dataset is used to make the results accurate and the comparisons equivalent. This is supported by the increased capability of the post-GSCV hyperparameter tuning model compared to the default hyperparameter setting. Then the researcher found an exciting discovery when conducting unit root tests on the data. The conclusions given by the KPSS and ADF tests are contradictory, even though the data has gone through a transformation using square roots. The ADF states that the data accept the stationary hypothesis, while the KPSS states the opposite. For this reason, researchers use square root differentiation to transform the data into stationary.

In addition, this study also encountered several obstacles in the pre-processing stage because the stock price data had many outliers after undergoing the data transformation stage. This condition indicates the distribution graph's extreme shape and the high kurtosis value. For this reason, efforts are made to smooth the values or normalize using scalers such as min-max scalers, robust scalers, standard scalers, and power transformers. However, these efforts did not produce results. This is because the outlier deviation is too apparent, so data indicated as an outlier must delete or handled using another method. In future research, paying attention to the data during pre-processing is necessary because it has many outliers. In addition, it is necessary to include exogenous variables and external variables, which are determined to involve many parties.

ACKNOWLEDGMENT

The authors thank the Department of Information Systems, Faculty of Computer Science, and Department for Research and Community Service, Universitas Amikom Yogyakarta, for supporting this research.

REFERENCES

[1] M. M. S. Saragih, T.Nurhaida, S. Sinaga, R. N. Ilham, and Faisal, "The impact of the Covid-19 pandemic on stock performance : Evidence from Indonesia," *Manag. Res. Behav. J.*, vol. 1, no. 1, pp. 1–6, 2021.

[2] H. Rezaei, H. Faaljou, and G. Mansourfar, "Stock price prediction using deep learning and frequency decomposition," Expert Systems with Applications, vol. 169, p. 114332, May 2021, doi: 10.1016/j.eswa.2020.114332.

[3] R. Chandra and Y. He, "Bayesian neural networks for stock price forecasting before and during COVID-19 pandemic," PLOS ONE, vol. 16, no. 7, p. e0253217, Jul. 2021, doi: 10.1371/journal.pone.0253217.

[4] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," Neural Computing and Applications, vol. 33, no. 10, pp. 4741–4753, Nov. 2020, doi: 10.1007/s00521-020-05532-z.

[5] H. T. H. Ton and T. K. Dao, "The Effects of Psychology on Individual Investors' Behaviors: Evidence from the Vietnam Stock Exchange," Journal of Management and Sustainability, vol. 4, no. 3, Aug. 2014, doi: 10.5539/jms.v4n3p125.

[6] E. Mulyani, H. Fitra, and F. F. Honesty, "Investment Decisions: The Effect of Risk Perceptions and Risk Propensity for Beginner Investors in West Sumatra," *Seventh Padang Int. ...*, vol. 192, no. Piceeba, pp. 49–55, 2021.

[7] W. Budiharto, "Data science approach to stock prices forecasting in Indonesia during Covid-19 using Long Short-Term Memory (LSTM)," Journal of Big Data, vol. 8, no. 1, Mar. 2021, doi:10.1186/s40537-021-00430-0.

[8] M. Shahvaroughi Farahani and S. H. Razavi Hajiagha, "Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models," Soft Computing, vol. 25, no. 13, pp. 8483–8513, Apr. 2021, doi: 10.1007/s00500-021-05775-5.

[9] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," Pattern Recognition, vol. 121, p. 108218, Jan. 2022, doi:10.1016/j.patcog.2021.108218.

[10] E. S. Abdulla, A. Hamdan, and H. Akeel, "The Impact of Artificial Intelligence on Financial Institutes Services During Crisis: A Review of the Literature," in *Digitalisation: Opportunities and Challenges for Business*, 2023, pp. 642–655.

[11] H. Yu, L. J. Ming, R. Sumei, and Z. Shuping, "A Hybrid Model for Financial Time Series Forecasting—Integration of EWT, ARIMA With The Improved ABC Optimized ELM," IEEE Access, vol. 8, pp. 84501–84518, 2020, doi: 10.1109/access.2020.2987547.

[12] Z. Li, J. Han, and Y. Song, "On the forecasting of high-frequency financial time series based on ARIMA model improved by deep

learning," Journal of Forecasting, vol. 39, no. 7, pp. 1081–1097, Mar. 2020, doi: 10.1002/for.2677.
[13] U. M. Sirisha, M. C. Belavagi, and G. Attigeri, "Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison," IEEE Access, vol. 10, pp. 124715–124727, 2022, doi: 10.1109/access.2022.3224938.
[14] T. C. Nokeri, "Forecasting Using ARIMA, SARIMA, and the Additive Model," in *Implementing Machine Learning for Finance: A Systematic Approach to Predictive Risk and Performance Analysis for Investment Portfolios*, Berkeley, CA: Apress, 2021, pp. 21–50.
[15] Z. Fang, X. Ma, H. Pan, G. Yang, and G. R. Arce, "Movement forecasting of financial time series based on adaptive LSTM-BN network," Expert Systems with Applications, vol. 213, p. 119207, Mar. 2023, doi: 10.1016/j.eswa.2022.119207.
[16] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional Neuro-Sequential ARFIMA-LSTM for Financial Market Forecasting," IEEE Access, vol. 8, pp. 71326–71338, 2020, doi: 10.1109/access.2020.2985763.
[17] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," Physica A: Statistical Mechanics and its Applications, vol. 519, pp. 127–139, Apr. 2019, doi:10.1016/j.physa.2018.11.061.
[18] Q. Gu, Y. Chang, N. Xiong, and L. Chen, "Forecasting Nickel futures price based on the empirical wavelet transform and gradient boosting decision trees," Applied Soft Computing, vol. 109, p. 107472, Sep. 2021, doi: 10.1016/j.asoc.2021.107472.
[19] T. Le, B. Vo, H. Fujita, N.-T. Nguyen, and S. W. Baik, "A fast and accurate approach for bankruptcy forecasting using squared logistics loss with GPU-based extreme gradient boosting," Information Sciences, vol. 494, pp. 294–310, Aug. 2019, doi:10.1016/j.ins.2019.04.060.
[20] F. Zhou, Q. Zhang, D. Sornette, and L. Jiang, "Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices," Applied Soft Computing, vol. 84, p. 105747, Nov. 2019, doi: 10.1016/j.asoc.2019.105747
[21] F. I. Durrah, Y. Yulia, T. P. Parhusip, and A. Rusyana, "Peramalan Jumlah Penumpang Pesawat Di Bandara Sultan Iskandar Muda Dengan Metode SARIMA (Seasonal Autoregressive Integrated Moving Average)," Journal of Data Analysis, vol. 1, no. 1, pp. 1–11, Sep. 2018, doi: 10.24815/jda.v1i1.11847.
[22] N. S. Arunraj, D. Ahrens, and M. Fernandes, "Application of SARIMAX Model to Forecast Daily Sales in Food Retail Industry," International Journal of Operations Research and Information Systems, vol. 7, no. 2, pp. 1–21, Apr. 2016, doi:10.4018/ijoris.2016040101.
[23] A. Khumaidi, R. Raafi'udin, and I. P. Solihin, "Pengujian Algoritma Long Short-Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung," *J. Telemat.*, vol. 15, no. 1, pp. 13–18, 2020.
[24] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," The Annals of Statistics, vol. 29, no. 5, Oct. 2001, doi:10.1214/aos/1013203451.
[25] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, vol. 2017-Decem, no. Nips, pp. 3147–3155.
[26] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions," International Journal of Forecasting, vol. 38, no. 4, pp. 1346–1364, Oct. 2022, doi: 10.1016/j.ijforecast.2021.11.013.
[27] T. Chen et al., "Prediction of Extubation Failure for Intensive Care Unit Patients Using Light Gradient Boosting Machine," IEEE Access, vol. 7, pp. 150960–150968, 2019, doi: 10.1109/access.2019.2946980.
[28] A. A. Taha and S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," IEEE Access, vol. 8, pp. 25579–25587, 2020, doi:10.1109/access.2020.2971354.
[29] M. Gan, S. Pan, Y. Chen, C. Cheng, H. Pan, and X. Zhu, "Application of the Machine Learning LightGBM Model to the Prediction of the Water Levels of the Lower Columbia River," Journal of Marine Science and Engineering, vol. 9, no. 5, p. 496, May 2021, doi:10.3390/jmse9050496.
[30] P. Pokhrel, "A LightGBM based Forecasting of Dominant Wave Periods in Oceanic Waters," in *Proceedings of ACM Conference on Information and Knowledge Management. (CIKM'21)*, 2018, vol. 9.
[31] A. M. Husein and M. Harahap, "Pendekatan Data Science untuk Menemukan Churn Pelanggan pada Sector Perbankan dengan Machine Learning," Data Sciences Indonesia (DSI), vol. 1, no. 1, pp. 8–13, Nov. 2021, doi: 10.47709/dsi.v1i1.1169.
[32] M. Chlebus, M. Dyczko, and M. Woźniak, "Nvidia's Stock Returns Prediction Using Machine Learning Techniques for Time Series Forecasting Problem," Central European Economic Journal, vol. 8, no. 55, pp. 44–62, Jan. 2021, doi: 10.2478/ceej-2021-0004.
[33] J. Li, "Monthly Housing Rent Forecast based on LightGBM (Light Gradient Boosting) Model," *NCCP Int. J. Intell. Inf. Manag. Sci.*, vol. 7, no. 6, pp. 2307–0692, 2018.
[34] J. Wang, T. Ji, and M. Li, "A Combined Short-Term Forecast Model of Wind Power Based on Empirical Mode Decomposition and Augmented Dickey-Fuller Test," Journal of Physics: Conference Series, vol. 2022, no. 1, p. 012017, Sep. 2021, doi: 10.1088/1742-6596/2022/1/012017.
[35] M. Ahmed et al., "Bubble Identification in the Emerging Economy Fuel Price Series: Evidence from Generalized Sup Augmented Dickey–Fuller Test," Processes, vol. 10, no. 1, p. 65, Dec. 2021, doi:10.3390/pr10010065.
[36] A. Kagalwala, "kpsstest: A command that implements the Kwiatkowski, Phillips, Schmidt, and Shin test with sample-specific critical values and reports p-values," The Stata Journal: Promoting communications on statistics and Stata, vol. 22, no. 2, pp. 269–292, Jun. 2022, doi: 10.1177/1536867x221106371.
[37] Marsani, Ani Shabri, Basri Badyalina, Nur Amalina Mat Jan, and Mohd Shareduwan Mohd Kasihmuddin, "Efficient Market Hypothesis for Malaysian Extreme Stock Return: Peaks over a Threshold Method," *Mat. Mjim*, vol. 38, no. 2, pp. 141–155, 2022.
[38] H. Alimohammadi and S. Nancy Chen, "Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis," Expert Systems with Applications, vol. 191, p. 116371, Apr. 2022, doi: 10.1016/j.eswa.2021.116371.
[39] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A Review on Outlier/Anomaly Detection in Time Series Data," ACM Computing Surveys, vol. 54, no. 3, pp. 1–33, Apr. 2021, doi: 10.1145/3444690.
[40] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," International Journal of Computers and Applications, vol. 44, no. 9, pp. 875–886, Sep. 2021, doi:10.1080/1206212x.2021.1974663.
[41] M. Adnan, A. A. S. Alarood, M. I. Uddin, and I. ur Rehman, "Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models," PeerJ Computer Science, vol. 8, p. e803, Feb. 2022, doi: 10.7717/peerj-cs.803.
[42] S. Chen, H. Zhu, W. Liang, L. Yuan, and X. Wei, "A Stock Index Prediction Method and Trading Strategy Based on the Combination of Lasso-Grid Search-Random Forest," in *Intelligent Computing and Block Chain*, 2021, pp. 431–448.