JOIV : Int. J. Inform. Visualization, 7(4) - December 2023 2255-2262

INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

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

Stock Price Movement Classification Using Ensembled Model of Long Short-Term Memory (LSTM) and Random Forest (RF)

Albertus Emilio Kurniajaya Gunawan [a,*], Antoni Wibowo [a]

[a] *Computer Department, Binus Graduate Program - Master of Computer Science Bina Nusantara University, Jakarta, Indonesia*
Corresponding author: [*]albertus.gunawan001@binus.ac.id

***Abstract*— Stock investing is known worldwide as a passive income available for everyone. To increase the profit possibly gained, many researchers and investors brainstorm to gain a strategy with the most profit. Machine learning and deep learning are two of these approaches to predicting the stock's movement and deciding the strategy to gain as much as possible. To reach this goal, the researcher experiments with Random Forest (RF) and Long Short-Term Memory (LSTM) by trying them individually and merging them into an ensembled model. The researcher used RF to classify the results from LSTM models obtained throughout the Hyperparameter Optimization (HPO) process. This idea is implemented to lessen the time needed to train and optimize each LSTM model inside the ensembled model. Another anticipation done in this research to overcome the time needed to train the model is classifying the return for longer periods. The dataset used in this model is 45 stocks listed in LQ45 as of August 2021 This research results in showing that LSTM gives better results than RF model especially when using Bayesian Optimization as the HPO method, and that the ensembled model can return better precision in predicting stocks in comparison to the LSTM model itself. Future improvement can focus on the model structure, additional model types as the ensemble model estimator, improvement on the model efficiency, and datasets research to be used in predicting the stock movement prediction.**

***Keywords*— Machine learning; deep learning; long short-term memory; random forest; ensembled model; bayesian optimization; random search; stock investing; classification.**

Manuscript received 13 Feb. 2023; revised 5 Aug. 2023; accepted 3 Sep. 2023. Date of publication 31 Dec. 2023.

I. INTRODUCTION

Stock investing has piqued everyone's interest nowadays because of how big the profit one can gain [1]. Research on stock predicting is still actively developed to predict stock patterns, with noisy environments and extremely complicated factors affecting the stock market becoming the challenges in this topic [2], [3]. In stock investing, the investor decides on three actions: to buy, to hold, or to sell the corresponding stock. There are two types of analysis: technical and fundamental[4]. Technical analysis focuses on the price and trading volume to predict the movement of the stock, while fundamental analysis predicts the long run of the stock using the underlying company characteristics [5]. In order to support these methods, big data will indeed be needed to help make decisions [6].

There are many research approaches to improve the accuracy of stock prediction. Some researchers improved their model by adding more input features or modifying them. In a Research led by Zheng Tan, momentum features improved their stock selection strategy performance significantly [5]. Another research conducted by Ghosh used intraday features as input features, which gives both LSTM and RF models a boost on their return [7]. Another input feature used in stock prediction methods is text-based opinions such as Twitter feed and news, oil and gold prices, and technical indicators such as RSI and bias ratio [8]–[11].

Furthermore, research done by Naik proved how useful the Boruta feature selection technique is in identifying a relevant technical indicator [12]. Research on DeepClue, a text-based model, gives a higher accuracy compared to LSTM and RF by using some text-based information such as financial news and social media [13]. Besides adding some features, there are other ways to improve the model's accuracy by modifying the used features. One method was adjusting the number of input features and target vectors [14]. Another method was done by applying volume up on the data, where the later the data, the more impact it will have compared to the previous ones to handle the biased data [15]. Besides the research on input features, many other studies also encourage adding and applying feature selection for their future works.

Some research is conducted comparing the accuracy between two or more models. In Qian's research with Google Stock as a dataset, LSTM and ARIMA are being compared, which results in LSTM giving a lower Root Mean Square Error (RMSE) compared to ARIMA model [6]. Another comparison was done on China Securities 100 index stock, comparing six different models of Deep Multilayer Perceptron (DMLP), LSTM, Convolutional Neural Network (CNN), RF, Support Vector Regression (SVR), and Autoregressive Integrated Moving Average (ARIMA) which resulting in LSTM gives the highest accuracy, followed by RF [16]. Another comparison was conducted on Support Vector Machine (SVM), Perceptron Neural Network, and Linear Regression (LR) by Parray to compare all three performances on stock price prediction as a time series data, where SVM outperformed the other models [17]. Patil did similar research, comparing graph-based models with ARIMA, where the graph-based model gave better accuracy, yet it was vulnerable to exploding gradient [18]. In research conducted by Nikou, LSTM outperforms the other two methods, SVR and LR [19]. Nabipour compared nine models, consisting of 5 different tree models: RF, LSTM, Artificial Neural Network (ANN), and Deep Neural Network (DNN). In their research, Adaboost was the best tree model with the highest accuracy, while overall, LSTM outperformed the other models in exchange for higher runtime [20]. Other research compares Back Propagation Neural Networks (BPNN) against Fuzzy algorithm, resulting in BPNN giving a better result [21]. In 2020, Ta V. created and compared models based on LSTM, LR and SVM, where LSTM gives the highest accuracy compared to the others [22].

Another type of stock predicting research focuses on modifying the models, either by combining a different model or experimenting with the model's hyperparameter. Many ensembled models are made by combining the model with another model or algorithm. Nti made an ensembled model of Decision Tree (DT), SVM, and Multilayer Perceptron (MLP) with some different ensembling techniques [23]. Other ensembled models are made from the Gbest algorithm and Artificial Bee Colony model, resulting in a model with high accuracy on all test environments compared to theormal Artificial Bee Colony model [24]. Another ensembled model was produced by combining Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), Autoregressive Moving Average (ARMA), and LSTM, taking each model's advantage on stock predicting such as CEEMDAN with its ability in data decomposition, LSTM with its ability to handle time series data, and ARMA to combine linear and non-linear model [25]. Agrawal introduced the Evolutionary Deep Learning Approach (EDLA) with LSTM, outperforming LR. SVM, and Enhanced LSTM (ELSTM) [26]. Lu presented another ensembled model of CNN, BiLSTM, and Attention Mechanism (AM), using CNN to extract the input data, BiLSTM to predict the extracted features, and AM to detect the relevance of the feature states with the prediction result. Pawar combined Recurrent Neural Network (RNN) with LSTM, making several different combined structures based on LSTM [27].

Previous research shows some interesting points to be considered in making stock predictions. The first point is the used data itself. The data used in stock predicting will affect the prediction result, including how the data is processed before use. The next point is model's effectiveness in each case. In stock prediction, the data used are mostly in time order, which means the data sequence will play a huge part in data prediction. LSTM and RF appeared a lot of times in stock, predicting high accuracy compared to other models. Third, there are ways to enhance the model performance, such as assembling several models by combining them, exploring the best hyperparameter set, and modifying the model itself based on the needs. Hence, the researchers of this paper will try to create a model based on those three points. The data used will be daily prices of stocks listed in LQ45 from August 2021 to January 2022 using Yahoo Financial library. The model used is an Ensemble of LSTM with its ability to handle sequenced data and RF to handle highly biased data. There will be 2 based methods of Hyperparameter Optimization (HPO) used in this research Random Search (RS) and Bayesian Optimization (BO). This research will use Google Colab Pro+ with Python3 as the platforms.

II. Materials and Methods

There are a lot of studies with different approaches to creating models to help the decision-making in stock investing. For example, the research conducted by Vijh focused on predicting the exact closing price for the stock, while the research conducted by Guptaocused on classifying the stock price movement instead [3], [28].

From the conducted research, one of the most used models is LSTM and RF. Research conducted by Ma Y compared Deep Multilayer Perceptron (DMLP), LSTM, Convolutional Neural Network (CNN), Support Vector Regression (SVR), RF, and Autoregressive Integrated Moving Average (ARIMA), LSTM and RF were the two best models out of these 6 [16]. On the other hand, another research was conducted by Nabipour, where the tree models were compared to the artificial neural network (ANN), deep neural network (DNN), and LSTM. From the tree models, Adaboost gave the best result out of the tree models, while LSTM still gave the best result out of all models tried. However, compared to tree models, LSTM consumed significantly more time. However, there was an interesting part in one of the research projects conducted previously, where an ensemble model of DNN, Gradient Boosted Tree (GBT) and RF gives off a better result than RF itself [20].

The data for stock price movement prediction can be divided into fundamental and technical data. The basic data that can be retrieved for technical data are open, close, high, low, adjusted close, and volume. This data can be processed into a lot of more useful information. In research conducted by Tan, there are some modified data such as moving average, momentum, volume acceleration, volatility, and standard deviation [5]. Meanwhile, Ghosh uses intraday return, return concerning the last closing price, and return concerning the opening price [7].

In building Machine Learning, two parameters exist in the model itself. The first one is the normal parameter inside the model, which is updated through the training process, whereas the second one is the parameter used to create the model itself, which is then called as hyperparameter [29].

This research will try to compare some Deep Learning models from both classification performance and investing strategy evaluation based on the classification results themselves using LQ45 list in August 2021 to January 2022. Before the experiment, the stock dataset will need to be checked and fixed first. The LQ45 dataset will be taken from Yahoo Finance using YahooFinancial library from 2010 to 2020. Stocks registered after the year 2010 will be removed first, and stocks with at least 5% missing data will also be removed from this research. The invalid data includes both missing data and anomalies like having a volume with zero value or less. Afterwards, the rest of the missing data will be rechecked once more. The common missing dates from all stock will be removed, and the rest will be filled using linear interpolation.

Next, the researcher will extract features from the base data. The features are shown in table 1. These features are taken from the previous work of Ghosh and Tan [5] and [7] and added with a new column of RSI, which measures the relative strength between the momentum of positive change and negative change [30]. After the feature extraction, the next step is to prepare the data by rescaling them between 0 and 1 and model them into the shape needed for training and testing. The dataset needed for each prediction model can be seen in Table 2. The dataset will then be ready to be used to train the models, which are divided into three groups: RF, LSTM, and Ensembled model of RF and LSTM.

TABLE I
FEATURES DESCRIPTION

Feature	Description
Intraday return	Difference between close price and open price in the same day
Overnight return	Difference between the close price and open price on the next day
Daily return	Difference between close prices between 2 days
Turnover_t	Moving average from t period
Close_0/close_t	Momentum to identify stock price trend
Close_t/close_0	Reversed momentum
Adjclose_0/ adjclose_t	Stock momentum without paying much attention to the latest price
Vol_0/vol_t	Volume acceleration to identify volume trend line
Volatility	Stock return volatility
Std	Standard deviation from stock price return
RSI	A technical indicator to support decision-making on buying or selling stock

To obtain a better result from the model, some trials and errors must be conducted against the model's hyperparameter. Thus, this research applied Hyperparameter Optimization with Bayesian Optimization and Random Search and compared the time and accuracy of both techniques. After all the classification results from all the models, an exploration will be conducted to check whether the prediction is invalid. This step will be done by checking the output classification for the testing model. Simply put, the model will be marked as invalid when it only gives a constant output, either sell (-1) or buy (1). When this condition happened to one of the models, it can be said that the model is not working properly, and the HPO failed to find a proper combination.

A. RF

This model is originally created by assembling some decision tree models. The principal behind this model is simple. The word "Random Forest" itself suggests that this model would aggregate some random trees. Random Forest works by building several trees, each with input features. Differing the input features can be done simply by removing some features, allowing each tree to be trained from a subset of features instead. Each tree will be trained with its own subset of input data and produce its results. The outcome from each tree created will be used as a "vote" to decide the final conclusion as output. This principle of using a lot of tree models is meant to reduce the effect of overfitting, which is suitable for stock data [31]. RF model is illustrated in Fig. 1.

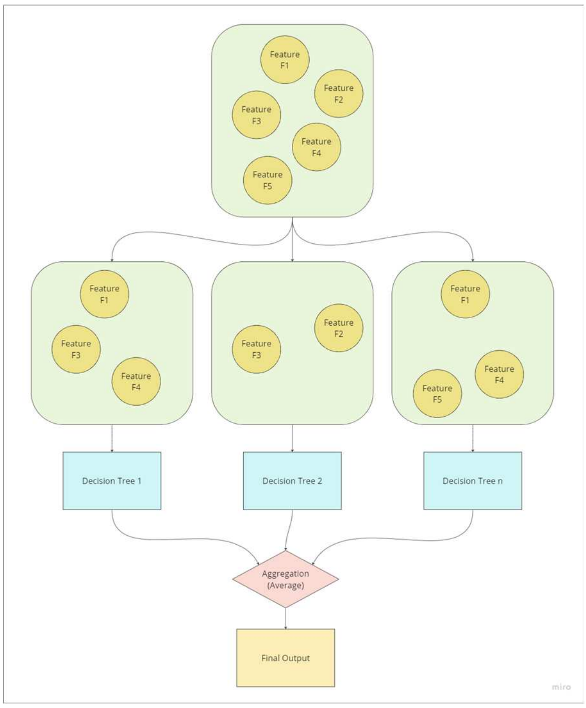

Fig. 1 Random Forest Structure

There are some extensions or modifications for Random Forest modeling. Weighted Forest is a modified Random Forest, where each tree will have its weight to highlight trees with more accurate predictions. Online Forest is another developed Random Forest, which does not necessarily need all the training sets to be accessible at once, enabling Online Forest to generate training data over time and stream data environments. These modified models show that Random Forest can be altered in many ways and adapted to the related environments [32].

B. LSTM

ANN implements human brain function by using interconnected neurons to process information [33]. RNN is an advanced version of ANN with a connected hidden layer that enables RNN to handle long-ranged dependencies datasets. However, in real practice, this model has proven to be effective [34]. The reason is a phenomenon called vanishing gradient and exploding gradient. Vanishing happens because the information's effectiveness decreases dramatically, while the exploding gradient increases the information's effectiveness, causing the model to be unable to work properly [35]. An illustration of this phenomenon can be found in Fig. 2.

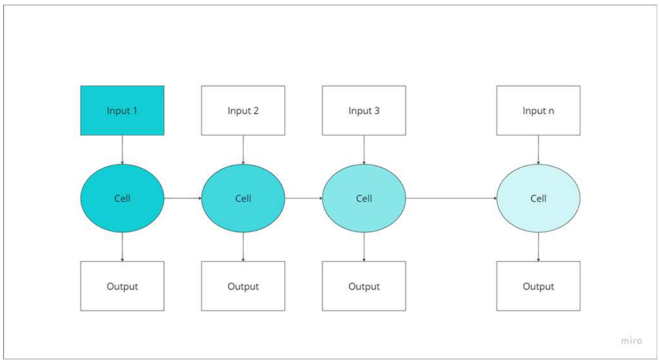

Fig. 2 Vanishing Gradient

LSTM was then created to solve those main problems using a memory block component. This block has three function gates that can filter the importance of each information [36]. The first gate, the input gate, filters the newest information. The next gate is called the forget gate, whose function is to remove the information no longer needed from the input gate. The last gate is called the output gate, which computes the output result and lets them flow into the hidden cell for the next input [37], [38]. The illustration for LSTM vanilla model can be seen in Fig. 3.

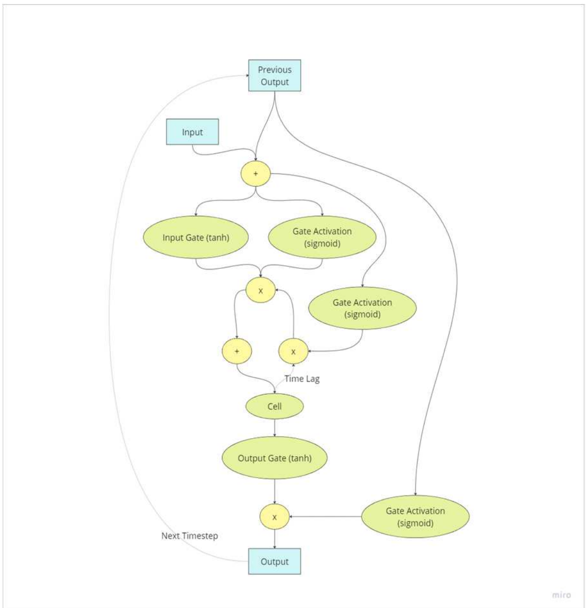

Fig. 3 LSTM Structure

C. *Ensembled Model*

The proposed model in this research is an ensembled model from LSTM and RF. In basic RF, there will be a collection of decision tree models, and the final output is voted from tallying all the trees' predictions. This research will modify RF by replacing decision trees with LSTM models instead. LSTM model to be used in this are LSTM model created through the hyperparameter optimization (HPO) process, which will have a low correlation from one to another even by having the same dataset. Of all the LSTM created in the HPO process, only x models with the highest validation accuracy will be used in the Ensembled Model. This method will help save the time and resources needed to optimize several LSTM models, which approximately will be the same time and resources needed to train and optimize the LSTM model. The output of this model itself will be -1 (sell) and 1 (buy). The workflow of the ensembled model is illustrated in Fig 4.

Adopting from RF, the ensembled model will return the output calculated from all x models. Two types of Ensembled models will be created based on the Random Forest modifications. The first one will use a vanilla Random Forest, which will return the average of each model's output as the final prediction. The second model will implement Weighted Tree, where each tree will be weighted based on accuracy. In this research, each LSTM models inside will be multiplied by its train accuracy, and the final output will have the average of the weighted output from each LSTM. When the final result is more than 0, the final prediction is to buy. Otherwise, when the final result is less than 0, the final prediction is to sell the stock.

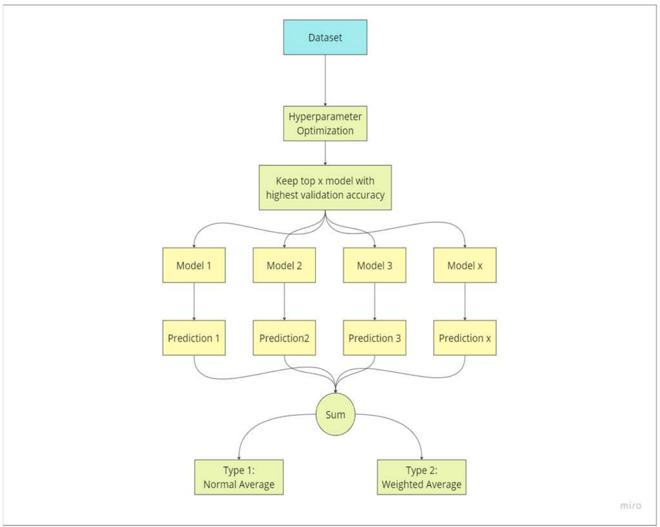

Fig. 4 Ensembled Model

D. *Bayesian Optimization*

Bayesian optimization is one of many methods that can solve functions without closed-form expression and is expensive to calculate. Bayesian has two important parts: updating the posterior distribution and maximizing the acquisition function [39]. Gaussian process (GP) is commonly applied by Bayesian optimization as a tool to update the posterior distribution. GP works by changing scalar value into a probability distribution using mean and covariance functions. This probability distribution becomes the posterior distribution, which will processed through the acquisition function to get the local optima from the posterior distribution. These steps are repeated until one of two conditions is fulfilled, whether the iteration process reached the designated limit, or the optimal value change is no longer significant [29].

E. *Grid Search*

Grid Search is an algorithm that finds the best combination of hyperparameters, which can be considered a brute force method. This algorithm can be divided into two main steps. The first step of Grid Search is to define the set of values for each hyperparameter to be explored. After having a set of values for each hyperparameter, the next step is to combine each value and try to create a model. This step will be repeated until all possible combinations are iterated [29]. Grid Search flow is illustrated in Fig. 5.

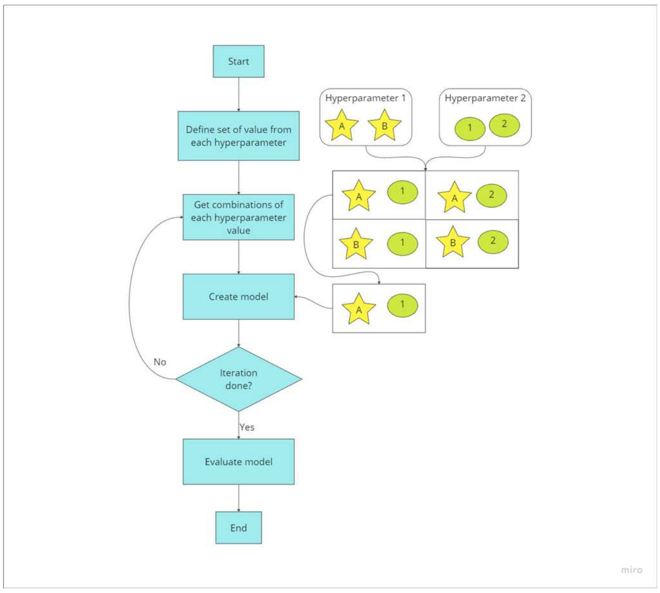

Fig. 5 Grid Search

Because of the second step, Grid Search is considered inefficient when creating a model with many hyperparameters. The more the hyperparameter is, the more the grid would increase exponentially, which also increases the iteration exponentially. In this case, Grid Search can only be effective and usable in cases with a small number of hyperparameters [29].

F. Random Search

Random Search is one of the algorithms used to optimize function. This Algorithm is the same as Grid Search, except random search picks the parameter combinations randomly instead of fixed positions. This algorithm has a higher chance of finding better optimization than a simple Grid Search [40].

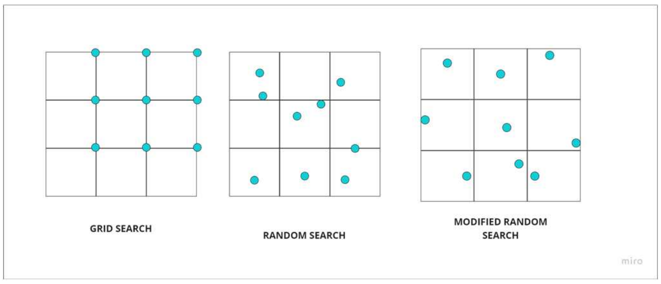

Fig. 6 Grid Search, Random Search, and Modified Random Search

In this research, the random search for LSTM will be modified a bit due to the lack of RAM available. Modified search is an ensembled method of Grid Search and Random Search. This modification is applied since the resources needed to run the HPO are limited. To make up for the lack of resources, Grid Search will be applied on Random Search to make sure the sampled set of hyperparameters is more scattered. The illustration for the search result is illustrated in Fig. 6.

The modified Random Search method is done by applying a Grid Search step to define each parameter's value set. Instead of defining a value for each parameter, this method will define a set of ranges instead. After defining the range set of each parameter, the next step is to pick each combination of value range sets for each parameter. Random Search will be applied in this step by randomizing each hyperparameter from the range value set picked in each hyperparameter. These steps are then repeated until all combinations are iterated or reached a time limit. The flow comparison of Random Search, Grid Search, and the modified Random Search can be observed in Fig. 7.

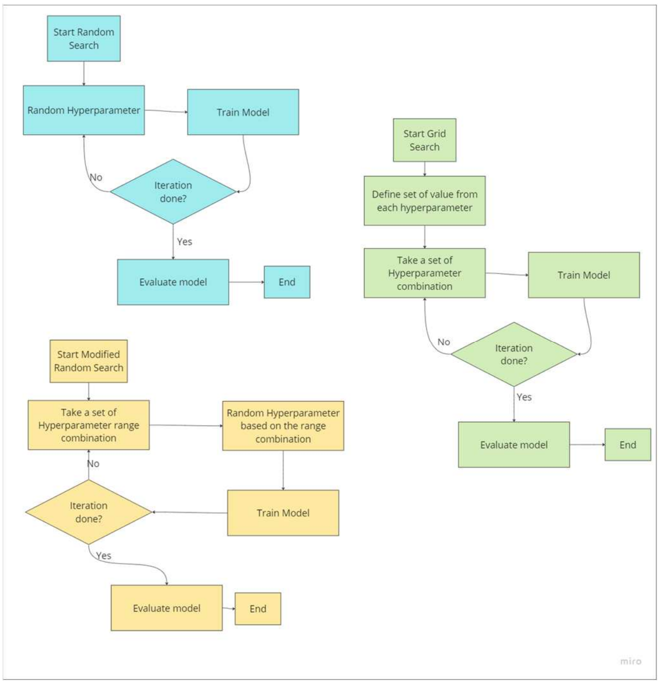

Fig. 7 Flow of Random Search, Grid Search, and Modified Random Search

III. Results and Discussion

Fig. 8 shows the main flow of this research. The first step of this research is to retrieve stock data from Yahoo Finance using a python library named YahooFinancial. The dataset used in this research is daily prices data ranging from 2010 to 2020 from stocks listed in LQ45 starting from August 2021. After retrieving the data, the next step is data preprocessing. In this step, stock with a lot of invalid and unusable data will be removed from the dataset used, such as negatively valued prices and missing data on some dates. Stocks with 5% missing data will be excluded from this research, while the other missing data will be replaced using interpolation using the adjacent dates. After the data cleaned up, the next step is to transform it into features mentioned in Table 1 and group into sequential data of 240 sequential data, with buy and sell classification as its output. This classification is done by comparing the close price of the latest training data against the close price of the next seven days.

The next step is to define the model used to predict the stock price. This step has two parts: Hyperparameter Optimization (HPO) and model base. The HPO used in this research are Random Search (RS) and Bayesian Optimization (BO), while the models used are Random Forest (RF), Long Short-Term Memory (LSTM), and ensembled model of RF and LSTM. These models will each be paired with each HPO method. After having the paired HPO and models, the training will be done per stock dataset. To shorten the time needed to run these models, each notebook will run 7 to 8 stocks in parallel, while there are five possible notebook to be executed at once using Google Colab Pro+ with high RAM capacity.

Each model will then predict the stock price movement and save it into an Excel file in Google Drive and compared against each other in terms of training time needed, accuracy, recall, precision, and F1 Score. Underfit models will also be observed here. Models that only predict 0 (sell) in all test days will be classified as underfit models.

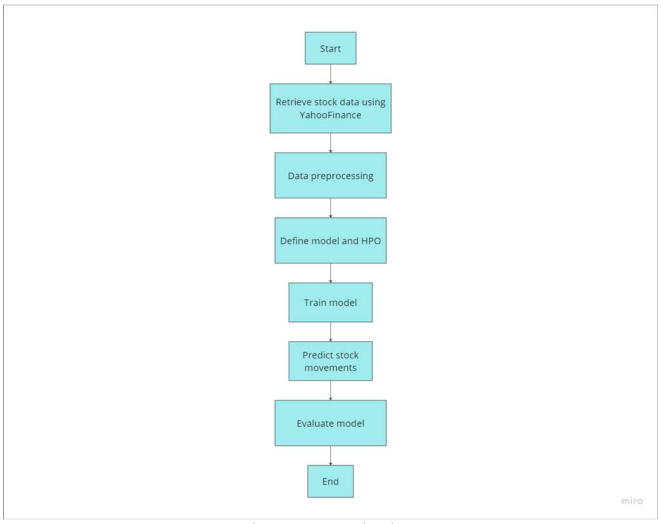

Fig. 8 Research Flow

TABLE II
BUY AND SELL DISTRIBUTION

Stock Name	Class Sell	Class Buy	Stock Name	Class Sell	Class Buy
ACES.JK	998	742	INTP.JK	1061	679
ADRO.JK	1080	660	ITMG.JK	1059	681
AKRA.JK	1036	704	JPFA.JK	1022	718
ANTM.JK	1131	609	JSMR.JK	1154	586
ASII.JK	1123	617	KLBF.JK	1105	635
BBCA.JK	1104	636	MEDC.JK	1078	662
BBRI.JK	1045	695	MNCN.JK	1010	730
BBTN.JK	1009	731	PGAS.JK	1067	673
BMRI.JK	1060	680	PTBA.JK	1027	713
BRPT.JK	1086	654	PWON.JK	978	762
BSDE.JK	1047	693	SMGR.JK	1090	650
CPIN.JK	992	748	SMRA.JK	1011	729
EXCL.JK	1077	663	TINS.JK	1118	622
GGRM.JK	1073	667	TKIM.JK	1171	569
HMSP.JK	1181	559	TLKM.JK	1104	636
ICBP.JK	1118	622	UNTR.JK	1031	709
INCO.JK	1031	709	UNVR.JK	1100	640
INDF.JK	1144	596	WIKA.JK	1011	729
INKP.JK	1041	699			

The first result of this research is the comparison of 8 different combinations of Hyperparameter Optimization (HPO) methods and classification models. There are 2 HPO methods that will be compared in this research: Bayesian Optimization and Random Search. As for the classification models, there are four models to be compared: RF, LSTM, and two types of Ensembled models. The first type of ensembled model will get the average of all the prediction output, while the second type uses a weighted average by adding the training accuracy as the weight of each prediction output. Of 37 predicted stocks, seven stocks will be used and analyzed in this research, as these seven stocks are the ones with the least underfit models, the result of which can be seen in Table III.

TABLE III
MODEL AND HPO COMPARISON

Model x HPO	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Training time (second)
RF with BO	63.630	0	0	0	4814.14
RF with RS	63.630	0	0	0	8427.38
LSTM with BO	62.746	51.023	32.110	33.815	30545.05
LSTM with RS	63.696	20.402	17.134	18.160	66571.34
Ensembled Model 1 with BO	63.630	0	0	0	30545.05
Ensembled Model 1 with RS	63.630	0	0	0	66571.34
Ensembled Model 2 with BO	63.630	0	0	0	30545.05
Ensembled Model 2 with RS	63.630	0	0	0	66571.34

From the result, some conclusions can be taken. The first one is that LSTM models are the ones that give out the result with a less underfit model when predicting the stocks, while RF and ensembled models failed to do so. BO hyperparameter gives faster results and higher F1 scores compared to the RS method. Continuing from this result, this research modified the ensembled model by reducing the models used as the estimator from 21 to 7, 5, and 3 models and comparing them. These ensembled models will only use BO as HPO as it gave off the better result compared to RS as HPO in previous results.

TABLE IV
MODIFIED ENSEMBLE MODEL COMPARISON

N estimator	Models	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
21	LSTM	62.746	51.023	32.110	33.815
	Ensembled model type 1	63.630	0	0	0
	Ensembled model type 2	63.630	0	0	0
7	LSTM	61.664	44.431	21.975	28.544
	Ensembled model type 1	63.794	50.222	21.029	21.666
	Ensembled model type 2	63.794	50.222	21.029	21.666
5	LSTM	62.844	46.431	29.740	32.321
	Ensembled model type 1	65.138	60.514	18.413	24.754
	Ensembled model type 2	65.138	60.514	18.413	24.754
3	LSTM	61.337	43.801	26.676	32.446
	Ensembled model type 1	63.106	45.996	23.633	29.942
	Ensembled model type 2	63.105	45.996	23.633	29.942

Table IV shows an interesting fact: the recall for the ensembled model is better from the LSTM baseline only when it adopted 5 LSTM, while the model with 3 and 7 LSTM as estimators has slightly higher recall compared to each LSTM baseline. It can be concluded that the number of LSTM estimators used in this ensembled model will affect how it performs, and it gives a higher recall in exchange for precision.

IV. Conclusion

This paper proposed Ensemble Model of LSTM and RF for stock price classifications of stock listed in LQ45 as per August 2021. To measure the performance of this ensembled model, this model was compared against a vanilla LSTM and RF. As a result, the best model to predict the stock movement classification was Ensembled model, with an accuracy of 65.138% and a precision of 60.514%. This ensembled model was based on 5 LSTM models from BO Hyperparameter Optimization process of LSTM, which produced 62.844% accuracy and 46.431% precision as the best model. This proves the proposed model gives a more precise output than the LSTM. Another point from this research was that BO completely outperforms RS as Hyperparameter Optimization from runtime and model quality. From the first round of this research with LSTM, using BO as HPO completely outperformed RS in accuracy, precision, and even recall, with the time needed less than half the time needed by RS model.

Further improvement can be done by focusing more on one of the processes, such as refining the dataset used by adding or removing more features, selecting stock with more price movement history, and even adding fundamental data or other things related to stock price movement like news. Another way to improve this research is by modifying the LSTM model used as the RF, adding the variant used as the RF tree nodes instead of just LSTM, or changing the RF into another tree model such as Gradient Boosted Tree.

References

[1] Z. Jiang, “Forecast of Stock Price,” Advances in Economics, Business and Management Research, pp. 1529–1539, 2022, doi: 10.2991/978-94-6463-052-7_169.

[2] D. Kumar, P. K. Sarangi, and R. Verma, “A systematic review of stock market prediction using machine learning and statistical techniques,” Materials Today: Proceedings, vol. 49, pp. 3187–3191, 2022, doi:10.1016/j.matpr.2020.11.399.

[3] R. Gupta and M. Chen, “Sentiment Analysis for Stock Price Prediction,” 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Aug. 2020, doi:10.1109/mipr49039.2020.00051.

[4] R. Rosillo, D. de la Fuente, and J. A. L. Brugos, “Technical analysis and the Spanish stock exchange: testing the RSI, MACD, momentum and stochastic rules using Spanish market companies,” Applied Economics, vol. 45, no. 12, pp. 1541–1550, Apr. 2013, doi:10.1080/00036846.2011.631894.

[5] Z. Tan, Z. Yan, and G. Zhu, “Stock selection with random forest: An exploitation of excess return in the Chinese stock market,” Heliyon, vol. 5, no. 8, p. e02310, Aug. 2019, doi:10.1016/j.heliyon.2019.e02310.

[6] H. Qian, “Stock Predicting based on LSTM and ARIMA,” Advances in Economics, Business and Management Research, pp. 485–490, 2022, doi: 10.2991/978-94-6463-036-7_72.

[7] P. Ghosh, A. Neufeld, and J. K. Sahoo, “Forecasting directional movements of stock prices for intraday trading using LSTM and random forests,” Finance Research Letters, vol. 46, p. 102280, May 2022, doi: 10.1016/j.frl.2021.102280.

[8] M.-C. Lee, “Research on the Feasibility of Applying GRU and Attention Mechanism Combined with Technical Indicators in Stock Trading Strategies,” Applied Sciences, vol. 12, no. 3, p. 1007, Jan. 2022, doi: 10.3390/app12031007.

[9] S. Krishna Lakshminarayanan and J. McCrae, "A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction," 2019.

[10] H. Maqsood et al., “A local and global event sentiment based efficient stock exchange forecasting using deep learning,” International Journal of Information Management, vol. 50, pp. 432–451, Feb. 2020, doi:10.1016/j.ijinfomgt.2019.07.011.

[11] N. Jing, Z. Wu, and H. Wang, “A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction,” Expert Systems with Applications, vol. 178, p. 115019, Sep. 2021, doi:10.1016/j.eswa.2021.115019.

[12] N. Naik and B. R. Mohan, “Stock Price Movements Classification Using Machine and Deep Learning Techniques-The Case Study of Indian Stock Market,” Communications in Computer and Information Science, pp. 445–452, 2019, doi: 10.1007/978-3-030-20257-6_38.

[13] L. Shi, Z. Teng, L. Wang, Y. Zhang, and A. Binder, “DeepClue: Visual Interpretation of Text-Based Deep Stock Prediction,” IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 6, pp. 1094–1108, Jun. 2019, doi: 10.1109/tkde.2018.2854193.

[14] Y. Song, J. W. Lee, and J. Lee, “A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction,” Applied Intelligence, vol. 49, no. 3, pp. 897–911, Oct. 2018, doi: 10.1007/s10489-018-1308-x.

[15] E. Koo and G. Kim, “A Hybrid Prediction Model Integrating GARCH Models With a Distribution Manipulation Strategy Based on LSTM Networks for Stock Market Volatility,” IEEE Access, vol. 10, pp. 34743–34754, 2022, doi: 10.1109/access.2022.3163723.

[16] Y. Ma, R. Han, and W. Wang, “Portfolio optimization with return prediction using deep learning and machine learning,” Expert Systems with Applications, vol. 165, p. 113973, Mar. 2021, doi:10.1016/j.eswa.2020.113973.

[17] I. R. Parray, S. S. Khurana, M. Kumar, and A. A. Altalbe, “Time series data analysis of stock price movement using machine learning techniques,” Soft Computing, vol. 24, no. 21, pp. 16509–16517, Apr. 2020, doi: 10.1007/s00500-020-04957-x.

[18] P. Patil, C.-S. M. Wu, K. Potika, and M. Orang, “Stock Market Prediction Using Ensemble of Graph Theory, Machine Learning and Deep Learning Models,” Proceedings of the 3rd International Conference on Software Engineering and Information Management, Jan. 2020, doi: 10.1145/3378936.3378972.

[19] M. Nikou, G. Mansourfar, and J. Bagherzadeh, “Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms,” Intelligent Systems in Accounting, Finance and Management, vol. 26, no. 4, pp. 164–174, Oct. 2019, doi:10.1002/isaf.1459.

[20] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. S., “Deep Learning for Stock Market Prediction,” Entropy, vol. 22, no. 8, p. 840, Jul. 2020, doi: 10.3390/e22080840.

[21] D. Zhang and S. Lou, “The application research of neural network and BP algorithm in stock price pattern classification and prediction,” Future Generation Computer Systems, vol. 115, pp. 872–879, Feb. 2021, doi: 10.1016/j.future.2020.10.009.

[22] V.-D. Ta, C.-M. Liu, and D. A. Tadesse, “Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading,” Applied Sciences, vol. 10, no. 2, p. 437, Jan. 2020, doi: 10.3390/app10020437

[23] I. K. Nti, A. F. Adekoya, and B. A. Weyori, “A comprehensive evaluation of ensemble learning for stock-market prediction,” Journal of Big Data, vol. 7, no. 1, Mar. 2020, doi: 10.1186/s40537-020-00299-5.

[24] H. Shah, N. Tairan, H. Garg, and R. Ghazali, “A Quick Gbest Guided Artificial Bee Colony Algorithm for Stock Market Prices Prediction,” Symmetry, vol. 10, no. 7, p. 292, Jul. 2018, doi:10.3390/sym10070292.

[25] P. Lv, Q. Wu, J. Xu, and Y. Shu, “Stock Index Prediction Based on Time Series Decomposition and Hybrid Model,” Entropy, vol. 24, no. 2, p. 146, Jan. 2022, doi: 10.3390/e24020146.

[26] M. Agrawal, P. Kumar Shukla, R. Nair, A. Nayyar, and M. Masud, “Stock Prediction Based on Technical Indicators Using Deep Learning Model,” Computers, Materials & Continua, vol. 70, no. 1, pp. 287–304, 2022, doi: 10.32604/cmc.2022.014637.

[27] K. Pawar, R. S. Jalem, and V. Tiwari, “Stock Market Price Prediction Using LSTM RNN,” Emerging Trends in Expert Applications and Security, pp. 493–503, Nov. 2018, doi: 10.1007/978-981-13-2285-3_58.

[28] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, “Stock Closing Price Prediction using Machine Learning Techniques,” Procedia Computer Science, vol. 167, pp. 599–606, 2020, doi:10.1016/j.procs.2020.03.326.

[29] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” Neurocomputing, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.

[30] S. Chen and L. Ge, “Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction,” Quantitative

Finance, vol. 19, no. 9, pp. 1507–1515, Jul. 2019, doi:10.1080/14697688.2019.1622287.
[31] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, “Predicting the direction of stock market prices using tree-based classifiers,” The North American Journal of Economics and Finance, vol. 47, pp. 552–567, Jan. 2019, doi: 10.1016/j.najef.2018.06.013.
[32] G. Biau and E. Scornet, "A Random Forest Guided Tour," 2015. Accessed: Aug. 15, 2022. [Online]. Available: https://arxiv.org/abs/1511.05741
[33] S.-C. Wang, “Artificial Neural Network,” Interdisciplinary Computing in Java Programming, pp. 81–100, 2003, doi:10.1007/978-1-4615-0377-4_5.
[34] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, “Deep Learning with Long Short-Term Memory for Time Series Prediction,” IEEE Communications Magazine, vol. 57, no. 6, pp. 114–119, Jun. 2019, doi: 10.1109/mcom.2019.1800155.
[35] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, “A Novel Connectionist System for Unconstrained Handwriting Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855–868, May 2009, doi:10.1109/tpami.2008.137.
[36] S. Chakraborty, J. Banik, D. Chatterjee, and S. Addhya, "Study of Dependency on number of LSTM units for Character based Text Generation models," 2020.
[37] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A Search Space Odyssey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/tnnls.2016.2582924.
[38] S. Badillo et al., “An Introduction to Machine Learning,” Clinical Pharmacology & Therapeutics, vol. 107, no. 4, pp. 871–885, Mar. 2020, doi: 10.1002/cpt.1796.
[39] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," Journal of Electronic Science and Technology, vol. 17, no. 1, pp. 26–40, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.
[40] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," 2012. [Online]. Available: http://scikit-learn.sourceforge.net.