International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Interconnected Stocks Examination for Predicting the Next Day's High on the Indonesian Stock Exchange

Andreas Werner Sihotang[1], Andrea Stevens Karnyoto[2], and Bens Pardamean[1,2]

[1] *Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480*
[2] *Bioinformatics and Data Science Research Center, Bina Nusantara University Jakarta, Indonesia, 11480*

E-mail address: andreas.sihotang@binus.ac.id, andrea.stevens@binus.edu, bpardamean@binus.edu

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: We observed in many WhatsApp/Telegram Indonesian stock market groups, but we didn't find any stock prediction method that utilizes interconnectivity between stocks. In this paper, we examined the interconnected stock dynamics in the IDX and used it to predict the next day's high. We employed a novel method called "Connected Stocks + Rolling Window Method" which uses both the temporal dynamics of the stock market and the interconnectedness of IDX's stocks. We explored the characteristics of the interconnected stocks by implementing three machine learning algorithms - K-nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF) - and found valuable insight. The experiment showed that several factors including a balanced threshold model and increased stock input size helped the performance of a model, while several factors including window size, additional features added, and using specific sectors as training data did not help the model's performance. The result also showed that several stocks like ANTM and ERAA show signs of interconnectedness and are influenceable while some like KLBF are hard to influence and show no sign of interconnectedness based on their results. This research contributes to a deeper understanding of stock market dynamics on the IDX, especially the characteristics of interconnected stocks on the IDX.
Keywords: Stock prediction, machine learning, support vector machine, random forest, indonesian stock market

1. INTRODUCTION

Based on our observations on various WhatsApp/Telegram Indonesian stock market groups, we didn't find any technique that explores the connectivity among the listed stocks in the Indonesian Stock Exchange (IDX). This investigation has motivated us to examine the relationships between stocks using a machine learning approach.

The stock market, one of the foundations in the economy, is a marketplace where investors buy and sell stocks. The concept of a stock market works so well, that by having a better understanding of it, you are shown to be able to predict economic cycles [1]. By default, everyone started to try and predict the flow of the stock, and thus the world of stock market prediction came into existence with its ever-growing tool of techniques and models [2].

At first, most tools that are models that people use as a guideline on stock prediction relied on traditional analysis utilizing features and macroeconomic indicators [3]. However, standard machine learning methods became more popular and were starting to be applied in this area due to their capabilities [4]. Several models, such as Random Forest and Support Vector Machines have been used to discern patterns and relationships between stock data [5], [6]. Other models, like logistic regression and K-nearest neighbor were mainly implemented due to their effective and simple way of classifying stock price movements [7]. Improvement on the world of stock prediction started to focus on time-series analysis due to how time itself can give context to a stock and that the ability to capture the temporal dynamics of stock price movements shows promise [8]. It can be seen why while traditional methods might have valuable insights, they often struggle capturing the market's volatility and non-linearity which machine learning has shown promise on doing and improving upon [9].

The Indonesian Stock Exchange (IDX), when compared to other major stock market indexes of other large countries, has a small market capitalization. Other than that, a study found that specific stock groups in the IDX were found to be volatile, which shows high risks [10]. Another paper found that specifically fiscal impacts

E-mail:author's email

were likely ineffective due to the government's usage of paying debt instead of investing showcasing that the country can impact itself negatively in its stock market [11]. Lastly, Purnomo and Rider [12] found that surprisingly foreign stocks like the U.S or Japan has a very small influence on Indonesia's stock market. All of this shows that IDX has a problem of being a volatile stock market to work with for investors, while also showing that it is not affected by foreign stocks, but instead its own policies and stocks which might be due to its small market capitalization [13].

We examined this issue and used the interconnected stock behavior that's in the IDX to predict the next day's high via binary classification. We predicted whether the next day's high is higher than 1.5% of today's close to ensure that we have a profit of 1% since the fee in total to pay for trading using Mirae Asset Sekuritas is 0.4%. We implemented a novel method called "Connected Stocks + Rolling Window" method which captures the temporal dynamic of stocks via rolling window and captures the interconnected stock dynamics via proper sequencing. To ensure that the method work on different types of machine learning model, we employed several machine learning models, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forests (RF).

Our key contributions are summarized as follows:

- We examined the interconnected property of stocks in IDX by predicting the next day's high of a stock.
- We implemented three different machine learning models to learn the data to ensure that the evaluation results will be generalized. Furthermore, we will gain insight regarding the characteristics of these three models when being used with potential interconnected stocks.
- We conducted extensive experiments to find the characteristics of the interconnected stocks in IDX and their ability to predict the next day's high by analyzing different variables / inputs and its impact to the models ability in predicting.

The paper is organized as follows. Section 2 provides an extensive review of related works, showcasing the different ways stock prediction is explored and relevant literature that unmasks the characteristic of the IDX. In Section 3, we show our methodology, including our flowchart, data collection, pre-processing, model training, and evaluation method of our model. In section 4, we describe our lists of experiments and tasks in detail, showcasing the different ways we experiment with the variables to learn the characteristics of the model. Section 5 displays the results of our experiment, with proper discussion to further give an understanding on interconnected stocks, IDX, and the model's performance. In section 6, we conclude the paper by summarizing the main findings and their implications in stock prediction on the IDX.

2. RELATED WORKS

A. The Stock Market

The stock market is a complicated area that both reflects and influences economic activities. A study that looked at the connection between stock markets and big economic indicators, found that looking at historical data can help predict how the market might change, especially considering key factors like inflation and industrial production growth [14]. The study found that these key factors work in predicting volatility in both long-term and short-term changes by using different models and show how closely linked the economy and the stock market are.

Another study approached this way of thinking regarding the stock market's role in making investment decisions and its ability to predict economic cycles [1]. By emphasizing the market's potential as a predictor of GNP components and the business cycle, the research challenges the idea that stock market ups and downs are just random noise. It highlights the importance of understanding the relationship between stock prices and investment decisions, considering things like required returns on equity and the cost of capital. Galeotti and Schiantarelli [15] explores the intricate relationship between stock market volatility and investment decisions, comparing fundamental and non-fundamental factors to it. They found that changes in investment are significantly correlated with movements in both fundamental and non-fundamental components of stock prices. However, a significant difference arises in their influence on investment decisions, with fundamentals having a more substantial impact compared to non-fundamentals. These findings emphasize how economic factors and stock market evaluations can influence financial decision-making.

B. Machine Learning in Stock Market Prediction

With the previous explanation and understanding of the stock market, it can be seen why machine learning is a powerful tool within the financial sector, particularly for the prediction and analysis of stock market prices [3]. The utilization of various machine learning paradigms, including supervised and unsupervised algorithms, ensemble methods, time series analysis algorithms, and deep learning models, has become commonplace in addressing stock price prediction challenges [3], [5], [6], [16], [17], [18].

The reason for using machine learning in stock prediction is its ability to use historical stock market data as a valuable source of information. Their predictive power comes from their ability to apply these patterns to predict future trends, offering valuable insights for making investment decisions [3]. Their adaptability allows them to find subtle and non-linear relationships which help provide understanding on the dynamic nature of stock market movements [3], [7].

Huang, Capretz, and Ho [17] did an innovative study on using machine learning for predicting stock prices. By analyzing a comprehensive dataset covering 22 years of quarterly financial data, the study revealed relevant findings based on fundamental analysis. The Random Forest model stood out by providing superior prediction results, proving itself as a powerful tool for forecasting stock prices. When feature selection was applied using Random Forest, it significantly improved the performance of other models where the combination of them into a unified framework even outperformed the benchmark DJIA index during testing in regard to their portfolio score. Leung, MacKinnon, and Wang [6] paper delved into business intelligence (BI) systems and structural support vector machines (SSVMs) for stock price prediction. The paper suggested using a minimum graph cutting algorithm to efficiently solve the optimization problem, drawing parallels between the SSVM's separation oracle and maximum a posteriori (MAP) inference. Their experiment shows the practicality and effectiveness of this method in predicting stock prices achieving higher accuracy compared to many existing systems according to domain experts. The main highlight

C. K-Nearest Neighbors (KNN) in Stock Market

K-Nearest Neighbors (KNN) is a flexible data mining technique widely used for classification tasks. Its core concept involves categorizing an unknown sample by considering the known classifications of its neighboring elements within a training set [19]. KNN, using a specific distance function, selects the k nearest neighbor to the element and classifies the class of the new element based on its neighbors and their distance to the element [20]. The parameter 'k' here denotes the number of neighbors to consider which its example can be seen in Fig. 1, where the illustrations denote two ‘k’ parameters, which are k = 3 and k = 5.

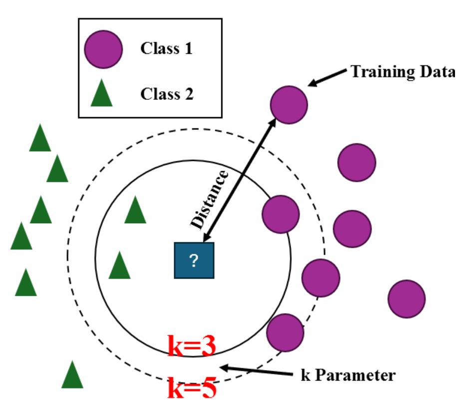

Figure 1 Illustration of K-Nearest Neighbor

KNN finds application in stock market analysis, particularly in prediction and classification tasks. In stock prediction, KNN commonly identifies the k nearest neighbors in the training dataset based on the Euclidean distance from the instance being classified [18].

Imandoust and Bolandraftar [21] explores the application of the KNN algorithm in economic forecasting, emphasizing its versatility across various domains, including stock market forecasting. Due to its robustness to noisy data, KNN can be effective even with large training dataset and with its simplicity, effectiveness, and flexibility it can be considered a valuable tool during stock prediction. Subha and Nambi [18] using these advantages of KNN trained their data and explored the predictability of stock index movement using the KNN algorithm, while drawing comparisons with the traditional Logistic Regression model. They achieved an overall %error score of 20.35% for the KNN Classifier, whereas Logistic Regression had a higher %error score of 45.89% showing the effectiveness of the KNN Classifier.

D. Support Vector Machines (SVM) in Stock Market

Support Vector Machines (SVM) is a powerful machine learning algorithm widely employed in various domains, including stock market prediction. While originally unpopular, SVM became popular when they showed they could do really well in practical tasks like recognizing digits, understanding images, and sorting text [22], [23]. A big strength of SVM is that it's particularly effective in situations where the relationship between input features and the output is complex and non-linear[22], [24]. The reason for this is that SVM operates by finding a hyperplane that best separates data points into different classes while maximizing the margin between these classes[22], [25]. This can be seen in Fig. 2, where the illustration showcases how by having a hyperplane with maximizing margin can classify large sums of data well [26], [27].

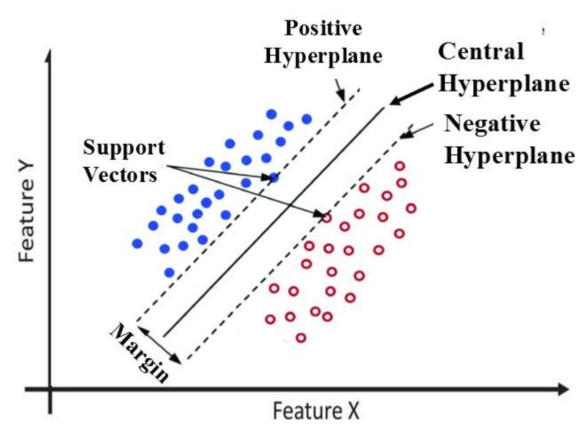

Figure 2 Illustration of Support Vector Machine

SVMs are well known to be effective at classifying because it’s able to find a good balance between two different ways of solving problems [28]. They figure out a straight line for making decisions, but they can also turn the data into a more complex form using something called kernels [25], [29]. The kernel function allows SVM to implicitly map the input features into a higher-dimensional space, making it possible to find a hyperplane that effectively separates the data [22], [25], [30].

SVM has been shown to work well in stock market prediction, proven by it being one of the best model when compared to other models [8], [31]. Ou and Wang [31] did a comparison of ten different data mining techniques to forecast the Hang Seng Index . The study compared multiple data mining techniques, where the result of the comparison shows that SVM is better than all the other models showing its superior predictive powers. Notably, SVM outshines LS-SVM in in-sample prediction, showcasing its advantages in accurately classifying training data which shows how good it is at understanding patterns. Qian [8] compared machine learning models (Logistic Regression, Multilayer Perceptron, and SVM), traditional models (ARIMA and GARCH), and a deep learning model called denoising auto-encoder (DAE) in the S&P 500 index. By using hit ratio and prediction as the evaluation method, they found that compared to other models SVM was the highest reaching 0.642. This paper also shows that SVM is compatible with the deep learning model. When combined, the model was able to achieve the highest hit rate reaching 0.672 showing the capabilities of SVM and its compatibility.

E. Random Forests (RF) in Stock Market

Random forests, as explained by Breiman [32], are an ensemble of decision trees, where each decision tree in the ensemble acts as a base classifier to determine the class label of an unlabeled instance through majority voting which can be seen illustrated by Fig. 3 [33], [34]. Since it's an ensemble model, Random forest's results are based on the majority of the results from every decision tree inside of it [35]. Additionally, from Fig. 3, since the model is used for a classification problem, the correct next step in the illustration after processing all decision trees are majority voting because averaging is used on regression problems.

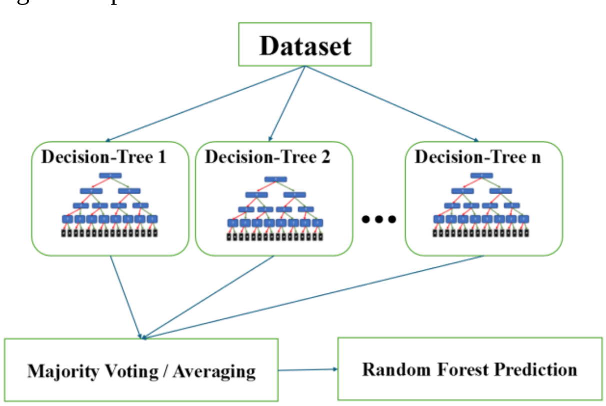

Figure 3 Illustration of Random Forests

The effectiveness of a random forest depends on the individual trees' strength and their correlation, meaning the rate of convergence from the model depends only on its strong features [36]. Random forests are also good at handling noisy data because they employ a random selection of features for node splitting, a feature that distinguishes them favorably from models like Adaboost [32], [36]. Breiman's research shows that random forests consistently perform well, especially when dealing with sparse data [32]. They're good at avoiding overfitting, reducing bias, and matching the accuracy of the Bayes rate on multiple datasets[32], [36].

Due to all of these capabilities, random forests application is also extended into the realm of stock market prediction. For example, Huang, Capretz, and Ho [17] was able to fully capitalize random forest's strength in stock market prediction by combining it with the understanding of fundamental analysis. Random forests as a model were able to beat both Feed-forward Neural Network (FNN) and Adaptive Neural Fuzzy Inference System (ANFIS) showcasing its strength under the right circumstances [17].

F. Indonesian Stock Exchange

TABLE I. COMPARISON BETWEEN DIFFERENT STOCK EXCHANGES'S STOCK MARKET INDEX

	IDX	Nasdaq	DJIA	Nikkei 225	DAX
Stock Count	911	3,418	30	225	40
Price	7,235.15	15,756.65	38,677.36	36,738.42	16,921.96
Total Sector	11	11	9	36	10

Before examining the characteristics of IDX, Table I. was created to show the difference between Indonesia's whole stock market compared to the big U.S. Index stock exchange Nasdaq and DJIA (Dow Jones Industrial Average), the Tokyo Stock Exchange's stock market index Nikkei 225, and DAX the index of the 40 major blue-chip companies in the Frankfurt Stock Exchange. Even with only Nasdaq being the stock exchange with a higher stock count than IDX, every single stock market index had a better price compared with IDX. This shows how the Indonesian stocks are smaller compared to other big stock exchanges, meaning that it is easier to be influenced.

There are several studies that try to explore the dynamics and intricacies of the Indonesian stock market which is called the Indonesian Stock Exchange (IDX). First, Herwany and Febrian [10] did an extensive research analyzing the volatility of the Islamic Stock in the IDX and found that it is heavily influenced by macroeconomic indicators during economic downturns. They found that these stocks were highly volatile and found that the risk-return relationship still needs to be researched further due to the current methods not being effective at minimizing the Islamic stocks risks. Another paper focused on examining the effects of fiscal and monetary policy in the Indonesian stock market [11]. They found that there's a positive stock price response in regard to monetary policy shocks, while there's a negative stock price response in regard to fiscal policy shocks. This indicates that fiscal policy is ineffective at influencing the economy which the paper suggests due to government's spending mainly used for paying debt rather than public finance investments.

Purnomo and Rider [12] had a crucial analysis on the impact of domestic and foreign shocks to the Indonesian stock market. Surprisingly, the paper found that there is no evidence that the Indonesian stock market is cointegrated with the U.S and Japanese stock market meaning low influence on the market. The paper also finds Indonesia's stock market to be influenced by regional markets meaning they are better stock market predictors compared to foreign stocks. Lastly, Gan Siew Lee and Djauhari [13] investigates 99 blue chip stocks in the IDX using network analysis approach and correlation networks analyzing the market's connectivity. By using a novel centrality measure, the overall centrality measure, which is the optimal linear combination of traditional centrality measures to summarize important information in the IDX, they were able to find high scoring stocks.

From these studies, it can be seen that there's evidence of volatility in the IDX [10], closely-related stocks [13], the government's policy has an impact on the stock market [11], and that it is unlikely to be influenced by foreign shocks [12]. These strings of potential reasons on the behavior of the Indonesian stock market shows promise on examining the connections between stocks in the IDX.

3. METHODOLOGY

In Methodology, there will be explanations in-depth regarding how the model is created through the framework in Fig. 4, while also illuminating on the novel method that we're proposing by using historical stock data in the IDX and the interconnected stocks as the main theory.

A. Data Collection

The dataset for this stock prediction came from Mirae Asset Sekuritas's software called HOTS30 that stores historical stock prices with features in IDX. We used several stocks that had the maximum total of 600 days in the stock market. The time period for these stocks starts at most from 8/9/2021 until 1/22/2024.

We stored several features alongside the basic features that HOTS30 provides at the beginning (Open, Low, High, Close, Volume) for further experimentation regarding how features impact the result of the model. After that, as shown in Fig. 5, we extracted these features: Open, Low, High, Close, Moving Average (Price and Volume), Bollinger Bands, Weighted Close, Volume, PDI, MDI, ATR, Roc, and RSI.

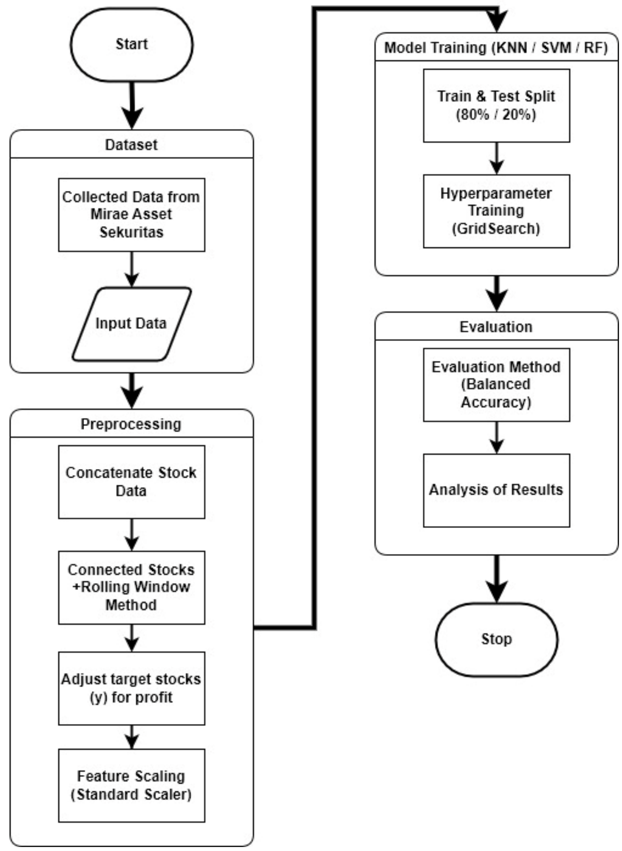

Figure 4 Main Research Framework

Figure 5 HOTS30 Dashboard

Table II. showcases an example of the five first indexes in the ANTM stock and its features. The data provided from Mirae Asset Sekuritas gave us the main four basic features which are Open, High, Low, and Close. Open and Close both refer to the opening and closing price of the stock during the trading day, while High and Low both refer to the highest and lowest price of the stock during the trading day. There are also other features that we extracted as mentioned before, however this is meant to be a simplified example of the input data that we got from HOTS30.

TABLE II. A SIMPLIFIED RESULT OF A STOCK EXTRACTION

Date	Open	High	Low	Close	Volume
1/22/2024	1,645	1,650	1,645	1,645	42
1/19/2024	1,670	1,705	1,640	1,645	1,382,762
1/18/2024	1,605	1,635	1,600	1,620	427,918
1/17/2024	1,620	1,620	1,600	1,605	223,405
1/16/2024	1,610	1,625	1,600	1,605	286,909

B. Pre-Processing

According to Fig. 4, pre-processing will be extended into three different techniques, however before that we will need to input the stocks data first as x. For every single stock that is used as an input and the features that are also used as an input, they will be put into a 1-D array together. As an example, the baseline model for this study will consist of four stocks: ANTM, ERAA, KLBF, and MIKA in which the baseline model also only uses a single feature which is 'Close'. Therefore, x consists of the closes from ANTM, ERAA, KLBF, and MIKA. After the pre-processed x has correctly been integrated, we proceed to the three stages of pre-processing.

1. Connected Stocks + Rolling Window Method

To grasp the inter-connected stocks in the IDX, we will be using a novel method combining "Connected Stocks Method" and "Rolling Window Method". By ensuring that the input (x) consists of the historical timeframe of each stock by using the rolling window while also consisting of multiple stocks at the same time, x will leverage the historical data and capture the temporal and inter-connected stock dynamics.

The "Connected Stocks Method" is inherently a simple method that means x will have sequential stocks between one-another in the input. This can clearly be seen in Fig. 6, where in both forms the rolling window subsample has two connected stocks which are stock A and Stock B. This means for example that if the input consists of two stocks A and B, while also consisting of two features 'Open' and 'Close' the ordered array of x would be the same as the array form in Fig. 6. This method's main usage is to capture the inter-connected stock dynamics to see if the stocks influence each other.

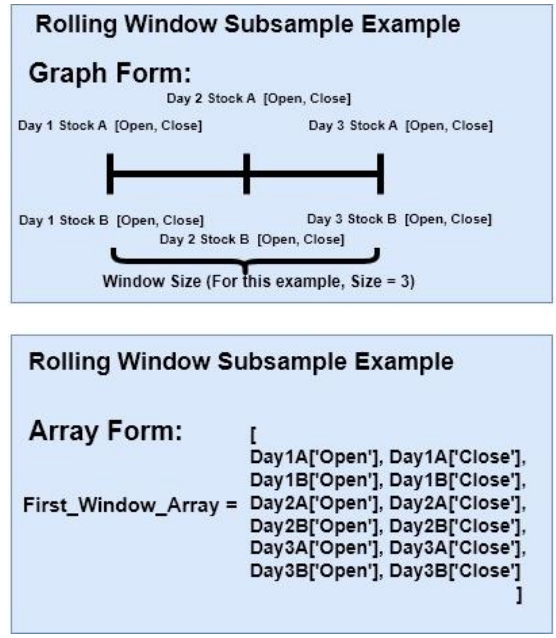

Figure 6 Rolling Window Subsample

To capture the temporal dynamics of the Indonesian stock market, we decided to use the rolling window method. As seen in Fig. 7, this method captures a subset of the timeframe from the whole duration which will all be combined to capture the changes in the movement of the stock market, where the size of the window is how many days you want to capture in a window (portrayed in the graph form in Fig. 6). This means we need to store an additional variable that captures the subset turning the input into a 2-D array. Further examination of this can be seen in Fig. 7, where the rolling window method is already combined with the connected stocks method and formed a 2-D array that adjusts to both the data size and window size.

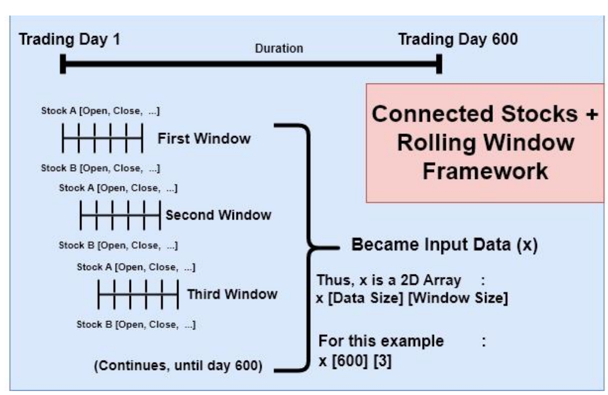

Figure 7 Connected Stocks + Rolling Window Framework

2. Adjust target stocks (y) for profit

Since the main benefit of stock prediction is the profitability obtained from it, there needs to be a target that can achieve that benefit. For this research, we decided to predict a variable y, where it performs a binary classification whether the next day's high is higher than 1.5% of today's close or not. Based on (1), we can calculate t which is the threshold. Once we have the value of t, we can then properly do a binary classification which is shown through (2).

$$t = \frac{next\ high - current\ close}{current\ close} * 100 \quad (1)$$

$$y = \begin{cases} 1, if\ t\ \geq 1.5 \\ 0, if\ t\ < 1.5 \end{cases} \quad (2)$$

The reason that we wanted t to be higher than 1.5% is because we wanted to account for the buy and sell fee provided by Mirae Asset Sekuritas [38]. The buy fee provided by them is 0.15%, while their sell fee is 0.25%. Both fees combined resulted in 0.4%, which means by subtraction we will get a profit of 1.1%. We would achieve our goal is to try to earn a profit of a minimum of 1% every day.

3. Feature Scaling

Feature scaling is crucial to employ standardization and ensure all features contribute equally to the model. For this model we decided on using Standard Scaler since it preserves the shape of the original distribution, which is better for KNN and SVM, while RF is relatively unaffected by scaling since it's a tree-based algorithm. So, we standardized all our input of x based on (3) where μ refers to the mean of x and σ refers to the standard deviation of x,

$$x_{standardized} = \frac{x - \mu}{\sigma} \quad (3)$$

C. Train Test Split

We performed a train test split, with the split being 80/20, meaning 80% of it is training data, meanwhile 20% of it is test data. Since we used train_test_split as a function from sklearn, we also implemented a random_state to ensure that our results are replicable. Each stock has a data size of 600, meaning that we separate them into 480 training data and 120 test data. With this in mind, since we will mostly experiment using four to five stocks to predict, which in total is around 2400 - 3000 data size, which is a good enough sample size for common machine learning models to learn the data.

D. Hyperparameter Tuning and Model Training (KNN, SVM, RF)

From previous chapters, we learned that an optimal hyperparameter can boost the performance of the model highly [20], [39]. Because of this, we're doing hyperparameter tuning with grid search, a simple technique that evaluates a model's performance for each combination of hyperparameters in a grid. Since there are three machine learning models that we use, that means we have three different parameters for each grid which can be seen in Fig. 8. After proper hyperparameter tuning, each model can be trained according to hyperparameters from hyperparameter tuning.

KNN

```
param_grid = {
    'n_neighbors': [3, 5, 7, 9,
                    11, 13, 15,
                    17, 19, 21],
    'weights': ['uniform', 'distance']
}
```

SVM

```
param_grid = {
    'C': [0.1, 1, 10, 100],
    'kernel': ['linear', 'poly', 'rbf'],
    'degree': [2, 3, 4]
}
```

RF

```
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'bootstrap': [True, False]
}
```

Figure 8 Parameters for each model

E. Model Evaluation

After the model is successfully trained, we can now evaluate the model's capabilities. For this model evaluation metric, we will be using balanced_accuracy_score from sklearn in which the mathematical model can be seen in (6). In that equation, sensitivity means the percentage of positive cases the model is able to detect, while specificity means the percentage of negative cases the model is able to detect.

After training the model, we assess its performance using a specific evaluation metric: balanced accuracy score from sklearn. Balanced accuracy considers two variables which are Sensitivity and Specificity. Sensitivity in (4) and Specificity in (5) measure the opposite of each other which balances it out for balanced accuracy. Sensitivity refers to a true positive rate, calculating the proportion of true positive identified by the classifier, while specificity refers to a true negative rate, calculating the proportion of true negative identified by the classifier.

$$\text{Sensitivity} = \frac{TP\ (True\ Positive)}{TP\ (True\ Positive) + FN\ (False\ Negative)} \quad (4)$$

$$\text{Specitivity} = \frac{TN\ (True\ Negative)}{TN\ (True\ Negative) + FP\ (False\ Positive)} \quad (5)$$

$$\text{Balanced Accuracy} = \frac{Sensitivity + Specificity}{2} \quad (6)$$

When training the model, we found that generally most of the data and results tend to be skewed showing a class imbalance problem. When a model has shown an imbalance class problem, the evaluation method cannot purely be by accuracy since the results would be skewed [40]. Because of that, to ensure that the data is accurately representing the model's performance we decided to use Balanced Accuracy (6).

In regard to evaluating the model itself, we found that 55% is a good benchmark being a good result that shows stock connectivity, meanwhile anything below that especially in the range of 50% or below shows that the stocks aren't impacted by other stocks.

4. EXPERIMENT AND TASKS

A. Experiment Setup

The experiments were conducted on a high-performance workstation featuring an Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz, 8GB of RAM, and an NVIDIA GeForce GTX 1650 GPU. This hardware configuration provided the computational power required for training and evaluating machine learning models efficiently. The machine learning models were implemented using Python programming language (version 3.9.7). We utilized popular libraries such as NumPy, Pandas, and Scikit-learn for data preprocessing, feature engineering, and model evaluation. All experiments were conducted using Python within a Jupyter Notebook environment.

B. Experimented Stocks

In Table III we provided the list of stocks that we're going to use for the experimentation. It can be seen that most of the stocks are from either the basic materials or financials sector. This is because the list of stocks that we're going to be using for this experiment is ordered. We also calculated their last known total sale, which is in Q3 2023, where most of these stocks have a similar range.

C. Description of Tasks

To obtain a comprehensive understanding regarding the dynamics of the interconnected stocks in IDX, we performed multiple experiments that focuses on different aspects of the training, starting from pre-processing where we test a variety of inputs, experiment using different features and its data size, experiment using model and stocks that has a balanced classification target data, consider using different window sizes, and consider sector-specific stock interconnection. By properly investigating IDX's stocks we will have a better understanding of how our method properly predicts stock prices.

1. Baseline Model

Since the training data that we use aren't similar to other relevant papers in this field, we created a baseline model that will be the base of comparison for every other experiment. All other models will share identical parameters with this baseline (window size, stock selection, features, etc.), except for one variable, which will be the target for experimental comparison purposes.

TABLE III. LISTS OF STOCKS EXPERIMENTED

Stock Name	**Company**	**Sector**	**Total Sales Q3 2024 (in Rupiah)**
ANTM	Aneka Tambang Tbk	Basic Materials	30.898T
ERAA	Erajaya Swasembada Tbk	Consumer Cyclicals	42.816T
KLBF	Kalbe Farma Tbk	Healthcare	22.561T
MIKA	Mitra Keluarga Karyasehat Tbk	Healthcare	3.156T
DUTI	Duta Pertiwi Tbk.	Properties and Real Estate	2.903T
DSSA	Dian Swastatika Sentosa Tbk	Energy	5.782T
ARTO	Bank Jago Tbk	Financials	0.8T
BBRI	Bank Rakyat Indonesia (Persero)	Financials	31.603T
BBTN	Bank Tabungan Negara (Persero)	Financials	1.426T
BFIN	BFI Finance Indonesia Tbk.	Financials	2.056T
BNGA	Bank CIMB Niaga Tbk.	Financials	3.896T
BRPT	Barito Pacific Tbk.	Basic Materials	18.617T
CMNT	Cemindo Gemilang Tbk.	Basic Materials	0.434T
MDKA	Merdeka Copper Gold Tbk.	Basic Materials	6.689T
TPIA	Chandra Asri Pacific Tbk.	Basic Materials	12.783T

The baseline model's input stocks are ANTM, ERAA, KLBF, and MIKA, using only the 'Close' feature as input. This means the input (x) for the baseline model includes the closing prices of these four stocks. For the novel connected stocks + rolling window method, we use the window size of four. The evaluation method for the baseline model and every other experiment will be the one discussed in Section 3 which is a balanced accuracy metric.

2. Testing using Different Features

Commonly, additional features as inputs help machine learning models improve their performance therefore there needs to be an experiment to show if it's true for this case. We created two additional models using two different features, which can be seen in Table IV. Models with the Common features will have five input features and are supposed to represent standard stock prediction models that only use the basic features. Advanced features include technical indicators into the mix which should help increase the model's performance. For advanced features, in total they have 29 input features.

TABLE IV. COMPARISON USING DIFFERENT FEATURES

Name	Features Used
Baseline Features	Close
Common Features	Open, High, Low, Close, Volume
Advanced Features	Open, Low, High, Close, Volume, Moving Average (Price and Volume), Bollinger Bands, Weighted Close, PDI, MDI, ATR, Roc, and RSI

3. Testing using a Balanced Threshold Model

In prior testing, particularly with SVM, we observed a tendency for one-sided predictions where every stock was forecasted to be either higher or lower than the threshold. These results suggest an imbalanced data classification. While we can't adjust our threshold model in our target stock (y) to guarantee a minimum 1% profit, we can explore what would happen if each stock had a more balanced target. This approach could mitigate the imbalance issue in the data.

As you can see in Table V, we have adjusted the threshold for every single stock in the baseline model to be balanced. This means, every single stock has a perfectly balanced data classification by adjusting the threshold to fit that criteria. From the table, you can see that every threshold when balanced is still positive, showing that every single stock's growth has in majority been positive.

TABLE V. LISTS OF BALANCED THRESHOLD FOR EACH STOCK

Name	Threshold(t)
Original Threshold	1.5
ANTM's Balanced Threshold	1.2
ERAA's Balanced Threshold	1.265
KLBF's Balanced Threshold	0.93
MIKA's Balanced Threshold	1.535

4. Testing using Different Window Size

Window size is a variable used in the rolling window method to assess the performance of a model in a particle timeframe. By exploring the effect of different window sizes, we can see whether it affects the model's performance. For this experiment, we decided to test these window sizes: 5, 10, 15, 20, and 60 days. Since we're trying to compare different window sizes, this means that the stock that we use as comparison must be the same. Instead of using one of the stocks as comparison, we use the average result of each stock as a comparison. This ensures the model's generalizability and that the result won't be skewed by a specific stock's permeability to a window size.

5. Testing using Different Stock Size

Normally, increasing the amount of input data to the model will increase the model's performance. However, in the case of our model, since the input data also considers interconnectivity between stocks there's a chance that adding more stocks reduces the probability of the stocks to influence each other, thus lowering the model's performance. Because of this, we're going to experiment with different stock sizes as inputs which the list is shown at Table VI.

TABLE VI. LISTS OF DIFFERENT STOCK SIZE EXPERIMENTS

Stock Lists	Number of Stocks
[ANTM, ERAA, KLBF]	3
[ANTM, ERAA, KLBF, MIKA]	4
[ANTM, ERAA, KLBF, MIKA, DUTI]	5
[ANTM, ERAA, KLBF, MIKA, DUTI, DSSA]	6

Like our experiment with different window sizes, the target that we're trying to compare is between stock sizes, meaning we must use the same stock as a comparison. Instead of using one of the stocks as comparison, we use the average result of each stock as a comparison. The stocks that we use as the predicted stocks will be ANTM, ERAA, and KLBF since we need to experiment the results on a model that has three as its stock size. The model will follow the structure of the baseline model, other than the input it uses.

6. Testing using a Specific Sector (Financial Sector)

In IDX, stocks are grouped into various sectors such as Healthcare, Financials, and others. Since our baseline model is composed of stocks from different sectors, we want to compare that to models that are purely trained using a specific sector. For this experiment, we will be using the Financial Sector as the specific sector using ARTO, BBRI, BBTN, BFIN, and BNGA as the input and target stocks.

7. Testing using a Combination of Specific Sectors(Basic Materials Sector)

Similar to the prior experiment, this one will analyze a group of stocks from the same sector, the Basic Materials Sector. We'll use BRPT, CMNT, MDKA, and TPIA as input and target stocks. Unlike before, we'll also merge the Basic Materials and Financials sectors as input stocks. The aim is to assess if combining sectors as input data improves the model's performance in predicting Basic Materials stocks. This means BRPT, CMNT, MDKA, TPIA, ARTO, BBRI, BBTN, BFIN, and BNGA will serve as input stocks.

5. RESULT AND DISCUSSION

In this section, we will be going over the results from the experimentation explained in the previous section. We will add another model which is the average of all the other Machine Learning models (KNN, SVM, RF) to better judge how each stock's interconnectivity truly is. The results from the table will follow our evaluation metric, balanced accuracy score. According to our evaluation method, if a stock is classified as having high interconnectivity, then it should at minimum be > 55%. If any stock has an evaluation score that is ⩽ 50%, it means that they're not connected at all.

A. Testing Results using Baseline Model

It can be seen from Table VII, that the baseline model performed well for ANTM and ERAA, reaching an average accuracy of 56% and 59% respectively. However, for KLBF and MIKA this isn't the case, only reaching 50% and 52% respectively. While MIKA performed terribly using KNN, it performed well using the others showing that the stock might be interconnected and KNN is just an outlier KLBF however is definitely not connected having consistently terrible performance on every model. Comparatively from each machine learning model, RF showcased the best performance against each model where only KLBF was inaccurately predicted.

TABLE VII. ACCURACY RESULT USING BASELINE MODEL

Machine Learning Model	Stock Name			
	ANTM	ERAA	KLBF	MIKA
KNN	**0.6054**	0.5979	0.5099	0.4742
SVM	0.5439	**0.5626**	0.5	0.5508
RF	0.5403	**0.614**	0.516	0.5646
Average	0.5632	**0.5915**	0.5086	0.5298

B. Testing Results using Common Features and Advanced Features

From the comparison Table VIII, it is found that additional features don't give consistent improvement to most stock's performance except MIKA. ANTM on average performed best when using advanced features, but ERAA on average performed best when using common features showing that it's an indecisive proof. KLBF has shown itself as a stock that is not interconnected, since it has consistently performed around 50% accuracy. MIKA however performed well compared to the baseline model, showing that it might be an interconnected stock that was trained badly by KNN on the baseline model. Regarding the algorithm itself, all three models performed well except at KLBF where RF performed the best with a slight edge.

TABLE VIII. ACCURACY COMPARISON USING DIFFERENT FEATURES

Features	Machine Learning Model	Stock Name			
		ANTM	ERAA	KLBF	MIKA
Common Features	**KNN**	0.5615	**0.6394**	0.4996	0.5492
	SVM	0.5425	**0.6121**	0.4521	0.5319
	RF	0.5294	**0.614**	0.5028	0.5457
	Average	0.5445	**0.6218**	0.4849	0.5423
Advanced Features	**KNN**	**0.5952**	0.5809	0.5221	0.5473
	SVM	0.5709	**0.5973**	0.4878	0.5696
	RF	**0.6424**	0.5821	0.4906	0.5909
	Average	**0.6029**	0.5868	0.5002	0.5693

C. Testing Results regarding Imbalanced Data

From Table IX, it is found that the model that uses a balanced threshold significantly improved the performance of each stock that wasn't interconnected yet. While ANTM and ERAA still performed well similarly as a stock, KLBF and MIKA found massive improvement as a target stock. KLBF performed the best out of everyone achieving an average accuracy of 58%, massively improving and showing interconnectedness when compared to previous results. MIKA also performed well, achieving an average of 54% accuracy, where its average is badly influenced because of sudden KNN drop-off which is the main reason why its average is under 55%. From the machine learning perspective, RF as usual performed the best out of all the machine learning models. SVM performed consistently well achieving results mostly above the threshold of 55% except at KLBF, and KNN seems to perform well but with random failed results at capturing interconnectivity in stocks such as MIKA at Table VIII.

TABLE IX. ACCURACY RESULT USING BASELINE MODEL WITH BALANCED THRESHOLD

Machine Learning Model	Stock Name			
	ANTM	ERAA	KLBF	MIKA
KNN	**0.5797**	0.5691	0.5781	0.4856
SVM	0.5517	**0.5982**	0.5204	0.5567
RF	0.5258	0.5703	**0.6562**	0.5944
Average	0.5524	0.5792	**0.5849**	0.5456

D. Testing Results with Different Window Size

Analyzing the stock comparison in Fig. 9, ANTM didn't perform well when the window size is between 10-20. ERAA as usual performed well under any experiments achieving an average higher than 55% on every single window size. KLBF performed consistently awful, except when window size is 10 due to KNN randomly performing well which now seems to be consistent as a trait for KNN. MIKA had a good performance, in which the model started to improve when window size is 15 and improved further. From the machine learning perspective, RF performed the best as usual, however it is strange to see SVM performed the worst at stocks like ANTM. KNN as usual has a good performance with slightly random results.

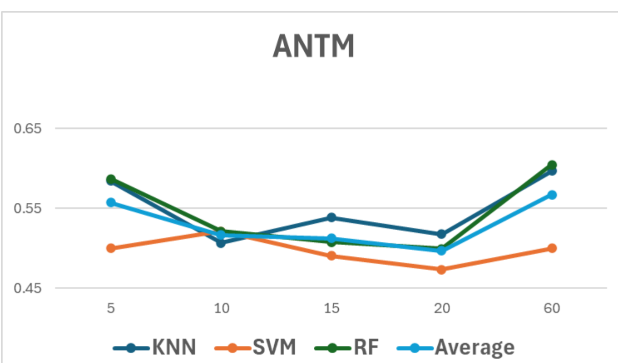

Figure 9(a) ANTM Comparison on Results with Different Window Size

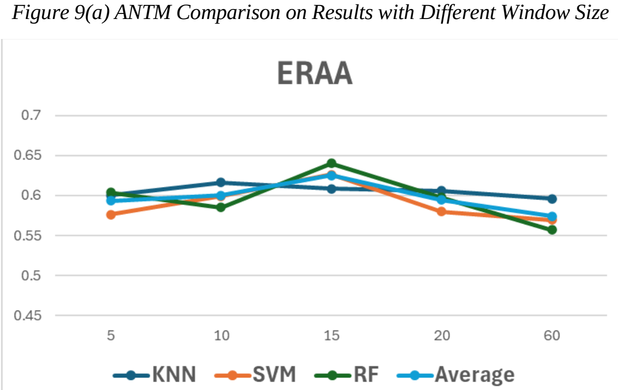

Figure 9(b) ERAA Comparison on Results with Different Window Size

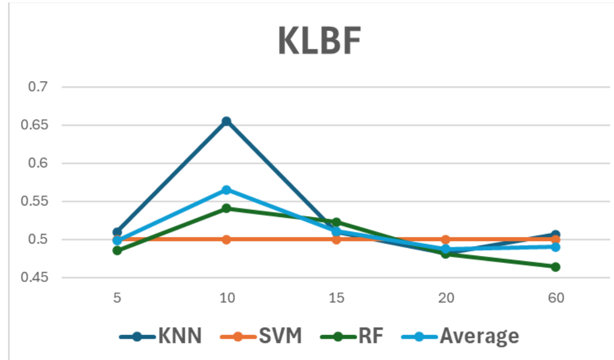

Figure 9(c) KLBF Comparison on Results with Different Window Size

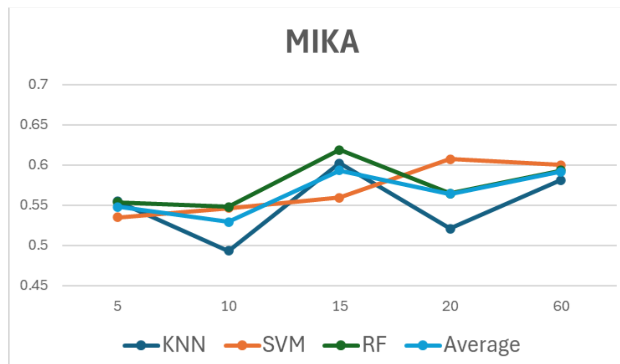

Figure 9(d) MIKA Comparison on Results with Different Window Size

Analyzing Table X, it can be seen that previous analysis is reflected on the table. With ANTM performing worse and MIKA performing better than normal it evens out when averaged. Other than these two, ERAA and KLBF remained consistent, meaning that when the stocks are averaged the score is relatively the same. There doesn't seem to be a pattern found on Table X, where the accuracy fluctuates with increased window size, suggesting that different window sizes don't affect the performance of the model. This is even further shown through different machine learning models, where they remained stagnant with only tiny differences.

TABLE X. ACCURACY RESULT USING BASELINE MODEL WITH DIFFERENT WINDOW SIZE FOR AVERAGE STOCKS

Machine Learning Model	Window Size				
	5	10	15	20	60
KNN	0.5622	**0.5677**	0.5646	0.5315	0.57
SVM	0.5277	0.5417	**0.544**	0.54	0.5423
RF	0.5572	0.5488	**0.5723**	0.5356	0.5547
Average	0.549	0.5527	**0.5603**	0.5357	0.5556

E. Testing Results with Different Stock Size

Analyzing the stocks comparison in Fig. 10, ANTM performed consistently well where on average it improves and almost reached 60% when using six stocks as input data. ERAA performed consistently well, this time on average reaching around 60% accuracy. KLBF on average performed below the criteria as usual, however has shown improvement with additional stock size.

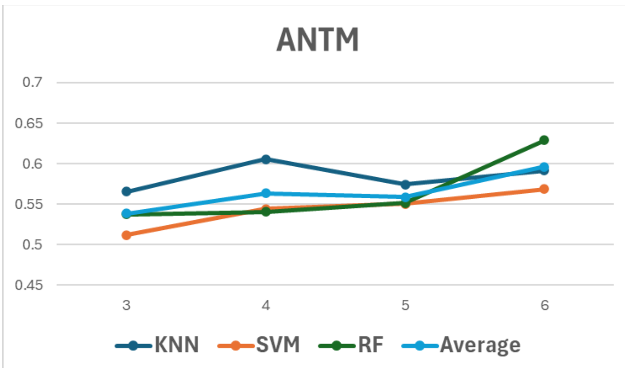

Figure 10(a) ANTM Comparison on Results with Different Stock Size

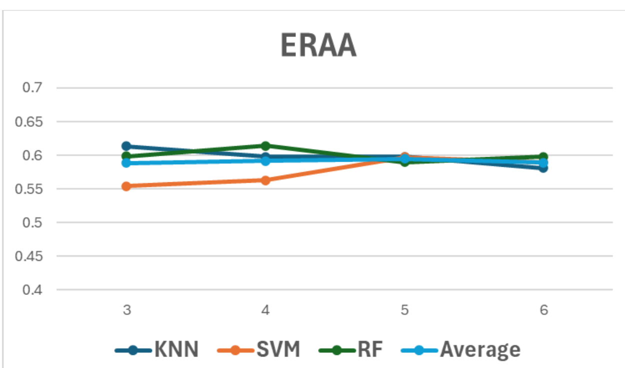

Figure 10(b) ERAA Comparison on Results with Different Stock Size

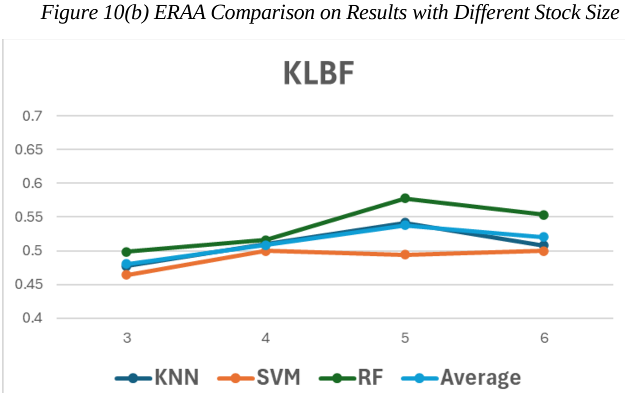

Figure 10(c) KLBF Comparison on Results with Different Stock Size

Analyzing Table XI, it can be seen that the previous stock comparison is reflected in this table showcasing growth. On average, the performance of the model increases as the stock size as input increases, giving a new insight on improving the model. Looking at the machine learning models itself, RF and KNN performed well, while SVM didn't perform as well when the stock size was small meaning it adapts better to bigger datasets.

TABLE XI. ACCURACY RESULT USING BASELINE MODEL WITH DIFFERENT STOCK SIZE FOR AVERAGE STOCKS

Machine Learning Model	Stock Size			
	3	4	5	6
KNN	0.5521	**0.5711**	**0.5711**	0.56
SVM	0.51	0.5355	**0.5471**	0.5526
RF	0.5447	0.5568	0.5729	**0.5934**
Average	0.5356	0.5545	0.5637	**0.5687**

F. Testing Result in a Specific Sector (Financial Sector)

Looking at Table XII, we can see each result of stocks in the financial sector. All five models performed terribly and not a single one hit the benchmark of 55% accuracy on average. Other than that, machine learning models also performed consistently terribly except for RF achieving two models above the benchmark. This table showcases that using specific sectors as input might ruin the interconnectivity of the stocks as they're volatile with each other and that RF is the best machine learning model so far comparatively between the three.

TABLE XII. ACCURACY RESULT USING FINANCIAL SECTOR MODEL

Machine Learning Model	Stock Name				
	ARTO	BBRI	BBTN	BFIN	BNGA
KNN	0.527	0.5135	0.4921	**0.5537**	0.5146
SVM	0.4908	0.5	0.5154	**0.5184**	0.5
RF	**0.5889**	0.4939	0.5205	0.5337	0.563
Average	**0.5355**	0.5024	0.5093	0.5353	0.5259

G. Testing Result in a Specific Sector and Combined Sector Models (Basic Material Sector)

Looking at Table XIII, it can be seen that on average the model mostly improved with additional stock inputs similar to Table XI. However, similar to Table XII as well, model trained using specific sectors have shown terrible performance with the exception for BRPT. When looking at the machine learning model, SVM and RF performed better with additional inputs, while KNN performed similarly through both comparisons. The table showcases that while there might be stocks like BRPT that are interconnected when trained using specific sectors achieving accuracy as high as 67%, generally the model will perform worse and not achieve the benchmark.

TABLE XIII. ACCURACY COMPARISON BETWEEN SPECIFIC SECTORS AND JOINT SECTORS

Stock Input	Machine Learning Model	Stock Name			
		BRPT	CMNT	MDKA	TPIA
Basic Materials Sector Input	KNN	**0.6293**	0.5458	0.4893	0.5124
	SVM	**0.6693**	0.4939	0.464	0.5377
	RF	**0.603**	0.5248	0.5429	0.5699
	Average	**0.6339**	0.5215	0.4988	0.54
Combined Sector Input (Basic Material + Financial)	KNN	**0.6556**	0.5489	0.4878	0.493
	SVM	**0.6793**	0.5609	0.5309	0.4933
	RF	**0.6889**	0.4945	0.464	0.5695
	Average	**0.6746**	0.5348	0.4943	0.5186

H. Summary of Testing Results

Implementing the novel method "Connected Stocks + Rolling Window Method" on several machine learning models with several experimentations has resulted in several interesting discoveries. Regarding the baseline stocks, ANTM and MIKA have shown improvement with additional help through features, balanced dataset, etc. ANTM and ERAA performed well, with ERAA consistently performing well in any experiment. KLBF performed the worst where it lacks change, showing its not interconnected, however with proper balancing data it has been shown that this stock can perform well. Here's several key points to summarize the test results:

- Implementing additional features didn't impact the model's performance in a positive way, only causing fluctuations of accuracy for each model and stock (except for MIKA).
- The balanced threshold model proved to significantly improve each stock's performance, proving the importance of having a balanced class of data.
- The differences of window size during experiment didn't impact the model's accuracy in any meaningful way.
- Increased stock size was able to improve the model's performance on average.
- Using specific sectors, instead of a variety of stocks, didn't improve the model, even shown to be worse where almost all of them didn't hit the minimum benchmark of 55%.
- Combining input data from sectors might help improve the model for stocks that are interconnected like BGPT, however it wasn't able to help stocks that weren't.

6. CONCLUSION

To deal with the volatility of the IDX, we examined the stocks interconnectedness using our novel method "Connected Stocks + Rolling Window Method" to predict the next day's high of stocks in IDX using several machine learning models (KNN, SVM, RF). We found that having balanced classification data improved the model's performance significantly. Using a higher amount of input stocks also improved the accuracy, especially interconnected stocks. The machine learning model that performed the best was found to be Random Forest. In the end, we successfully have shown effects of interconnectedness in the Indonesian Stock Exchange and were able to predict the next day's high using several stocks including ANTM, ERAA, and BGPT having accuracies higher than 55% most of the time through the use of the stock's interconnectedness.

REFERENCES

[1] S. Fischer and R. C. Merton, "Macroeconomics and finance: The role of the stock market," National Bureau of Economic Research, 1984. Accessed: Feb. 06, 2024. [Online]. Available: https://www.nber.org/papers/w1291

[2] D. P. Gandhmal and K. Kumar, "Systematic analysis and review of stock market prediction techniques," *Computer Science Review*, vol. 34, p. 100190, 2019.

[3] G. Sonkavde, D. S. Dharrao, A. M. Bongale, S. T. Deokate, D. Doreswamy, and S. K. Bhat, "Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications," *IJFS*, vol. 11, no. 3, p. 94, Jul. 2023, doi: 10.3390/ijfs11030094.

[4] M. Usmani, S. H. Adil, K. Raza, and S. S. A. Ali, "Stock market prediction using machine learning techniques," in *2016 3rd international conference on computer and information sciences (ICCOINS)*, IEEE, 2016, pp. 322–327.

[5] P. P. S. Kohli, S. Zargar, S. Arora, and P. Gupta, "Stock Prediction Using Machine Learning Algorithms," in *Applications of Artificial Intelligence Techniques in Engineering*, vol. 698, H. Malik, S. Srivastava, Y. R. Sood, and A. Ahmad, Eds., in Advances in Intelligent Systems and Computing, vol. 698. , Singapore: Springer Singapore, 2019, pp. 405–414. doi: 10.1007/978-981-13-1819-1_38.

[6] C. K.-S. Leung, R. K. MacKinnon, and Y. Wang, "A machine learning approach for stock price prediction," in *Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14*, Porto, Portugal: ACM Press, 2014, pp. 274–277. doi: 10.1145/2628194.2628211.

[7] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," *Expert Systems with Applications*, vol. 184, p. 115537, Dec. 2021, doi: 10.1016/j.eswa.2021.115537.

[8] X.-Y. Qian, "Financial Series Prediction: Comparison Between Precision of Time Series Models and Machine Learning Methods." arXiv, Dec. 07, 2018. Accessed: Feb. 05, 2024. [Online]. Available: http://arxiv.org/abs/1706.00948

[9] I. Bhattacharjee and P. Bhattacharja, "Stock price prediction: a comparative study between traditional statistical approach and machine learning approach," in *2019 4th international conference on electrical information and communication technology (EICT)*, IEEE, 2019, pp. 1–6.

[10] A. Herwany and E. Febrian, "Portfolio volatility of Islamic and conventional stock: The case of Indonesia stock market," *Journal of Governance and Regulation*, vol. 2, no. 4, 2013.

[11] M. Jusoh, M. A. S. Zaidi, and R. D. Handoyo, "Impact of monetary policy and fiscal policy on Indonesian stock market," *Expert Journal of Economics*, vol. 3, no. 2, pp. 113–126, 2015.

[12] B. Purnomo and M. W. Rider, "Domestic and foreign shocks and the indonesian stock market: time series evidence," *Georgia State University*, pp. 1–28, 2012.

[13] Gan Siew Lee and M. A. Djauhari, "Network topology of Indonesian stock market," in *2012 International Conference on Cloud Computing and Social Networking (ICCCSN)*, Bandung, West Java: IEEE, Apr. 2012, pp. 1–4. doi: 10.1109/ICCCSN.2012.6215721.

[14] R. F. Engle, E. Ghysels, and B. Sohn, "Stock Market Volatility and Macroeconomic Fundamentals," *Review of Economics and Statistics*, vol. 95, no. 3, pp. 776–797, Jul. 2013, doi: 10.1162/REST_a_00300.

[15] M. Galeotti and F. Schiantarelli, "Stock market volatility and investment: Do only fundamentals matter?," *Economica*, pp. 147–165, 1994.

[16] Y. Song, "Stock trend prediction: Based on machine learning methods," PhD Thesis, UCLA, 2018. Accessed: Feb. 06, 2024. [Online]. Available: https://escholarship.org/uc/item/0cp1x8th

[17] Y. Huang, L. F. Capretz, and D. Ho, "Machine Learning for Stock Prediction Based on Fundamental Analysis," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, Orlando, FL, USA: IEEE, Dec. 2021, pp. 01–10. doi: 10.1109/SSCI50451.2021.9660134.

[18] M. V. Subha and S. T. Nambi, "Classification of Stock Index movement using k-Nearest Neighbours (k-NN) algorithm," *WSEAS Transactions on Information Science & Applications*, vol. 9, no. 9, pp. 261–270, 2012.

[19] A. Mucherino, P. J. Papajorgji, and P. M. Pardalos, "k-Nearest Neighbor Classification," in *Data Mining in Agriculture*, vol. 34, in Springer Optimization and Its Applications, vol. 34. , New York, NY: Springer New York, 2009, pp. 83–106. doi: 10.1007/978-0-387-88615-2_4.

[20] G. Batista and D. F. Silva, "How k-nearest neighbor parameters affect its performance," in *Argentine symposium on artificial intelligence*, Citeseer, 2009, pp. 1–12. Accessed: Feb. 05, 2024. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=b83da505b60e32469152b986cbd6199842403b11

[21] S. B. Imandoust and M. Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background," vol. 3, no. 5, 2013.

[22] V. Kecman, "Support Vector Machines – An Introduction," in *Support Vector Machines: Theory and Applications*, vol. 177, L. Wang, Ed., in Studies in Fuzziness and Soft Computing, vol. 177. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1–47. doi: 10.1007/10984697_1.

[23] K. Purwandari, R. B. Perdana, J. W. Sigalingging, R. Rahutomo, and B. Pardamean, "Automatic Smart Crawling on Twitter for Weather Information in Indonesia," *Procedia Computer Science*, vol. 227, pp. 795–804, 2023.

[24] R. E. Caraka, R. C. Chen, T. Toharudin, M. Tahmid, B. Pardamean, and R. M. Putra, "Evaluation performance of SVR genetic algorithm and hybrid PSO in rainfall forecasting," *ICIC Express Lett Part B Appl*, vol. 11, no. 7, pp. 631–639, 2020.

[25] J. M. Moguerza and A. Muñoz, "Support Vector Machines with Applications," *Statist. Sci.*, vol. 21, no. 3, Aug. 2006, doi: 10.1214/088342306000000493.

[26] K. Purwandari, J. W. Sigalingging, T. W. Cenggoro, and B. Pardamean, "Multi-class weather forecasting from twitter using machine learning aprroaches," *Procedia Computer Science*, vol. 179, pp. 47–54, 2021.

[27] R. E. Caraka, S. A. Bakar, B. Pardamean, and A. Budiarto, "Hybrid support vector regression in electric load during national holiday season," in *2017 International Conference on Innovative and Creative Information Technology (ICITech)*, IEEE, 2017, pp. 1–6.

[28] I. N. Setiawan, R. Kurniawan, B. Yuniarto, R. E. Caraka, and B. Pardamean, "Parameter optimization of support vector regression using Harris hawks optimization," *Procedia Computer Science*, vol. 179, pp. 17–24, 2021.

[29] K. Purwandari, T. W. Cenggoro, J. W. C. Sigalingging, and B. Pardamean, "Twitter-based classification for integrated source data of weather observations," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, p. 271, 2023.

[30] M. Isnan, G. N. Elwirehardja, and B. Pardamean, "Sentiment Analysis for TikTok Review Using VADER Sentiment and SVM Model," *Procedia Computer Science*, vol. 227, pp. 168–175, 2023.

[31] P. Ou and H. Wang, "Prediction of Stock Market Index Movement by Ten Data Mining Techniques," *MAS*, vol. 3, no. 12, p. p28, Nov. 2009, doi: 10.5539/mas.v3n12p28.

[32] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[33] K. Fawagreh, M. M. Gaber, and E. Elyan, "Random forests: from early developments to recent advancements," *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 602–609, Dec. 2014, doi: 10.1080/21642583.2014.956265.

[34] B. Pardamean, H. Soeparno, A. Budiarto, B. Mahesworo, and J. Baurley, "Quantified self-using consumer wearable device: predicting physical and mental health," *Healthcare Informatics Research*, vol. 26, no. 2, pp. 83-92, 2020.

[35] T.W. Cenggoro, B. Mahesworo, A. Budiarto, J. Baurley, T. Suparyanto, and B. Pardamean, "Features importance in classification models for colorectal cancer cases phenotype in Indonesia," *Procedia Computer Science*, vol. 157, pp. 313–320, 2019.

[36] G. Biau, "Analysis of a random forests model," *The Journal of Machine Learning Research*, vol. 13, pp. 1063–1095, 2012.

[37] R. Genuer, J.-M. Poggi, and C. Tuleau, "Random Forests: some methodological insights." arXiv, Nov. 21, 2008. Accessed: Feb. 05, 2024. [Online]. Available: http://arxiv.org/abs/0811.3619

[38] "Customer Services | Mirae Asset Sekuritas." Accessed: Feb. 08, 2024. [Online]. Available: https://m.miraeasset.co.id/mobile/custSvc/faqDtlR.do

[39] R. Behera, S. Das, S. Rath, S. Misra, and R. Damasevicius, "Comparative Study of Real Time Machine Learning Models for Stock Prediction through Streaming Data," *jucs*, vol. 26, no. 9, pp. 1128–1147, Sep. 2020, doi: 10.3897/jucs.2020.059.

[40] N. Dominic, Daniel, T.W. Cenggoro, A. Budiarto, and B. Pardamean, "Transfer learning using inception-resnet-v2 model to the augmented neuroimages data for autism spectrum disorder classification," *Commun. Math. Biol. Neurosci.*, vol. 2021, p. Article ID 39, 2021.

Andreas Werner Sihotang, A first-year student pursuing a Master's in Computer Science at Bina Nusantara University, is incredibly passionate about developing machine learning models and understanding data science. He aims to focus his studies on developing and innovating machine learning and deep learning model for stock prediction.

Andrea Stevens Karnyoto earned his Bachelor's degree from STMIK Dipanegara, a Master's degree from Hasanuddin University, and a Ph.D. from Harbin Institute of Technology. His Ph.D. studies focused on natural language processing, including Large Language Modeling (LLM), the latest transformers architecture, topic-based methods, and graph neural networks. Aside from working in Artificial Intelligence (AI), he has a web and desktop programming background. During his career in computer science, he created many information systems for various organizations. In addition, he has received two grants for research and two for community service from the Ministry of Education, Culture, Research, and Technology.

Bens Pardamean has over thirty years of global experience in information technology, bioinformatics, and education, including a strong background in database systems, computer networks, and quantitative research. His professional experience includes being a practitioner, researcher, consultant, entrepreneur, and lecturer. His current research interests are in developing and analyzing genetic data in cancer studies and genome-wide association studies (GWAS) for agriculture genetic research. After successfully leading the Bioinformatics Research Interest Group, he currently holds a dual appointment as the Director of the Bioinformatics & Data Science Research Center (BDSRC) and as a Professor of Computer Science at the University of Bina Nusantara (BINUS) in Jakarta, Indonesia. He earned a doctoral degree in informative research from the University of Southern California (USC), as well as a master's degree in computer education and a bachelor's degree in computer science from California State University, Los Angeles.