Feature Engineering for Machine Learning-Based Trading Systems Using Decision Tree, Random Forest, and Gradient Boosting

Nugroho Agus Haryono, Yuan Lukito, Aditya Wikan Mahastama
Department of Informatics, Universitas Kristen Duta Wacana, Yogyakarta, Indonesia

***Abstract*—Machine learning-based trading systems require the selection and creation of features that crucially determine the performance level of the trading system. This study introduces an asset-specific, correlation-based feature selection approach for machine learning–based stock trading models. The research conducts a systematic evaluation of the influence of lookup period, the number of features from technical analysis, and feature selection on the performance of trading systems using tree-based algorithms: Decision Tree, Random Forest, and Gradient Boosting. The performance of the trading system was measured using the backtesting method, with metrics such as total return, win rate ratio, and profit factor. The research steps included selecting stocks with the largest market capitalization in the financial sector, which are included in the banking index. Historical data on the prices of these stocks was obtained from Yahoo! Finance for the years 2014-2025. The historical data was then divided into two parts, namely the in-sample dataset (2014-2024 time period) and the out-of-sample dataset (2025 time period). Each part of the data was supplemented with features from technical analysis and several other additional features. Trading signals are determined based on a profit target of +4% and a loss limit of –2% in a lookup period of 2 to 10 days. The results show that the ML strategy consistently outperforms the buy-and-hold strategy, with Gradient Boosting generating the highest return (37.443%). Spearman correlation-based feature selection per stock improves the performance of the strategy compared to uniform features.**

***Keywords*—Feature engineering; machine learning; trading system; decision tree; Random Forest; Gradient Boosting**

I. Introduction

Predicting prices in financial markets to help build trading systems requires sophisticated methods to detect market direction and take advantage of price volatility in the market to profit from trading. The complexity of prices in the market encourages the integration of machine learning algorithms into trading systems to improve trading system performance [1]. Machine learning is capable of analyzing large and complex datasets, detecting and predicting patterns, and making predictions based on historical trends [2]. Accurate predictions play an important role in making trading decisions and measured risk management [3]. Research on the stock market is most widely studied by researchers, followed by the foreign exchange (Forex) market and cryptocurrency trading. In developing predictive models using artificial intelligence techniques, technical analysis is more widely used than fundamental analysis. Technical analysis is developed based on statistical methods that describe stock price movements over a certain period. In financial markets, technical analysis aims to find price patterns and stock price trends that can be used by traders or investors to make decisions. The performance of a trading strategy can be validated using several testing methods. Backtesting is a popular and considered a robust assessment method [4]. The performance of portfolios built with machine learning can be measured using excess return and Sharpe ratio metrics [5], [6]. The accuracy of predictions generated by a model can be tested using backtesting to look at the performance results of a trading system before it is implemented in real trading [7].

In building a trading model, algorithms such as decision tree, ensemble learning Gradient Boosting, and Random Forest bagging techniques can be employed, which have considerable prediction accuracy and effectiveness [8]. The decision tree algorithm was chosen for its simplicity, ease of implementation, and ease of interpretation [9]. Metode Ensemble learning methods, such as Random Forest and Gradient Boosting, perform well in terms of accuracy, precision, and recall [10].

In the study of machine learning-based trading applications, several things have to be carried out, including: building a machine learning model using historical stock price data, adding features and providing classification labels (buy/hold/sell) based on target profit and target loss rules within a certain period, training the model using training data and testing data, using the model to classify data that will be used for backtesting, and measuring the performance of the trading strategy applied using a backtesting library using trading metrics, such as total return, win rate, and profit factor [11].

While many international studies have been conducted, there remains a research gap in the context of developing country stock markets, including Indonesia: few studies have systematically compared: 1) the effect of varying lookup period (e.g. 2-10 days), 2) the effect of the number of technical features (1–6 features) and feature selection methods (unique features per issuer vs. uniform features) on 3) actual trading performance evaluated using backtesting metrics such as total return, win rate, and profit factor. This study addresses this gap with an experimental study of the 15 largest capitalized bank stocks on August 31, 2025 on the Indonesia Stock Exchange, to assess the combination of feature engineering and Decision

Tree, Random Forest, and Gradient Boosting algorithms, and compare the results against a buy-and-hold strategy.

This research adopts an experimental methodology to systematically examine how variations in lookup periods, feature dimensionality, and feature selection methods influence machine learning–based stock trading performance using out-of-sample backtesting. This research differs from prior work by adopting a correlation-based, asset-specific feature selection scheme rather than a uniform feature set for all stocks in machine learning–based trading models. The main contributions of this study are summarized as follows: 1) systematic evaluation of the effect of lookup period ranging from 2 days to 10 days and the number of technical features ranging from 1 feature to 7 features on the performance of machine learning-based trading, 2) comparison of two feature selection approaches—features selected per stock using Spearman's correlation compared to a uniform set of features for all stocks. The evaluation was conducted using out-of-sample backtesting analysis (data from January 1 to August 31, 2025) with practical trading metrics, namely total return, win rate, and profit factor, thus making the results relevant for practitioners and researchers who intend to implement machine learning on the Indonesian stock market.

II. Literature Review

A. Machine Learning for Stock Trading Systems

Studies related to stock market prediction and trading systems have developed rapidly using machine learning (ML) algorithms. High and varying volatility among issuers, non-linearity, and changing market dynamics make trading decisions a significant challenge. A systematic review found that studies of ML algorithms in the stock market are classified into supervised learning (regression and classification) and unsupervised learning (clustering). A number of ML algorithms for model building include neural networks, support vector machines, fuzzy theory, deep learning, random forest, and decision trees, as well as hybrid algorithms [12].

For instance, Anwar et al. (2024) reviewed that ML algorithms are capable of identifying stock market trends more accurately than traditional methods. Methods in machine learning, such as the Ensemble method, can be used by both investors and traders to analyze stock market trends. The selection of specific technical features as inputs for the model significantly affects the results of stock market trend predictions [10]. Machine learning models make it possible to combine various features to capture complex and nonlinear patterns in stock market forecasting. Model efficiency may be improved by selecting more powerful features [13].

B. Feature Engineering and Technical Indicators

The key factors in ML model performance are the quality and relevance of features (feature engineering). In the context of the stock market, many studies have explored the use of technical indicators (e.g., moving average, RSI, Aroon, VWMA minus price, slope, volume) as representations of market signals. Each indicator provides a unique perspective, offering additional insights into market dynamics that may not be revealed by daily stock prices. In their study, Fozap used Moving Averages (SMAs and EMAs), Bollinger Bands, Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and On-Balance Volume (OBV) as inputs for the ML model they built. These indicators represent momentum, trend, and trading volume [14]. In their study on Dow30 index stocks, Aksehir (2022) used 15 technical features, which were RSI, Williams's % R, WMA, EMA, SMA, HMA, Triple EMA, CCI, CMO, MACD, PPO, ROC, CMFI, DMI, and PSI as inputs for the CNN-TA model used to predict stock trading actions on the following day [15]. A study examined the integration of technical indicators with ML and found that this combination significantly improved accuracy compared to only ML or only technical analysis, especially when market volatility was high [16].

Selecting the optimal technical indicators is an important part of predicting the stock market, but there is currently no consensus on which indicators are the most suitable. Technical indicators are a crucial component of financial forecasting, as they provide a quantitative measure of market dynamics, including price trends, volatility levels, and trading volume. These indicators allow analysts to identify the direction and intensity of stock price movements, and have been widely integrated into machine learning models as a representation of historical price behavior [17]. Research conducted by Mostafavi and Hooman put a number of indicators into categories of momentum, trend, volatility, and volume. From each category, technical indicators that gave the best performance were obtained. Furthermore, research could be conducted for different markets, for the development of hybrid models combining machine learning with traditional technical analysis, and for the effect of time horizons on model performance [18]. Traditional technical indicators may have limitations in their individual predictive value, however when combined with machine learning methods, these indicators have the potential to provide more comprehensive insights into market movement characteristics [19].

C. Ensemble Tree-based Learning in the Context of Trading

Decision tree-based algorithms remain very popular in financial applications due to their interpretability, relative tolerance to different feature scales, and ability to handle non-linear relationships well.

- Decision Tree is a graphical-based decision-making method that uses probability analysis to evaluate the risks and feasibility of projects, depicted as tree branches. This method is intuitive and easy to understand visually. Decision trees can be used as comprehensive classification models for stock prediction and risk assessment [2].

- Random Forest is an ensemble algorithm that combines multiple decision trees, each trained on a randomly selected subset of data and features. The prediction results from all trees are combined through averaging to improve accuracy. This algorithm is effective for high-dimensional data and is able to reduce overfitting by incorporating random elements in the training process. Therefore, Random Forest is often used for large-scale datasets and complex problems [2]. In their study of China's A-share market covering 3,000 companies using 8 years of historical data, Wu (2024) found that the Random Forest-based feature selection method demonstrated superior performance with higher prediction accuracy compared to traditional approaches [20].

- Gradient Boosting is a boosting method that iteratively improves the errors of previous models and is known to perform well with tabular data. Priel (2024) conducted research to select stocks for value investing in the US stock market by utilizing corporate financial features. Through preliminary research, it was discovered that the Random Forest and GB ML methods consistently yielded higher average precision and recall values compared to Logistic and Deep Learning models [21]. In their study towards articles in Scopus and Web of Science between 2011 and 2022 concerning stock market applications, Htun (2023) discovered that RF and SVM methods are the most popular Machine Learning methods [22].

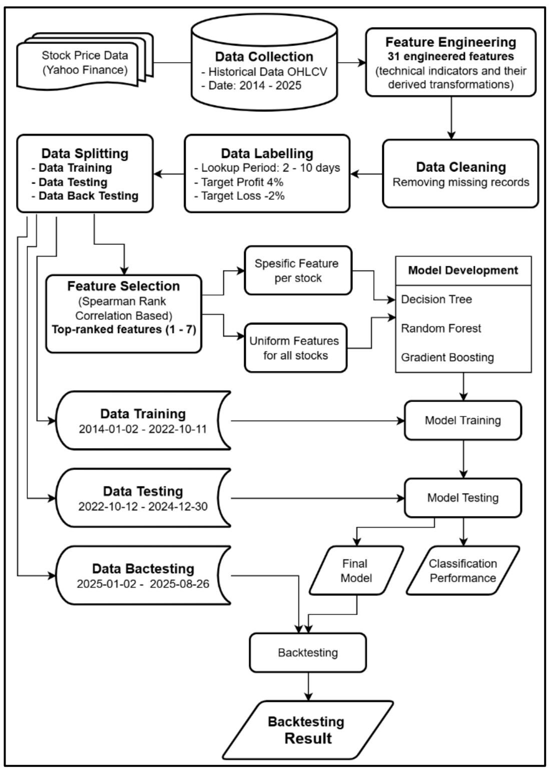


Fig. 1. Research workflow.

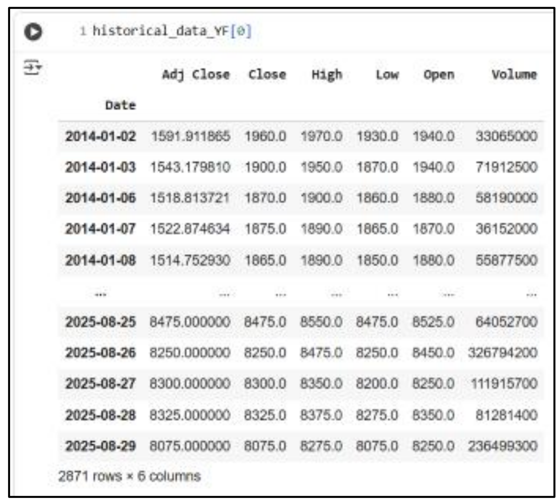
From the preceding literature review, it is possible to identify several gaps that remain open: 1) variations in lookup period: many studies use a single or arbitrary horizon, but few systematically test the range of lookup period in the context of ML trading signals. 2) The number and combination of features: although feature engineering has been widely discussed, systematic experiments on “how many features” and “whether features are uniform versus specific to each stock issuer” are still limited. 3) Trading system evaluation (not merely price prediction): many studies focused on price or direction prediction, but few linked prediction results to trading performance (cumulative return, win rate, profit factor) in the out-of-sample period, 4) emerging market context: studies specific to Indonesian stocks or emerging markets remain relatively scarce, hence research on 15 Indonesian bank stocks could provide additional values.

III. Research Methodology

This study aims to analyze the impact of feature engineering on the performance of machine learning-based trading systems for large-cap banking stocks listed on the Indonesia Stock Exchange. The research methodology design consists of data collection, feature construction, model training, and out-of-sample backtesting to comprehensively evaluate model performance, as shown in Fig. 1.

A. Data Collection and Preprocessing

Stock price data was obtained from Yahoo Finance for the top 15 banking stocks by market capitalization in Indonesia, covering the period from January 1, 2014, to August 31, 2025. The data obtained includes Adj Close, Close, High, Low, Open, and Volume prices, as presented in Fig. 2.



```
1 historical_data_YF[0]
```

Date	Adj Close	Close	High	Low	Open	Volume
2014-01-02	1591.911865	1960.0	1970.0	1930.0	1940.0	33065000
2014-01-03	1543.179810	1900.0	1950.0	1870.0	1940.0	71912500
2014-01-06	1518.813721	1870.0	1900.0	1860.0	1880.0	58190000
2014-01-07	1522.874634	1875.0	1890.0	1865.0	1870.0	36152000
2014-01-08	1514.752930	1865.0	1890.0	1850.0	1880.0	55877500
...	...	...	...	...	...	...
2025-08-25	8475.000000	8475.0	8550.0	8475.0	8525.0	64052700
2025-08-26	8250.000000	8250.0	8475.0	8250.0	8450.0	326794200
2025-08-27	8300.000000	8300.0	8350.0	8200.0	8250.0	111915700
2025-08-28	8325.000000	8325.0	8375.0	8275.0	8350.0	81281400
2025-08-29	8075.000000	8075.0	8275.0	8075.0	8250.0	236499300

2871 rows × 6 columns

Fig. 2. Example of BBCA.jk historical data.

The list of the top 15 banks by market capitalization in Indonesia, as of September 1, 2025, is provided in Table I.

TABLE I. List of the Top 15 Bank Stocks in Indonesia

#	*Issuer Code*	*Issuer Name*	*Market Capitalization (millions Rp.)*
1	BBCA.jk	PT. Bank Central Asia	985.976.046
2	BBRI.jk	PT. Bank Rakyat Indonesia	604.720.389
3	BMRI.jk	PT. Bank Mandiri	434.933.185
4	BBNI.jk	PT. Bank Negara Indonesia	162.423.639
5	BRIS.jk	PT. Bank Syariah Indonesia	125.010.400
6	BNLI.jk	PT. Bank Permata	112.885.657
7	BNGA.jk	PT. Bank CIMB Niaga	42.738.850
8	MEGA.jk	PT. Bank Mega	38.627.560
9	BBHI.jk	PT. Allo Bank Indonesia	34.659.828
10	NISP.jk	PT. Bank OCBC NISP	31.320.336

#	Issuer Code	Issuer Name	Market Capitalization (millions Rp.)
11	ARTO.jk	PT. Bank Jago	30.772.086
12	PNBN.jk	PT. Bank Pan Indonesia	27.314.683
13	BINA.jk	PT. Bank Ina Perdana	26.992.768
14	BDMN.jk	PT. Bank Danamon Indonesia	24.336.138
15	BTPN.jk	PT. Bank SMBC Indonesia	22.143.474

The dataset is divided into two main subsets:

- In-sample dataset (2014–2024): used for ML model training and testing. From this time period, 80% of the data was used as the training set and 20% as the testing set.
- Out-of-sample dataset (2025): used for backtesting the model's strategy against market conditions that the model has never observed before.

All data underwent cleaning to remove data that had no value due to the impact of feature calculations, such as the moving average feature for a certain period.

B. Feature Engineering

Feature engineering is performed by adding technical indicators and derivative features that theoretically represent market trends, momentum, and volatility. The list of indicators used in this study consists of commonly used technical indicators and derivative indicators prepared by the authors. The list of indicators used and their descriptions are described in Table II.

TABLE II. LIST OF 31 TECHNICAL INDICATORS USED

Name of Indicator	Description	Formula
apo (Absolute Price Oscillator)	apo attempts to quantify market momentum	APO = absolute(Long Cycle (Slow Moving Avg) - Short Cycle (Fast Moving Average))
aroon_up	to measure how many periods have passed since price has recorded an n-period high	Aroon-Up (14) = ((14 - Days Since 14-day High)/14) x 100
aroon_down	to measure how many periods have passed since price has recorded an n-period low	Aroon-Down (14) = ((14 - Days Since 14-day Low)/14) x 100
aroon_diff	represents the difference between aroon_up and aroon_down	Aroon_diff = aroon_up – aroon_down
aroon_diff_ma5	represents the 5-moving average of aroon_diff	Aroon_diff_ma5 = MA5(aroon_diff)
adx	measuring the amount of movement in a single direction	ADX = 100 times the Exponential Moving Average of the Absolute Value of (+DI - -DI) / (+DI + -DI) DI=Directional Indicator
cci	attempts to identify "overbought" and "oversold" levels relative to a mean.	CCI (20) = (Typical Price - 20 Period SMA of TP) / (.015 x Mean Deviation) Typical Price (TP) = (High + Low + Close)/3
inertia	Inertia is the rvi smoothed by the Least Squares MA.	Inertia = Least Squares Moving Average of Relative Volatility Index
kst	attempts to capture trends using a smoothed indicator of four different smoothed ROCs	KST =(ROCMA1 x 1) + (ROCMA2 x 2) + (ROCMA3 x 3) + (ROCMA4 x 4)
kst_sig	Kst smoothed by n-periode SMA	Kst_sig = n-Period SMA of the KST
kst_diff	represents the difference between kst and kst_sig	Kst_diff=kst – kst_sig
macd_hist	Moving Average Convergence Divergence Histogram. MACD can be used to identify aspects of a security's overall trend.	MACD Line: (12-day EMA - 26-day EMA) Signal Line: 9-day EMA of MACD Line MACD Histogram: MACD Line - Signal Line
momentum	Momentum attempts to quantify speed by using the differences over a bar length	Mom(t,n)=close(t)-close(t-n)
Obv (on balance volume)	obv is used in technical analysis to measure buying and selling pressure.	Obv(t)=obv(t-1)+volume, if price(t)>price(t-1), and Obv(t)=obv(t-1)-volume if price(t)<price(t-1), and obv(t)=obv(t-1) if price doesn't change
ppo	shows the relationship between two exponential moving averages in percentage terms.	PPO=100 x (12-periode EMA – 26-periode EMA)/26-periode EMA
ppo_signal	9-period EMA of PPO	Ppo_signal = 9-period EMA of PPO
rsi	RSI used to attempts to quantify "velocity" and "magnitude" of directional price movements	RSI = 100 – 100/ (1 + RS) RS = Average Gain of n days UP / Average Loss of n days DOWN
ma5_rsi	5-periode moving average of rsi	ma5_rsi = 5-periode moving average of rsi
slope	Calculates a rolling slope.	Slope=coeficient or slope of linear regression line
stoch_slowk_diff_ma5	5-periode moving average of stoch_slowk_diff	stoch_slowk_diff_ma5= 5-periode moving average of stoch_slowk_diff
tsi	attempts to identify short-term swings in trend direction	Tsi=(price change double smoothed/absolute prcie change double smoothed)/100
tsi_signal	Tsi line smoothed by 12-periode EMA	Tsi_signal=12-period EMA of the TSI line.
Volume	Total Volume perdagangan	Volume perdagangan
vwma_minus_price	Computes a weighted average	3-day vwma=(c1.v1+c2.v2+c3.v3

Name of Indicator	Description	Formula
	using price and volume.	)/ (v1+v2+v3) c=close price v=volume vwma_minus_price = vwma – close price
vwma_minus_price_ma5	5-periode moving average of vwma_minus_price	vwma_minus_price_ma5 = 5-periode moving average of vwma_minus_price
f1	the deviation of the On-Balance Volume (OBV) from its 5-period moving average.	f1=obv – MA5(obv)
f2	the deviation of the Close Price from from its 5-period moving average.	f2=Close – MA5(Close)
f3	represents the difference between rsi line and MA5(rsi)	f3=rsi – MA5(rsi)
f4	the difference between the ppo and ppo signal line	f4 = ppo - ppo signal line
f5	Value of ppo histogram	f5 = ppo histogram
f6	Difference of stoch_slowk and stoch_slowd	Stoch_slowk_diff= stoch_slowk – stoch_slowd

The number of features used in creating the model varied from 1 to 7 depending on the experimental design. Feature selection was based on the Spearman rank correlation value with respect to the target column (return lookup period).

Two approaches were used to evaluate the impact of feature selection:

1) Feature-specific per stock: Each stock has a different combination of features, which are selected based on their best Spearman's correlation.

2) Uniform features across stocks: All stocks use the same combination of features based on the highest aggregate correlation results.

C. Label Construction

The target label is constructed based on buy, hold, and sell signals calculated from price returns during a specific lookup period. The lookup period was varied from 2 days to 10 days to observe the sensitivity of the strategy to the time horizon. Returns were calculated using the Formula (1).

$$R_t = (Close_{t+lookup} - Close_t)/Close_t \quad (1)$$

with $Close_t$ is the closing price on day t.

Labels are constructed based on profit–loss rules:

- Buy, if Rt ≥ profit target (in this study, a value of 4% is used)
- Sell, if Rt ≤ loss target (in this study, a value of -2% is used)
- Hold for values between those two boundaries.

This approach reflects common practice in rule-based signal labeling trading systems.

D. Data Splitting

The data is split into three parts, with the following description.

- Data from January 1, 2014 to December 31, 2024 is used for training and testing data, with 80% of the data allocated for training and 20% for testing.
- Data from January 1, 2025 to August 31, 2025 is used as out-of-sample data to test trading performance through backtesting.

E. Model Development

Three main algorithms are used for trading signal classification:

1) Decision Tree: single tree model using Gini impurity.

2) Random Forest: bagging-based ensemble model, built several independent trees and combined the results for stability. In this study, 100 estimators and max_depth=4 were used.

3) Gradient Boosting: boosting-based ensemble model, training trees sequentially to minimize residual error. In this study, 100 estimators, max_depth=4, and learning_rate=0.1 were used.

For each algorithm, the model is trained separately on each of these combinations:

- Number of features, varied between 1 and 7 specific features per issuer.
- Lookup period, varied between 2 and 10 days.

After the best number of features and the best lookup period were obtained, testing was conducted for 1 to 7 features using a uniform feature approach for all issuers. The resulting model was used to classify buy–hold–sell signals in the out-of-sample (2025) data.

F. Backtesting Evaluation

Backtesting is used to evaluate the performance of strategies generated by the model against real data from 2025. This procedure is performed using the backtesting.py library, which enables portfolio simulation based on model signals.

The steps include:

1) Read the signal of the model prediction results on the out-of-sample data.

2) Implement trading strategy based on the following rules:

- Buy if signal = Buy
- Sell if signal = Sell
- Hold if signal = Hold

3) Calculate trading performance using the following metrics:

- Total return (%)
- Win rate (%)
- Profit factor (ratio of gross profit to gross loss)

4) Compare the results of the ML strategy against the Buy-and-Hold (BH) strategy return as a baseline benchmark.

The entire process was carried out over 15 major bank stocks, and the results were then averaged to obtain the aggregate performance of each algorithm and feature configuration.

Through the employed research design, this study enables a systematic evaluation of the relationship between feature complexity, time horizon coverage, and the effectiveness of machine learning algorithms in formulating sustainable and profitable trading strategies.

IV. Experimental Analysis and Result

This section presents the results of experiments on the implementation of machine learning models in signal-based trading systems (buy, hold, sell), focusing on 15 large-cap banking stocks on the Indonesia Stock Exchange. The evaluation was conducted to examine the effect of variations in the lookup period, the number of features used, and the type of algorithm on system performance, both in terms of signal classification accuracy and financial performance based on backtesting results.

A. Experimental Setup

The experiment was conducted on 15 large-cap banking stocks listed on the IDX. Each stock was tested with the combination of parameters given in Table III.

TABLE III. Variation in Testing Parameter Values

Parameter	***Variations***
Lookup period	2–10 days
Number of features	1–7 features
Algorithms	Decision Tree, Random Forest, Gradient Boosting
Training/validation time period	Jan 1, 2014 – Dec 31, 2024
Backtesting time period	Jan 1, 2025 – Aug 31, 2025
Evaluation	Return, Win Rate, Profit Factor

Two feature selection approaches were applied: 1) stock-specific features based on the highest Spearman correlation, and 2) uniform features for all stocks. Table IV provides list of features having the highest correlation with issuers while using a 3-day lookup period, with the number of features equal to 4.

TABLE IV. List of Selected Features Per Issuer

Issuer	**Selected features**			
	1	***2***	***3***	***4***
BBCA.	vwma_minus_price_ma5	vwma_minus_price	Volume	aroon_down
BBRI	Volume	vwma_minus_price	vwma_minus_price_ma5	aroon_down
BMRI	vwma_minus_price	vwma_minus_price_ma5	aroon_down	kst_sig
BBNI	Volume	f1	f2	aroon_down
BRIS	adx	Volume	f1	tsi_signal
BNLI	slope	f3	f2	vwma_minus_price_ma5
BNGA	kst_diff	f2	macd_hist	momentum
MEGA	aroon_down	vwma_minus_price	vwma_minus_price_ma5	stoch_slowk_diff_ma5
BBHI	f1	Volume	tsi_signal	rsi
NISP	vwma_minus_price	vwma_minus_price_ma5	aroon_down	cci
ARTO	tsi	tsi_signal	ma5_rsi	rsi
PNBN	Volume	aroon_diff_ma5	aroon_up	aroon_diff
BINA	slope	f2	momentum	macd_hist
BDMN	slope	f2	DeltaRSI	f3
BTPN	aroon_diff_ma5	inertia	vwma_minus_price_ma5	aroon_diff

B. Model Performance on Classification Metrics

Model performance on testing data (in-sample) was evaluated using standard classification metrics. Table V shows the average accuracy of one to seven features for each lookup period. Fig. 3 shown the bar chart displaying accuracy scores using specific features for each stock.

TABLE V. Classification Performance

Lookup Period	**Accuracy**			
	Decision Tree	***Random Forest***	***G Boosting***	***Average***
2	61%	79%	75%	72%
3	56%	74%	70%	67%
4	52%	69%	65%	62%
5	48%	66%	62%	58%
6	46%	62%	58%	55%
7	44%	60%	55%	53%
8	44%	58%	53%	52%
9	43%	57%	52%	51%
10	42%	55%	49%	49%
Average	49%	64%	60%	58%

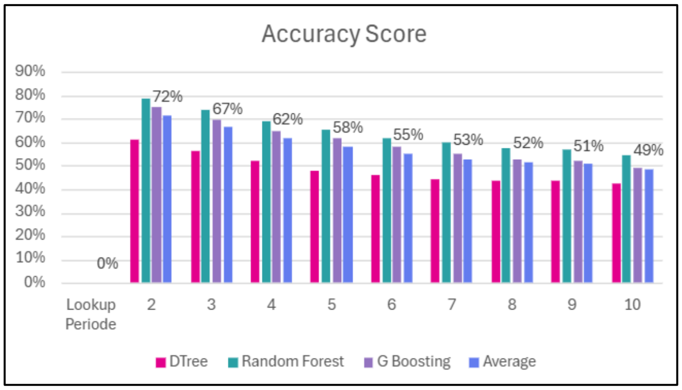


Fig. 3. Accuracy score per lookup period.

From the resulting accuracy chart, it is observed that the model is capable of predicting for relatively short lookup periods. The Random Forest model shows relatively better results, followed by Gradient Boosting, and then Decision Tree. Fig. 4 shows an example of the BBCA.JK stock accuracy matrix for the Random Forest model with a 3-day lookup period and 4 features.

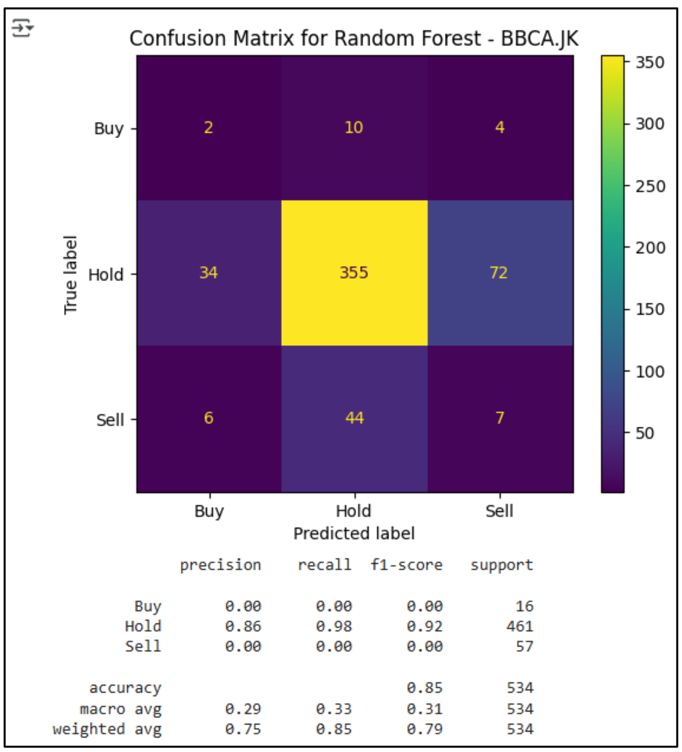


Fig. 4. Confusion matrix for random forest – BBCA.JK.

Fig. 5 exemplifies one of the Random Forest estimator trees for BBCA stock using 5 features and a 4-day lookup period.

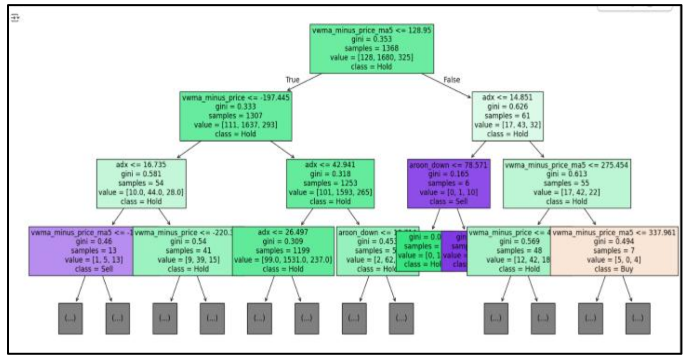


Fig. 5. Tree example of BBCA historical data.

C. Comparative Backtesting Results

The main evaluation was conducted using backtesting on out-of-sample data (January–August 2025 time period). The average aggregate backtesting results from total return calculations when compared to Buy & Hold for 15 stocks are presented in Table VI.

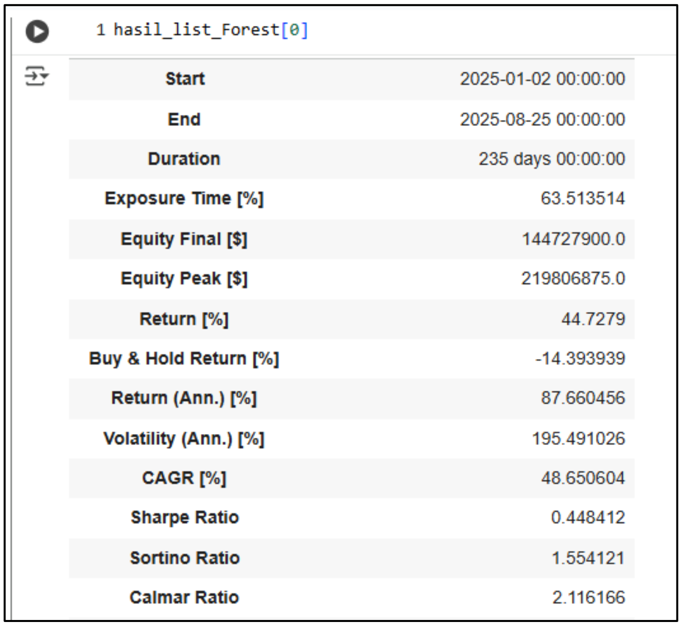
TABLE VI. BACKTESTING RESULTS

Strategy	***Total Return [%]***	***Win Rate [%]***	***Profit Factor***
Buy & Hold	15,13	-	-
Decision Tree	31,50	44,78	1,73
Random Forest	16,62	56,74	4.07
Gradient Boosting	37,44	46,85	3,18

The results show that all ML-based strategies outperform the Buy & Hold strategy in terms of total return.

- Gradient Boosting delivers the highest performance with an average total return of 37.44.3%, a win rate of 46.85%, and a profit factor of 3.18, indicating good profit consistency. While the win rate is not particularly high, the profit factor is quite substantial. This is because the trading system is built using a risk reward ratio of 1:2, with a profit target of 4% and a loss target of -2%.
- Decision Tree ranked second with a total return of 31.50%.
- Random Forest provides a stable but more conservative result (16.62%).

Fig. 6 displays an example of BBCA stock backtesting results using 5 features and a 4-day lookup period. Fig. 7 illustrates the trading curve on the BBCA.jk chart.



```
1 hasil_list_Forest[0]
```

Start	2025-01-02 00:00:00
End	2025-08-25 00:00:00
Duration	235 days 00:00:00
Exposure Time [%]	63.513514
Equity Final [$]	144727900.0
Equity Peak [$]	219806875.0
Return [%]	44.7279
Buy & Hold Return [%]	-14.393939
Return (Ann.) [%]	87.660456
Volatility (Ann.) [%]	195.491026
CAGR [%]	48.650604
Sharpe Ratio	0.448412
Sortino Ratio	1.554121
Calmar Ratio	2.116166

Fig. 6. Backtesting result for BBCA.jk.

Fig. 7. Plotting backtesting results for BBCA.jk.

D. Lookup Period Analysis

An evaluation was conducted to assess the sensitivity of the strategy to the time horizon based on the lookup period, analyzing variations in the lookup period from 2 days to 10 days. The average total return results presented in Table VII show that the system performed best in the 4-day to 7-day lookup period. The highest achievement was in the 5-day lookup period, which generated an average system return of 77.54%. This shows that a five-day period is the optimal horizon for capturing momentum patterns and price reversals in Indonesian banking stocks. A lookup period that is too short (less than 4 days) produces noisy signals, while a period that is too long (more than 7 days) loses sensitivity to short-term fluctuations.

TABLE VII. TOTAL RETURN PER LOOKUP PERIOD

Lookup Period	Total Return [%]				
	Decision Tree	***Random Forest***	***Gradient Boosting***	***Average***	***Buy& Hold***
2	3,83	2,87	25,99	10,90	16,15
3	26,81	7,93	33,02	22,59	16,06
4	19,76	8,27	47,35	25,13	15,87
5	112,80	21,49	98,32	77,54	14,83
6	35,01	23,27	37,26	31,85	15,06
7	18,91	25,38	36,58	26,96	15,55
8	8,76	23,79	30,13	20,89	14,12
9	31,39	17,70	6,44	18,51	14,78
10	26,20	18,87	21,83	22,30	14,93
Average	31,50	16,62	37,44	28,52	15,26

E. Feature Dimensionality Impact

The ensuing experiment evaluated the impact of the number of features on the performance of the trading system.

TABLE VIII. TOTAL RETURN PER NUMBER OF FEATURES

Number of Features	Total Return [%]			
	Decision Tree	***Random Forest***	***Gradient Boosting***	***Average***
1	10,02	8,98	11,88	10,29
2	29,74	20,31	31,31	27,12
3	38,83	30,82	44,92	38,19
4	89,72	15,77	32,93	46,14
5	23,78	12,75	35,10	23,87
6	15,98	16,15	70,82	34,32
7	12,41	11,56	35,10	19,69
Average	31,50	16,62	37,44	28.52

The results in Table VIII show that unsatisfactory performance occurs when the number of features used is 1 and 7. For the number of features between 2 and 6, ML performance is generally good and stable, giving the total return a range of 23% to 46%, with the highest total return of 46.14% achieved when there were 4 features used.

F. Feature-Specific vs. Uniform Feature Selection

The selection of uniform features for all issuers was also conducted in this study. The aggregation of features with the highest correlation that are frequently used by issuers when using a lookup period of 3, with a total of 4 features, is shown in Fig. 8.

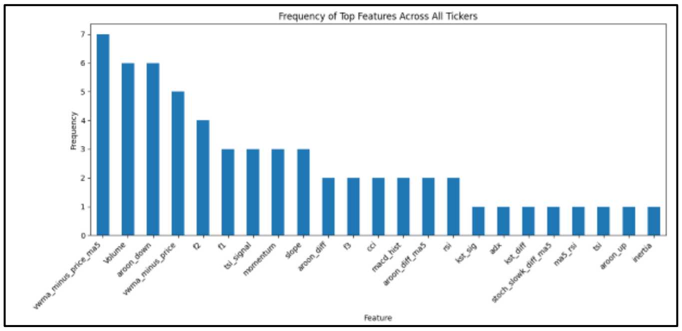


Fig. 8. Frequency of top features across all stocks.

This study uses a 4-day lookup period with uniform features ranging from 1 to 7 features that are most frequently used in aggregate for all issuers. A 4-day lookup period was chosen because its total return value is the median value of all total returns for the lookup period, its returns are relatively more stable, and it has good accuracy. The list of uniform features that are most frequently selected in aggregate for each number of features is provided in Table IX.

TABLE IX. LIST OF UNIFORM FEATURES PER NUMBER OF FEATURES

Number of features	*Uniform Features*
1	Volume
2	Volume, vwma_minus_price
3	Volume, vwma_minus_price, vwma_minus_price_ma5
4	Volume, vwma_minus_price, vwma_minus_price_ma5, aroon_down
5	Volume, vwma_minus_price, vwma_minus_price_ma5, aroon_down, f1
6	Volume, vwma_minus_price, vwma_minus_price_ma5, aroon_down, f1, aroon_up
7	Volume, vwma_minus_price, vwma_minus_price_ma5, aroon_down, f1, aroon_up, kst_sig

The results of accuracy research using uniform features compared to specific features with a 4-day lookup period are provided in Table X. The results of backtesting using uniform features are provided in Table XI.

TABLE X. CLASSIFICATION PERFORMANCE FOR UNIFORM FEATURES

Number of features	**Accuracy (Lookup period = 4 days)**				
	Decision Tree	***Random Forest***	***Gradient Boosting***	***Average for Uniform Features***	***Average for Specific Features***
1	0.53	0.70	0.65	0.63	0.65
2	0.51	0.69	0.65	0.62	0.62
3	0.51	0.69	0.65	0.62	0.61
4	0.50	0.68	0.64	0.61	0.61
5	0.50	0.68	0.64	0.61	0.61
6	0.51	0.69	0.63	0.61	0.62
7	0.51	0.69	0.64	0.61	0.62
Average	0.51	0.69	0.64	0.61	0.62

TABLE XI. TOTAL RETURN FOR UNIFORM FEATURES

Number of features	**Total Return [%] (Lookup period = 4 days)**				
	Decision Tree	***Random Forest***	***Gradient Boosting***	***Average for Uniform Features***	***Average for Specific Features***
1	17.55	-4.87	10.31	7.66	10.56
2	18.71	6.95	10.04	11.90	36.84
3	36.76	-1.46	21.98	19.09	2.61
4	4.74	1.06	56.38	20.72	39.49
5	16.94	-0.58	12.73	9.70	35.48
6	79.54	-7.70	76.91	49.58	39.63
7	11.11	4.34	50.26	21.90	11.28
Average	26.48	-0.32	34.09	20.08	25.13

The average return of specific features was 25.13%, which was 5.05% higher than the average return of uniform features, which was 20.08%. These results suggest that selecting specific features for each stock (based on the best Spearman correlation) produces higher performance than using uniform features.

These findings indicate that the characteristics of each stock, such as volatility, trend, and momentum, require a specific combination of features for each stock in order for the model to capture relevant stock patterns. The feature-specific approach per stock has proven to be more adaptive to the distinct characteristics of different stocks.

G. Discussion

The experiment results confirmed a number of important findings, as follows:

- The feature engineering and feature selection processes play a crucial role in determining the success of machine learning (ML) models in the context of stock trading.
- A lookup period of two to four days is sufficient to capture price movements effectively, as reflected in the accuracy rates obtained by the model.
- In terms of prediction accuracy, the Random Forest algorithm shows superior performance. However, in backtesting against out-of-sample data, the Gradient Boosting algorithm produces higher total returns.
- The use of specific feature selection for each issuer has proven to improve signal accuracy and overall portfolio performance.

From a practical perspective, the developed ML-based trading system shows significant potential as a decision support tool for active investors in the Indonesian capital market, particularly in the banking sector, which is characterized by medium volatility and high trading volume. The following experiment evaluates the impact of the number of features on trading system performance. This study does not consider transaction costs and only tests within a predetermined time frame. Therefore, further testing is needed to anticipate changes in market direction.

V. Conclusion and Future Work

A. Conclusion

This study has developed and evaluated a machine learning-based stock trading system using three main algorithms, specifically Decision Tree, Random Forest, and Gradient Boosting. Through conducting experiments on 15 large capitalization banking stocks on the Indonesia Stock Exchange during the period of 2014–2025, this study highlights the importance of feature engineering, the number of features, and the lookup period in affecting the performance of trading strategies.

The results showed that the combination of feature engineering techniques and tree-based ensemble algorithms consistently improves system performance compared to the Buy & Hold strategy. Out of the three algorithms tested, Gradient Boosting showed the best performance with an average total return of 37.44%, outperforming Decision Tree (31.50%), Random Forest (16.62%), and Buy & Hold (15.26%). These findings indicate the superiority of the boosting approach in capturing stock price movement patterns in emerging markets.

In addition, a lookup period of two to four days was able to identify price patterns. A feature selection approach tailored specifically to each stock—based on the highest Spearman correlation—proved to be more effective than a uniform feature approach. This confirms the importance of an adaptive strategy in selecting relevant features to capture the unique characteristics of each stock.

Collectively, this study provides an essential contribution to the AI for Trading literature, particularly in the context of emerging markets. The empirical results suggest that correlation-based feature selection enhances the robustness, adaptability, and interpretability of automated trading systems, supporting their use as effective decision-making tools for active investors.

B. Limitations and Future Work

While the results are substantial, this study has several limitations:

- It focused solely on the banking sector, therefore, generalizations to other sectors require further testing.
- It does not incorporate fundamental macroeconomic factors such as interest rates or inflation, which can affect stock price movements.
- The model has not been evaluated under extreme market conditions, i.e., crisis or periods of extreme volatility.

These limitations provide opportunities for expansion in future research.

Acknowledgment

We would like to express our gratitude to the Faculty of Information Technology at Universitas Kristen Duta Wacana for supporting the authors through funding and research facilities, thus allowing the research to be conducted completely.

References

[1] H. S. AlQahtani, M. J. Alhaddad, and M. Jarrah, "Comprehensive Analysis of Machine and Deep Learning Models for Stock Market Prediction." [Online]. Available: www.ijacsa.thesai.org

[2] M. Özcan, "Forecasting in stock market with Machine Learning: A State of Art," in 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, Jul. 2023, pp. 1–7. doi: 10.1109/ICCCNT56998.2023.10307692.

[3] Z. Shi, O. Ibrahim, H. Ilyana, and C. Hashim, "A Novel Hybrid HO-CAL Framework for Enhanced Stock Index Prediction," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 16, no. 7, p. 333, 2025, [Online]. Available: www.ijacsa.thesai.org

[4] F. Dakalbab, M. A. Talib, Q. Nasir, and T. Saroufil, "Artificial intelligence techniques in financial trading: A systematic literature review," Journal of King Saud University - Computer and Information Sciences, vol. 36, no. 3, p. 102015, Mar. 2024, doi: 10.1016/j.jksuci.2024.102015.

[5] C. Wang, "Stock return prediction with multiple measures using neural network models," Financial Innovation, vol. 10, no. 1, p. 72, Jun. 2024, doi: 10.1186/s40854-023-00608-w.

[6] J. P. Steyn, E. Gilbert, and S. Viviers, "The low-volatility effect in African frontier equity markets," Investment Analysts Journal, vol. 53, no. 3, pp. 189–206, Jul. 2024, doi: 10.1080/10293523.2024.2361986.

[7] H. Bala Malami et al., "Generating a Trading Strategy Using Candlestick Patterns with Machine Learning," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 16, no. 9, pp. 320–325, 2025, [Online]. Available: www.ijacsa.thesai.org

[8] E. M. Ferrouhi and I. Bouabdallaoui, "A comparative study of ensemble learning algorithms for high-frequency trading," Sci Afr, vol. 24, p. e02161, Jun. 2024, doi: 10.1016/j.sciaf.2024.e02161.

[9] Y. Wang and T. Yampaka, "Predicting Stock Price Bubbles in China Using Machine Learning," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 15, no. 11, pp. 415–425, 2024, [Online]. Available: www.ijacsa.thesai.org.

[10] A. Anwar, D. Ameya, A. Patil, M. S. Choudari, D. Chetna, and C. S. Kiran, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Machine Learning Insights for Stock Market Trend Identification," International Journal of Intelligent Systems and Applications in Engineering, vol. 2024, no. 10s, pp. 567–575, 2025, [Online]. Available: https://orcid.org/0000-0002-6756-6514.

[11] F. Sherasiya, "Developing A Machine Learning-Based Options Trading Strategy For The Indian Market," International Journal For Multidisciplinary Research, vol. 7, no. 4, Jul. 2025, doi: 10.36948/ijfmr.2025.v07i04.50375.

[12] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," Expert Syst Appl, vol. 197, p. 116659, Jul. 2022, doi: 10.1016/j.eswa.2022.116659.

[13] M. Patsiarikas, G. Papageorgiou, and C. Tjortjis, "Using Machine Learning on Macroeconomic, Technical, and Sentiment Indicators for Stock Market Forecasting," Information, vol. 16, no. 7, p. 584, Jul. 2025, doi: 10.3390/info16070584.

[14] F. M. P. Fozap, "Hybrid Machine Learning Models for Long-Term Stock Market Forecasting: Integrating Technical Indicators," Journal of Risk and Financial Management, vol. 18, no. 4, p. 201, Apr. 2025, doi: 10.3390/jrfm18040201.

[15] Z. D. Akşehir and E. Kiliç, "How to Handle Data Imbalance and Feature Selection Problems in CNN-Based Stock Price Forecasting," IEEE Access, vol. 10, pp. 31297–31305, 2022, doi: 10.1109/ACCESS.2022.3160797.

[16] E. Arif, S. Suherman, and A. P. Widodo, "Integration of Technical Analysis and Machine Learning to Improve Stock Price Prediction Accuracy," Mathematical Modelling of Engineering Problems, vol. 11, no. 11, pp. 2929–2943, Nov. 2024, doi: 10.18280/mmep.111106.

[17] F. M. P. Fozap, "Hybrid Machine Learning Models for Long-Term Stock Market Forecasting: Integrating Technical Indicators," Journal of Risk and Financial Management, vol. 18, no. 4, p. 201, Apr. 2025, doi: 10.3390/jrfm18040201.

[18] S. M. Mostafavi and A. R. Hooman, "Key technical indicators for stock market prediction," Machine Learning with Applications, vol. 20, p. 100631, Jun. 2025, doi: 10.1016/j.mlwa.2025.100631.

[19] A. Deep, A. Shirvani, C. Monico, S. Rachev, and F. Fabozzi, "Risk-Adjusted Performance of Random Forest Models in High-Frequency Trading," Journal of Risk and Financial Management, vol. 18, no. 3, p. 142, Mar. 2025, doi: 10.3390/jrfm18030142.

[20] Z. Wu, Z. Feng, and B. Dong, "Optimal Feature Selection for Market Risk Assessment: A Dimensional Reduction Approach in Quantitative Finance," Journal of Computing Innovations and Applications, vol. 2, no. 1, pp. 20–31, Jan. 2024, doi: 10.63575/CIA.2024.20103.

[21] R. Priel and L. Rokach, "Machine learning-based stock picking using value investing and quality features," Neural Comput Appl, vol. 36, no. 20, pp. 11963–11986, Jul. 2024, doi: 10.1007/s00521-024-09700-3.

[22] H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," Financial Innovation, vol. 9, no. 1, p. 26, Jan. 2023, doi: 10.1186/s40854-022-00441-7.