# Leveraging on Synthetic Data Generation Techniques to Train Machine Learning Models for Tenaga Nasional Berhad Stock Price Movement Prediction

Nur Aliah Syahmina Mohd Nazarudin
New Energy Division,
Tenaga Nasional Berhad, Malaysia
aliah.nazarudin@tnb.com.my

Nor Hapiza Mohd Ariffin
MIS Department,
Sohar University, Oman
nariffin@su.edu.om

Ruhaila Maskat
College of Computing, Informatics and Mathematics,
Universiti Teknologi MARA Shah Alam, Malaysia
ruhaila256@uitm.edu.my

**Abstract:** *This study employs machine learning models to explore stock price prediction for Tenaga Nasional Berhad (TNB), Malaysia's primary electricity provider. It addresses the limitations of previous studies by incorporating various input variables, including the stock market, technical, financial, and economic data. This study also tackles the issue of imbalanced class distribution due to small datasets of stock market data by generating synthetic data using Synthetic Minority Over-Sampling Technique (SMOTE) and Generative Adversarial Network-Synthetic Minority Over-Sampling Technique (GAN-SMOTE) techniques. The performance of four classifier models (random forest, Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), and Artificial Neural Network (ANN) is evaluated without any synthetic data and with synthetic data generated. The SMOTE-ANN model is the best-performing model, exhibiting superior accuracy of 93%, F1-Score of 92%, precision of 90%, recall of 94%, and specificity of 92%. Overall, this research provides valuable insights into TNB stock price movements, offers a solution for imbalanced class distribution, and identifies the top-performing model for predicting TNB stock price movement. These findings are relevant to investors, analysts, and organisations in the utility sector.*

**Keywords:** *Stock price prediction, machine learning, synthetic data generation.*

*Received; October 24, 2023, accepted, May 09, 2024*
*https://doi.org/10.34028/iajit/21/3/11*

## 1. Introduction

The stock market plays a crucial role in the economy, offering investment opportunities and driving economic growth [13]. The stock market also provides investors with a wide range of opportunities to diversify their portfolios and potentially earn investment returns [16]. Consequently, predicting the movement of stock prices has become an area of interest for analysts, investors, and organisations involved in the stock market. Tenaga Nasional Berhad (TNB) is Malaysia's largest electricity utility company listed on the Bursa Malaysia stock exchange [21]. Like other stock markets, its share price is influenced by the dynamics of supply and demand and various factors such as economic conditions, industry trends, company-specific news, and global events.

The stock price movement prediction is challenging due to its non-stationary and volatile nature [9], as its mean and variance vary over time, and it may have the presence of trend and seasonality [16], which can lead to high-risk investments. The inability to accurately predict the stock price movement can result in a loss of interest in potential investors [16]. In addition, the time-series nature of stock price data results in relatively small datasets [8], limiting data available for training and testing machine learning models [6]. Small datasets can also lead to imbalanced class distributions, affecting the performance of stock price prediction models [15]. Additionally, the behaviour and predictability of stock markets differ between developed and emerging markets [23] due to different compositions of industries and higher economic growth rates in emerging markets [4].

Previous studies have employed various machine learning and deep learning models for stock market price and movement prediction, including Support Vector Machine (SVM), random forest, Extreme Gradient Boosting (XGBoost), Artificial Neural Network (ANN), and Long-Short Term Memory (LSTM).

SVM is a supervised learning algorithm used for classification by finding a boundary that separates the data points into different classes or predicts the value of the output variable. The boundary that separates these two classes are called hyperplane, and it must be far from any point in the dataset. It can handle non-linearly separable data using kernel tricks. Studies have shown

that SVM can handle high-dimensional data and model non-linear relationships between input and output variables, making it suitable for stock price prediction [3, 17].

Random forest is a bagging ensemble learning algorithm based on decision trees. It is known for handling large amounts of data and high dimensionality. Random Forest uses Bootstrap to train a group of trees [14]. The estimation of the best split is done based on the number of randomly chosen features in the dataset, which accounts for boosting its performance. The determinative decision is taken after the majority voting from the constructed trees. Random forest effectively predicts stock prices, with studies reporting high accuracy and low prediction errors [17].

XGBoost, on the other hand, is an improved version of the gradient boosting algorithm, an ensemble method that combines multiple decision trees to improve the model's predictive performance. It is also a robust boosting ensemble learning algorithm that uses gradient descent to reduce overfitting while using many trees [14]. Previous studies have shown that XGBoost can handle high-dimensional data and model non-linear relationships between input and output variables, making it suitable for stock price prediction. Finally, ANN is a supervised learning algorithm based on the human brain's structure for prediction, classification, and control tasks. It is also a prominent subset of machine learning algorithm that is usually single or multi-layer nets that are fully connected [17]. During the past few decades the ANNs have shown great applicability in time series prediction especially in time-series prediction [5].

SVM and Random Forest are two commonly used machine learning models [7, 20], and the model performance was often compared with ANN as it has shown its ability its ability to learn non-linear relationships between input and output variables [17]. Model performance varies depending on the behaviour of stock market data and the choice of input variables used to feed into the models.

Most studies use only stock market data as input variables, and only a few studies address imbalanced class distribution [18], which can negatively impact the performance of classification machine learning models in stock price movement prediction [15]. Carvajal-Patiño and Ramos-Pollán [6], and Zhang *et al*. [22] suggested using generative models to create synthetic time-series data to increase dataset size for model training, but average model accuracy remains below 60% due to imbalanced class distribution. Additionally, few studies have combined stock market data variables with technical, financial, and economic variables for stock price prediction in emerging markets, with the majority focusing on developed markets [2, 12].

This study aims to address these issues using synthetic data generation to increase dataset size, overcome imbalanced class distribution, and incorporate all significant variables influencing TNB stock price movement as input features for classification machine learning models to predict stock price movement in an emerging market. This study will use classification machine learning models to predict the stock price movement of TNB using a combination of real and synthetic data.

The following sections of this paper are organised as follows. Chapter Two offers an extensive review of relevant literature, encompassing previous studies on various aspects of the stock market, such as market types, key variables, and the application of machine learning algorithms for predicting future market movements. This chapter also highlights the strengths and limitations of these approaches. Chapter Three outlines the methodology employed in this study, detailing data collection and pre-processing techniques, as well as the modeling strategies utilized for predicting stock price movements. Additionally, it discusses the evaluation metrics used to gauge the performance of the models, along with descriptions of the data sources and datasets utilized, and any assumptions or limitations inherent in the analysis. Chapter Four provides a detailed synthesis of the study's findings, utilizing descriptive statistics, tables, and graphs to present a comprehensive analysis of the dataset. This chapter aims to address the research questions and objectives clearly and thoroughly. Finally, Chapter Five concludes the paper and outlines potential avenues for future research.

## 2. Related Works

In a study by Deng *et al*. [11], stock market data from Rebar Futures, a Chinese emerging market steel company, was utilised. Technical data derived from the market was used as input variables. The authors employed a sliding window technique and utilised Synthetic Minority Over-Sampling Technique (SMOTE) for oversampling the minority class. SMOTE-NSGA-II-XGBoost, where NSGA-II stands for Non-dominated Sorting Genetic Algorithm II, emerged as the best-performing model, achieving the highest hit ratio of 53.95%, while XGBoost without SMOTE achieved 48.98%. The accuracy is still relatively lower, although SMOTE was utilised to balance the class distribution. The authors recommended incorporating investor sentiment features, exploring alternative algorithms, and addressing class imbalance to improve predictive modelling in similar financial markets. This study adapted the sliding window technique, explored SMOTE and Generative Adversarial Network-Synthetic Minority Over-Sampling Technique (GAN-SMOTE) for class imbalance, and assessed the performance of XGBoost for prediction accuracy of stock price movement.

Ghasemieh and Kashef [14] used stock market data and 18 derived technical data to predict next-day stock prices for 83 stocks. SVM was the best-performing machine learning model with 67% accuracy, and

Feedforward Neural Network (FFNN) outperformed LSTM in deep learning with 68% accuracy. Despite effective feature selection techniques, accuracy remained below 70%. This study would explore SVM's potential for predicting TNB stock price movement using the same feature selection techniques and investigates the effectiveness of these techniques combined with SMOTE and GAN-SMOTE.

Nabipour *et al*. [17] utilised continuous and binary technical data to predict stock price movements in diverse stock market groups on the Tehran stock exchange. Comparing various machine learning models, they identified ANN as the most accurate at 87.56%, and Recurrent Neural Network (RNN) outperformed LSTM in deep learning at 90.12% accuracy. The study highlighted the significant enhancement of classifier performance through binary-type input technical variables. Consequently, this study adopted their techniques by encoding technical variables into binary data types for input variables in the modelling process and included ANN as one of the classifier models.

Another study by Das *et al*. [10] used stock market data from the S and P 500 index to predict whether to buy, hold, or sell stocks. They employed a sliding window technique and oversampled the minority class using SMOTE. Their study compared the performance of SMOTE-tree bag and SMOTE-Random forest, with SMOTE-random forest achieving the best performance at 99.56% Area Under the ROC Curve (AUC). The authors aimed to explore the adaptability of the combined SMOTE-Random forest model to other large-scale datasets, extending its utility beyond the S and P 500 stock market dataset. Thus, this study adapted the sliding window technique, explored SMOTE for addressing the class imbalance issues, and included random forest as one of the classifier models.

Lastly, Carvajal-Patiño and Ramos-Pollán [6], aimed to predict the price velocity of market prices using real and synthetically generated datasets. They employed Variational Autoencoder (VAE) and GAN techniques for data generation and compared the performance of random forest models. The combination of GAN-random forest achieved the highest accuracy of 59%, which could be further improved. The authors recommended exploring diverse trading market signals, incorporating technical indicators, and external economic variables employing advanced generative models, adopting robust predictive models (such as RNN), and addressing the class imbalance in enhancing stock market analysis and prediction. Since they recommended balancing synthetically generated data, this study aimed to experiment with other techniques, such as SMOTE, for balancing class distribution after synthetic data is generated using GAN.

# 3. Methods and Materials

## 3.1. Data Collection

This study used 11 years of historical stock data and financial variables extracted from S and P Capital IQ [19] to predict TNB's stock price movement. S and P Capital IQ provides public access to extensive financial data, analytics, and research, serving as a valuable resource for academia, investors, analysts, and businesses. The extracted dataset consists of 2,870 rows with 17 attributes, including TNB stock market data and financial, economic, and technical variables. The dataset was imported into Jupyter Notebook using the Pandas library. Notably, the analysis covered the year 2020, allowing evaluation of TNB's performance during the volatile COVID-19 pandemic. Table 1 describes the data type, description and structure of each variable used in this study.

## 3.2. Data Preparation

The list of variables, along with their respective categories, data types, and counts, is presented in Table 1 below. This study addressed missing values using three scenarios. Firstly, rows missing due to stock market closure were removed. Secondly, columns with more than 30% missing rows were dropped. Finally, the remaining missing values in the time-series dataset were filled with the previous day's values using the fillna (method='ffill') function. Missing rows in the Open Price variable were replaced with the previous day's Close Price value. Both open price and close price are in Malaysian Ringgit (MYR), while the crude oil price is in United States Dollar (USD).

Table 1. Data description.

| Variable Category | Name of Variable | Data Type | Count |
|---|---|---|---|
| Stock Market Data Variable | Trading Date | Date | 2870 |
| | Close Price | Float | 2704 |
| | Volume | Float | 2704 |
| | Open Price (MYR) | Float | 2694 |
| | High Price (MYR) | Float | 1261 |
| | Low Price (MYR) | Float | 1261 |
| Financial Variable | Dividend Yield (%) | Float | 2696 |
| | Price-to-Earnings Ratio (Multiples) | Float | 2647 |
| Economic Variable | Crude Oil Price (USD) | Float | 2771 |
| | Long-Term Yield | Float | 2870 |
| | Forex Rate MYR:USD | Float | 2870 |
| Technical Variable | Relative Strength Index | Float | 2704 |
| | Volatility % | Float | 2704 |
| | Stochastic %K | Float | 2704 |
| | Stochastic %D | Float | 2704 |
| | Moving Average Convergence Divergence (MACD) | Float | 2704 |
| | Simple Moving Average (SMA) | Float | 2704 |

This study aims to develop a classification model for predicting TNB stock price movement. Hence, two

predicted variables were derived, and each classification model was evaluated based on their performance in predicting these two variables separately. Table 2 shows the two predicted variables used in this study. Note that 1 means the stock price movement is upward, while 0 means the stock price movement is downward.

Table 2. Predicted variables.

| Predicted Variable | Binary Condition |
|---|---|
| Daily Stock Price Change | 1: Current day closing price > Current day opening price, 0: otherwise |
| Next-Day Stock Price Movement | 1: Next-day closing price > Current day closing price, 0: otherwise |

Next, this study encoded technical variables into binary data types before using them as input variables for the modelling. Table 3 shows the binary condition of each newly derived variable. Consequently, all continuous technical variables were dropped from the data frame. As a result, the new data frame had 15 input variables and four predicted variables, with 2,704.

Table 3. Binary condition of technical variables.

| Predicted Variable | Binary Condition |
|---|---|
| Volatility | 1: Next day volatility > Current day volatility<br>0: Next day volatility < Current day volatility |
| Relative Strength Index (RSI) | 1: RSI < 30<br>0: RSI > 70 |
| Stochastic K% | 1: Current day Stochastic K% > Previous day Stochastic K%, 0: Otherwise |
| Stochastic D% | 1: Current day Stochastic D% > Previous day Stochastic D%, 0: Otherwise |
| Simple Moving Average (SMA) | 1: If current stock closing price is > SMA, 0: otherwise |
| Moving Average Convergence Divergence (MACD) | 1: Current day MACD > Previous day MACD, 0: Otherwise |

This study employed Pearson's correlation coefficient and Backward Elimination techniques to identify and select the most relevant features for the prediction models. By leveraging these techniques, the study aimed to reduce dataset complexity, identify significant variables that influence TNB stock price movement, and enhance the overall performance of the prediction models.

This study performed the calculation of Pearson's correlation coefficient between each variable using the corr() function from Pandas library to determine the relationship between input variables and all two predicted variables. Equation (1) shows Pearson's correlation coefficient formula [1], where x and y are input and predicted variables, respectively while n is number of observations. Any input variables with a correlation value of less than 0.2 will be removed for each predicted variable.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (1)$$

In addition, backward elimination was used to check each feature's relevance by building a linear model with and without the presence of the input variables [14]. In this feature selection method, the p-value indicates the significance of each variable in the model. Variables with p-values above a threshold (often 5%) are removed iteratively until only statistically significant variables remain in the final model [14]. Finally, the input variables chosen for the model training were based on Pearson's correlation coefficient between each input variable and each target variable and the backward elimination technique. Any input variable with a correlation coefficient greater than 0.2 and input variable with a p-value lower than 0.05 based on linear regression of the backward elimination technique was used for training the models. This combination of techniques would ensure that the model was trained with only the most relevant and significant input variables that influence the movement of TNB stock price, leading to better prediction model performance.

This study utilised a sliding window technique for the dynamic arrangement of the training and testing sets for the TNB stock price movement prediction. This technique involved dividing the dataset into a series of fixed-size windows, each representing a subset of data [11]. Figure 1 shows a sliding window technique applied in this study, and Table 4 shows the specific date ranges and sample sizes of four training and testing datasets. Note that the whole testing period lasted for one year. This study used the TimeSeriesSplit class from the scikit-learn library to split the dataset into training and testing sets using the sliding window technique.

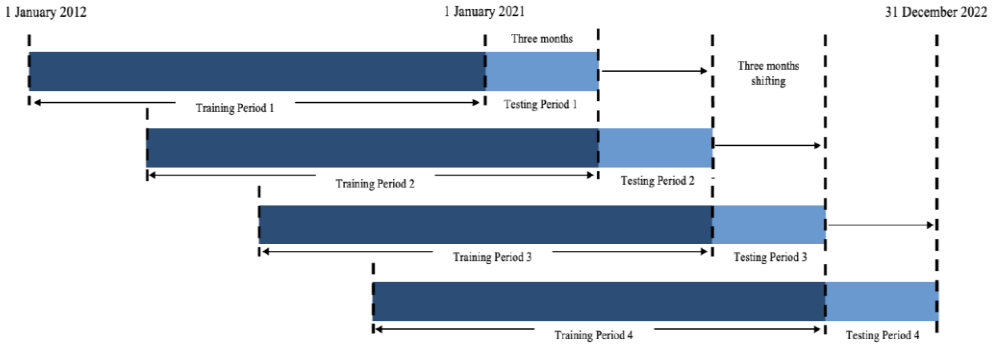


Figure 1. Sliding window for the dynamic arrangement of the training and testing periods.

Table 4. Training and testing dataset.

| Training dataset period | Sample size | Testing dataset period | Sample size |
|---|---|---|---|
| January 2012-December 2021 | 2,461 | January 2022- March 2022 | 61 |
| April 2012-March 2022 | 2,463 | April 2022- June 2022 | 59 |
| July 2012-June 2022 | 2,458 | July 2022-September 2022 | 63 |
| October 2012-September 2022 | 2,459 | October 2022-December 2022 | 60 |

## 3.3. Synthetic Data Generation Techniques

Based on Figures 2 and 3, the distributions of the stock price movement for the two predicted variables are slightly imbalanced. To overcome imbalanced class distribution due to relatively small sample sizes of TNB stock prices, this study generated synthetic data using SMOTE and GAN-SMOTE techniques to be used in the training of machine learning models.

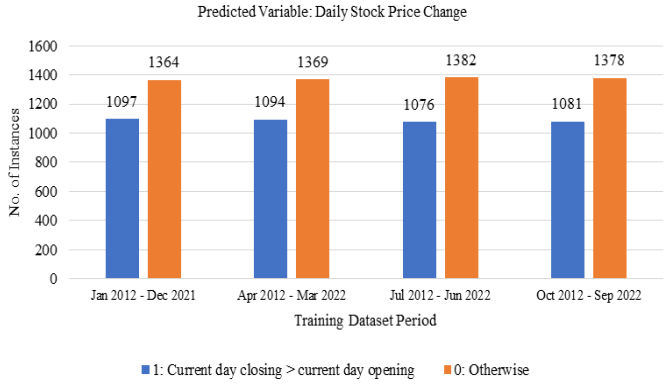


Figure 2. Class distribution of predicted variable: daily stock price change.

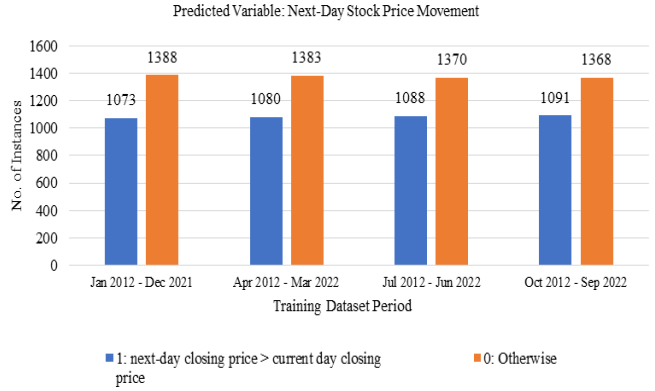


Figure 3. Class distribution of predicted variable: next-day stock price movement.

The SMOTE step was performed using the SMOTE class from the imblearn library. The SMOTE function from the imblearn.over_sampling module was applied to generate synthetic samples for the minority class, indicating that the minority class was oversampled. This approach aimed to balance the class distribution in the training sets, and the effectiveness of SMOTE technique in mitigating class imbalance was evaluated in the later stages of the research.

The GAN-SMOTE technique is achieved by combining GAN and SMOTE methods to generate synthetic data matching the minority class distribution. The GAN consisted of a generator and a discriminator. The generator generated synthetic samples, while the discriminator learned to distinguish between real and synthetic samples. The make_generator and make_discriminator functions were used to create the generator and discriminator, respectively The GAN technique generated synthetic samples to be combined with the original ones. Subsequently, SMOTE was applied to the combined dataset for further class imbalance correction and data augmentation. The approach aimed to improve resampled data balance and enhance machine learning model generalisation. The effectiveness of GAN-SMOTE in mitigating class imbalance was evaluated in the later stages of the research.

## 3.4. Modelling

This study used a classification approach to predict TNB stock price movement. Four machine learning models were performed in this study, including random forest, SVM, XGBoost and ANN.

Each machine learning model was trained on the training sets using real data and a combination of real and synthetic data generated by SMOTE and GAN-SMOTE. The input variables were chosen for the model training based on Pearson's correlation coefficient between each input variable and each target variable and the Backward Elimination technique. The performance of the machine learning models in predicting TNB stock price movements was evaluated using the predicted variables: daily stock price change and next-day stock price movement.

## 3.5. Experiment Settings

### 3.5.1. Experiment 1: Evaluating the Performance of Machine Learning Models without Synthetic Data Generation

In Experiment 1, the performance of machine learning models in predicting TNB stock price movement was assessed without utilising synthetic data generation techniques. The machine learning models were employed with hyperparameter tuning to achieve this objective. A Random Forest classifier was created using the random forest classifier class from the scikit-learn library, while an SVM classifier model was created using the SVC class. The XGBoost classifier model was created using the XGBClassifier class from the XGBoost library, and the ANN classifier model was created using the Sequential class from the Keras library. The primary hyperparameters of each classifier were specified and summarised in Tables 5 to 8 below.

Table 5. Range of Hyperparameter values Set for Random Forest.

| Parameter | Description | Values |
|---|---|---|
| n_estimators | Number of trees in the forest | [100, 200, 300, 500] |
| criterion | Criterion used for splitting | Default: GINI |
| max_depth | Maximum depth of the tree | [7, 8, 9, 10, 15, None] |
| min_samples_split | Minimum number of samples required to split an internal node | [2, 5, 10] |
| min_samples_leaf | Minimum number of samples required to be at a leaf node | [1, 2, 4] |
| max_features | Number of features to consider when looking for the best split | [‘auto’, ‘sqrt’] |
| max_leaf_nodes | Maximum number of leaf nodes | Default: None |
| bootstrap | Method for sampling data points (with or without replacement) | Default: True |
| oob_score | Whether to use out-of-bag samples to estimate the generalization accuracy | Default: False |
| n_jobs | Number of parallel jobs to run | Default: None |

Table 6. Range of hyperparameter values set for SVM.

| Parameter | Description | Values |
|---|---|---|
| C | Regularisation parameter | [0.1, 1.0, 10] |
| kernel | Kernel type used in algorithm | [Poly, rbf] |
| degree | Degree of polynomial kernel function (‘poly’) | [2, 3] |
| gamma | Kernel coefficient | 1/ (n_features * var_features) |
| coef() | Independent term in kernel function | 0.0 |
| shrinking | Whether to use the shrinking heuristic | Default: True |
| probability | Whether to enable probability estimates. This must be enabled prior to calling fit, will slow down that method as it internally trains a cross validation of the SVC for each probability | Default: False |
| tol | Tolerance for stopping criterion | Default: 1e-3 |
| class_weight | Weight of class | Default: 1 |
| random_state | Seed of the psudo random number generator | Default: None |

Table 7. Range of Hyperparameter Values Set for XGBoost.

| Parameter | Description | Values |
|---|---|---|
| booster | Type of booster to use | Default: gbtree |
| n_estimators | Number of trees to fit | [50, 100, 200] |
| max_depth | Maximum depth of the tree for base learners | [3, 5, 7, 10] |
| min_child_weight | Minimum sum of instance weight (hessian) needed in a child | Default: 1 |
| learning_rate | Learning rate | [0.1, 0.01] |
| gamma | Minimum loss reduction required to make a further partition on a leaf node of the tree | Default: 0 |
| subsample | Subsample ratio of the training instances | Default: 1 |
| colsample_bytree | Subsample ratio of columns when constructing each tree | Default: 1 |
| colsample_bylevel | Subsample ratio of columns for each split, in each level. | Default: 1 |
| reg_lambda | L2 regularization term on weights, increase this value will make model more conservative | Default: 1 |
| reg_alpha | L1 regularization term on weights, increase this value will make model more conservative | Default: 0 |
| scale_pos_weight | Control the balance of positive and negative weights, useful for unbalanced classes | Default: 1 |
| random_state | Seed of the random number generator | Default: 0 |
| n_jobs | Number of parallel threads used to run XGBoost | Default: 1 |
| objective | Specify the learning task and the corresponding learning objective or a custom objective function to be used | binary : logistic |

Table 8. Range of Hyperparameter Values Set for ANN.

| Parameter | Description | Values |
|---|---|---|
| hidden_layer_sizes | Tuple specifying the number of neurons in each hidden layer | [16, 32, 64] |
| activation | Activation function for the hidden layer | [relu, sigmoid] |
| solver | Algorithm for weight optimization | [adam, rmsprop] |
| alpha | L2 penalty (regularization term) parameter | Default: 0.0001 |
| batch_size | Size of mini-batch for stochastic optimizers | [16, 32] |
| epochs | Number of times the training set is iterated | [30, 50, 100] |
| learning_rate | Learning rate schedule for weight updates | Default: “constant” |
| learning_rate_init | Initial learning rate | Default: 0.001 |
| power_t | The exponent for inverse scaling learning rate | Default: 0.5 |
| max_iter | Maximum number of iterations | Default: 200 |
| shuffle | Whether to shuffle samples in each iteration | Default: True |
| random_state | Seed for random number generator | Default: None |
| tol | Tolerance for the optimization | Default: 0.0001 |
| verbose | Whether to print progress messages | Default: False |
| warm_start | When set to True, reuse the solution of the previous call to fit as initialization | Default: False |
| momentum | Momentum for gradient descent update | Default: 0.9 |
| nesterovs_momentum | Whether to use Nesterov's momentum | Default: True |
| early_stopping | Whether to use early stopping to terminate training when validation score is not improving | Default: False |
| validation_fraction | The proportion of training data to set aside as validation set for early stopping | Default: 0.1 |
| beta_1 | Exponential decay rate for estimates of first moment vector in adam, should be in [0, 1) | Default: 0.9 |
| beta_2 | Exponential decay rate for estimates of second moment vector in adam, should be in [0, 1) | Default: 0.999 |

The GridSearchCV class was used to identify the optimal set of hyperparameters by training and evaluating all possible combinations within a specified range of values for each parameter. The best-performing parameter combination on the validation set was selected, and a new classifier was instantiated and fitted to the training data using these optimised parameters.

#### 3.5.2. Experiment 2: Evaluating Performance of Machine Learning Models with Synthetic Data Generation Technique-SMOTE

In experiment 2, the objective was to assess the performance of machine learning models in predicting TNB stock price movement using the synthetic data generation technique, SMOTE. Each of the four machine learning models was evaluated with the SMOTE technique to achieve this objective. To evaluate the effectiveness of the SMOTE technique, the classifier models were trained on both the original training sets without SMOTE and the resampled training sets with SMOTE. By comparing the performance of these models, insights into the effectiveness of the SMOTE technique in addressing imbalanced class distribution issues were obtained. Like experiment 1, the GridSearchCV class was employed for each model to perform a grid search and identify the best hyperparameter values. The grid search involved defining a range of values for each hyperparameter and evaluating all possible combinations of these values. The optimal set of hyperparameters that achieved the best performance on the validation set was selected for each model.

#### 3.5.3. Experiment 3: Evaluating Performance of Machine Learning Models with Synthetic Data Generation Technique–GAN-SMOTE

In experiment 3, the objective was to evaluate the performance of machine learning models in predicting TNB stock price movement using the synthetic data generation technique called GAN-SMOTE, which combines the power of GANs and SMOTE to produce synthetic data that closely matches the minority class distribution. The experiment involved applying the GAN-SMOTE technique to the four machine learning models. The GAN-SMOTE algorithm generated synthetic samples for the minority class, addressing the issue of imbalanced class distribution and improving the models' ability to generalise. To evaluate the effectiveness of the GAN-SMOTE technique, the classifier models were trained on both the original training sets without GAN-SMOTE and the resampled training sets with GAN-SMOTE. By comparing the performance of these models, valuable insights into the effectiveness of the GAN-SMOTE technique in addressing imbalanced class distribution issues were obtained. Like the previous experiments, the GridSearchCV class was utilised for each model to perform a grid search and identify the best hyperparameter values.

#### 3.5.4. Model Evaluation

This study later compared the performance of machine learning models trained using real data and a combination of synthetic and real data using several evaluation metrics derived from a confusion matrix. A confusion matrix is a fundamental tool in the evaluation of machine learning models, particularly in classification tasks. It provides a concise summary of the model's predictive performance by organizing the outcomes into four categories: true positives, true negatives, false positives, and false negatives. These components allow for a detailed assessment of the model's accuracy, precision, recall, and F1-score, which were used to assess the model performance [17]. Table 9 shows the description of the confusion matrix, while Table 10 shows the evaluation metrics description and formula [17]. Additionally, the effectiveness of the models in predicting the TNB stock price movement based on all two predicted variables was evaluated. The selection of the best-performing model for each experiment is determined by identifying the model with the highest value for each evaluation metric.

Table 9. Description of confusion matrix.

| Value | Description |
|---|---|
| True Positive (TP) | Instances correctly predicted as positive by the model. |
| False Positive (FP) | Instances incorrectly predicted as positive by the model |
| True Negative (TN) | Instances correctly predicted as negative by the model. |
| False Negative (FN) | Instances incorrectly predicted as negative by the model |

Table 10. Evaluation metrics description and formula.

| Metric | Description | Formula |
|---|---|---|
| Accuracy | The proportion of correct predictions out of the total predictions made by the model. | $\frac{TP + TN}{TP + TN + FN + FP}$ |
| Precision | The ratio of true positive predictions to the total predicted positive instances, indicating the model's ability to correctly identify positive cases. | $\frac{TP}{TP + FP}$ |
| Recall | The ratio of true positive predictions to the total actual positive instances, representing the model's ability to capture all positive cases. | $\frac{TP}{TP + FN}$ |
| Specificity | The ratio of true negative predictions to the total actual negative instances, measuring the model's ability to correctly identify negative cases. | $\frac{TN}{TN + FP}$ |
| F1-Score | The harmonic mean of precision and recall, providing a balanced measure of a model's performance across both precision and recall metrics | $\frac{2 * (Precision * Recall)}{Precision + Recall}$ |

## 4. Results and Discussions

### 4.1. Feature Engineering-Significant Input Variables

Based on Pearson's Correlation Coefficient and Backward Elimination techniques in choosing the significant input variables that influenced the movement of TNB stock price, it was observed that for both

predicted variables, the selected input variables were the same: SMA signal, volatility rate, close price, open price, stochastic %D, and MACD Signal. The study identified key variables for predicting stock price movement, including technical indicators like SMA signal, stochastic %D signal, and MACD signal, which assist traders in analysing market trends. The volatility rate was also found to be important in capturing market fluctuations. Additionally, Close and Open Prices were significant factors, reflecting the influence of market trends and sentiment on stock prices.

## 4.2. Synthetic Data Generation Techniques

The application of SMOTE successfully balanced the class distribution of the predicted variable in the training datasets, as shown in Figures 4 and 5. This achievement significantly enhanced the representativeness and robustness of the dataset, laying a solid foundation for training predictive models. Consequently, this balanced dataset played a pivotal role in improving the model's performance in accurately forecasting the changes in stock prices based on the provided conditions.

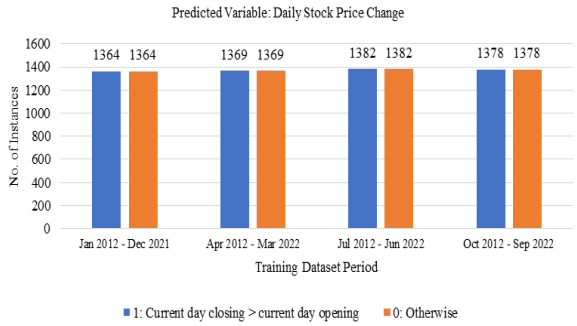


Figure 4. Class distribution of daily stock price change after SMOTE.

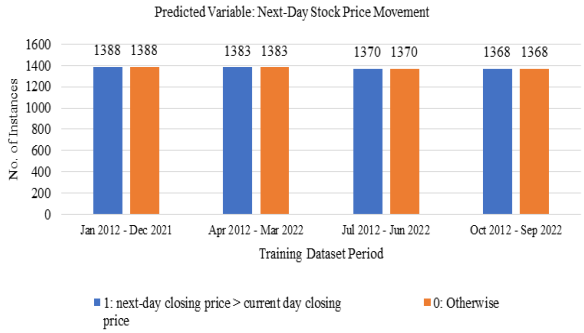


Figure 5. Class distribution of next-day stock price movement after SMOTE.

In addition, applying the GAN technique substantially increased the number of samples for both predicted variables. Subsequently, the SMOTE algorithm was applied to the GAN-generated samples to tackle the class imbalance issues, as shown in Figures 6 and 7.

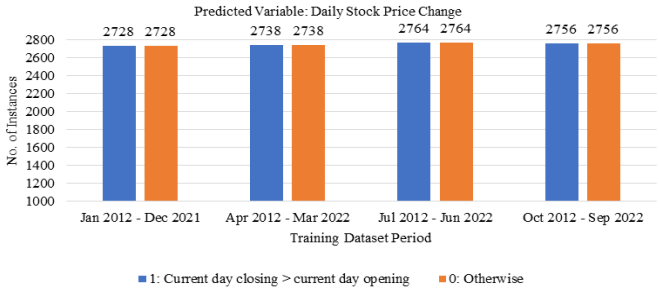


Figure 6. Class distribution of daily stock price change after GAN-SMOTE.

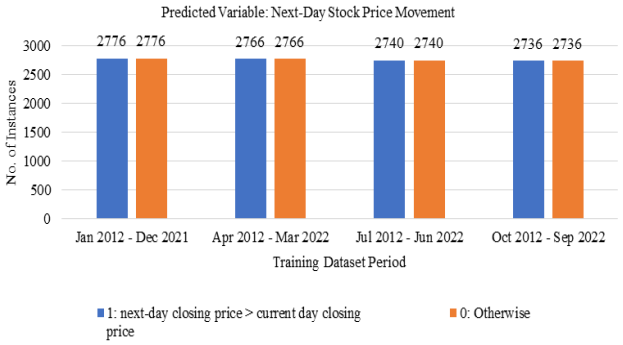


Figure 7. Class distribution of next-day stock price movement after GAN-SMOTE.

GAN-SMOTE approach harnessed the strengths of both the GAN and SMOTE techniques. It leveraged the GAN's ability to generate diverse and realistic synthetic samples while capitalising on SMOTE's capability to oversample the minority class. As a result, the resulting dataset achieved an equal representation of the minority and majority classes.

## 4.3. Experiment 1

This experiment aimed to assess the predictive capabilities of machine learning models without using synthetic data generation. As presented in Table 11, multiple metrics were used to evaluate the models' performance in predicting the Daily Stock Price Change.

Table 11. Performance evaluation of predicting daily stock price change.

| A. | Experiment 1 | | | | Experiment 2 | | | | Experiment 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | **XGBoost** | **Random forest** | **SVM** | **ANN** | **SMOTE-XGBoost** | **SMOTE-random forest** | **SMOTE-SVM** | **SMOTE-ANN** | **GAN-SMOTE-XGBoost** | **GAN-SMOTE-random forest** | **GAN-SMOTE-SVM** | **GAN-SMOTE-ANN** |
| **Accuracy** | 82% | 72% | 83% | 78% | 83% | 74% | 83% | 93% | 77% | 73% | 80% | 78% |
| **F1-score** | 77% | 63% | 81% | 74% | 82% | 71% | 81% | 92% | 74% | 70% | 77% | 73% |
| **Precision** | 84% | 69% | 83% | 79% | 81% | 72% | 82% | 90% | 76% | 70% | 77% | 83% |
| **Recall** | 74% | 58% | 79% | 70% | 84% | 72% | 80% | 94% | 75% | 73% | 78% | 65% |
| **Specificity** | 89% | 81% | 85% | 86% | 75% | 76% | 94% | 92% | 79% | 74% | 81% | 89% |
| **Best model** | **SVM** | | | | **SMOTE-ANN** | | | | **GAN-SMOTE-SVM** | | | |

These five-evaluation metrics indicate that SVM performed well in predicting the Daily Stock Price Change, achieving the highest accuracy, F1-Score, and recall among the models. XGBoost emerged as the next-best performing model, achieving the highest precision and specificity scores in predicting the Daily Stock Price Change. Random Forest showed relatively lower performance, while ANN exhibited intermediate performance.

The superior performance of SVM and XGBoost in predicting the daily stock price change could be attributed to their respective strengths and characteristics. SVM's ability to handle complex relationships and non-linear data using kernel functions allows it to effectively separate classes in the feature space, resulting in high accuracy, precision, recall, and specificity scores. Additionally, SVM's capability to find decision boundaries in high-dimensional spaces contributes to its strong predictive performance.

Based on these findings, it can be concluded that SVM is more suitable for predicting the Daily Stock Price Change on the given dataset due to its superior performance across multiple evaluation metrics. Its ability to handle complex relationships, leverage boosting techniques, and find decision boundaries in high-dimensional spaces contributes to its predictive accuracy and reliability in this context.

In the next-day stock price movement prediction, as indicated in Table 12, the models achieved high accuracy scores ranging from 77% to 79%, indicating overall correctness in predicting stock price movements. Random Forest attained the highest accuracy, followed by XGBoost, ANN, and SVM. The high accuracy of random forest, XGBoost and ANN indicates that the models have a relatively low rate of incorrect predictions, providing valuable insights into the potential direction of the next-day stock price movement.

Table 12. Performance evaluation of predicting next-day stock price movement.

| **B.** | **Experiment 1** | | | | **Experiment 2** | | | | **Experiment 3** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | XGBoost | Random forest | SVM | ANN | SMOTE-XGBoost | SMOTE-Random forest | SMOTE-SVM | SMOTE-ANN | GAN-SMOTE-XGBoost | GAN-SMOTE-Random forest | GAN-SMOTE-SVM | GAN-SMOTE-ANN |
| Accuracy | 78% | 79% | 77% | 78% | 80% | 78% | 79% | 81% | 80% | 76% | 80% | 79% |
| F1-Score | 74% | 74% | 77% | 77% | 78% | 76% | 77% | 79% | 78% | 73% | 79% | 75% |
| Precision | 84% | 83% | 76% | 78% | 83% | 81% | 80% | 88% | 83% | 81% | 77% | 86% |
| Recall | 67% | 70% | 77% | 76% | 75% | 71% | 67% | 73% | 73% | 67% | 78% | 70% |
| Specificity | 88% | 87% | 77% | 78% | 84% | 85% | 92% | 89% | 86% | 84% | 80% | 89% |
| Best Model | **XGBoost** | | | | **SMOTE-ANN** | | | | **GAN-SMOTE-SVM** | | | |

All models showed similar F1 scores, signifying a balanced trade-off between correctly identifying positive cases and minimising false predictions. This indicates that these models can provide reliable predictions of stock price movements while minimising the chances of false predictions, allowing traders and investors to make more informed decisions. XGBoost exhibited the highest precision, closely followed by Random Forest, while SVM had the lowest precision. In stock price prediction, a lower precision score means a higher likelihood of false alarms, which means predicting a stock price increase when it may not occur.

SVM achieved the highest recall, with ANN closely behind. This indicated that SVM and ANN were better at accurately capturing positive cases than XGBoost and Random Forest. They have a higher likelihood of correctly capturing instances of stock price increases. On the other hand, the lower recall scores for XGBoost and Random Forest suggest a comparatively lower ability to capture positive cases accurately. Meanwhile, XGBoost had the highest specificity, followed by Random Forest, showing the models performed relatively better in accurately identifying instances of stock price decreases.

Considering the overall performance across multiple evaluation metrics, XGBoost is the best-performing model as it consistently demonstrated strong accuracy, precision, recall, and specificity performance, followed by random forest. However, it is important to note that XGBoost had a relatively lower recall than SVM and identifying positive cases, SVM and ANN may offer a more balanced performance. It would be beneficial to weigh the trade-offs between precision and recall and accurately assess the relative importance of capturing positive cases. Additionally, exploring ways to enhance ANN. Therefore, when prioritising accurately model as it consistently demonstrated strong accuracy, precision, recall, and specificity performance, followed by Random Forest. However, it is important to note that XGBoost had a relatively lower recall than SVM and ANN. Therefore, when prioritising accurately identifying positive cases, SVM and ANN may offer a more balanced performance. It would be beneficial to weigh the trade-offs between precision and recall and accurately assess the relative importance of capturing positive cases. Additionally, exploring ways to enhance the recall performance of XGBoost or considering an ensemble approach using multiple models might be worth investigating.

In summary, experiment 1 evaluated the performance of machine learning models in predicting the daily stock price change and next-day stock price movement without incorporating synthetic data generation techniques. In predicting the daily stock price change, SVM is the best-performing model achieving an accuracy of 83%, an F1-Score of 81%, a precision of 83%, a recall of 79%, and a specificity of 85%. For predicting the next-day stock price movement,

XGBoost emerged as the best-performing model, with an accuracy of 78%, an F1-Score of 74%, a precision of 84%, a recall of 67%, and a specificity of 88%.

### 4.4. Experiment 2

This experiment assessed the impact of incorporating balanced datasets on predictive model performances. SMOTE-ANN emerged as the top-performing model for predicting both daily stock price change and next-day stock price movement, achieving the highest scores across all evaluation metrics, as shown in Table 11 and Table 12. The superior performance of SMOTE-ANN could be attributed to its ability to capture complex non-linear relationships in the data using deep learning techniques. With multiple interconnected neurons, ANN has the flexibility and capacity to learn intricate patterns and representations, resulting in accurate TNB stock price movement predictions.

SMOTE-XGBoost and SMOTE-SVM also demonstrated strong performance, benefiting from boosting and support vector machine algorithms. However, SMOTE-Random forest exhibited relatively lower performance, which can be attributed to its limitation in capturing complex relationships within the data. Despite using an ensemble approach with multiple decision trees, which can reduce overfitting and improve generalisation, it struggled to capture intricate nonlinear relationships present in financial time series data, such as stock price movements. It is worth noting that the high accuracy and performance of SMOTE-ANN could be attributed to the SMOTE technique applied to address the class imbalance. SMOTE helps to generate synthetic samples of the minority class, boosting the representation of positive cases in the training data and improving the model's ability to capture their patterns. Overall, the results suggest that SMOTE-ANN, with its strong performance in accuracy, F1-Score, precision, recall, and specificity, is the most suitable model for predicting both the Daily Stock Price Change and Next-Day Stock Price Movement in this experiment.

### 4.5. Experiment 3

This experiment aimed to assess the impact of incorporating GAN-SMOTE-generated balanced samples on predictive model performance and to determine the extent of improvement achieved by these augmented samples. In predicting daily stock price change, GAN-SMOTE-SVM exhibited superior performance, achieving high accuracy, precision, and recall while effectively capturing positive and negative cases. GAN-SMOTE-XGBoost improved accuracy and precision over time through its boosting technique, and GAN-SMOTE-ANN excelled in capturing complex patterns and relationships in the datasets. However, GAN-SMOTE-Random forest showed relatively lower performance, possibly due to challenges introduced by the GAN-SMOTE technique and the model's limitation, as discussed in experiment 2, which resulted in lower accuracy and precision scores. Based on the overall evaluation metric, GAN-SMOTE SVM demonstrates the best overall performance among the listed models for predicting the next-day stock price movement. GAN-SMOTE-SVM achieves the highest accuracy, tied with GAN-SMOTE-XGBoost. Although GAN-SMOTE-ANN achieves the highest specificity score at 89%, GAN-SMOTE-SVM still demonstrates strong performance with a specificity score of 80% therefore, considering the overall performance based on the provided metrics, GAN-SMOTE-SVM stands out as the model with the best performance for predicting the next-day stock price movement.

### 4.6. Experiment 1 vs Experiment 2 vs Experiment 3

Tables 5 and 6 highlight the superior performance of the best model from Experiment 2 compared to experiment 1 and Experiment 3 in predicting both variables. Incorporating the SMOTE technique in experiment 2 improves the model's ability to address the class imbalance and capture underlying data patterns. Among the models in experiment 2, SMOTE-ANN stands out with the highest accuracy, F1-Score, precision, recall, and specificity values. These results demonstrate the effectiveness of SMOTE-ANN in capturing intricate patterns associated with TNB stock price movement. The lower performance of GAN-SMOTE in experiment 3 may be due to increased training complexity, potential inaccuracies in synthetic samples, and the need for fine-tuning. Despite these challenges, GAN-SMOTE offers benefits like diversity and realism in samples, particularly for imbalanced datasets. However, in this study, the SMOTE technique in experiment 2 outperformed the GAN-SMOTE technique in Experiment 3. The combination of SMOTE-ANN holds promise for predicting Daily Stock Price Change and next-day stock price movement, achieving an impressive 93% accuracy in predicting daily stock price change and a respectable 82% accuracy in predicting next-day stock price movement.

It is crucial to distinguish the accuracy of predicting daily stock price change from next-day stock price movement, as these tasks have distinct characteristics and conditions. When predicting the daily stock price change, models analyse intraday shifts influenced by market sentiment, news releases, and trading activity, comparing closing and opening prices. This allows them to identify patterns and accurately determine price exceedance or shortfall. However, predicting the next-day stock price movement involves forecasting price changes from the current day to the subsequent day, considering unpredictable factors like after-hours news releases, earnings announcements, and economic reports. These uncertainties introduce challenges and

make precise next-day price predictions more difficult.

## 5. Conclusions and Future Works

Overall, this study successfully identified significant factors influencing TNB stock price movement, overcame imbalanced class distribution due to small sample sizes of TNB stock price through synthetic data generation, and determined which model performs best in TNB stock price movement prediction.

This study's findings offer valuable insights with practical applications. For investors, the identified factors influencing TNB stock price movements provide essential indicators to assess risks and returns, enabling them to manage their portfolios effectively. TNB can leverage the insights to inform corporate strategy and financial planning, enhancing investor confidence and market value. Policymakers can use accurate stock price predictions to understand the economy, formulate policies, and foster economic growth. Additionally, collaboration with financial institutions facilitates the implementation of these findings into decision support systems, trading algorithms, and risk management, adding value to their services.

While this study contributes to TNB stock price movement prediction, there are limitations and potential future research directions. The study only considers the stock market, technical, financial, and economic variables. Future research should include additional variables like market sentiment, macroeconomic indicators, industry-specific factors, and news sentiment to understand TNB stock price dynamics better. While SMOTE-ANN has demonstrated significant performance, further exploration and evaluation of advanced machine learning techniques are warranted. Future research should consider delving into deep learning algorithms, RNNs, and hybrid models to enhance predictive accuracy and capture intricate and complex patterns within stock price data. Furthermore, this study focused on SMOTE and GAN-SMOTE techniques, leaving room for future investigations into more advanced synthetic data generation methods such as WGAN and TimeGANs. Exploring these advanced techniques could enhance the quality and diversity of generated data, further improving the performance of predictive models.

## Acknowledgement

Our acknowledgements go to the research and community service of the College of Computing, Informatics, and Mathematics, Universiti Teknologi MARA, Shah Alam, Selangor.

## References

[1] Asuero A., Sayago A., and González G., "The correlation coefficient: An overview," *Critical Reviews in Analytical Chemistry*, vol. 36, no. 1, pp. 41-59, 2006. DOI: 10.1080/10408340500526766

[2] Bahrami A., Shamsuddin A., and Uylangco K., "Out-of-Sample Stock Return Predictability in Emerging Markets," *Accounting and Finance*, vol. 58, no. 3, pp. 727-750, 2018. https://doi.org/10.1111/acfi.12234

[3] Bathla G., "Stock Price Prediction using LSTM and SVR," *in Proccedings of the 6th International Conference on Parallel, Distributed and Grid Computing*, Waknaghat, pp. 211-214, 2020. doi: 10.1109/PDGC50313.2020.9315800

[4] Bekaert G. and Harvey C., "Emerging Markets Finance," *Journal of Empirical Finance*, vol. 10, no 1-2, pp. 3-55, 2003. https://doi.org/10.1016/S0927-5398(02)00054-3

[5] Bozic J. and Djordje B,. "Financial Time Series Forecasting Using Hybrid Wavelet-Neural Model," *The International Arab Journal of Information Technology*, vol. 15, no. 1, pp. 50-57, 2018.

[6] Carvajal-Patiño D. and Ramos-Pollán R., "Synthetic Data Generation with Deep Generative Models to Enhance Predictive Tasks in Trading Strategies," *Research in International Business and Finance*, vol. 62, pp. 101747, 2022. https://doi.org/10.1016/j.ribaf.2022.101747

[7] Chaajer P., Shah M., and Kshirsagar A., "The Applications of Artificial Neural Networks, Support Vector Machines, and Long-Short Term Memory for Stock Market Prediction," *Decision Analytics Journal*, vol. 2, pp. 100015, 2022. https://doi.org/10.1016/j.dajour.2021.100015

[8] Chatzis S., Siakoulis V., Petropoulos A., Stavroulakis E., and Vlachogiannakis N., "Forecasting Stock Market Crisis Events Using Deep and Statistical Machine Learning Techniques," *Expert Systems with Applications*, vol. 112, pp. 353-371, 2018. https://doi.org/10.1016/j.eswa.2018.06.032

[9] Chopra R. and Sharma G., "Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda," *Journal of Risk and Financial Management*, vol. 14, no. 11, pp. 526, 2021. https://doi.org/10.3390/jrfm14110526

[10] Das T., Khan A., and Saha G., "Classification of Imbalanced Big Data Using SMOTE with Rough Random Forest," *International Journal of Engineering and Advanced Technology*, vol. 9, pp. 5174-5184, 2019. DOI: 10.35940/ijeat.B4096.129219

[11] Deng S., Zhu Y., Huang X., Duan S., and Fu Z., "High-Frequency Direction Forecasting of the Futures Market Using a Machine-Learning-Based Method," *Future Internet*, vol. 14, no. 6, pp. 180, 2022. https://doi.org/10.3390/fi14060180

[12] Dhafer A., Nor F., Hashim W., Shah N., Bin-

Khairi K., and Alkawsi G., "A NARX Neural Network Model to Predict One-Day Ahead Movement of the Stock Market Index of Malaysia," *in Proccedings of the 2nd International Conference on Artificial Intelligence and Data Sciences*, Ipoh, pp. 1-7, 2021. DOI: 10.1109/AiDAS53897.2021.9574394

[13] Eapen J., Bein D., and Verma A., "Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction," *in Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference*, Las Vegas, pp. 0264-0270, 2019. doi: 10.1109/CCWC.2019.8666592

[14] Ghasemieh A. and Kashef R., "Deep Learning Vs. Machine Learning in Predicting the Future Trend of Stock Market Prices," *in Proccedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Melbourne, pp. 3429-3435, 2021. DOI: 10.1109/SMC52423.2021.9658938

[15] Kumar P., Bhatnagar R., Gaur K., and Bhatnagar A., "Classification of Imbalanced Data: Review of Methods and Applications," *in Proccedings of the IOP Conference Series: Materials Science and Engineering*, Sanya, pp. 012077, 2021. doi:10.1088/1757-899X/1099/1/012077

[16] Ling L. and Belaidan S., "Stock Market Price Movement Forecasting on Bursa Malaysia using Machine Learning Approach," *in Proceedings of the 14th International Conference on Developments in eSystems Engineering*, Sharjah, pp. 102-108, 2021. https://doi.org/10.1109/DeSE54285.2021.9719534

[17] Nabipour M., Nayyeri P., Jabani H., Shahab S., and Mosavi A., "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data: A Comparative Analysis," *IEEE Access*, vol. 8, pp. 150199-150212, 2020. DOI: 10.1109/ACCESS.2020.3015966

[18] Ravikumar S. and Saraf P., "Prediction of Stock Prices Using Machine Learning Regression, Classification Algorithms," *in Proccedings of the International Conference for Emerging Technology*, Belgaum, pp. 1-5, 2020. doi: 10.1109/INCET49848.2020.9154061.

[19] S&P Capital IQ. (2022). Tenaga Nasional Berhad: Public Company Profile, Last Visited, 2024.

[20] Sarumpaet N., Indwiarti., and Rohmawati A., "Performance Comparison Between Support Vector Regression and Long Short-Term Memory for Prediction of Stock Market," *in Proccedings of the 10th International Conference on Information and Communication Technology*, Bandung, pp. 168-173. 2022. DOI: 10.1109/ICoICT55009.2022.9914839

[21] Tenaga Nasional Berhad. (2022). About TNB: Corporate Profile. https://www.tnb.com.my/about-tnb/corporate-profile/, Last Visited, 2024.

[22] Zhang K., Zhong G., Dong J., Wang S., and Wang Y., "Stock Market Prediction Based on Generative Adversarial Network," *Procedia Computer Science*, vol. 147, pp. 400-406, 2019. https://doi.org/10.1016/j.procs.2019.01.256

[23] Zhao X., "The Prediction of Apple Inc. Stock Price with Machine Learning Models," *in Proccedings of the 3rd International Conference on Applied Machine Learning*, Changsha, pp. 222-225, 2021.

**Nur Aliah Syahmina Mohd Nazarudin** is an Asset Management Analyst in the New Energy Division at Tenaga Nasional Berhad, Malaysia. She earned her Master of Data Science from Universiti Teknologi MARA Shah Alam, Malaysia, in 2023 supervised by Dr Ruhaila Maskat. Her Bachelor of Electrical Engineering was from McGill University, Canada, in 2019. Her research interests include investment and financial data analysis, with a focus on applying machine learning techniques to financial markets. Additionally, she is pursuing the Financial Modeling and Valuation Analyst (FMVA®) certification through the Corporate Finance Institute (CFI) to further develop her expertise in financial modeling, valuation, and data-driven decision-making.

**Nor Hapiza Mohd Ariffin** is an Assistant Professor at the MIS Department in the Faculty of Business at Sohar University, Oman. She obtained her Ph.D. in Science and System Management from Universiti Kebangsaan Malaysia in 2010. She is mostly interested in studying Blockchain technology, Information systems, and Digital Business. In addition, she possesses two professional certificates from Microsoft Azure AI and Data Fundamental and is a certified associate developer in Blockchain.

**Ruhaila Maskat**, currently serving as a senior lecturer at the College of Computing, Informatics and Mathematics, Universiti Teknologi MARA Shah Alam, Malaysia, attained her Ph.D. in Computer Science from the University of Manchester, United Kingdom, in 2016. Initially focused on Pay-As-You-Go dataspaces, her research interests transitioned to Data Science, leading to her role as an EMC Dell Data Associate. Additionally, she holds four professional certifications from RapidMiner in machine learning and data engineering. Notably, Dr. Maskat received the Kaggle BIPOC grant.