INTERNATIONAL MULTI-AWARD WINNING INSTITUTION FOR SUSTAINABILITY

CSCI 4304: MACHINE LEARNING
SEM 1 25/26

Group G

Group Report

A Novel Particle Swarm–Optimized LSTM Hybrid Model for Bursa Stock Index Forecasting

Lecturer: Assoc. Prof. Ts. Dr. Amelia Ritahani binti Ismail

Section: 2

GROUP NAME:	Group G	Contributions
MEMBER NAMES AND MATRIC NUMBERS	1. Muhamad Adam Muhaimin bin Shahbani (2319255)	100%
	2. Muhammad Bashyar bin Kamarul Anuar (2313927)	100%
	3. Mohammad Hakim bin Mahyudin (2313673)	100%
	4. Arif Haikal bin Shamsul Kamarul (2318237)	100%

1.0 Abstract

Stock market prediction is a difficult task due to the non-linear, noisy, and chaotic nature of financial time-series data. Traditional forecasting methods always struggle in capturing long-term dependencies. Meanwhile, Long Short-Term Memory (LSTM) networks is one of the modern deep learning model that rely on manual selection of hyperparameters which is inefficient. This project will implement a hybrid forecasting model that integrates Particle Swarm Optimization (PSO) with LSTM networks to predict the trends of the Bursa Malaysia stock index. The primary objective of this project is to utilize PSO to obtain and optimize LSTM hyperparameters such as the number of hidden layers and learning rates to improve the model's predictive accuracy and convergence speed. We predict that the PSO-LSTM hybrid will achieve superior reliability and lower error rates (RMSE and MAPE) compared to the baseline models.

2.0 Introduction

The stock market is a key tool for investing and creating wealth as well as a gauge of the state of the economy.. However, stock price movements often have high volatility, uncertainty, and non-linearity that make accurate forecasting difficult. For investors and financial analysts, the ability to predict market trends effectively is important for optimized trading timing and maximizing returns on investment.

In recent decades, machine learning techniques such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM) have been widely adopted to address financial forecasting. Furthermore, the LSTM network has become a powerful tool because of its ability to retain information over long periods through memory cells and gate mechanisms. However, there is a limitation to this model. Their performance is too sensitive to hyperparameter settings. These parameters are determined by using trial and error or user experience which does not guarantee the optimal solution and can lead to overfitting.

To solve these problems, this project proposes a hybrid machine learning model that integrates Particle Swarm Optimization (PSO) with Long Short-Term Memory (LSTM) networks. By utilizing PSO's evolutionary search capabilities, the model will find the best hyperparameters for the LSTM network. This hybrid aims to remove the trial-and-error process of model configuration to improve prediction accuracy and computational efficiency.

For this project, we have selected the historical daily trading data of the FTSE Malaysia KLCI. The reason we chose this dataset because the most accurate indicator of the Malaysian stock market which is a trustworthy stand-in for the nation's more general economic patterns. Furthermore, it captures a wide range of market phases from 2019 to 2025. Also, the dataset contains high-quality daily OHLC (Open, High, Low, Close) and volume data allows for rich feature engineering. From this dataset, we plan to generate reliable predictions of the future daily closing prices of the KLCI index, extract long-term dependencies and recurrent patterns buried within the 2,451 days of trading data, optimize set of hyperparameter configurations, and compare the performance of 4 models (LSTM, Hybrid PSO-LSTM, Naïve Bayes, Decision Tree).

3.0 Literature Review

Long Short-Term Memory (LSTM) networks have become a dominant methodology in financial time series forecasting due to its specialized memory cells and gating mechanisms, which effectively mitigate the vanishing gradient problem and capture long-term temporal dependencies in stock market data (Fischer & Krauss, 2018; Ji et al., 2021). Fischer and Krauss (2018) stated that LSTM networks significantly outperform memory-free classifiers such as random forests, deep neural networks, and logistic regression when predicting directional movements of S&P 500 constituent stocks from 1992 to 2015, achieving superior risk-adjusted returns by identifying stocks with high volatility and short-term reversal patterns. Despite their architectural advantages, the predictive efficacy of standard LSTMs is very effective in the precise selection of hyperparameters such as the learning rate, look-back window, and number of hidden units where traditional trial-and-error tuning often results in sub-optimal convergence and overfitting . To overcome these optimization challenges, we applying the integration of Particle Swarm Optimization (PSO), which is a metaheuristic algorithm inspired by swarm intelligence, and demonstrate that by automating the search for optimal hyperparameter configurations within high-dimensional spaces.The hybrid PSO-LSTM model also significantly reduces manual tuning efforts and enhances model generalization, consistently produce a superior performance metrics such as lower Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and higher coefficients of determination (R^2) compared to standard LSTM, support vector regression, and standard PSO-LSTM across multiple major stock indices .

The selection of the PSO-LSTM hybrid model for predicting the Bursa Malaysia stock index is further justified when contrasted with traditional machine learning baselines such as Naïve Bayes and Decision Trees. Due to its conditional independence assumption, Naïve Bayes is fundamentally not suitable for time-series forecasting because it treat each feature that contributes to classification independently given the class label a property that fails to account for the sequential momentum, volatility clustering, and autocorrelation that define market trends where successive price values are inherently dependent through temporal dynamics (Ashari et al., 2013). Similarly, Decision Trees and their ensemble variants such as Random Forests often struggle with temporal continuity,they rely on static partitioning of data into regions with piecewise constant predictions and generally lack the ability to predict trends beyond the range of values seen during training which become a critical limitation known as extrapolation failure where any input outside the observed training range will simply return the prediction of the nearest boundary region that making them unable to extend trends or capture values beyond the minimum or maximum target values encountered during training. Therefore, the PSO-LSTM model is the preferred approach for this project, as it uniquely combines the capacity to model complex, nonlinear sequential dependencies with an evolutionary optimization mechanism that ensures more robust and accurate predictions than static classification or regression alternatives.

4.0 Data Analysis

Firstly,Exploratory Data Analysis was conducted in order to check data quality and guide the cleaning logic . Initial inspections by using the `info()` function to reveal that raw dataset treat key financial metrics treated as object (string) types rather than numerical floats due to special characters and abbreviations. The `head()` function was also utilized to visually verify the data structure and identifying that the original file was sorted in reverse chronological order (Newest to Oldest), which would have disrupted time-series forecasting.

After that,The data preprocessing stage was followed up for converting raw historical records into a format that compatible with the LSTM neural network. The "Date" column was converted to datetime objects, and the entire dataset was sorted in ascending chronological order to ensure the model learned time-dependent patterns correctly. Next,non-numeric format was remove by stripping commas from the "Price," "High," "Low," and "Open" columns, removing the percentage sign from "Change %," and parsing the "Vol." (Volume) column to convert "K" (thousands) and "M" (millions) suffixes into actual integers number.

Finally,for the modeling phase, the target "Price" variable was normalized using a `MinMaxScaler` to compress values into a [0, 1] range to enhance training stability. Beside that,a sliding window algorithm was also applied to structure the data into sequences with a 20-day look-back period, which reshaped into 3D arrays and split into 70% training and 30% testing sets.

5.0 Experimental Setup

5.1 Computational Environment

We conducted the experiments using Google Colab cloud environment, this is to make sure reproducibility, and we can also leverage Google T4 GPU acceleration for deep learning tasks. We used the following core libraries to implement the model:

- **Data Manipulation:** pandas and numpy for handling the time-series dataset.
- **Machine Learning:** sklearn for Naive Bayes (GaussianNB), Decision Tree (DecisionTreeRegressor), and performance metrics (accuracy_score, classification_report).
- **Deep Learning:** tensorflow.keras for constructing and training the LSTM networks .
- **Visualization:** matplotlib.pyplot and seaborn for plotting loss curves, confusion matrices, and trend lines.

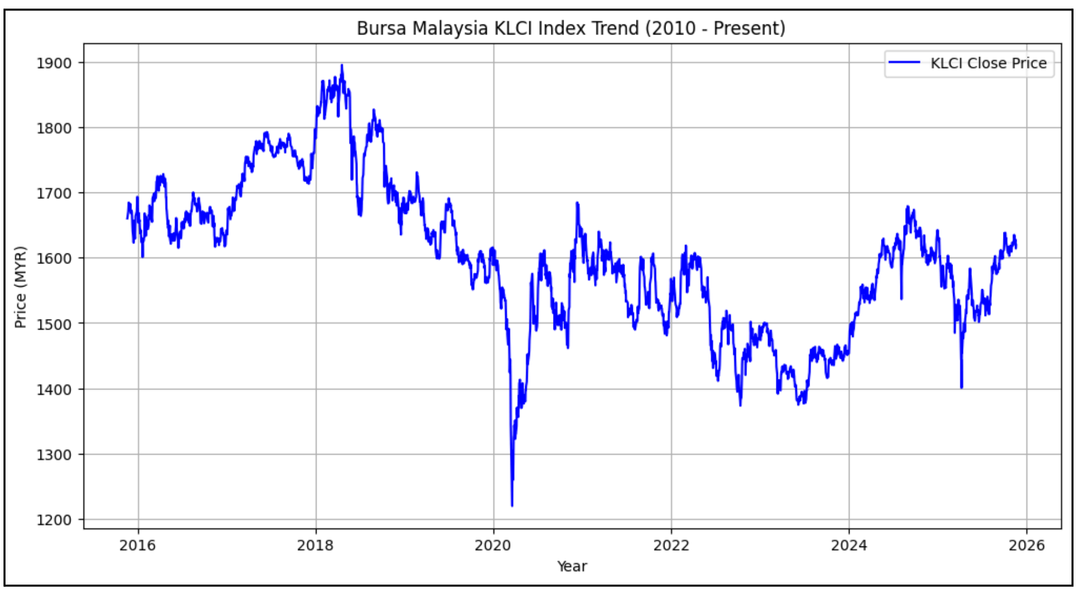
5.2 Data Collection and Preprocessing

For the study, we chose FTSE Malaysia KLCI, which is stock index that tracks 30 largest companies listed on Bursa. The dataset period covers the period from 2010 to present. It consists of 2,451 daily records.

5.2.1 Common Preprocessing

For all models, raw data cleaning was performed:

- **Cleaning:** The commas from the price columns are removed, and volume suffixes (like K for thousands and M for millions) were converted to numeric float values.
- **Date Parsing:** The Date column was converted to datetime objects and sorted chronologically.



Above is the graph of Bursa KLCI Index daily prices after cleaning.

5.2.2 Feature Engineering by Model Type

Because we are using 4 models to compare which performs the best with stock price prediction, some of the models solve different problems than the others.

A. Regression Models (LSTMs and Decision Tree)
    - **Target:** Exact Closing Price (MYR)
    - **Scaling:** We used Min-Max Scaling to normalized the data to [0,1].

- **Windowing**: To capture sequential patterns, we used 20 days sliding window.
- **Split:** 70% Training / 30% Testing

B. Classification (Naive Bayes)

- **Target:** It uses binary Classification.If tomorrow's price is higher than today, we assigned the value of 1, if it is not then assign 0.
- **Input Features:** Instead of raw prices, we calculate market state indicators
    - Open-Close: Daily Price movement range
    - High-low: Daily volatility
    - Vol_Increase: Binary flag (1 if volume increase vs yesterday, if not 0)
- **Split:** 80% Training and 20% Testing (Sequential split).

5. 3 Model Architectures

As discussed, we developed four models that could help us benchmark performance across both forecasting precision and directional accuracy of the stock market index.

5.3.1 Baseline 1: Naive Bayes (Trend Classifier)

We implemented Gaussian Naive Bayes classifier to establish a probabilistic baseline to predict market direction. The model assumes that the features follow normal distribution.

- Algorithm: sklearn.naive_bayes.GaussianNB
- Objective: Maximize classification accuracy (Up vs. Down)

5.3.2 Baseline 2: Decision Tree Regression

A Decision Tree Regressor was implemented for the non-deep learning baseline for price prediction and the random state was fixed at 42 to ensure results are deterministic.

5.3.3 Baseline 3: Standard LSTM

Before implementing PSO-Optimized LSTM, we implemented the standard LSTM architecture first. It consists of two LSTM layers (100 and 20 neurons) and followed by a Dense output layer. Adam optimizer was used with the learning rate of 0.001 and trained for 64 epochs.

5.3.4 Proposed model: PSO-Optimized LSTM

This is the key point of the study. In order to optimize LSTM, Particle Swarm Optimization (PSO) was used to search for the best hyperparameters.

- Swarm settings: 3 particles, 3 iterations
- Search space: Nodes (10-200), Epochs (10-100), Learning rate (0.001 - 0.01).
- Resulting architecture: The PSO converged on a deeper architecture (68 and 147 neurons) trained for 92 epochs.

5.4 Experimental Configuration Summary

The table below summarizes the final settings used for comparative analysis. Note that the hyperparameters of the PSO-Optimized LSTM model was chosen by the PSO.

Feature	Standard LSTM	PSO-Optimized	Decision Tree	Naive Bayes

		LSTM		
Task Type	Regression (Price)	Regression (Price)	Regression (Price)	Classification (Direction)
Input	20-Day Window	20-Day Window	20-Day Window	Engineered Features
Train/Test Split	70% / 30%	70% / 30%	70% / 30%	80% / 20%
Layer 1	100 Neurons	61 Neurons	N/A	Gaussian Distribution
Layer 2	20 Neurons	135 Neurons	N/A	N/A
Learning Rate	0.001	0.0058	N/A	N/A
Epochs	64	90	N/A	N/A

5.5 Evaluation Metrics

For the evaluation, we used two different metrics for different model type

For Price Prediction (LSTM and Decision Tree)
- RMSE (Root Mean Squared Error): This measures the standard deviation of prediction errors
- MAE (Mean Absolute Error): This measures the average absolute difference
- MAPE (Mean Absolute Percentage Error): Measures error in absolute number
- R^2 Score: Measures goodness of fit

For Trend Direction (Naive Bayes)
- Accuracy: The percentage of correct Up/Down predictions
- Confusion Matrix: To visualize true positives vs false positives
- Classification Report: Precision, Recall and F1-Score

6.0 Deployment of Model

6.1 Overview

The deployment of the model is to transfer the PSO-Optimized LSTM model from our coding in Google Colab to a real world application where users can actually use it to predict stock prices. There are a few steps required to deploy the model as a web-based tool. This process ensures that the optimized hyperparameters we found during training are preserved and used correctly for new predictions.

6.2 Step 1: Exporting Trained Model

Firstly, we saved the trained model of the project so it can be run elsewhere. We need to export two specific files:

- **The Trained Model (.h5 file):** We saved the final Keras model that contains the best parameters found by the PSO algorithm. This ensures the deployment model uses the same configuration that achieved the RMSE of 10.1.
- **The Scaler Object (scaler.pkl):** During our data analysis, we used a MinMaxScaler to normalize stock prices between 0 and 1. We must save this scaler because the model does not understand raw prices. If we do not use the same scaler to transform new data, the predictions will be inaccurate.

6.3 Step 2: Developing the Inference Engine

We built a Python script called app.py that acts as the brain for our application. This script will do the preprocessing steps:

1. **Input Handling:** The system will accept a list of the past 20 days of closing prices, which corresponds to the look-back window we used during training.
2. **Preprocessing:** The script will automatically normalize these 20 values using the saved scaler and reshape them into the 3D format, which the LSTM network requires.
3. **Prediction:** The model will generate a prediction in the normalized range from 0 to 1.
4. **Inverse Transformation:** Finally, the system will convert the result back into a currency value (MYR) so the user can read it.

6.4 Step 3: User Interface and Hosting

We used Gradio to make a user-friendly web interface. This allows us to create a web interface without the implementation of HTML or CSS coding.

- **The Interface:** The users will see a text box to paste recent stock prices and a Predict button.
- **Hosting:** We host the application on Hugging Face Spaces. It is a free platform that supports TensorFlow and provides a permanent URL. This is to ensure it is easier to share with examiners and future researchers. Here is the URL: https://srilimau123-bursa-stock-predictor.hf.space/?logs=container&__theme=system&deep_link=duKFZMV4084

6.5 Maintenance Plan

Since the stock market is always changing, a static model might lose accuracy over time. In the future, we will retrain the model over times with the latest Bursa Malaysia data to ensure it continues to capture new market volatility and trends.

7.0 Results and Discussion

7.1 Performance Comparison: Price Prediction Models

Our primary objective is to minimize the error between the predicted and actual closing prices of Bursa Malaysia KLCI. The table below presents the quantitative results of the three regression models.

Model	RMSE	MAE	MAPE	R2 Score
Decision Tree (Baseline)	16.71	8.4	0.83	0.95
Standard LSTM	13.01	12.63	0.64	0.97
PSO-Optimized LSTM	10.1	8.37	0.48	0.98

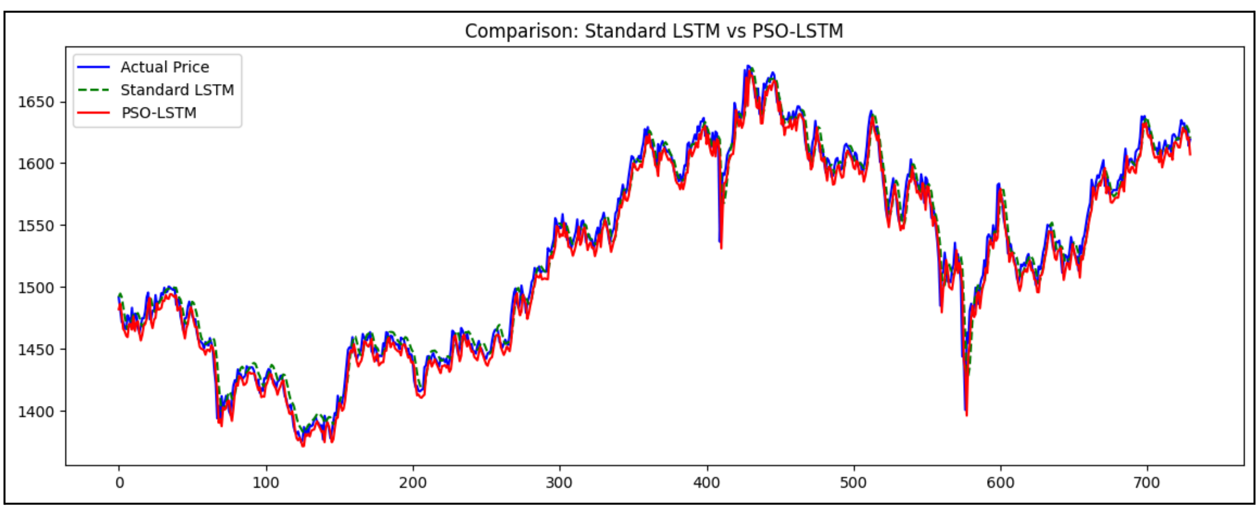
From the above table, we can see that the Decision Tree baseline yields the highest error (RMSE 16.71). This means simple regression trees struggle and are not able to capture complex temporal dependencies of the stock data. The standard LSTM improved even better than the decision tree with RMSE 13.01. This confirms that Recurrent Neural Networks (RNNs) like LSTM are more effective for time series tasks.

However, the PSO-Optimized LSTM showed the most superior performance with lowest RMSE of 10.1. That is 22% error reduction compared to Standard LSTM, and 40% compared to normal Decision Trees. This means, using novel and evolutionary algorithms like PSO to automate hyperparameter tuning instead of doing manually is proven to be more effective.

7.2 Standard vs. Optimized Prediction

7.2.1 Standard vs. Optimized Prediction

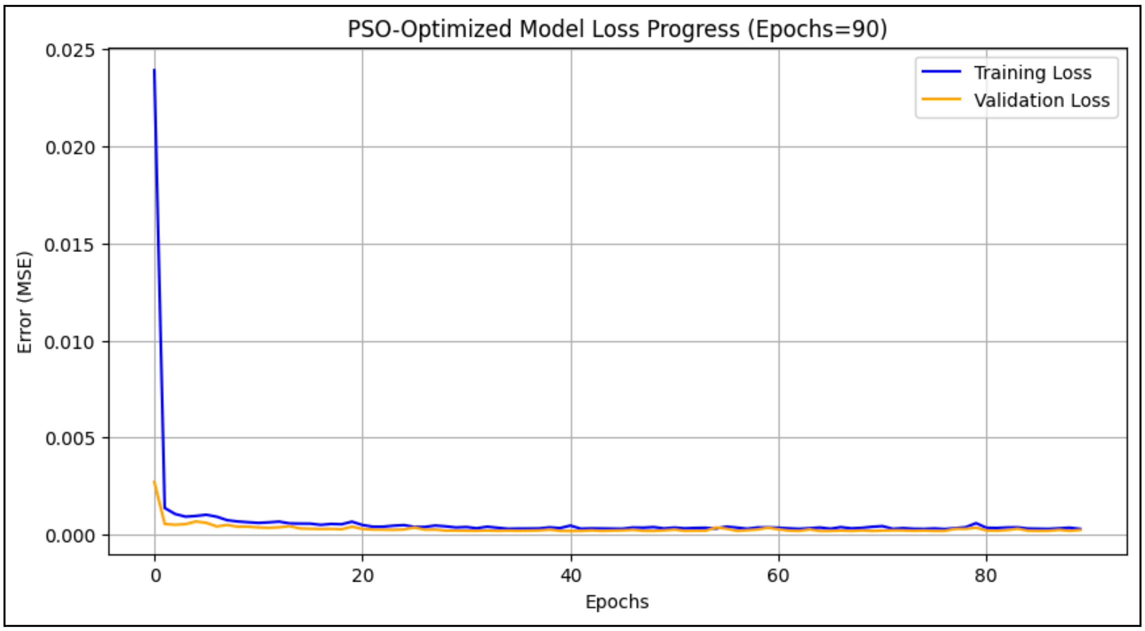
The figure below illustrates the forecasting capability of Standard LSTM vs. the PSO-Optimized LSTM against the actual market data.



As we can see, the Standard LSTM (green-dashed line) does follow the general trend but we can notice the lag during sharp market corrections, and it reacts a little bit slower compared to general trend. In contrast, the PSO-Optimized LSTM (red line) tracks the Actual Price (blue line) more closely, especially during volatile periods. This means that the deeper architecture and hyperparameters found by PSO (147 neurons vs. 20) allowed the model to learn more detailed market features.

7.2.2 Training Convergence and Stability

The figure below compares the training loss curves for the standard LSTM.



The model less was investigated because of overfitting concern, as the model follows the actual price trend really closely (with RMSE 10.1), but as we can see from the loss curve, it indicates healthy convergence without significant overfitting. The validation loss stabilised around epochs 40-50, but the PSO algorithm extended the training to 90 to extract more gains in accuracy without degrading generalization in performance.

7.3 Trend Classification Performance (naive Bayes)

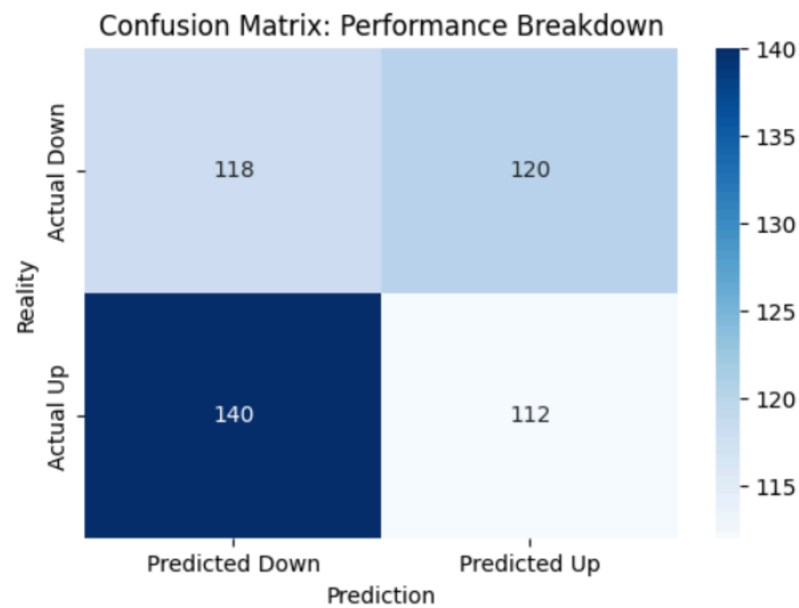
While LSTMs predicted the exact price, the other model we trained, Naive Bayes was tasked with prediction the ‘direction’ of the price or Buy and Sell.

Below are the Classification Report and Confusion Matrix of the Naive Bayes.

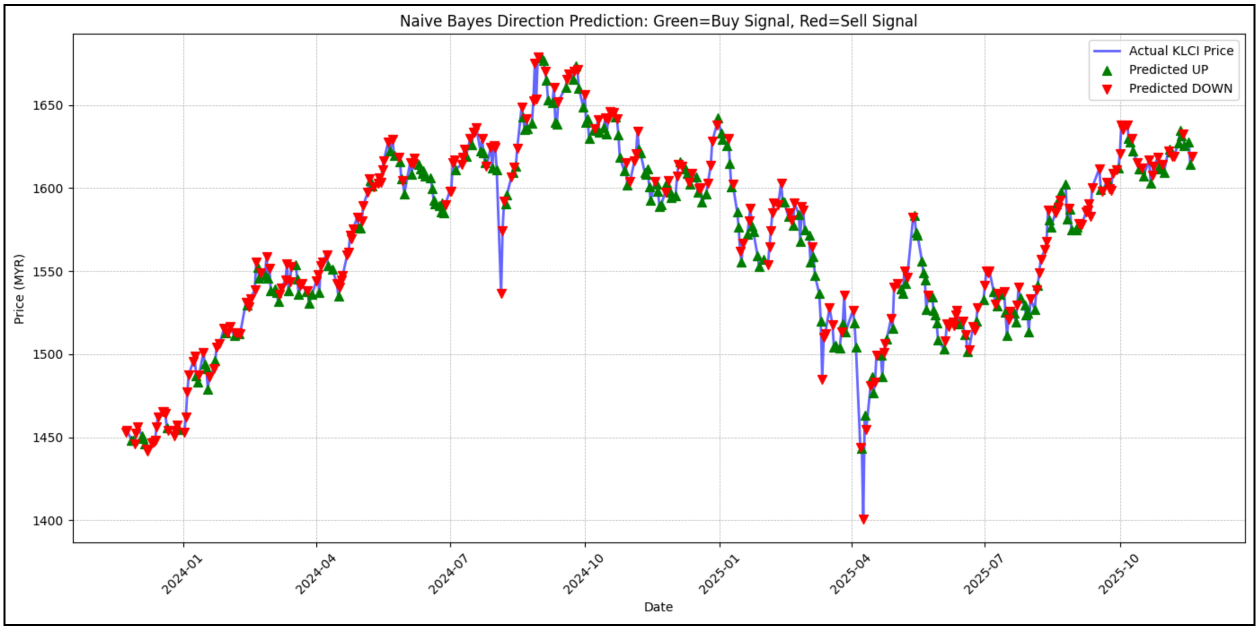
```
--- Classification Report ---
              precision    recall  f1-score   support

           0       0.46      0.50      0.48       238
           1       0.48      0.44      0.46       252

    accuracy                           0.47       490
   macro avg       0.47      0.47      0.47       490
weighted avg       0.47      0.47      0.47       490
```



As seen, the Naive Bayes model achieved an accuracy of only 46.94%, which is lower than random chance (50%). This underperformance exactly highlights the limitations of "Independence Assumption" in Naive Bayes. Naive Bayes assumes market features such as Open, High, Volume are independent, when in reality they are highly correlated to each other.



The above figures visualizes the Naive Bayes model prediction. It rapidly generates whipsaw, or rapidly changing signals between Buy and Sell, failing to capture sustained trends. Not only that, the predictions are not consistent and off, giving buy signals when the market is selling off and vice versa.

7.4 Discussion

As we have shown and discussed, PSO-LSTM model is the winner of the 4 models in predicting stock prices. The superior performance of the model can be attributed to two key factors:

1. Automated Architecture Search: The classical method of finding hyperparameters is doing so manually, which relies on “rules of thumbs”, such using powers of 2 like 64 or 128 neurons. PSO explored continuous search space and discovered non-standard configuration (like 68 or 147 neurons) that is not usually set and tested by humans manually as it seems like an arbitrary number.
2. Balanced Complexity: The Standard LSTM was likely to be underfitting (too simple) with only 20 neurons in the second layer. The PSO algorithms identified a significantly larger second layer (147 neurons) to train the model the non-linear volatility of the Bursa Malaysia KLCI index.

8.0 Conclusion

From the project, we learned a lot of things. First of all, we learned that Machine Learning could be fun too. We never knew it is fun to do research and to delve into different models, and implement up-to-date and hybrid approaches. All the members worked together to make the project successful. It would not have been successful just because of one or two members, but everyone contributed their time and energy into the project. We were always fascinated by stock predictions. We found the research about implementing the hybrid approach PSO to LSTM and started working on it. Bashyar and Hakim worked together to train standard LSTM and PSO-LSTM on the Bursa KLCI index. Adam helped us build the model using Naive Bayes, while Arif built the baseline for our comparison, which is Decision Tree Model. Each of us spent the time during the hectic final weeks to complete and document the results of the study, as well as recording videos to explain even further our project.

Most importantly, through the project, we successfully fulfilled our primary objective which is to validate the effectiveness of using hybrid method such as Particle Swarm Optimization with Long Short Term Memory (LSTM) networks to forecast the Bursa Malaysia KLCI index prices. And as seen, it significantly outperformed traditional models like Naive Bayes and Decision Trees. By making the complex process of tuning hyperparameters automated, the PSO-Optimized LSTM achieved the lowest Root Mean Squared Error (RMSE) of 10.1, a 22% improvement over the standard LSTM, and 40% if compared to decision tree, proving its superior ability to capture the market's non-linear volatility and time dependencies compared to static baselines. Our study also highlights the limitation of using traditional methods of hyperparameters tuning and the classifiers' assumptions of data independence in financial time-series contexts. At the end, we deployed the high accuracy model we have trained on the web.

For future work, we have to prioritize maintenance routine to address the stock market that is always evolving. Static models are prone to concept drift and they can lose accuracy over time, therefore we must establish a schedule to periodically retrain the model with the latest Bursa Malaysia data. This continuous updating process will make sure our web-based tool remains robust against new volatility patterns and emerging trends, therefore making it remain relevant to real-world users.

References

Abdelfattah, B. A., Darwish, S., & Elkaffas, S. M. (2024). Comparison of the model's performance using LSTM, neural networks, Support Vector Machine, and naïve Bayes. Journal of Electrical Systems, 20(3), 856–868.

Albeladi, K., Zafar, B., & Mueen, A. (2023). Time series forecasting using LSTM and ARIMA: A comparative analysis. *International Journal of Advanced Computer Science and Applications, 14*(1), 314–319. https://doi.org/10.14569/IJACSA.2023.0140133

Ashari, A., Paryudi, I., & Tjoa, A. M. (2013). Performance comparison between Naïve Bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool. *International Journal of Advanced Computer Science and Applications, 4*(11), 33–39. https://doi.org/10.14569/IJACSA.2013.041105

Cao, R. (2024). Stock price prediction using deep-learning models: CNN, RNN, and LSTM. *SHS Web of Conferences, 196*, 02004. https://doi.org/10.1051/shsconf/202419602004

Dhafer, A. H., & Nor, F. M. (2023). Prediction of FTSE Bursa Malaysia KLCI stock market using LSTM recurrent neural network. *International Journal of Advanced Computer Science and Applications, 14*(1), 314–319.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

He, B., Gong, E., Li, L., & Yang, Y. (2023). A stock price prediction method based on LSTM and K-Means. *Frontiers in Science and Engineering, 3*(6), 44–57. https://doi.org/10.54691/fse.v3i6.5121

Ji, Y., Liew, A. W. C., & Yang, L. (2021). A novel improved particle swarm optimization with long-short term memory hybrid model for stock indices forecast. *IEEE Access, 9*, 23660–23671. https://doi.org/10.1109/ACCESS.2021.3056713

Jiao, Y., Li, G., & Liu, Y. (2023). Future stock price prediction based on Bayesian LSTM in CRSP. *Proceedings of the 2023 International Conference on E-commerce and Digital Economy (ICEDE 2023)*, 155–163. Atlantis Press.

Khalil, R. A. B., & Bakar, A. A. B. U. (2023). A comparative study of deep learning algorithms in univariate and multivariate forecasting of the Malaysian stock market. *Sains Malaysiana, 52*(3), 993–1009. https://doi.org/10.17576/jsm-2023-5203-22

Pardeshi, K., Gill, S. S., & Abdelmoniem, A. M. (2023). Stock market price prediction: A hybrid LSTM and sequential self-attention based approach. *arXiv preprint arXiv:2308.04419*.

Shakawi, A. M. H., & Shabri, A. (2024). Improving prediction of Bursa Malaysia stock index using time series and deep learning hybrid model. In *Advances in Intelligent Computing Techniques and Applications* (pp. 119–128). Springer. https://doi.org/10.1007/978-3-031-59711-4_11

Soon, P. C., Tan, T. P., Chan, H. Y., & Gan, K. H. (2025). Stock trend prediction using multi-attention network on domain-specific and domain-general features in news headline. Pertanika Journal of Science & Technology, 33(2). https://doi.org/10.47836/pjst.33.2.13

Tian, J. (2025). Forecasting of share prices based on hybrid model of CNN and LSTM: A multi-factor approach. *Proceedings of the 1st International Conference on E-commerce and Artificial Intelligence (ECAI 2024)*, 222–228. https://doi.org/10.5220/0013213700004568

Wang, Z., Sun, Y., & Zhao, Z. (2024). Enhancing stock index prediction: A hybrid LSTM-PSO model for improved forecasting accuracy. *PLOS ONE, 19*(3), e0310296. https://doi.org/10.1371/journal.pone.0310296