Comparative Analysis of Traditional Machine Learning Models (SVM, KNN, and Linear Regression) for KSE 100 Stock Price Forecasting

Aldin Febriansyah[1], Ervin Yohannes[2]

[1,2] Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya
[1]aldin.20005@mhs.unesa.ac.id
[2]ervinyohannes@unesa.ac.id

***Abstract*—The erratic volatility of stock prices presents a significant challenge for analysts and investors when making informed investment decisions. Although the Efficient Market Hypothesis suggests that price prediction is theoretically impossible, numerous studies indicate that predictive models can yield high-quality results. This research compares the effectiveness of three traditional machine learning algorithms—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Linear Regression (LR)—in forecasting the daily stock prices of the KSE 100 Index from the Pakistan Stock Exchange (PSX). The study utilized 3,221 daily closing prices recorded between February 22, 2008, and February 23, 2021. The models were implemented in Python and optimized through hyperparameter tuning using GridSearchCV. To ensure robust evaluation, five distinct data-splitting techniques were employed: a chronological split of 2020 and proportional splits of 80:20, 75:25, and 70:30. Performance was measured using MSE, RMSE, MAE, MAPE, and Accuracy metrics. The findings reveal that Linear Regression (LR) consistently delivered the best and most stable performance across all testing schemes. LR achieved its highest accuracy of 97.9% and lowest error (MSE 0.000404) in the 70:30 split, while maintaining a 97.3% accuracy in the 2020 test data. In contrast, KNN was the most sensitive model, with accuracy dropping to 92.2% in the 30% test scheme. These results underscore that LR is the most accurate and dependable option for stock price time-series prediction among these traditional models, proving that simpler models can remain highly competitive.**

***Index Terms*— Stock Price Forecasting, Machine Learning, Linear Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN).**

I. INTRODUCTION

The stock market is an important economic indicator that reflects confidence in the economy [1]. In 2023, there were 12.16 million capital market investors in Indonesia, up 18% from the year before, according to data from PT Bursa Efek Indonesia (BEI). [2]. This indicates a growing awareness among the Indonesian public regarding stock investment. Stock investment offers the potential for profit through capital gains and dividends, but faces the main challenge of high price volatility [3]. Unstable price fluctuations cause investors to face uncertainty in making investment decisions [4]. Therefore, predicting stock price movements is an important aspect to minimize risk and optimize profits.

Although the Efficient Market Hypothesis states the difficulty of accurately predicting stock prices [5], various studies show that predictive models with high accuracy can yield satisfactory results. Because machine learning can identify hidden trends in past data, it has become a popular method for stock price prediction [6]. Support Vector Machine (SVM), Linear Regression, and K-Nearest Neighbors (KNN) are a few common machine learning methods used in stock prediction [7]. SVM is helpful for both classification and regression by locating an optimal hyperplane [8]. Continuous value prediction is a good use for linear regression, which models the linear connection between variables [9]. KNN assumes that comparable data have similar properties and bases its predictions on the k nearest neighbors [10]. Every algorithm has advantages and disadvantages when it comes to managing time series stock price data.

It is anticipated that the results of this study will help traders, investors, and financial analysts choose and apply the best machine learning algorithms to facilitate more successful and knowledgeable investment decision-making. This study's primary contributions are as follows:

1) A comprehensive comparison between SVM, Linear Regression, and KNN algorithms with various hyperparameter configurations to identify the most accurate and reliable approach in stock price prediction;
2) A thorough analysis that highlights which algorithm configurations provide the greatest prediction accuracy utilizing important performance measures including MAPE, MSE, RMSE, and MAE;
3) Providing practical insights to help financial practitioners select and implement the most effective machine learning-based predictive models to improve their investment strategies.

In Section II, we will discuss related works. The proposed method, which includes SVM, Linear Regression, and KNN architectures, will be presented in Section III. Section IV displays the experiments. Section V concludes with recommendations for further research.

II. LITERATURE REVIEW

Stock price movement prediction using machine learning has become a rapidly growing research topic. Various methods have been applied with varying accuracy levels across different capital markets. This literature review examines recent studies that use machine learning techniques for stock price prediction.

Singh used four distinct data subsets to apply eight machine learning algorithms (SGD, ANN, RF, LR, SVM, AdaBoost, KNN, and DT) in a thorough investigation of the Nifty 50 index. According to the study's findings, Artificial Neural

Network (ANN) and Linear Regression (LR) consistently produced the best results across all data subsets, with RMSE of 36.87, MAE of 25.72, and R^2 of 0.999 [11]. Pratama & Bawonosari compared LSTM and XGB for predicting the stock of PT. Bank Mandiri Tbk. (BMRI). LSTM achieved an accuracy of 98.23%, while XGB reached 96.79%, indicating that LSTM was slightly superior in the Indonesian capital market context [12]. Hwase & Fofanah developed a mobile application called Ethiopia Coffee Prices Predictor (ECPP) to predict coffee and sesame prices using LR, XGB, and LSTM. LSTM showed the best performance, especially on larger datasets, indicating superior capability in handling the complexity of time series data [13]. Nagar et al. studied MAANG (Meta, Amazon, Apple, Netflix, Google) stocks by comparing LSTM and SVR. LSTM demonstrated better performance with an RMSE of 7.04 for Meta's stock compared to SVR's RMSE of 14.61, proving LSTM's superiority in predicting large technology stocks [14]. Indika et al. explored ensemble methods for NASDAQ stock prediction using SVM, LSTM, LR, Random Forest, and various ensemble techniques. The blending ensemble produced the best performance with an average ranking of 1.60, showing that combining algorithms can enhance prediction accuracy [15]. Karim & Rasheed examined the Pakistan Stock Exchange KSE100 by comparing ANN, SVM, and Decision Tree. Their study found that SVM showed almost zero prediction error on Tuesdays, Wednesdays, and Thursdays, indicating highly accurate predictions on those days. Conversely, ANN performed worse than SVM and Decision Tree [16]. Saboor et al. used the KSE 100 (Pakistan), DSE 30 (Bangladesh), and BSE Sensex (India) regional stock indexes to compare deep learning methods with traditional machine learning. While SVR, Random Forest, and k-Nearest Neighbor were examples of classic machine learning techniques, deep learning techniques included LSTM, GRU, LSTM + Attention, GRU + Attention, and LSTM + GRU + Attention. The findings demonstrated that RNN with attention mechanisms consistently outperformed conventional machine learning techniques in terms of closing price prediction performance for all three indices. [17].

III. METHODOLOGY

A. *Research Data*

This study utilizes a comprehensive dataset derived from the Pakistan Stock Exchange (PSX), specifically focusing on the daily closing price movements of the KSE 100 Index. The observation period spans from February 22, 2008, to February 23, 2021, comprising a total of 3,221 distinct data points. As the primary benchmark for the Pakistani economy, the KSE 100 Index represents the top 100 companies by market capitalization, selected based on high trading volume and capitalization across diverse industrial sectors such as energy, finance, and telecommunications. Historically, the PSX was established in 2016 following the merger of the Karachi, Lahore, and Islamabad Stock Exchanges.

Regarding feature engineering, the dataset initially includes columns for Date, Open, High, Low, Close, Change, and Volume. However, to optimize the predictive capability of the machine learning model, a rigorous selection process was applied. Attributes such as "Date," "Change," and "Volume" were excluded to minimize potential noise. Instead, the model relies exclusively on the OHLC (Open, High, Low, Close) parameters. Empirical evidence suggests that these specific characteristics provide sufficient temporal information for accurate forecasting without introducing significant bias, thereby enhancing the overall reliability of the prediction model.

TABLE 1
SAMPLE OF PAKISTAN STOCK EXCHANGE DATASETS

Date	Open	High	Low	Close	Change	Volume
23-Feb-21	31,722	31,801	31,597	31,626	-21	718,19..
22-Feb-21	31,875	31,959	31,613	31,648	-204	721,95..
19-Feb-21	31,749	31,904	31,749	31,851	91	694,79..
18-Feb-21	32,050	32,105	31,746	31,760	-289	577,83..
…	…	…	…	…	…	…
22-Feb-08	10,634	10,635	10,546	10,607	0	313,08..

B. *Data Analyst Technique*

In order to guarantee the analytical rigor and predictive validity of the applied machine learning architectures—specifically Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Linear Regression (LR)—this study implements a multifaceted data preparation pipeline designed to transform raw historical data into a refined state suitable for high-precision forecasting. The methodological framework commences with a meticulous data preprocessing phase, wherein the temporal dimension is standardized by converting the 'Date' column into a uniform datetime format utilizing the Pandas library, followed by a strict ascending chronological sort to preserve the sequential integrity essential for time-series analysis. Concurrently, to ensure computational compatibility and numerical precision, the process involves a thorough sanitization of non-date columns, which entails the removal of non-numeric artifacts such as commas and the systematic conversion of all feature variables into the float64 data type. Furthermore, to prevent the introduction of bias or noise that could compromise model convergence, rigorous validation checks are conducted to identify and eliminate any missing values or duplicate entries within the dataset.

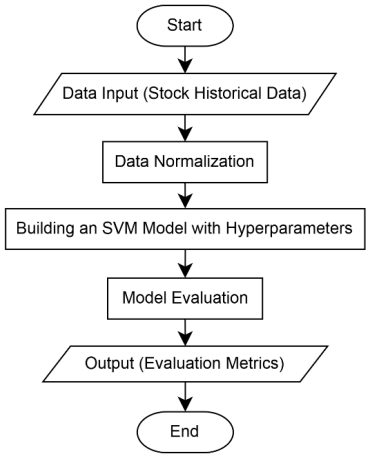
Subsequent to the cleaning phase, a sophisticated feature engineering process is executed to extract latent patterns from the historical price movements; this involves restructuring the raw time-series data into a supervised learning format by generating lagged variables and moving averages, with the subsequent day's closing price explicitly designated as the target variable for prediction. regarding the experimental evaluation scheme, the dataset undergoes a chronological partitioning strategy where data preceding the year 2020 is utilized for model training, whereas the data from the year 2020 is isolated as a distinct testing set to evaluate performance on unseen data—notably, records extending into 2021 were deliberately excluded to maintain a consistent

testing window. Finally, to address the inherent sensitivity of distance-dependent algorithms like KNN and SVM to varying magnitudes of data, all numerical features are normalized to a strict 0–1 range using the Scikit-Learn MinMaxScaler, after which the processed data is converted into NumPy arrays to facilitate seamless integration within the computational framework.

C. *Methodology*

1) *Support Vector Machine (SVM)*

Support Vector Regression (SVR), a variant of SVM, is employed in this study for its effectiveness in handling high-dimensional time-series data by finding a function that minimizes error within a specified threshold (ε). SVM separates data into two or more classes using a hyperplane that is adjusted to have the largest margin between the classes. As seen in Fgr. 1, the first step in using SVM is selecting a kernel, which determines how the data is converted into a higher-dimensional space to enable easier separation. SVM is renowned for its capacity to attain excellent accuracy and great generalization performance in a variety of applications [18]. Reducing the number of processes during training can yield more optimum outcomes, despite the algorithm's comparatively high training time. SVM is widely used in domains including prediction, picture classification, medical diagnosis, text analysis, and outlier identification. It has demonstrated success in managing massive datasets. [19].

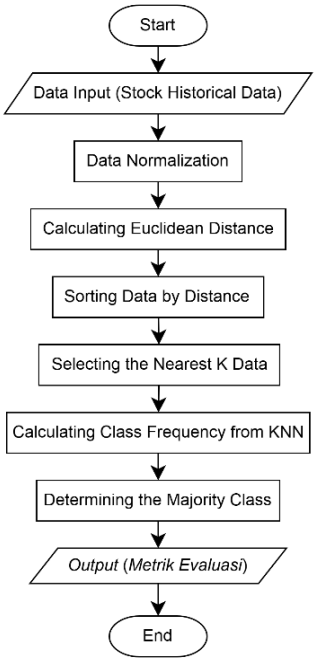


Fgr. 1 Process Flow in SVM Model.

The working principle of SVM is to find the optimal hyperplane that can separate data classes by maximizing the margin between the closest data points from different classes. This algorithm uses the concept of support vectors, which are the data points located on the margin boundaries that determine the hyperplane [20].

2) *K-Nearest Neighbors (KNN)*

K-Nearest Neighbors (KNN) is recognized as a seminal non-parametric technique in supervised machine learning, tracing its theoretical origins to the work of Fix and Hodges (1951) with further advancements by Cover [21]. The core mechanism of KNN involves the retrieval of the k most similar samples from the reference data to evaluate an unobserved query point. Consequently, the classification of the unknown sample is assigned based on the prevailing class among these identified neighbors, a process visually depicted in Fgr. 2 below.

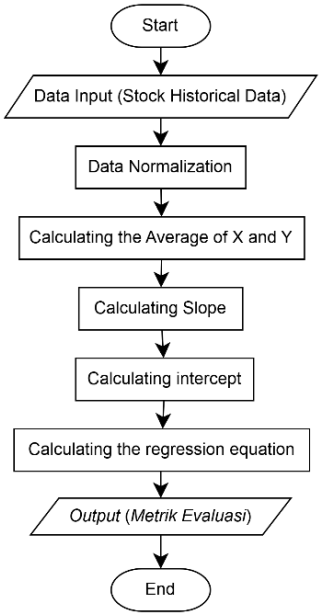


Fgr. 2 Process flow in the KNN Model.

As the primary tuning parameter that affects prediction accuracy, the value k is crucial to the KNN algorithm's performance. The optimal value of k can be determined through various methods such as cross-validation or bootstrap procedures [22]. KNN has the advantage of easy implementation but also has weaknesses in computational complexity, especially for large datasets.

3) *Linear Regression (LR)*

Linear Regression serves as a fundamental statistical framework designed to model the linear correlation between input variables (predictors, denoted as X) and output variables (targets, denoted as Y). The core objective of this methodology is to derive a linear equation that accurately represents the relationship between these variables, thereby facilitating the forecasting of the target variable based on predictor values. In scenarios where the model incorporates multiple predictor variables, the technique is designated as Multiple Linear Regression [23]. To establish the optimal fit, regression coefficients are typically estimated utilizing the Least Squares method, which mathematically minimizes the sum of squared differences between the regression line and the observed data points. The operational workflow of this technique, illustrating the predictive relationship between the independent and dependent variables, is schematically depicted in Fgr. 3.



Fgr. 3 Process Flow in the Linear Regression Model.

Linear regression is one of the most fundamental and commonly utilized machine learning algorithms in a variety of numerical prediction applications because of its simplicity and ease of comprehension of results [24]. This approach may be expanded to incorporate non-linear connections using feature transformations, but it functions best when there is a linear relationship between the input features and the desired output.

IV. Experiments and Result

This section uses machine learning models (SVM, KNN, and Linear Regression) to forecast stock prices using the Pakistan Stock Exchange (KSE 100) dataset. The solution uses the Python programming language and many auxiliary libraries, including Scikit-Learn for model development, Pandas for data processing, NumPy for numerical computations, and Matplotlib for performance evaluation and display.

A. *Hyperparameter Setting and Evaluation Metric on SVM*

Data Preparation and Experimental Design Prior to the construction of the Support Vector Machine (SVM) model, the dataset underwent a comprehensive preparation phase comprising preprocessing, normalization via MinMaxScaler, and partitioning. To evaluate the model's robustness, the study employs distinct data-splitting strategies, specifically:

- Chronological Splitting: Utilizing data from 2008–2019 for training and the year 2020 for testing.
- Proportional Splitting: Evaluating three randomized ratios of training-to-testing data: 80:20, 75:25, and 70:30.

Model Optimization and Training Following data preparation, the SVM architecture was optimized for stock price forecasting using GridSearchCV with 3-fold cross-validation (cv=3). This technique systematically explored a defined hyperparameter space to identify the configuration yielding the highest accuracy. The search parameters included:

- Kernel: 'linear', 'sigmoid'
- Degree: 3, 5
- Coef0: 0, 3, 7
- Gamma: 0.001, 0.1, and inverse number of features (1/n_features)
- C (Regularization): 1, 10, 100

The grid search identified the optimal hyperparameter combination as: C = 100, coef0 = 0, degree = 3, gamma = 0.001, and kernel = 'sigmoid'. Consequently, the Support Vector Regression (SVR) model was instantiated with these parameters and trained on the training set (X_train, y_train).

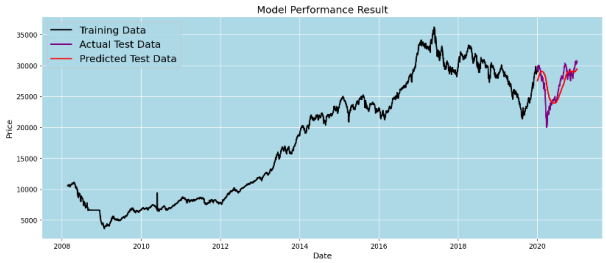
Predictions were subsequently generated using the test set (X_test), and performance was rigorously quantified using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Accuracy. The empirical results across the defined splitting scenarios are summarized in Table 2.

TABLE 2
SVM Evaluation Results

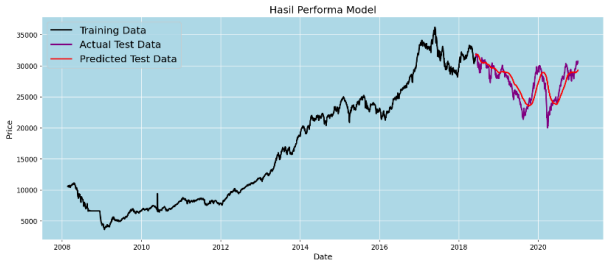
Model	MSE	RMSE	MAE	MAPE	Acc
2020	0.003462	0.058840	0.043233	0.064693	93.5%
20%	0.001049	0.032236	0.026108	0.035858	96.4%
25%	0.002304	0.047995	0.036797	0.052161	94.8%
30%	0.002216	0.046965	0.036294	0.049985	95.0%

The quantitative analysis presented in Table 2 identifies the 80:20 data partition (20% test data) as the optimal configuration for the SVM model, yielding a peak accuracy of 96.41%. This finding suggests a positive correlation between the volume of the training set and the model's predictive efficacy, indicating that the SVM architecture requires a substantial proportion of training data to maximize learning performance.

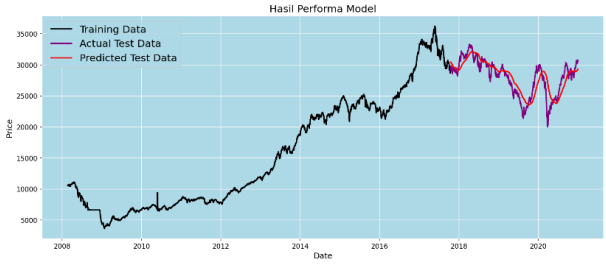
To complement the statistical metrics, Figures 4 through 7 present the time-series forecasting results for the SVM model. These plots depict the trajectories of the model under the various data-splitting scenarios, graphically overlaying the Predicted Test Data (red line) against the Actual Test Data (purple line) and the historical Training Data (black line). This visualization provides a qualitative assessment of the model's trend-following capabilities, allowing for a direct inspection of the alignment and divergence between the projected stock prices and actual market values.



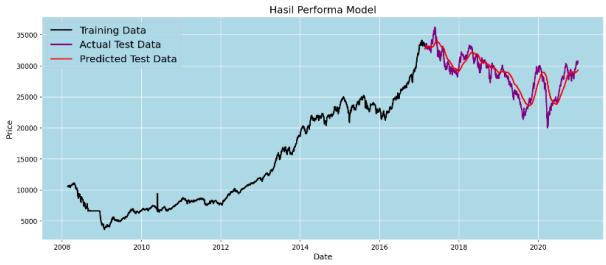
Fgr. 4 Results of SVM Test Data Evaluation for 2020.



Fgr. 5 SVM Evaluation Results for 20% Test Data.



Fgr. 6 SVM Evaluation Results for 25% Test Data.



Fgr. 7 SVM Evaluation Results for 30% Test Data.

B. *Hyperparameter Setting and Evaluation Metric on KNN*

Methodological Framework The analytical procedure for the K-Nearest Neighbors (KNN) model adheres to a structured data preparation pipeline, encompassing data processing, normalization, and model construction. Consistent with the approach applied to the SVM modeling, this study employed five distinct data-splitting scenarios to evaluate performance under varying conditions.

Hyperparameter Tuning To maximize predictive accuracy, the model was configured using GridSearchCV, a rigorous parameter tuning technique incorporating 3-fold cross-validation (cv=3). The optimization process systematically explored the following hyperparameter space:

- n_neighbors: 9, 10, 11, 50
- weights: 'uniform', 'distance'
- algorithm: 'ball_tree', 'kd_tree', 'brute'
- leaf_size: 1, 2, 20, 50, 200

The grid search identified the optimal configuration as: algorithm = 'brute', leaf_size = 1, n_neighbors = 50, and weights = 'distance'. Subsequently, the KNeighborsRegressor was trained on the training partitions (X_train) and validated against the test sets (X_test). Performance was quantified using standard metrics: MSE, RMSE, MAE, MAPE, and Accuracy.

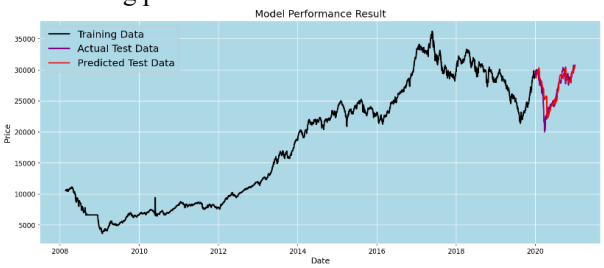
Performance Analysis As presented in Table 3, the evaluation highlights a distinct correlation between training data volume and model efficacy.

TABLE 3
KNN EVALUATION RESULTS

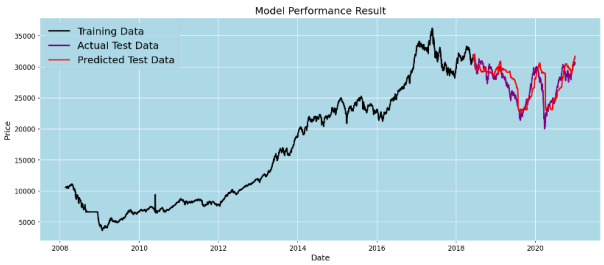
Model	MSE	RMSE	MAE	MAPE	Acc
2020	0.001777	0.042157	0.028423	0.043753	95.6%
20%	0.002976	0.054549	0.039793	0.057366	94.3%
25%	0.003042	0.055162	0.043346	0.060198	94.0%
30%	0.004854	0.069671	0.058356	0.077593	92.2%

The optimal performance was achieved using the 2020 chronological split, which recorded a peak accuracy of 95.6%. This finding indicates that the KNN algorithm requires a substantial training corpus to facilitate optimal learning. Conversely, proportional splits resulted in lower accuracy, with the 30% test split yielding the poorest result of 92.2%.

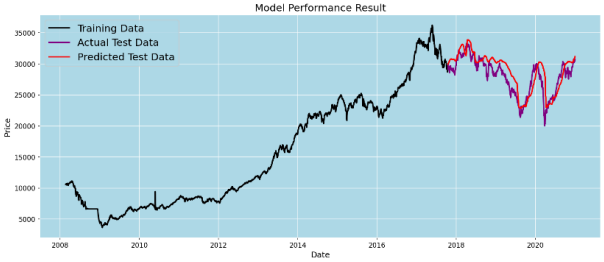
To complement the numerical evaluation metrics (MSE, RMSE, MAE, MAPE, and Accuracy), a visual analysis of the model's forecasting performance is presented. Figures 8 through 11 depict the time-series trajectories of the KNN model under the various data-splitting scenarios. These plots graphically overlay the Predicted Test Data (red line) against the Actual Test Data (purple line) and the historical Training Data (black line). This visualization provides a qualitative assessment of the model's fit, clearly illustrating how closely the predicted values track the actual market trends across the different testing periods.



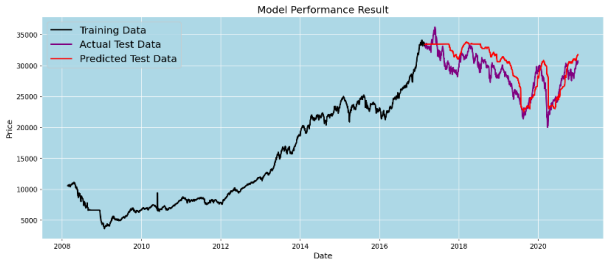
Fgr. 8 Result of KNN Test Data Evaluation for 2020.



Fgr. 9 KNN Evaluation Results for 20% Test Data.



Fgr. 10 KNN Evaluation Results for 25% Test Data.



Fgr. 11 KNN Evaluation Results for 30% Test Data.

C. *Hyperparameter Setting and Evaluation Metric on LR*

Methodological Framework The data preparation pipeline for the Linear Regression (LR) analysis replicates the rigorous standards applied to the SVM and KNN models, encompassing data processing, normalization, and partitioning across five distinct scenarios. To ensure model robustness, the architecture was fine-tuned using GridSearchCV with 3-fold cross-validation (cv=3). The optimization process scrutinized the following parameter space:

- alpha: 0, 0.01, 1, 10, 100
- l1_ratio: 0, 0.01, 1
- fit_intercept: True, False

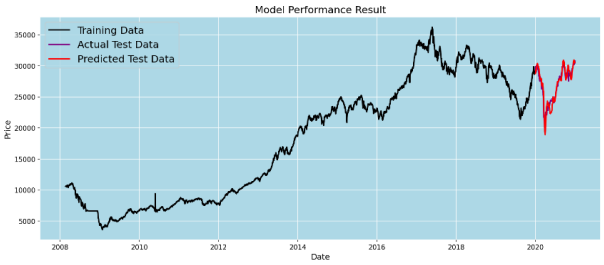
The grid search determined the optimal hyperparameter configuration to be: alpha=0, fit_intercept=True, and l1_ratio=0.0. Utilizing these parameters, the model was trained employing the ElasticNet approach on the training datasets (X_train) and validated against the test sets (X_test). The assessment relied on standard performance metrics: MSE, RMSE, MAE, MAPE, and Accuracy.

TABLE 4
LR EVALUATION RESULTS

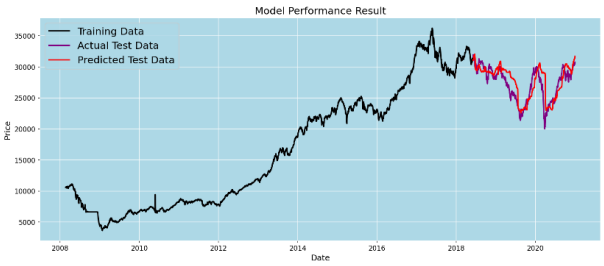
Model	MSE	RMSE	MAE	MAPE	Acc
2020	0.000598	0.024464	0.018302	0.027100	97.3%
20%	0.000492	0.022177	0.017227	0.024317	97.6%
25%	0.000416	0.020395	0.015753	0.021727	97.8%
30%	0.000404	0.020110	0.015671	0.020908	97.9%

As detailed in Table 4, the empirical results demonstrate that the Linear Regression model achieved its peak performance under the 70:30 proportional split (30% test data), recording an accuracy of 97.9%. This finding indicates that, unlike the distance-based models, the Linear Regression framework exhibited superior generalization capabilities with a larger allocation of testing data.

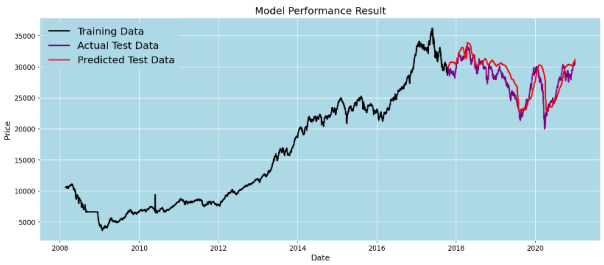
Visual Analysis of Model Performance To complement the statistical evaluation, Figures 12 through 15 present the time-series forecasting trajectories of the LR model. These plots graphically overlay the Predicted Test Data (red line) against the Actual Test Data (purple line) and the historical Training Data (black line). This visualization allows for a direct inspection of the alignment between projected values and actual market trends, confirming the model's high precision and minimal divergence across the evaluated data-splitting scenarios.



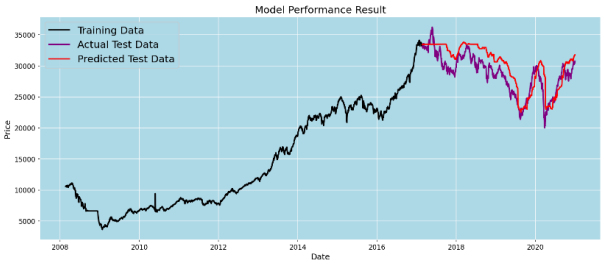
Fgr. 12 Results of LR Test Data Evaluation for 2020.



Fgr. 13 LR Evaluation Results for 20% Test Data.



Fgr. 14 LR Evaluation Results for 25% Test Data.



Fgr. 15 LR Evaluation Results for 30% Test Data.

D. *Result*

In this study, three distinct algorithms—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Linear Regression (LR)—were utilized to assess model performance on the target dataset. The evaluation outcomes are summarized in five tables, each corresponding to a specific data-splitting strategy. These results serve as the basis for analyzing the dataset and benchmarking the effectiveness of each model.

1) *Splitting with Test Data from the year 2020*

TABLE 5
EVALUATION RESULTS OF THE 3 MODELS ON TEST DATA FROM 2020

Model	MSE	RMSE	MAE	MAPE	Acc
SVM	0.00346	0.05884	0.04323	0.06469	93.5%
KNN	0.00178	0.04216	0.02842	0.04375	95.6%
LR	0.00060	0.02446	0.01830	0.02710	97.3%

As summarized in Table 5, the analysis of the 2020 testing window reveals that the Linear Regression (LR) model yielded the most favorable outcomes, distinctively outperforming the SVM and KNN algorithms. With an accuracy of 97.29%, LR minimized predictive deviation effectively, as evidenced by an MSE of 0.00060 and an MAE of 0.01830. The associated RMSE (0.02446) and MAPE (0.02710) values further attest to the model's precision in tracking price movements. Conversely, the non-linear models struggled to match this level of accuracy. KNN ranked second with an accuracy of 95.62%, whereas SVM proved to be the least robust. The SVM model's performance was characterized by the highest error rates—specifically an MSE of 0.003462 and MAE of 0.04323—and a consequent accuracy of only 93.53%.

2) *Splitting with 20% Test Data*

TABLE 6
EVALUATION RESULTS OF THE 3 MODELS ON 20% TEST DATA

Model	MSE	RMSE	MAE	MAPE	Acc
SVM	0.00104	0.03224	0.02611	0.03586	96.4%
KNN	0.00298	0.05455	0.03979	0.05737	94.3%
LR	0.00049	0.02218	0.01723	0.02432	97.6%

The quantitative assessment of the 20% test split, as outlined in Table 6, confirms the superior predictive capability of the Linear Regression (LR) model relative to its counterparts. LR not only attained the highest accuracy of 97.57% but also demonstrated exceptional precision, evidenced by a minimal Mean Squared Error (MSE) of 0.00049. This robustness is further substantiated by consistently low values across auxiliary error metrics, including RMSE (0.02218), MAE (0.01723), and MAPE (0.02432). In terms of comparative performance, the Support Vector Machine (SVM) model secured the second-tier position with an accuracy of 96.41% and an MSE of 0.00104. Conversely, the K-Nearest Neighbors (KNN) model exhibited the least favorable performance metrics in this specific scheme, recording the lowest accuracy of 94.26% alongside the most substantial error magnitude (MSE 0.00298).

3) *Splitting with 25% Test Data*

TABLE 7
EVALUATION RESULTS OF THE 3 MODELS ON 25% TEST DATA

Model	MSE	RMSE	MAE	MAPE	Acc
SVM	0.00230	0.04800	0.03680	0.05216	94.8%
KNN	0.00304	0.05516	0.04335	0.06020	94.0%
LR	0.00042	0.02040	0.01576	0.02173	97.8%

The results derived from the 25% test data partition, as summarized in Table 7, establish the Linear Regression (LR) model as the most robust predictor, significantly surpassing the performance of both SVM and KNN. LR attained a peak accuracy of 97.83% while consistently maintaining the lowest error magnitudes across all indicators, specifically recording an MSE of 0.00042, RMSE of 0.02040, MAE of 0.01576, and MAPE of 0.02173. These minimal deviation metrics underscore the model's exceptional precision and its ability to closely align with actual data points. In distinct contrast, the alternative models demonstrated inferior predictive capabilities. The SVM model secured an intermediate ranking with an accuracy of 94.78% and an MSE of 0.00230. However, the KNN algorithm proved to be the least effective in this testing scheme, yielding the lowest accuracy (93.98%) and the highest Mean Squared Error at 0.00304.

4) *Splitting with 30% Test Data*

TABLE 8
EVALUATION RESULTS OF THE 3 MODELS ON 30% TEST DATA

Model	MSE	RMSE	MAE	MAPE	Acc
SVM	0.00221	0.04697	0.03629	0.04999	95.0%
KNN	0.00485	0.06967	0.05836	0.07759	92.2%
LR	0.00040	0.02011	0.01567	0.02091	97.9%

The performance assessment of the 30% test data partition, as detailed in Table 8, reveals that the Linear Regression (LR) model outperformed both SVM and KNN by the most significant margin observed in this study. LR achieved a superior accuracy of 97.9092% and consistently maintained the lowest error rates across all metrics, including an MSE of 0.000404, RMSE of 0.020110, and MAPE of 0.020908. These figures indicate a minimal degree of divergence between the projected values and the actual dataset. In contrast, the KNN model exhibited the most severe performance decline in this scenario, recording the lowest accuracy (92.2406%) and the highest MSE (0.004854). Meanwhile, the SVM model occupied an intermediate position with an accuracy of 95.0015%. Synthesizing these findings, LR emerges as the most reliable option, whereas KNN demonstrates a high sensitivity to data splitting arrangements, evidenced by its sharp decline as the test ratio increased.

Synthesizing the empirical results across all data-splitting methodologies, the Linear Regression (LR) model consistently established its dominance, delivering the highest accuracy and minimizing error metrics throughout the testing phases. In terms of comparative stability, the Support Vector Machine (SVM) generally exhibited greater resilience than the K-Nearest Neighbors (KNN) algorithm. Notably, the KNN model displayed significant sensitivity to the dataset partitioning structure, evidenced by a marked deterioration in performance as the test data ratio was expanded to 30%. Consequently, these findings substantiate the conclusion that, among the evaluated architectures, Linear Regression constitutes the most robust and reliable approach for stock price forecasting.

V. CONCLUSION

Based on the comparative analysis of traditional Machine Learning model performances (SVM, KNN, and LR) across various data splitting schemes, the Linear Regression (LR) model showed the most consistent and superior performance in predicting stock prices. LR consistently recorded the highest accuracy and lowest error rates across all testing schemes, ranging from the 2020 Test Data (accuracy 97.3%,

MSE 0.00060) to the 30% scheme (highest accuracy 97.9%, lowest MSE 0.000404). This consistency indicates that LR is a robust and stable model against changes in test data proportions. In contrast, the SVM and KNN models performed lower and less stably. SVM showed accuracy varying between 93.5% and 96.4%, while KNN was the most vulnerable, with accuracy sharply dropping from 95.6% to the lowest 92.2% in the 30% scheme, where it also recorded the highest MSE (0.004854). Overall, these findings underline that LR is the most accurate and reliable choice among these three models for time series prediction, proving that simpler models can still be highly competitive.

REFERENCES

[1] [1] L. R. br Simbolon, T. D. Liani, H. R. Anwar, and E. A. I. br Naibaho, “Analisis Efisiensi Pasar Modal di Indonesia,” BISMA Bus. Manag. J., vol. 2, pp. 1–10, 2023, [Online]. Available: https://databoks.katadata.co.id/

[2] [2] I. I. S. Exchange, “Melalui Berbagai Pencapaian Tahun 2023, Pasar Modal Indonesia Tunjukkan OptimismeHadapi Tahun 2024,” Exchange, IDX Indonesia Stock. Accessed: May 05, 2024. [Online]. Available: https://www.idx.co.id/en/news/press-release/2080

[3] [3] M. Hisam, “Menavigasi Volatilitas Pasar: Wawasan Tentang Instrumen Keuangan dan Strategi Investasi,” J. Ekon. dan Perbank. Syariah, vol. 02, no. April, pp. 315–328, 2024, doi: 10.32806/ke534p70.

[4] [4] R. Zapar, D. Pratama, K. Kaslani, C. L. Rohmat, and F. Faturrohman, “Penerapan Model Regresi Linier Untuk Prediksi Harga Saham Bank Bca Pada Bursa Efek Indonesia,” JATI (Jurnal Mhs. Tek. Inform., vol. 8, no. 1, pp. 196–202, 2024, doi: 10.36040/jati.v8i1.8215.

[5] [5] S. Maddodi and K. G. N. Kumar, “Stock Market Forecasting: a Review of Literature,” IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. Vol 5, No, p. 11, 2021, [Online]. Available: https://ojs.stmikpringsewu.ac.id/index.php/ijiscs/article/download/1064/pdf

[6] [6] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, “Machine Learning and Deep Learning Approaches for CyberSecurity: A Review,” IEEE Access, vol. 10, no. Ml, pp. 19572–19585, 2022, doi: 10.1109/ACCESS.2022.3151248.

[7] [7] A. A. Rismayadi, R. W. Febrianto, A. R. Raharja, and I. Hariyanti, “Perbandingan Kinerja Metode Machine Learning Support Vector Machine (SVM), Random Forest, dan K-Nearest Neighbors (KNN) dalam Prediksi Harga Saham Apple,” Media Inform., vol. 23, no. 3, pp. 152–160, 2024, doi: 10.37595/mediainfo.v23i3.299.

[8] [8] K. V. Vijay and P. B. Narayan, “SVM-Based Stock Market Price Prediction Methods: An Advanced Review,” i-manager’s J. Comput. Sci., vol. 10, no. 3, p. 13, 2022, doi: 10.26634/jcom.10.3.19183.

[9] [9] J. M. Sangeetha and K. J. Alfia, “Financial stock market forecast using evaluated linear regression based machine learning technique,” Meas. Sensors, vol. 31, no. October 2023, p. 100950, 2024, doi: 10.1016/j.measen.2023.100950.

[10] [10] J. Tanuwijaya and S. Hansun, “LQ45 stock index prediction using k-nearest neighbors regression,” Int. J. Recent Technol. Eng., vol. 8, no. 3, pp. 2388–2391, 2019, doi: 10.35940/ijrte.C4663.098319.

[11] [11] D. G. Singh, “Machine Learning Models in Stock Market Prediction,” Int. J. Innov. Technol. Explor. Eng., vol. 11, no. 3, pp. 18–28, 2022, doi: 10.35940/ijitee.c9733.0111322.

[12] [12] B. Pratama and L. Y. Banowosari, “Perbandingan Metode Extreme Gradient Boosting (Xgboost) Dengan Long Short-Term Memory (LSTM) Untuk Prediksi Saham PT. Bank Mandiri Tbk. (BMRI),” J. Econ. Bussines Account., vol. 7, pp. 5631–5636, 2024.

[13] [13] T. K. Hwase and A. J. Fofanah, “Machine Learning Model Approaches for Price Prediction in Coffee Market using Linear Regression, XGB, and LSTM Techniques,” Int. J. Sci. Res. Sci. Technol., vol. 8, no. 6, pp. 10–48, 2021, doi: 10.32628/ijsrst218583.

[14] [14] N. Nagar, P. K. Jatav, M. Gupta, and A. Limone, “Performance Comparison of LSTM and SVR Models in Predicting Stock Prices,” J. Harbin Eng. Univ., vol. 44, no. 7, pp. 1–5, 2023.

[15] [15] A. Indika, N. Warusamana, E. Welikala, and S. Deegalla, “Ensemble Stock Market Prediction using SVM, LSTM, and Linear Regression,” TechRxiv, no. September 2019, pp. 1–6, 2021, doi: 10.36227/techrxiv.16626019.v1.

[16] [16] A. Karim and A. Rasheed, “Forecasting Modeling of Day of the Week Calendar Anomalies in Pakistan Stock Exchange: An Artificial Intelligence Perspective,” Bull. Bus. Econ., vol. 13, no. 2, pp. 436–447, 2023, doi: https://doi.org/10.61506/01.00351.

[17] [17]A. Saboor, A. Hussain, B. L. Y. Agbley, A. U. Haq, J. P. Li, and R. Kumar, “Stock Market Index Prediction Using Machine Learning and Deep Learning Techniques,” Intell. Autom. Soft Comput., vol. 37, no. 2, pp. 1325–1344, 2023, doi: 10.32604/iasc.2023.038849.

[18] [18] B. Gaye, D. Zhang, and A. Wulamu, “Improvement of Support Vector Machine Algorithm in Big Data Background,” Mathematical Problems in Engineering, vol. 2021. 2021. doi: 10.1155/2021/5594899.

[19] [19] Q. An, S. Rahman, J. Zhou, and J. J. Kang, “A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges,” Sensors, vol. 23, no. 9, 2023, doi: 10.3390/s23094178.

[20] [20] Y. Restiani and J. Purwadi, “Support Vector Machine for Classification: A Mathematical and Scientific Approach in Data Analysis,” J. Penelit. Pendidik. IPA, vol. 10, no. 11, pp. 9896–9903, 2024, doi: 10.29303/jppipa.v10i11.8122.

[21] [21] R. K. Halder, M. N. Uddin, M. A. Uddin, S. Aryal, and A. Khraisat, “Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications,” Journal of Big Data, vol. 11, no. 1. 2024. doi: 10.1186/s40537-024-00973-y.

[22] [22] B. G. Marcot and A. M. Hanea, “What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?,” Computational Statistics, vol. 36, no. 3. pp. 2009–2031, 2021. doi: 10.1007/s00180-020-00999-9.

[23] [23] D. Alita, A. D. Putra, and D. Darwis, “Analysis of classic assumption test and multiple linear regression coefficient test for employee structural office recommendation,” IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 15, no. 3, p. 295, 2021, doi: 10.22146/ijccs.65586.

[24] [24] D. Kinaneva, G. Hristov, P. Kyuchukov, G. Georgiev, P. Zahariev, and R. Daskalov, “Machine Learning Algorithms for Regression Analysis and Predictions of Numerical Data,” HORA 2021 - 3rd Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc., no. June, 2021, doi: 10.1109/HORA52670.2021.9461298.