Journal of Applied Data Sciences
Vol. 5, No. 4, December 2024, pp. 2079-2091
ISSN 2723-6471

Implementation of Stacking Technique Combining Machine Learning and Deep Learning Algorithms Using SMOTE to Improve Stock Market Prediction Accuracy

Ibnu Rasyid Munthe[1,*], Bhakti Helvi Rambe[2], Fauziah Hanum[3], Ade Trya Amanda[4], Anita Sri Rejeki Hutagaol[5], Richi Harianto[6]

[1,2,3,4,5]*Universitas Labuhanbatu, Jl. Sisingamangaraja, Labuhanbatu and 21418, Indonesia*

[6]*Institut Teknologi dan Sains Padang Lawas, Jl. Lintas Gunungtua-Padang Sidimpuan, Padang Lawas Utara and 22740, Indonesia*

(Received: July 13, 2024; Revised: September 14, 2024; Accepted: October 3, 2024; Available online: November 7, 2024)

Abstract

This study introduces a stacking technique that integrates machine learning (ML) and deep learning (DL) algorithms to enhance the accuracy of stock market trend predictions. The stacking model utilizes XGBoost and Random Forest as base models from the ML domain, while Logistic Regression and LSTM (Long Short-Term Memory) function as meta models to optimize predictive accuracy. A significant challenge in stock market data is class imbalance, where certain trends, such as stock price drops, are underrepresented. To mitigate this, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to generate synthetic data for the minority class. This approach helps the model better capture patterns from the underrepresented data while preserving essential information from the majority class. The implementation of SMOTE, coupled with the stacking technique, yielded a substantial improvement in prediction accuracy. The results showed that the Random Forest algorithm achieved an accuracy of 85% with precision, recall, and F1-score all at 85%, while XGBoost and Logistic Regression achieved accuracies of 82% and 81% respectively. For the deep learning models, LSTM reached an accuracy of 83%, while the Stacking Meta Model with LSTM achieved an accuracy of 83% with slightly better precision and recall at 84%. The stacking model, with Logistic Regression as the meta model, ultimately achieved the highest accuracy of 86%, outperforming individual models such as SVM (Support Vector Machine), LSTM, Random Forest, and Logistic Regression (LR). These findings demonstrate the efficacy of combining SMOTE with stacking to address data imbalance and improve stock market predictions. The novelty of this study lies in the integration of advanced ML and DL models within a stacking framework to handle class imbalance in financial datasets. Future research will explore the deployment of this model in a real-time web-based application to support investor decision-making in stock market trend analysis.

Keywords: Stock Market, Machine Learning, Deep Learning, Stacking, SMOTE

1. Introduction

The Jakarta Composite Index (JCI) measures the price performance of stocks listed on the Indonesia Stock Exchange (IDX) and serves as a key indicator in the Indonesian capital market, calculated using the Market Value Weighted Average Index [1]. Also known as the ICI or IDX Composite, the JCI provides an overall view of stock performance on the IDX, with data updated daily from Monday to Friday between 09:00 and 16:00 WIB, ensuring timely and accurate information [2]. For investors, the JCI acts as a guide in making investment decisions, such as whether to buy, sell, or hold stocks [3]. A rising JCI often prompts investors to sell shares to secure profits, though some may hold onto their stocks in hopes of further price increases [2]. Conversely, a declining JCI indicates bearish market conditions, where investors are advised to buy undervalued stocks with the expectation of future price recovery, or to employ a cut-loss strategy to minimize losses [4]. The term "stock bubble" is used to describe an unusual surge in stock prices, which can lead to sharp price corrections, highlighting the need for investor caution [5].

News related to stock movements in Indonesia has always been an interesting topic to analyze. This is reflected in various previous studies that have discussed the Indonesian stock market. One study conducted by [6] utilized machine

***Corresponding author: Ibnu Rasyid Munthe (ibnurasyidmunthe@gmail.com)**
DOI: https://doi.org/10.47738/jads.v5i4.421

learning to identify stock market trends using time series data. Ensemble methods such as random forest and gradient boosting were used to improve the model's performance in analyzing stock market movements. Another study [7] also examined the stock market by utilizing machine learning and deep learning to understand the factors influencing stock market trend movements. The results show that many important factors can affect the rise and fall of stock values. Additionally, there are studies that predict stock market value using sentiment analysis to calculate polarity scores. The results of this sentiment analysis were then used to predict stock prices using a random forest algorithm, which proved efficient in providing predictions based on sentiment data [8].

One of the main challenges in research related to predicting stock market movements is the limited generalizability of the developed models. While machine learning and deep learning methods may demonstrate good performance on Indonesian stock market data, these models might not achieve the same level of accuracy in other stock markets that possess different characteristics and dynamics. Differences in market structure, trading volume, as well as unique economic and political factors in each country can significantly affect the model's performance. Furthermore, models built with specific historical data might not be able to capture sudden changes or new patterns emerging in other markets. This suggests that a model effective in one geographical or temporal context may not be directly generalizable to another context without additional adjustments or validations. Therefore, further research is needed to explore ways to improve model generalization, such as expanding the dataset coverage, applying transfer learning techniques, or utilizing models that are more adaptive to changing market conditions.

This study aims to address this gap by proposing a novel stacking technique that integrates XGBoost and Random Forest as base models with Logistic Regression and LSTM as meta-models. By incorporating the SMOTE technique to handle data imbalance, this research offers a more comprehensive approach to predicting stock market trends in Indonesia. The novelty of this study lies in its exploration of stacking combinations and the application of deep learning models in the meta-layer, which has not been extensively investigated in previous studies. This approach is expected to enhance prediction accuracy and provide more reliable insights for investors in making informed decisions based on stock market trends.

Logistic Regression (LR) was chosen as the first meta-model due to its simplicity and effectiveness in combining predictions from multiple base models. LR is also known for its high interpretability, making it easier to understand the contribution of each base model to the final result. Moreover, LR can avoid overfitting because it has fewer parameters compared to more complex models, making it suitable for data with limited variation [9]. Meanwhile, LSTM was selected as the second meta-model due to its superior ability to handle sequential data and capture complex temporal patterns. In the context of stock market prediction, which is often influenced by historical trends and recurring patterns, LSTM can store long-term information that can enhance prediction accuracy. By using LSTM as the meta-model, it is expected to better combine predictions from the base models, especially in capturing patterns that cannot be detected by traditional models [10].

In this research, the analysis of stock market rises and falls reported in online media is conducted using machine learning and deep learning approaches. The SMOTE is used to overcome data imbalance, where the amount of data for some classes is significantly less compared to others. This imbalance can lead the model to be more likely to predict the majority class, ultimately decreasing prediction accuracy for the minority class [11]. Therefore, the use of the SMOTE technique is expected to improve the model's performance in predicting stock market movements.

To address this challenge, the study employs a combination of XGBoost and Random Forest as base models and evaluates the performance of two stacking methods: one with Logistic Regression as the meta model and another with LSTM as the meta model. This research contributes to a deeper understanding of how these ensemble methods can be optimized for stock market prediction in Indonesia and provides a basis for further research in integrating different machine learning and deep learning models to enhance forecasting performance.

2. Literature Review

Previous studies have discussed testing datasets with various algorithms, both single and combined. Table 1 is a previous study.

Table 1. Previous Research

No	Researcher	Dataset	Algorithm	Accuracy
1	[12]	Media Sosial	LSTM	72.85%
2	[13]	Stock Market	Random Forest	81.2%
3	[11]	Diabetes	SVM using SMOTE	Without SMOTE: 83% With SMOTE: 85.4%
4	[14]	Air Quality	Random Forest with SMOTE	Without SMOTE: 82% With SMOTE: 90%
5	[15]	Stock Price	Random Forest	58%

The first study [12] applied the LSTM model on a social media dataset, achieving an accuracy of 72.85%. In the second study [13], Random Forest was employed to predict stock market trends, resulting in an accuracy of 81.2%, showing a strong performance for this dataset. In the third study [11], focused on diabetes prediction, the SVM algorithm was used with and without SMOTE. Without SMOTE, the accuracy was 83%, while applying SMOTE increased the accuracy to 85.4%, demonstrating SMOTE's effectiveness in handling imbalanced data. In the air quality prediction study [14], Random Forest was used with and without SMOTE. The accuracy improved from 82% without SMOTE to 90% with SMOTE, further showing the importance of balancing data using SMOTE in improving model performance. Lastly, in a study predicting stock prices [15], Random Forest was used, but the accuracy was only 58%, highlighting the complexities involved in stock price prediction.

In this research, SMOTE was used to address data imbalance, where the minority class has significantly fewer samples than the majority class. SMOTE generates new synthetic samples for the minority class by interpolating between existing minority class data points. Unlike traditional oversampling methods, which merely duplicate existing data, SMOTE creates more representative variations in the minority class data, allowing machine learning models to learn better patterns and reduce bias towards the majority class. In this study, SMOTE was applied before using algorithms like XGBoost, Random Forest, and LSTM to enhance predictive accuracy on imbalanced datasets. The results showed a significant improvement in model accuracy, particularly in the diabetes and air quality datasets, where accuracy increased from 83% to 85.4% and 82% to 90%, respectively.

Additionally, the study employed the stacking technique, which combined XGBoost, Random Forest, and LSTM as base models, with Logistic Regression (LR) as the meta-model, alongside LSTM as an additional meta-model. Stacking leverages the strengths of each base algorithm to produce more accurate final predictions. The combination of SMOTE and stacking techniques proved to be effective in improving predictive performance on imbalanced datasets, particularly in disease prediction scenarios requiring high accuracy for minority classes.

This study not only demonstrates that SMOTE can significantly improve machine learning model performance on imbalanced datasets, but also that combining stacking techniques with powerful algorithms like XGBoost, Random Forest, and LSTM can further optimize predictive accuracy. This offers a strong contribution to the development of more reliable and effective predictive models for both medical and non-medical applications.

3. Methodology

Figure 1 is a methodology flow that is used to facilitate the conduct of this research.

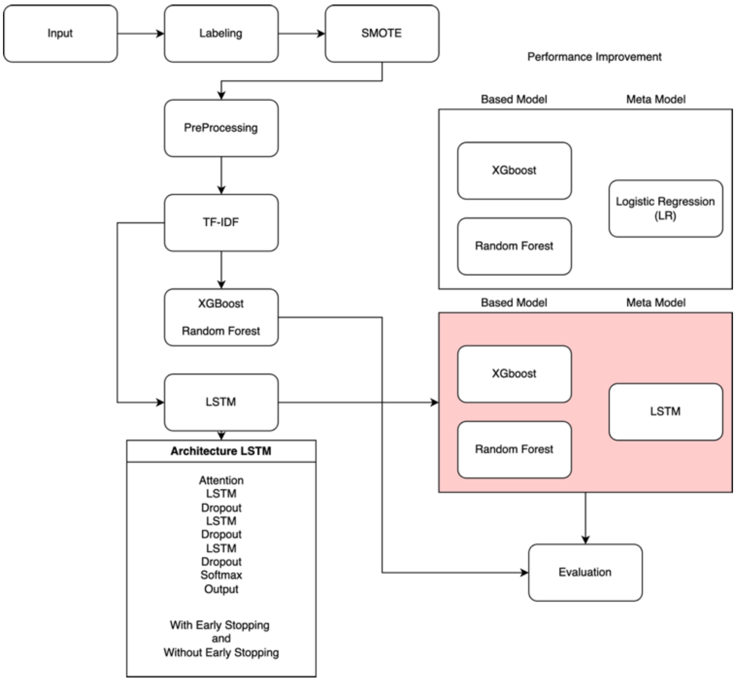

Figure 1. Research Methodology

3.1. Input Data

This figure illustrates the flow of the data analysis process starting with Data Input. In this context, the dataset used is a collection of news related to stock price movements. This data will then be processed to produce a prediction model that can be used to predict whether stock prices will rise or fall based on the information in the news. In the research, 335 records were processed.

3.2. Labelling

After the data is entered, the next step is Labeling. At this stage, each news data in the dataset is labeled based on whether the news indicates the stock price will “Go Down” or “Go Up”. This labeling process is important because the machine learning model to be built requires labeled data to train the model. However, in this research, there is an imbalance of data, where the down labeling is only 335 records and up is 714 records. Figure 2 is the actual label distribution.

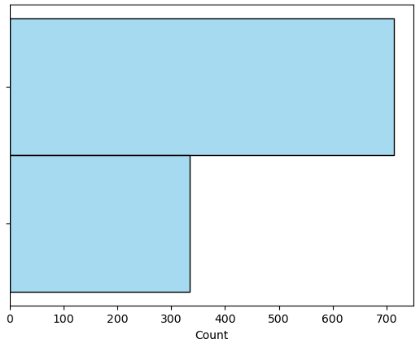

Figure 2. Label Distribution

3.3. SMOTE

After labeling, the SMOTE technique is used to overcome the imbalance class problem. Class imbalance occurs when the amount of data from one class, it is seen that the up labels are much more than the down labels. SMOTE helps overcome this problem by synthesizing additional data for the minority class, so that the data distribution becomes more balanced and the model can learn better. Figure 3 is the dataset that has been balanced with the oversampling technique.

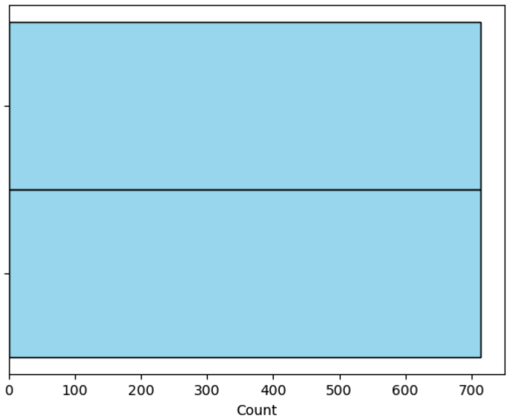

Figure 3. Balanced Dataset

Oversampling is preferred over undersampling in this context for several reasons. First, undersampling reduces the size of the majority class by removing instances, which can lead to the loss of valuable information and potentially degrade the model's performance. In contrast, oversampling techniques like SMOTE increase the number of instances in the minority class by generating synthetic samples. This helps preserve all the information in the majority class while also enhancing the representation of the minority class.

Moreover, SMOTE has the advantage of reducing overfitting, which is often associated with random oversampling techniques. Unlike random oversampling, which simply duplicates minority class instances, SMOTE generates new, synthetic instances by interpolating between existing ones. This approach not only balances the dataset but also helps the model learn more generalized decision boundaries, as the synthetic samples are spread throughout the feature space.

In stock market prediction, maintaining the complexity and integrity of the dataset is crucial for capturing the subtle patterns that influence price movements. By using SMOTE, the study ensures that the model is exposed to a diverse range of scenarios without losing important information, which would be a risk with undersampling. This, in turn, leads to improved prediction performance and a more robust model that can better handle the inherent volatility and unpredictability of the stock market.

3.4. Preprocessing

The preprocessing stage is a crucial step in preparing text data for use in machine learning models. In this study, several common preprocessing techniques were applied, including tokenization, removal of punctuation, conversion of all text to lowercase, and removal of stop words [16]. The first step in this research involves data cleaning, which is used to remove links, symbols, and other unnecessary elements [17]. The subsequent preprocessing step is case folding, which is applied to convert all letters in the document to lowercase, accepting only characters from 'a' to 'z'. Non-alphabetic characters are removed and considered as delimiters [18].

The next step is tokenizing, which serves as the backbone for various digital applications, enabling machines to process and understand large amounts of textual data. By breaking down the text into manageable pieces, tokenization facilitates more efficient and accurate data analysis. Following the tokenizing stage is filtering, which is used to extract significant words from the tokenized output [19]. Common words that typically appear and do not carry meaningful information are referred to as stopwords, such as conjunctions like "and," "that," "which," "after," and others. The removal of stopwords can reduce the index size and processing time, and it can also lower the noise level. However, stopping does not always improve retrieval performance. Carelessly constructing a stopword list (referred to as a stoplist) can degrade the performance of the Information Retrieval (IR) system. There is no conclusive evidence that the use of stopping will always enhance retrieval performance, as results tend to vary across different studies [20]. In addition to these techniques, this study also employed stemming to further optimize the text representation.

Stemming was used to reduce words to their root forms by removing suffixes such as “-ing,” “-ed,” or “-s.” This technique helps reduce the complexity of the data by consolidating morphological variations of the same word into a simpler form. For example, the words “running,” “runner,” and “runs” are all converted to the root word “run.” By using stemming, the model can focus on the core meaning of different words that share the same root, thus reducing redundancy and improving efficiency during model training [21].

Stemming was chosen over lemmatization in this study because it is simpler and faster to implement and can yield sufficient results in the context of sentiment analysis and stock market trend prediction, where word variations typically do not significantly affect the overall meaning of the text. Additionally, the lemmatization process, which is more complex and requires additional information such as part of speech (POS) tagging, was not used due to considerations of time efficiency and computational resources [22].

After the stemming process, the processed text was then converted into numerical representation using the TF-IDF (Term Frequency-Inverse Document Frequency) technique. TF-IDF is used to measure the importance of a word in a document relative to the set of other documents in the dataset. A higher TF-IDF value indicates that the word is more important in the context of a particular document. This representation is then used as input for the machine learning models.

3.5. TF-IDF

TF-IDF is a widely used feature extraction method in text processing, which converts textual data into numerical values that can be utilized by machine learning algorithms. The TF-IDF value is calculated by multiplying two metrics: term frequency (TF), which measures how often a word appears in a document, and inverse document frequency (IDF), which measures how unique or rare the word is across the entire dataset. This combination allows TF-IDF to capture not only the relevance of a word within a specific document but also its importance across the whole dataset [23].

The choice of TF-IDF over other methods such as word embeddings (e.g., Word2Vec or BERT) in this study was based on several considerations. First, TF-IDF is simpler and more interpretable compared to word embeddings [24]. It provides a straightforward way to quantify word importance within a document, where each value in the TF-IDF vector clearly represents the significance of a specific word, making it easier to understand the contribution of each term to the model's predictions [25]. Additionally, TF-IDF is better suited for small to medium-sized datasets like the one used in this study [26]. Word embeddings such as Word2Vec or BERT require larger datasets to be effectively trained because they are designed to capture complex semantic relationships between words. In contrast, TF-IDF does not rely on word context or large corpora, making it more appropriate for datasets with limited size and scope [27].

Moreover, methods like BERT and Word2Vec are computationally intensive, both in terms of training and inference. For instance, BERT utilizes deep neural networks that require significant computational resources, which may not be feasible in all research environments. On the other hand, TF-IDF is computationally efficient, making it a more practical choice for studies with limited resources. While word embeddings are capable of capturing deeper semantic meanings and contextual relationships between words, TF-IDF offers a good balance between complexity, interpretability, and computational efficiency, making it suitable for use in this study.

3.6. Modeling with Machine Learning

At this stage, the data that has been represented with TF-IDF is used to build a prediction model using two Machine Learning algorithms, namely XGBoost and Random Forest. Both algorithms are well-known in the machine learning world for their ability to handle complex datasets and provide accurate predictions [28], [29].

3.7. LSTM

In addition to using traditional machine learning models, this research also involves LSTM, a type of artificial neural network that is particularly effective for processing sequence data such as text or time series [30]. Figure 1 shows the LSTM architecture used, which involves multiple LSTM layers with dropouts in between to prevent overfitting, as well as a softmax layer at the end to output predictions. Testing was performed with the application of Early Stopping, a technique used to stop training when the model begins to show signs of overfitting [31].

3.8. Stacking with Logistic Regression (LR) Meta model

The first stacking technique is performed using XGBoost and Random Forest as base models, and Logistic Regression as a meta model. In this stacking technique, predictions from the base model will be used as input for the meta model, which then provides the final prediction [32]. Logistic Regression was chosen as the first meta model because it is simple yet effective in combining predictions from several base models [9].

3.9. Stacking with LSTM Meta model

The second stacking uses LSTM as the meta model. Like the first stacking, XGBoost and Random Forest are used as base models, but the prediction results from these base models are further processed by the LSTM model. The use of LSTM as a meta model is intended to capture complex patterns that may not be captured by simpler meta models such as Logistic Regression.

3.10. Evaluation

The last stage in this flow is Evaluation. Once the model has been trained and optimized, it is evaluated to see how well it can analyze stock price movements based on the news data. This evaluation is done to measure the performance of the model, for example in terms of accuracy, precision, recall, and other evaluation metrics, to ensure that the model is reliable for real-world use [33]. In this study, two main techniques were used to prevent overfitting: SMOTE and early stopping for the LSTM model. Overfitting is a common issue, especially when working with relatively small datasets, as the model tends to learn patterns and noise specific to the training data, which can lead to poor performance on new, unseen data.

First, SMOTE was applied to address the problem of data imbalance, which is one of the factors contributing to overfitting. By balancing the number of instances between the majority and minority classes, SMOTE helps the model to learn patterns from the minority class as well, instead of only focusing on the majority class. This technique enhances the model's robustness and its ability to generalize better to new data.

Second, for the LSTM model, early stopping was used as a preventive measure against overfitting. Early stopping monitors the model's performance on a validation set during training and automatically stops training when the model's performance starts to degrade or stagnate, indicating signs of overfitting. By halting the training process at the right time, early stopping prevents the model from over-learning the training data, thus preserving better performance on validation and test data.

Although this study did not extensively employ techniques like cross-validation, regularization, or hyperparameter tuning, the combination of SMOTE and early stopping was sufficient to mitigate the risk of overfitting within the context of the dataset used. These two techniques provide a balanced approach, maintaining the complexity and performance of the model while effectively predicting stock market trends. For future research, additional techniques such as cross-validation and hyperparameter tuning could be explored to further enhance the model's resistance to overfitting and to ensure more generalizable results on a larger scale.

4. Results and Discussion

4.1. Result

The first test results are using stacking with XGBoost and Random Forest based algorithms, for the meta model using logistic regression. Figure 4 is the classification report of stacking using LR.

```
              precision    recall  f1-score   support

        naik       0.85      0.87      0.86       220
       turun       0.86      0.84      0.85       209

    accuracy                           0.86       429
   macro avg       0.86      0.86      0.86       429
weighted avg       0.86      0.86      0.86       429

Accuracy : 0.8578088578088578
-------------------------------------------
```

Figure 4. Classification Report Stacking using LR

Figure 4 displays the results of the evaluation report of the binary classification model, which focuses on two classes: “up” and ‘down’. The metrics used to evaluate the model's performance include precision, recall, F1-score, and support. Precision measures how precise the model's prediction is when it says that data belongs to a certain class. For example, a precision of 0.85 for the “up” class indicates that 85% of all “up” predictions made by the model actually belong to the “up” class. On the other hand, the precision for the “down” class is slightly higher at 0.86.

Recall measures how good the model is at finding all instances that actually belong to a particular class. A recall of 0.87 for the “up” class indicates that the model successfully found 87% of all “up” instances in the data. For the “down” class, the recall is slightly lower at 0.84. F1-Score, which is the harmonic mean of precision and recall, gives a more balanced picture of the model's performance. The F1-score of 0.86 for the “up” class shows that the model has a good balance between precision and recall. The F1-score for the “down” class is slightly lower at 0.85. Finally, the support shows the number of instances in the data that are actually in each class. There are 220 instances for the “up” class and 209 instances for the “down” class. Overall, the model showed an accuracy of 0.86, which indicates a fairly good performance in predicting the “up” and “down” classes in the dataset used.

This research also tested with the basic algorithm which can be seen in table 2.

Table 2. Comparison Based Algorithm

No	Algorithm	Accuracy	Precision	Recall	F1-Score
1	Random Forest	85%	85%	85%	85%
2	XGBoost	82%	82%	82%	82%
3	Logistic Regression	81%	81%	81%	81%

Table 2 compares the performance of three machine learning algorithms namely Random Forest, XGBoost, and Logistic Regression. Based on key metrics such as accuracy, precision, recall, and F1-Score. The results from the table show that Random Forest has the best performance with 85% for all metrics. This indicates that Random Forest is very effective in predicting the classes in the dataset with low error rate and good consistency in prediction. Meanwhile, XGBoost performed slightly lower than Random Forest, with all metrics at 82%. While XGBoost is known to be strong in handling data with many features and interactions between features, in this context, Random Forest still excels in terms of overall performance.

Then Logistic Regression, which has a metric of 81% for all parameters, shows the lowest performance among the three algorithms. Logistic Regression, although simpler and faster, does not seem to be able to handle data complexity as well as Random Forest and XGBoost. Furthermore, when these three models are compared with a stacking approach, which combines the strengths of multiple models, the end result can exceed the performance of any single model. In stacking, the predictions from Random Forest, XGBoost, and Logistic Regression are used together to make the final prediction, and get a 1% improvement to 86%. Thus, although Random Forest has shown the best performance individually, the combination of models through stacking can provide superior results. Then this study also uses ROC (Receiver Operating Characteristic).

ROC is used to evaluate the performance of a classification model, particularly in the context of binary classification. It helps measure the model's ability to distinguish between positive and negative classes at various thresholds [34]. By displaying the relationship between the True Positive Rate (TPR) or Sensitivity and the False Positive Rate (FPR) at different threshold values, the ROC curve allows us to understand how well the model can predict the positive class without generating too many false positives. The AUC (Area Under the Curve) value derived from the ROC curve provides an overall indication of the model's performance. AUC ranges from 0.5 to 1, where a value of 0.5 indicates a model that is no better than random guessing, while a value of 1 indicates a perfect model. ROC is particularly useful for selecting and optimizing the appropriate threshold for a classification model and comparing the performance of several classification models. Figure 5 shows the ROC result of the Stacking Technique (Base Model: XGBoost & Random Forest. Meta Model: Logistic Regression).

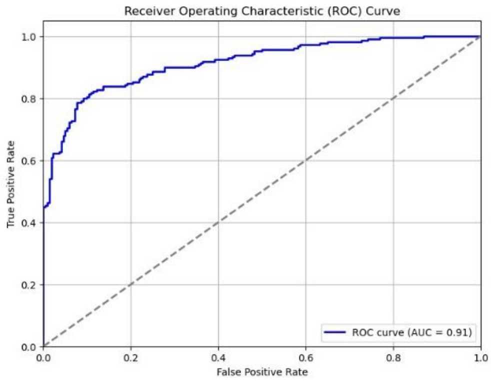

Figure 5. ROC

The ROC curve displayed indicates that the model has an AUC value of 0.91, which demonstrates a very good ability to distinguish between positive and negative classes. An AUC value close to 1 show that the model is able to identify most of the positive cases with a low error rate. Although the model's accuracy is 86%, this high AUC indicates that the model not only performs well overall but also effectively handles class imbalance.

Furthermore, this research also compared using deep learning algorithm, LSTM. In addition, it also uses the stacking technique using the LSTM meta model. Table 3 is a comparison with LSTM.

Table 3. Comparison Deep Learning Algorithm

No	Algorithm	Accuracy	Precision	Recall	F1-Score
1	LSTM	83%	83%	83%	83%
2	Stacking Meta Model With LSTM	83%	84%	84%	83%

Table 3 compares the performance between the LSTM model and the Stacking approach with the LSTM Meta Model based on several evaluation metrics, namely accuracy, precision, recall, and F1-Score. The LSTM model itself performed quite well with 83% for all metrics, indicating a consistent ability to predict the data. However, when LSTM was used as a meta model in the stacking approach, there was a small increase in precision and recall, which rose to 84% each, although accuracy and F1-Score remained the same at 83%. This shows that stacking with LSTM as a meta model is able to provide a slight improvement in identifying and capturing the correct class more precisely and consistently compared to stand-alone LSTM. This approach could provide an advantage in scenarios where prediction accuracy and the ability to capture all instances of a particular class are critical.

In this study, the evaluation of model performance focuses on four main metrics: accuracy, precision, recall, and F1-Score, as presented in table 1. These metrics were chosen because they are standard measures used to assess the predictive capability and overall effectiveness of classification models, especially in contexts where the balance between correctly identifying positive and negative classes is crucial. However, we acknowledge that these metrics alone may not provide a comprehensive assessment of the model's suitability for real-world applications, particularly in terms of computational complexity, training time, and scalability.

The decision to prioritize these four metrics was based on their relevance to the specific goals of this study, which primarily aims to evaluate the accuracy and reliability of different models in predicting stock market trends. These metrics provide clear insights into how well the models can classify data points correctly and handle imbalanced classes, which are critical aspects for ensuring the practical utility of the models in a predictive context.

Other important performance metrics, such as computational complexity, training time, and scalability, were not included in table 1 because the primary focus of this study was to compare the predictive performance of different machine learning models. The computational resources and time constraints of the research environment also influenced this choice, as conducting extensive experiments to evaluate these additional metrics would require significantly more resources and time than was available. In practical scenarios, these additional metrics indeed play a crucial role in determining the feasibility of deploying the model at scale. For future work, we plan to extend the evaluation by considering these factors to provide a more holistic assessment of the model's applicability in real-world environments. This will include measuring the training time required for each model, the computational resources needed for inference, and the scalability of the models when applied to larger datasets.

By focusing initially on accuracy, precision, recall, and F1-Score, this study aimed to establish a foundational understanding of each model's classification performance. Expanding the evaluation framework to include computational complexity, training time, and scalability will provide deeper insights and support the implementation of these models in practical, real-world applications.

4.2. Discussion

The results of stacking with Logistic Regression as a meta model, which obtained an accuracy of 86%, refer to the previous discussion where Random Forest and XGBoost were used as base models. In this approach, Logistic Regression as a meta model managed to combine the predictions from both base models more effectively, resulting in higher accuracy compared to individual models such as Random Forest, XGBoost, or even LSTM. When compared to stacking using LSTM as a meta model, which only achieved 83% accuracy, the approach with Logistic Regression clearly performed better. This shows that although LSTM is a powerful model, in this scenario, Logistic Regression as a meta model is better able to integrate the predictive power of Random Forest and XGBoost, thus providing superior results. Meanwhile, Random Forest as an individual model still performs very well with 85% accuracy, but is still slightly inferior to the stacking results using Logistic Regression. Therefore, the stacking approach with Logistic Regression as a meta model proved to be able to utilize the power of the base models more optimally, resulting in better accuracy and overall performance than the other models discussed. The following comparisons with previous research are presented in table 4.

Table 4. Comparison With Previous Research

No	Researcher	Algorithm	Accuracy
1	[17]	SVM	84%
2	[35]	BiLSTM	79%
3	[10]	CNN-LSTM	85%
4	[18]	Random Forest	71%
5	[36]	Logistic Regression	84%
6	This Study	Stacking Technique (Based Model: XGBoost & Random Forest. Meta Model: Logistic Regression)	86%

Table 4 compares the accuracy results of various studies that use different algorithms to perform sentiment analysis. Research by Panji Bintoro, et al [17] used SVM and achieved an accuracy of 84.03%. Meanwhile, Ade Romahony, et al [35] with the BiLSTM approach obtained 79% accuracy, which is the lowest accuracy among all the studies listed. Research by Anam, et al [10] using the CNN-LSTM combination managed to achieve a fairly high accuracy of 85%. In contrast, research by Muhaddisi, et al [18] with Random Forest showed lower accuracy than the others, which was 71%. Satya Abdul Halim Bahtiar, et al [36] used Logistic Regression and managed to get an accuracy of 84.58%, slightly higher than that obtained by Panji Bintoro with SVM. The study under discussion, which used a stacking technique with XGBoost and Random Forest as base models and Logistic Regression as a meta model, achieved the highest accuracy among all studies, at 86%. These results show that the stacking approach with Logistic Regression as a meta model is more effective in improving prediction accuracy compared to the use of individual models such as SVM, BiLSTM, CNN-LSTM, or Random Forest.

The results of this study are planned to be implemented in the form of a web-based application with a graphical user interface (GUI). This application will serve as a tool for investors to identify stock market trends in real-time. By utilizing the developed stacking model, this application will allow users to input the latest market data and receive predictions on stock price movements quickly and accurately. The web application will provide features that enable investors to monitor stock price movements and receive optimal action recommendations, such as when to buy, sell, or hold stocks based on the model's predictions. With an interactive and user-friendly interface, this application will assist investors in making more timely and informed investment decisions, thereby minimizing potential losses due to unforeseen market fluctuations. In addition, the application will be equipped with risk alert features that notify users of significant market changes or high potential losses. This feature is expected to help investors take preventive actions and reduce the risk of losses. Thus, the implementation of the stacking model in the form of a web application will not only make it easier for investors to access market predictions but also provide support in managing investment risks.

Further development of this application could also include integration with stock trading platforms, allowing investment decisions to be automated based on the model's predictions. This implementation is expected to help investors access relevant information more easily and make quicker decisions, thereby increasing profit opportunities and reducing the risk of losses [37], [38]. Through this application, the results of this study can be directly utilized by investors to support their investment decisions while providing a practical solution to address the challenges of applying stock market prediction models in the real world.

5. Conclusion

This research successfully demonstrated that the use of stacking technique with Logistic Regression as a meta model provides the most accurate results in predicting stock market trends in Indonesia, with accuracy reaching 86%. This technique outperformed various other machine learning and deep learning algorithms, including Random Forest, XGBoost, and LSTM, tested individually. The use of SMOTE to address data imbalance was also shown to significantly improve model performance. These results indicate that the stacking approach is able to integrate the strengths of multiple models, thus providing more accurate and reliable predictions.

For future research, it is recommended to further explore the combination of other models in the stacking technique, as well as consider the use of more diverse data, including fundamental and technical data, to improve prediction accuracy. In addition, the application of more complex deep learning methods and testing with real-time data may provide better results and be more relevant to the ever-changing market dynamics. Further research can also explore the impact of various optimization techniques on the performance of stacking models.

6. Declarations

6.1. Author Contributions

Conceptualization: I.R.M., B.H.R., F.H., A.T.A., A.S.R.H., dan R.H.; Methodology: A.S.R.H.; Software: I.R.M.; Validation: I.R.M., A.S.R.H., dan R.H.; Formal Analysis: I.R.M., A.S.R.H., dan R.H.; Investigation: I.R.M.; Resources: A.S.R.H.; Data Curation: A.S.R.H.; Writing Original Draft Preparation: I.R.M., A.S.R.H., dan R.H.; Writing Review and Editing: A.S.R.H., I.R.M., dan R.H.; Visualization: I.R.M.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] U. Septepana, P. L. Obrien, A. F. Surbakti, F. Astuty, and W. A. Ginting, "Analysis of Macroeconomic Factors that Influence the Movement of the Composite Stock Price Index (IHSG) on the Indonesian Stock Exchange for the Period 2013-2023," *IJEBIR*, vol. 3, no. 4, pp. 142–155, 2024.

[2] Fuad and I. Yuliadi, "Determinants of the Composite Stock Price Index (IHSG) on the Indonesia Stock Exchange," *Journal of Economics Research and Social Sciences*, vol. 5, no. 1, pp. 27–41, 2021, doi: 10.18196/jerss.v5i1.11002.

[3] C. Kumala Dewi and R. Masithoh Haryadi, "Volatility Analysis Of Indonesia Composite Index During COVID-19," *JEM*, vol. 15, no. 2, pp. 138–148, 2021, doi: 10.30650/jem.v15i2.2225.

[4] A. Logitama, L. Setiawan, and A. Hayat, "Control Behaviors Affecting Investors Investment Decision Making (Studies on Students at Higher Education South Kalimantan)," *International Journal of Economics, Business, and Accounting Research (IJEBAR)*, vol. 5, no. 1, pp. 278–291, 2021.

[5] A. S. Rakhmat and M. H. Fahamsyah, "Literature Review on Detecting Bubbles in Stock Market," *East Asian Journal of Multidisciplinary Research*, vol. 2, no. 4, pp. 1499–1504, Apr. 2023, doi: 10.55927/eajmr.v2i4.3828.

[6] A. Anwar, A. A. Patil, S. Choudari, Chetna, and C. S. Kiran, "Machine Learning Insights for Stock Market Trend Identification," *International Journal of Intelligent Systems and Applications in Engineering IJISAE*, vol. 12, no. 10s, pp. 567–575, 2024.

[7] S. L. V. V. D. Sarma, D. V. Sekhar, and G. Murali, "Stock market analysis with the usage of machine learning and deep learning algorithms," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 552–560, Feb. 2023, doi: 10.11591/eei.v12i1.4305.

[8] S. Kompella and K. C. Chilukuri, "Stock Market Prediction Using Machine Learning Methods," *International Journal of Computer Engineering and Technology*, vol. 10, no. 3, pp. 20–30, 2019.

[9] A. Desfiandi and B. Soewito, "Student Graduation Time Prediction Using Logistic Regression, Decision Tree, Support Vector, And Adaboost Ensemble Learning," International Journal of Information System and Computer Science) IJISCS, vol. 7, no. 3, pp. 195–199, 2023, doi: 10.56327/ijiscs.v7i2.1579.

[10] M. K. Anam, S. Defit, Haviluddin, L. Efrizoni, and M. B. Firdaus, "Early Stopping on CNN-LSTM Development to Improve Classification Performance," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, 2024, doi: 10.47738/jads.v5i3.312.

[11] N. G. Ramadhan, "Comparative Analysis of ADASYN-SVM and SMOTE-SVM Methods on the Detection of Type 2 Diabetes Mellitus," *Scientific Journal of Informatics*, vol. 8, no. 2, pp. 276–282, Nov. 2021, doi: 10.15294/sji.v8i2.32484.

[12] M. A. Nurrohmat and A. SN, "Sentiment Analysis of Novel Review Using Long Short-Term Memory Method," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 13, no. 3, pp. 209–218, Jul. 2019, doi: 10.22146/ijccs.41236.

[13] W. Khan, M. A. Ghazanfar, M. A. Azam, A. Karami, K. H. Alyoubi, and A. S. Alfakeeh, "Stock market prediction using machine learning classifiers and social media, news," *J Ambient Intell Humaniz Comput*, vol. 13, no. 7, pp. 3433–3456, Jul. 2022, doi: 10.1007/s12652-020-01839-w.

[14] F. P. Arifianti and A. Salam, "XGBoost and Random Forest Optimization using SMOTE to Classify Air Quality," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 1, pp. 1–8, Nov. 2024, doi: 10.26877/asset.v6i1.18136.

[15] Z. Akhter and H. Raza, "Stock Price Prediction Using Machine Learning: Evidence from Pakistan Stock Exchange," *International Journal of Business & Management*, vol. 19, no. 1, pp. 1–26, Jun. 2024, doi: 10.52015/nijbm.v19i1.197.

[16] N. Garg and K. Sharma, "Text pre-processing of multilingual for sentiment analysis based on social network data," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 776–784, Feb. 2022, doi: 10.11591/ijece.v12i1.pp776-784.

[17] P. Bintoro, T. H. Andika, A. F. Yulia, and P. Widiandana, "Sentiment Analysis on Twitter Using Machine Learning Approach," *International Journal of Software Engineering and Informatics*, vol. 1, no. 1, pp. 33–39, 2023.

[18] A. Muhaddisi, B. N. Prastowo, and D. U. K. Putri, "Sentiment Analysis With Sarcasm Detection On Politician's Instagram," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 4, pp. 349, Oct. 2021, doi: 10.22146/ijccs.66375.

[19] A. Hossain, M. Karimuzzaman, M. M. Hossain, and A. Rahman, "Text mining and sentiment analysis of newspaper headlines," *Information (Switzerland)*, vol. 12, no. 10, pp. 1-15, Oct. 2021, doi: 10.3390/info12100414.

[20] N. W. Susanto and H. Suparwito, "SVM-PSO Algorithm for Tweet Sentiment Analysis #BesokSenin," *Indonesian Journal of Information Systems (IJIS)*, vol. 6, no. 1, pp. 36–47, 2023, doi: 10.24002/ijis.v6i1.7551.

[21] W. Rifai and E. Winarko, "Modification of Stemming Algorithm Using A Non Deterministic Approach To Indonesian Text," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 13, no. 4, pp. 379–388, Oct. 2019, doi: 10.22146/ijccs.49072.

[22] Z. Abidin, A. Junaidi, and Wamiliana, "Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review," *Journal of Information Systems Engineering and Business Intelligence*, vol. 10, no. 2, pp. 217–231, Jun. 2024, doi: 10.20473/jisebi.10.2.217-231.

[23] V. Talasila, M. V Mohan, and N. M. R, "Enhancing Text-to-Image Synthesis with an Improved Semi-Supervised Image Generation Model Incorporating N-Gram, Enhanced TF-IDF, and BOW Techniques," *International Journal of Intelligent Systems and Applications in Engineerin*, vol. 11, no. 7s, pp. 381–397, 2023.

[24] S. Akuma, T. Lubem, and I. T. Adom, "Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3629–3635, 2022, doi: 10.1007/s41870-022-01096-4.

[25] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, pp. 1-68, 2019, doi: 10.3390/info10040150.

[26] A. Amalia, O. S. Sitompul, E. B. Nababan, and T. Mantoro, "A Comparison Study of Document Clustering using Doc2Vec Versus TFIDF Combined with LSA for Small Corpora," *J Theor Appl Inf Technol*, vol. 98, no. 17, pp. 3644-3657, Sep. 2020.

[27] D. S. Asudani, N. K. Nagwani, and P. Singh, "Impact of word embedding models on text analytics in deep learning environment: a review," *Artif Intell Rev*, vol. 56, no. 9, pp. 10345–10425, Sep. 2023, doi: 10.1007/s10462-023-10419-1.

[28] D. Tarwidi, S. R. Pudjaprasetya, D. Adytia, and M. Apri, "An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach," *MethodsX*, vol. 10, no. 1, pp. 1–12, Aug. 2023, doi: 10.1145/2939672.2939785.

[29] L. K. Shrivastav and R. Kumar, "An Ensemble of Random Forest Gradient Boosting Machine and Deep Learning Methods for Stock Price Prediction," *Journal of Information Technology Research*, vol. 15, no. 1, pp. 1–19, Nov. 2021, doi: 10.4018/jitr.2022010102.

[30] N. B. Aji, Kurnianingsih, N. Masuyama, and Y. Nojima, "CNN-LSTM for Heartbeat Sound Classification," *International Journal on Informatics Visualization*, vol. 8, no. 2, pp. 735–741, 2024, doi: 10.62527/joiv.8.2.2115.

[31] M. Vilares Ferro, Y. Doval Mosquera, F. J. Ribadas Pena, and V. M. Darriba Bilbao, "Early stopping by correlating online indicators in neural networks," *Neural Networks*, vol. 159, no. 1, pp. 109–124, Feb. 2023, doi: 10.1016/j.neunet.2022.11.035.

[32] B. L. V. S. R. Krishna, V. Mahalakshmi, and G. K. M. Nukala, "A Stacking Model for Outlier Prediction using Learning Approaches," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 2s, pp. 629–638, 2023.

[33] L. K. Ramasamy, S. Kadry, Y. Nam, and M. N. Meqdad, "Performance analysis of sentiments in Twitter dataset using SVM models," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2275–2284, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2275-2284.

[34] F. S. Nahm, "Receiver operating characteristic curve: overview and practical use for clinicians," *Korean J Anesthesiol*, vol. 75, no. 1, pp. 25–36, Feb. 2022, doi: 10.4097/kja.21209.

[35] A. Romadhony, S. Al Faraby, R. Rismala, U. N. Wisesti, and A. Arifianto, "Sentiment Analysis on a Large Indonesian Product Review Dataset," *Journal of Information Systems Engineering and Business Intelligence*, vol. 10, no. 1, pp. 167–178, 2024, doi: 10.20473/jisebi.10.1.167-178.

[36] S. A. H. Bahtiar, C. K. Dewa, and A. Luthfi, "Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling," *Journal of Information Systems and Informatics*, vol. 5, no. 3, pp. 915–927, Aug. 2023, doi: 10.51519/journalisi.v5i3.539.

[37] J. Grudniewicz and R. Ślepaczuk, "Application of machine learning in algorithmic investment strategies on global stock markets," *Res Int Bus Finance*, vol. 66, no. 1, pp. 1–24, Oct. 2023, doi: 10.1016/j.ribaf.2023.102052.

[38] Y. Irawan, "Decision Support System For Employee Bonus Determination With Web-Based Simple Additive Weighting (SAW) Method In PT. Mayatama Solusindo," *Journal of Applied Engineering and Technological Science*, vol. 2, no. 1, pp. 7–13, Nov. 2020, doi: 10.37385/jaets.v2i1.162.