Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Event Detection Optimization Through Stacking Ensemble and BERT Fine-tuning For Dynamic Pricing of Airline Tickets

NUR ALAMSYAH, SAPARUDIN, and ANGELINA PRIMA KURNIATI
School of Computing, Telkom University, Bandung 40257, Indonesia

Corresponding author: Saparudin (e-mail: saparudin@telkomuniversity.ac.id).

This work supported by the Telkom University's Research and Community Service (PPM) Directorate through the Doctoral Dissertation Research (PDD) Grant and also the PPM Directorate of Bandung Informatics and Business University.

ABSTRACT Dynamic pricing of airline tickets in competitive markets requires innovation that responds to market changes. Dynamic pricing is also influenced by public events, such as sporting events, music concerts, and more. This study aims to increase airline ticket revenue by optimizing flight occupancy during events. The data used is obtained from social media platform Twitter (X), with eight event classifications: soccer events, music concerts, volcanic eruptions, earthquakes, riots, floods, motorcycle racing, and others (non-events). We used a stacking ensemble method for data labeling and fine-tuned the BERT model for event detection. The stacking ensemble method achieved an accuracy rate of 0.99, while the fine-tuned BERT model produced an accuracy rate of 0.94. These results show a significant contribution to improving the accuracy and effectiveness of dynamic pricing. These findings not only offer a solution to the dynamic pricing challenge but also open opportunities to increase revenue by understanding event sentiment, providing competitiveness and flexibility in a dynamic market. With a focus on accurate event detection, this research paves the way for the development of more intelligent and adaptive dynamic pricing models by combining the strengths of the Stacking Ensemble labeling technique and BERT model fine-tuning to improve model accuracy.

INDEX TERMS Dynamic Pricing, Event Detection, Stacking Ensemble, BERT Model Fine-Tuning

I. INTRODUCTION

THE dynamics of airplane ticket prices have become a significant subject of discussion, especially on social media platforms [1]. Recent studies show a strong correlation between the occurrence of events in a region and their impact on airfares to that destination. [2]. However, the current dynamic airfare pricing systems have not fully utilized the potential of social media platforms [3]. The increase in ticket sales associated with events, and the resulting rise in revenue, underscores the importance of integrating social media insights into dynamic pricing models [4]. Therefore, event detection through social media is a critical factor in this evolving dynamic pricing paradigm. Recognizing the significant influence of social media sentiment on flight occupancy, this study emphasizes the role of event detection and its impact on seat occupancy.

Various studies reveal that the conventional approaches to airfare pricing have not fully exploited the potential of event detection [5]. These studies highlight the gap between existing practices and the optimal potential that could be achieved. Regarding event detection, there is a pressing need to delve deeper into how information from social media can be effectively integrated into dynamic pricing strategies [6]. This gap presents a significant opportunity to enhance the accuracy, responsiveness, and adaptability of existing pricing models. Therefore, this research aims to bridge this gap and propose an innovative solution to maximize airfare revenue through more efficient and intelligent event detection.

Event detection from Twitter data promises invaluable insights into ongoing events worldwide [7]. However, extracting meaningful signals from the unstructured and often ambiguous nature of tweets remains a challenge [8]. Traditional methods face difficulties due to the complexity of data labeling and model limitations [9]. One of the primary challenges in event detection on Twitter is the accuracy of data labeling [10].Manual data labeling, often performed by humans, is prone to errors and inconsistencies, reducing the accuracy of event detection models [11]. Additionally, manual labeling is a time-consuming and costly process [12].

Another major challenge is the limitation of traditional models [13]. Classification models such as Random Forest, Decision Trees, and Support Vector Machines often struggle to capture the subtle nuances of language used in tweets to describe events [14]. Consequently, these models may make errors in identifying events, particularly when slang or complex, rare events are involved [15]. Beyond labeling complexity and model limitations, another challenge in event detection on Twitter is the diversity of language, which can lead to ambiguity, varied interpretations, and the fast pace of information dissemination. Furthermore, cultural and contextual differences can influence how people communicate events on Twitter event [16].

Several studies have addressed model development for Twitter (X) data labeling. Hasan *et al.* [17], developed an automatic labeling model for Twitter data using regular expressions, enhanced by a custom word library generated through manual tweet analysis. Erdman *et al.* [18], highlighted the benefits of automatic labeling models, which effectively eliminate manual labeling costs. Nirbhaya *et al.* [19], conducted a multi-class labeling study on the Twitter account of the Metro Jakarta Police Department, categorizing traffic flow data into labels for smooth, congested, weather conditions, and traffic accidents. Bhardwaj *et al.* [20], compared four approaches—keyword search-based (Plain-Seed-Query), information retrieval-based (Temporal Query Expansion), Word2Vec embeddings, and semantic retrieval (ArmaTweet)—across six event classifications, finding that ArmaTweet outperformed other methods in five categories.

Neruda *et al.* [21], conducted a study on traffic event detection from Twitter using a combination of CNN and BERT models. This research demonstrated that by optimizing kernel size and the number of filters in CNN, the model can outperform bidirectional long-short-term memory networks (LSTMs), multilayer perceptrons (MLPs), Random Forest, and SVM models in terms of F1 score and generalization accuracy. In our study, we focus on event detection using Indonesian Twitter (X) data across several event categories, including soccer matches, motorcycle races, concerts, earthquakes, volcanic eruptions, riots, floods, and others. We gathered this data directly from Twitter using the API, ensuring the privacy of users and utilizing the data solely for research purposes.

In the context of dynamic airline ticket pricing, this research is limited to event detection, which is a notable constraint. While event detection plays a crucial role in understanding factors influencing ticket demand and optimizing pricing strategies, it is only one component of the dynamic pricing process. Other factors, such as departure and arrival times, ticket price, and class, were not explored in this study. Although our findings provide valuable insights into leveraging event sentiment for improved pricing decisions, they represent only a partial understanding of the complexities involved in airfare dynamic pricing strategies.

This paper makes the following key contributions:

- Stacking Ensemble Method for Data Labeling: Developed a stacking ensemble method for data labeling using Random Forest Classifier (RFC), Support Vector Classifier (SVC) and Voting Classifier (VC) achieving high accuracy rates.
- Fine-Tuned BERT Model for Event Detection: Fine-tuned the BERT model for multi-class event detection, specifically tailored for Indonesian Twitter (X) data.
- Comprehensive Evaluations: Conducted comprehensive evaluations, including learning curves and cross-validation, to ensure model robustness and generalization.
- Comparative Analysis: Provided a detailed comparative analysis with existing methods, showcasing the superior performance and effectiveness of the proposed approach.

The scalability of the proposed method is a major advantage. By utilizing the ensemble stacking model and refining BERT, the approach can be adapted to different regions and event types. The flexibility of the model allows for retraining with new datasets, ensuring that it captures the unique linguistic and contextual nuances of different environments. This adaptability is crucial for applying the model across different scenarios with minimal modifications.

II. RELATED WORKS

We explored previous studies on event detection using Twitter data, such as the research conducted Alfalqi *et al.* [22], which investigates event detection using datasets composed of Twitter images. This study applied Active Learning (AL) techniques to reduce the amount of manual labeling required during disasters. Through Active Learning, the model achieved an accuracy and recall of 0.98, compared to 0.97 without Active Learning.

Ramachandran *et al.* [23], demonstrated that selecting the optimal combination of features can significantly improve event detection performance. Their study examined several combinations of features (words, POS, TFIDF) using Naive Bayes and Decision Tree algorithms. For instance, the combination of words, POS, and TFIDF achieved a precision of 0.97, recall of 0.95, and F1-Score of 0.96 with Naive Bayes. Bhardwaj *et al.* [24], employed a Human-in-the-Loop Rule Discovery method, combining human and AI contributions to rule discovery for event detection in microposts. The resulting accuracy ranged from 0.71 to 0.86, depending on the iteration and mode settings.

Other studies, such as the one conducted by Rezaei *et al.* [25], focused on event detection on Twitter using deep learning classification and multi-label clustering. Their approach used Hierarchical Attention Networks (HAN) for deep learning classification, with evaluations yielding an Area Under Curve (AUC) of 0.98. Dhiman *et al.* [16], developed an approximate graph-based global event detection model for Twitter data, addressing the inherent uncertainty in the platform. Their proposed method involved modeling Twitter data as a Sentence Graph using JoSE, a spherical vector representation that captures contextual information.

Another study on multi-event detection was conducted by Goyal *et al.* [26], This study aimed to detect events from a stream of tweets, generate storylines, and summarize relevant information to enhance understanding and extract useful insights from vast and diverse Twitter (X) data. The researchers employed a framework called Mythos for event detection, which combines methods such as keyword precision, recall, and F-measure. Their approach combines clustering and matching techniques with ground truth topics to extract pertinent information. Yavari *et al.* [27], proposed another event detection method that uses specialized techniques for tweet analysis, including stepwise clustering and comparison of system output keywords with event keywords in the dataset. Their results showed that the average accuracy of the proposed method for predicting events three weeks before the event was 0.71, increasing to 0.81 and 0.85 two weeks and one week before the event, respectively. On the day of the event, the prediction and detection accuracy reached 0.87.

Chen *et al.* [28], conducted a study aimed at enhancing the accuracy and efficiency of COVID-19 vaccine adverse event detection using multi-label classification and various label selection strategies. The OvsR topic-based method achieved an optimal accuracy of up to 0.98, while the accuracy of the AA method using topic-based labels increased to 0.87. However, deep learning methods like LSTM and BERT showed lower performance, with accuracies of 0.71 and 0.64, respectively. Alfalqi *et al.* [22], demonstrated the potential of Active Learning (AL) in reducing the manual labeling required during disaster events, thereby minimizing human intervention and training sample review. They combined Federated Learning with Active Learning to improve model efficiency and performance in detecting emergency events from social media data. This approach ensures data privacy by training models locally and then aggregating the results, significantly enhancing the scalability and efficiency of the detection process. Inspired by their findings, we adopted a similar Federated Learning framework to handle large-scale Twitter data. By implementing their Active Learning strategy, we selectively labeled only the most informative data points, thus reducing labeling effort while maintaining high model performance.

Ramachandran *et al.* [23], highlighted the importance of comprehensive feature analysis in improving event detection on Twitter. They showed that analyzing various tweet features, such as text content, user interactions, and temporal patterns, can significantly enhance detection accuracy. In our approach, we incorporate their feature analysis techniques by extracting and analyzing similar features from Twitter data. We consider user engagement metrics (likes, retweets, replies) and temporal patterns (tweet frequency, timing) to improve our model's ability to detect relevant events accurately.

The reviewed literature highlights significant advancements in event detection using various methodologies, including machine learning, deep learning, and hybrid approaches. For instance, studies utilizing deep learning methods such as CNN and BERT models have demonstrated the potential for high accuracy in specific contexts. However, challenges remain in handling diverse and unstructured data from social media platforms like Twitter (X). Our research builds on these foundations by integrating a stacking ensemble method for data labeling with fine-tuning of the BERT model for multi-class event detection. This approach addresses the limitations identified in previous studies, such as manual labeling errors and model limitations in capturing language nuances. By doing so, we aim to enhance the accuracy and robustness of event detection, ultimately contributing to more effective dynamic pricing strategies in the airline industry.

III. PROPOSED METHOD

In event detection research, we employ two primary methods for data labeling, namely Term Frequency-Inverse Document Frequency (TF-IDF) and Stacking Ensemble, to construct a labeling model on Twitter data (X). Inspired by Alfalqi *et al.* [22], we incorporated Active Learning (AL) techniques within our stacking ensemble method. By utilizing AL, we reduced the reliance on manual labeling, allowing the model to actively select the most informative samples for labeling. This approach not only enhances labeling efficiency but also reduces human error, leading to a more accurate and robust model for event detection. The integration of AL techniques enables the model to handle large volumes of data more efficiently, which is crucial for real-time event detection using social media data.

The TF-IDF method is a feature extraction technique that assigns a weight to each word based on its frequency in a document and the inverse of its frequency in the entire document collection [29]. This approach generates a vector representation that captures the unique characteristics of each tweet [30]. On the other hand, the Stacking Ensemble method combines multiple machine learning models [31], allowing us to leverage the strengths and unique capabilities of each model. This approach improves overall performance and mitigates the weaknesses of individual models [32]. Based on the findings of Ramachandran *et al.* [23], we employ the TF-IDF technique for feature extraction. TF-IDF assigns weights to words according to their frequency and uniqueness, yielding a more informative feature representation. This technique helps capture the unique attributes of each tweet, thereby enhancing the model's ability to distinguish between relevant and irrelevant information during event detection. Incorporating TF-IDF into our model results in improved performance metrics, including precision, recall, and F1-score, as demonstrated in the study by Ramachandran *et*

TABLE 1. Event Detection Related Work

Author	Objectives	Method	Result
Alfalqi. *et al.* [22]	Combining federative and active learning to detect emergency events	Federated Learning dan Active Learning	Accuracy and recall around 0.98 With Active Learning
Ramachandran. *et al.* [23]	Improving event detection on Twitter through feature analysis	Analisis fitur dengan algoritma Naive Bayes dan Decision Tree	Precision to 0.97, recall 0.95, F1-Score 0.96
Bhardwaj. *et al.* [24]	Combining human and AI contributions in rule discovery for event detection	Human-in-the-loop approach with machine learning	Accuracy ranges between 0.71 to 0.86
Rezaei. *et al.* [25]	Detecting events on Twitter using deep learning classification and clustering	Deep learning classification with HAN and other methods	The AUC for the Opt-HAN model reached 0.98
Goyal. *et al.* [26]	Understand and summarize key information from the tweet stream	Twitter-based event detection and storyline generation	Effective in event detection and storyline generation
Yavari. *et al.* [27]	Predicting events using Twitter data analysis	Special techniques for tweet analysis	Average precision 0.71 increased to 0.81 dan 0.85 close to the event
Chen. *et al.* [28]	Improving COVID-19 vaccine adverse event detection	Multi-label classification with diverse label selection strategies	Optimal accuracy up to 0.98

al. [23]. Additionally, we apply the Transformer architecture, particularly fine-tuning the BERT (Bidirectional Encoder Representations from Transformers) model, for event detection on Twitter data that has been labeled using the stacking ensemble models. Transformers, and specifically BERT, are capable of understanding the global context of text by simultaneously processing information from all words in a sentence [33]. Fine-tuning is performed to adapt the BERT model to the context and characteristics of Indonesian Twitter data, resulting in accurate event classification across diverse classes [34].

A. STACKING ENSEMBLE METHOD

In the stacking ensemble method, we began by creating a dataset. We crawled Twitter (X) data using the Tweet Harvest algorithm, which is available for research purposes. First, we created a Twitter (X) account, then used the Tweet Harvest algorithm by adding the token ID from our Twitter (X) account. To systematically collect and process the Twitter (X) data, we implemented the following pseudocode, which outlines the steps for setting up the environment, crawling the data, and processing the collected tweets.

We collected data between January 1, 2023, and November 30, 2023, amassing 3,451 tweets based on various search keywords such as riots, soccer, floods, volcanoes, earthquakes, motorcycle racing, concerts, and others. After collecting the data, we manually labeled the tweets with the categories: ’riots’, ’soccer’, ’floods’, ’volcanoes’, ’earthquakes’, ’motorcycle_race’, ’concerts’, and ’others’. The class labeled as ’others’ had the least number of tweets because it overlapped with other topics. Any Twitter (X) data that did not clearly belong to one of the specified categories was assigned to the ’others’ class.

In conducting this study, we adhered to ethical guidelines to ensure that the data collection process from Twitter (X) was carried out responsibly. Specifically, we complied with Twitter’s API usage policies, which govern the lawful and ethical use of data obtained from the platform. We also took measures to anonymize any personal information that could be linked to specific individuals to protect user privacy. This included removing or obfuscating any user identifiers (such as usernames or user IDs) from the dataset before analysis. Additionally, the data collected was solely used for academic research purposes and was not shared or used in a manner that could harm or exploit the individuals whose tweets were included in the dataset. These ethical considerations were crucial in ensuring that our research not only meets academic standards but also respects the rights and privacy of social media users.

```
Algorithm 1: CrawlTwitterData
Input:
    filename, search_keyword, limit, token
Output:
    Displayed DataFrame, Printed number of tweets

SetupEnvironment()
    Install pandas
    Install nodejs
    return

CrawlData(filename, search_keyword, limit, token)
    Execute Tweet Harvest algorithm
        parameters:
            -o filename
            -s search_keyword
            -l limit
            --token token
    Save crawled data to filename
    return

ReadAndDisplayData(filename)
    Read CSV file into DataFrame df
    Display DataFrame df
    num_tweets ← len(df)
    Print "Jumlah tweet dalam dataframe adalah" num_tweets
    return

end
```

FIGURE 1. Pseudocode For The Twitter (X) Data Crawling

After the data was collected, we performed several preprocessing steps to ensure the quality and usability of the data. Initially, the data was checked for missing values and duplicates. No missing or duplicate data were found during this step. The text data was then cleaned to remove special characters, such as @, #, emojis, and URLs, as well as converting all text to lowercase. This cleaning process was implemented using the Indonesian NLTK library [35] to ensure the data was standardized. Stopwords,

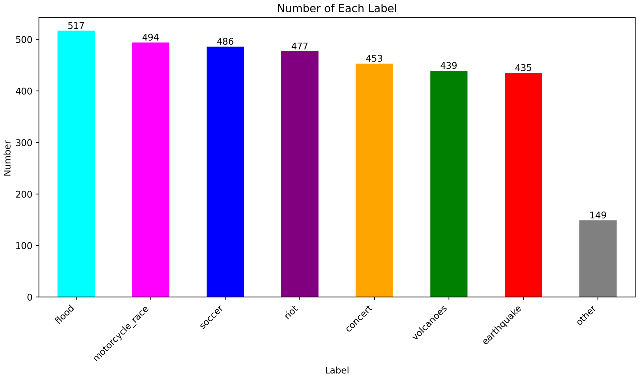

FIGURE 2. Number of Labels in the Dataset

TABLE 2. Description of twitter dataset (X)

Data Period	1 Januari 2023 - 30 November 2023
Amount Of Data	3.580
Label	motorcycle_racing, flood, football, concert, riot, volcano eruption, earthquake, other
Features	Created_at, id_str, full_text, quote_count, reply_count, retweet_count, favorite_count, lang, user_id_str, conversation_id_str, username, tweet_url, clean_text, processed_text, label
Language	Indonesian

which are common words that do not contribute meaningful information to the analysis, were also removed to refine the dataset further. The cleaned text data was then stored in a new column labeled 'processed_text' for subsequent analysis. Following the preprocessing steps, the dataset was split into training and testing sets. Specifically, 80% of the data was allocated to training, and 20% was reserved for testing. The train_test_split function from the scikit-learn library in Python was employed to perform this split. This function was chosen due to its capability to randomly shuffle the data before splitting, ensuring that the resulting subsets are representative of the overall dataset. Randomization is a fundamental technique in machine learning as it helps to minimize selection bias and avoids scenarios where the model might learn artifacts from the data ordering rather than the underlying patterns.

To further enhance the robustness of model evaluation, we employed stratified sampling during the train-test split. Stratified sampling ensures that each subset of data maintains the same class distribution as the original dataset, which is especially important in cases of imbalanced datasets where certain classes might be underrepresented. Without stratification, there is a risk that the training or testing set may disproportionately represent certain classes, leading to skewed model performance. In such cases, the model could perform well on overrepresented classes while underperforming on others. By implementing stratified sampling, we preserved the proportional representation of each class across both training and testing sets. This approach ensures that the model is trained on a dataset that mirrors the diversity and balance of the full dataset, allowing it to learn patterns that generalize well across all classes. Moreover, during testing, the model's performance can be more accurately assessed across different classes, providing a clearer understanding of its strengths and limitations.

In this study, we recognized the potential biases that could arise due to imbalances in the dataset, particularly with certain event classes being more prevalent than others. For example, classes such as 'concert' and 'football' were more frequent, which could cause the model to overfit on these classes while underperforming on less frequent ones. By applying stratified sampling during the train-test split, we ensured that the class distribution remained consistent across both sets, minimizing the risk of the model becoming biased toward more common classes and improving its ability to generalize to less frequent events.

From a technical perspective, stratified sampling works by dividing the dataset into strata, or layers, based on the unique classes present in the data. Within each stratum, the train_test_split function then randomly selects samples according to the specified split ratio (80/20 in this case), ensuring that each subset reflects the overall class distribution. This technique is particularly effective in avoiding overfitting, as it prevents the model from being trained or tested on a non-representative sample, thereby improving the reliability and validity of the model's performance metrics.

After preprocessing the data, we converted the text data from the cleaned data columns in the training DataFrame into a numeric feature matrix using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. TF-IDF is a statistical measure that evaluates the importance of a word within a document relative to a collection of documents. This technique is particularly useful for distinguishing words that are significant in specific contexts, as it assigns higher weights to terms that are frequent in a particular document but rare across the document collection. By applying TF-IDF, we generate a vector representation of the tweets that effectively captures their unique characteristics, which is crucial for accurate event detection. This approach is supported by studies like Bok *et al.* [36], which demonstrated the effectiveness of TF-IDF in an efficient graph-based event detection scheme on social media. TF-IDF assigns a numerical weight to each word in a document based on how often the word appears in that document and how unique it is across the entire document collection. The goal of using TF-IDF is to provide a more informative feature representation. Words that appear frequently but are common across many documents are assigned lower weights, while words that are less frequent but specific to a particular document receive higher weights. The resulting TF-IDF feature matrix is then used as input to train classification models, such as Random Forest and Support Vector Classifier, to predict the category label corresponding to each text in the dataset. We present the formula for the TF-IDF process as follows:

$$TF-IDF(t,d,D) = TF(t,d) * IDF(t,D) \tag{1}$$

Where $TF(t,d)$ is the frequency of the word (term frequency) from the word t in the document d. $IDF(t,D)$ is the inverse of the document frequency $(\text{inverse document frequency})$ from the word t in the document D. $TF(t,d)$ can be calculated by :

$$\text{TF}(t,d) = \frac{\text{number of occurrences of the word } t \text{ in the document } d}{\text{total number of words in the document } d} \tag{2}$$

The $IDF(t,D)$ formula can be calculated by:

$$\text{IDF}(t,D) = \log\left(\frac{\text{Total number of documents in collection } D}{\text{Number of documents containing term } t+1}\right) + 1 \tag{3}$$

Next is the data labeling process, the results of which will be used for the labeling training process according to the type of event. At this stage, we use the stacking ensemble technique. Stacking ensemble is a machine learning approach that combines multiple base models to enhance predictive performance. This technique leverages the strengths of each individual model while mitigating their weaknesses, leading to improved overall accuracy [37]. The method is referred to as "stacking" because it involves two layers of models: the base classification model layer and the main classification model layer, also known as the meta-classifier. The architecture of the stacking ensemble method we use is illustrated in Figure 2, which shows the stacking ensemble architecture

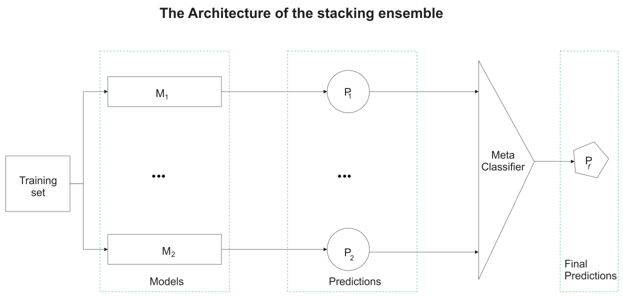

FIGURE 3. Stacking ensemble architecture

Here is the revised version of your paragraph, edited for proper grammar and IEEE-style writing:

In the base model, we use two classification models: the Random Forest Classifier (RFC) model as M_1 and the Support Vector Classifier (SVC) model as M_2. The RFC was configured with 100 estimators (n_estimators=100) due to its robustness in handling large feature sets and complex data, while the SVC was employed with a linear kernel and a regularization parameter C set to 1 (C=1). These hyperparameters were selected based on prior experimentation and are considered standard for these models. The RFC model is advantageous for its stability and resistance to overfitting, making it effective for our diverse dataset. However, it may struggle with linear patterns in the data a gap effectively filled by the SVC model, which excels in both linear and non-linear classifications. For the main model, we use the Voting Classifier (VC) as a meta-classifier. To enhance the predictive performance of our model, we combined the outputs of the RFC and SVC models using the VC, which operates on a soft voting mechanism. This means the final prediction is based on the averaged predicted probabilities from both base models.

This approach leverages the individual strengths of RFC and SVC, improving the overall classification accuracy. By integrating the complementary capabilities of these models, the VC mitigates the weaknesses inherent in each, resulting in a more robust and adaptive predictive model. Additionally, this ensemble method helps reduce the risk of overfitting by balancing the contributions of both base models. The RFC is robust in handling many features and complex data, with high prediction stability due to the aggregation of many randomized decision trees, and it is resistant to overfitting by combining predictions from numerous trees [38]. However, RFC has its limitations, particularly in handling linear patterns and data with dominant linear relationships [39]. The SVC model, on the other hand, excels at handling both linear and non-linear relationships between features and targets, is effective in high-dimensional feature spaces, and can handle outliers well [40].

Despite these strengths, the SVC model is prone to overfitting, especially when using complex kernels, and its performance can degrade with a large number of features or data samples [41]. By understanding the weaknesses and strengths of each base model, we can effectively combine them to complement and cover each other's shortcomings. While the RFC may be less effective in handling linear patterns, the SVC addresses this weakness with its ability to manage both linear and non-linear patterns. Similarly, while the SVC is prone to overfitting, this issue is mitigated by the RFC's predictive stability and robustness when handling complex data.

Combining the predictions from the base models—RFC and SVC—using voting rules allows the model to leverage the unique strengths of each base model, ultimately improving overall accuracy [42]. By combining multiple models, the Voting Classifier (VC) helps reduce the risk of overfitting if one model tends to overfit on certain training data [43]. The VC works by using the predicted probabilities of the base models, which enables it to make more informed decisions by considering the contributions of both RFC and SVC in making the final prediction [44]. As a meta-classifier, the VC can analyze the predictions from the RFC and SVC models collectively, assigning weights to each prediction based on the majority vote or probability. Meta-classifier models like the VC provide the advantage of combining the expertise of two different underlying models [45]. In this case, the VC makes decisions based on the majority vote or the prediction probabilities of the RFC and SVC models. This enables the VC model to mitigate the weaknesses of each base model while capitalizing on their strengths, resulting in a more robust and adaptive predictive model.

From the stacking ensemble model we developed, we derived a mathematical formula for the final prediction of the

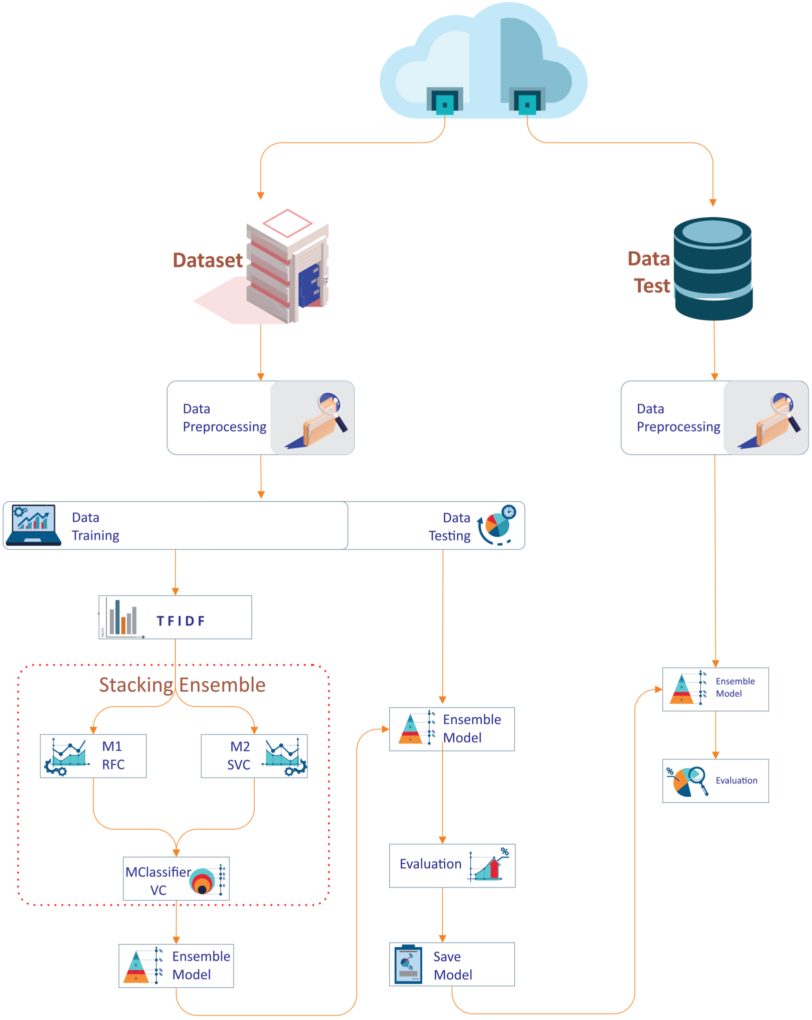

FIGURE 4. Stacking ensemble labeling proposed method

ensemble stacking model. Beginning with the base models, RFC and SVC, the basic RFC model is built on the random forest algorithm to predict the class of the data. The mathematical formula for the basic RFC model is as follows:

$$P_{\text{RFC}}(y|x) = 1/N * P_j(y|x) \tag{4}$$

Where N is the number of trees in the forest, $P_j(y|x)$ is the probability of class y for data x in the j tree. Then the mathematical formula for calculating the class prediction of the SVC model is as follows:

$$P_{\text{SVC}}(y|x) = sign(f(x)) \tag{5}$$

Where y is the class to be predicted, x the data to be classified and $f(x)$ is the hyperplane that separates the positive and negative classes. Meanwhile, the mathematical formula for the VC model is as follows:

$$P_{\text{VC}}(y|x) = w_i * P_i(y|x) \tag{6}$$

Where w_i is the weight of the i model, and $P_i(y|x)$ is the probability of class y for data x in the i model. In the case of stacking ensemble created using two basic models, namely the RFC basic model as M1 and SVC as M2 and VC as the main model (meta classifier). Therefore, we will get the final prediction math formula (p_f) as follows:

$$pf = \arg\max(w_{VC} * P_{VC}(y|x) + w_{M1} * P_{M1}(y|x) \\ + w_{M2} * P_{M2}(y|x)) \tag{7}$$

Where p_f is the final prediction, argmax the function that finds the value of the argument (x) that maximizes the function $f(x)$, w_{VC} denotes the weights for the Voting Classifier (VC) meta classifier, $P_{VC}(y|x)$ denotes the probability of class y for data x according to the VC model, w_{M1} is the weight for the Random Forest Classifier (RFC) base model as M1, $P_{M1}(y|x)$ denotes the probability of class y for data x according to M1 (RFC) model, w_{M2} denotes the weights for Support Vector Classifier (SVC) base model as M2 and $P_{M2}(y|x))$ is the probability of class y for data x according to M2 (SVC) model. So overall, this formula shows that the final prediction (p_f) is determined by selecting the class that has the highest combined value. The combined value is calculated by multiplying the probability of each model

(P_{VC}, P_{M1}, P_{M2}) with its weight (w_{VC}, w_{M1}, w_{M2}), then summing the multiplication results of all models and finding the class that maximizes the combined value using the argmax function. So it can be concluded that the main model (meta classifier) in this case is VC trains its model using class predictions from the base model (M1 and M2) as input, then VC predicts classes for new data by calculating the most votes from the base model predictions.

We also tested the model using a learning curve, which is a graph that shows how the model's overall performance changes as the amount of training data increases. The learning curve is useful for evaluating overfitting or underfitting and determining the optimal size of the training data. It provides insights into how the model's performance evolves with additional data. Testing with a learning curve is crucial because it gives a quick overview of the model's performance as the dataset size grows. It helps identify overfitting or underfitting, determines the optimal dataset size, and offers an evaluation of model performance at different data sizes. This information is valuable for model parameter optimization and validating cross-validation results. By analyzing the learning curve, developers can make informed decisions about whether model adjustments or changes in the training strategy are needed, ensuring that the model adapts well to larger datasets.

The learning curve indicates that the ensemble model continues to improve its performance as the number of training data examples increases. We evaluated accuracy metrics using a robust 10-fold cross-validation approach, resulting in an average cross-validation score of 0.92 on independent data. This method ensures that our model's performance is not only consistent but also generalizable across different subsets of the data. The stacking ensemble model demonstrated superior performance compared to individual models such as Random Forest (RF) and Support Vector Machine (SVM), achieving an accuracy rate of 0.99 on the test data. These results highlight the effectiveness of the stacking ensemble approach in handling complex classification tasks.

The stacking ensemble method was chosen due to its ability to leverage the strengths of multiple models, thereby improving overall prediction accuracy and robustness. By combining models such as Random Forest and Support Vector Machine, the stacking ensemble mitigates the weaknesses of individual models and enhances performance through collective learning. Additionally, fine-tuning the BERT model allows for precise event detection by utilizing its powerful language understanding capabilities, specifically tailored to the context of Indonesian Twitter data. This combination of methods ensures that our dynamic pricing model is both accurate and adaptive to real-time social media insights.

B. BERT MODEL FOR EVENT DETECTION

The Transformer model consists of two main parts: encoder and decoder [46]. The encoder analyzes the input text and generates an internal representation, while the decoder uses this representation to generate outputs [47]. The BERT (Bidirectional Encoder Representations from Transformers) model is a Transformer model that has been trained for text classification tasks [48]. BERT has emerged as a powerful tool for natural language processing (NLP) tasks due to its ability to generate contextually rich embeddings. As highlighted by Bano *et al.* [49], integrating BERT with models like BiGRU enables the capturing of both local and global contexts, significantly enhancing model performance in extractive summarization tasks. Moreover, the challenge of using BERT for long document summarization, due to its input length restrictions, has been addressed by Bano *et al.* [50], who proposed a novel architecture that divides long documents into smaller chunks. Each chunk is processed through BERT to generate sentence embeddings, which are then fed into an attention-based encoder-decoder framework, enabling the model to capture global context and generate highly accurate summaries. This approach was rigorously tested on scholarly datasets like arXiv and PubMed, where it consistently outperformed state-of-the-art models, demonstrating its effectiveness in handling complex and lengthy texts.

The success of this method underscores BERT's adaptability, making it particularly suitable for tasks like detecting traffic events in social media data. BERT is a deep learning model designed for NLP tasks, and it is particularly effective in understanding the context of words within a sentence by processing information bidirectionally [51]. The model is trained on large text and code datasets [52], and consists of two main components: the encoder and decoder [53]. In our experiments, we used the Transformer technique with the BERT model for multi-class classification tasks. Specifically, we used the IndoBERT-Base-p1 model from Hugging Face, a version of the BERT model pre-trained on an Indonesian corpus. IndoBERT-Base-p1 is specifically designed for NLP classification tasks in Indonesian. For our multi-class task, we fine-tuned the classification layer of the model using the event classes prepared in the Indonesian language. Hugging Face, a popular platform for machine learning models, provides various pre-trained models such as IndoBERT, which can be customized for specific applications like text classification. The data used for our event detection experiments was labeled in the earlier preprocessing steps. Our proposed method for detecting events is illustrated in Figure 4.

First, we initialize the tokenizer and model using *indobert-base-p1*. We use the BERT tokenizer for the Indonesian language from the Hugging Face Transformers library. The text data was tokenized to a maximum sequence length of 128 tokens, ensuring uniform input dimensions. This tokenized text was then converted into numerical representations, which were fed into the BERT model. The tokenizer is responsible for converting the text into a numerical representation that the BERT model can process. We chose the *indobert-base-p1* model because it is a state-of-the-art language model for Indonesian, based on the BERT architecture. The next step involves tokenizing and converting the text into a numerical format that the BERT model can interpret. First, we set

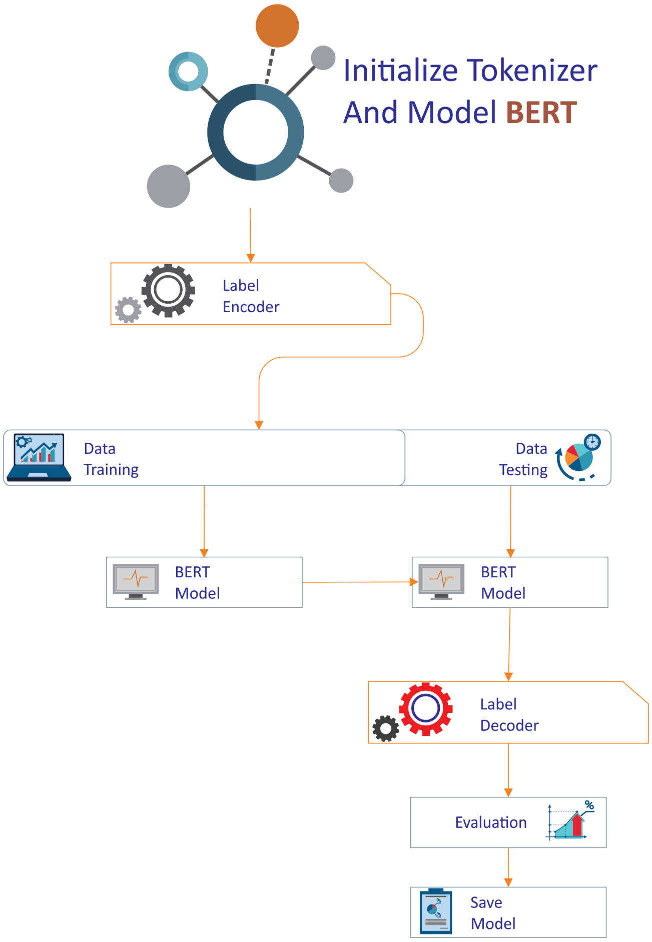

FIGURE 5. Event Detection Proposed Method

max_length as the maximum length of tokens allowable length of tokens in a sample. Then, we initialize two empty lists, input_ids and attention_masks, which will be populated with the tokenization results for each text. Using a loop, we iterate through each text in the 'full_text' column of the dataset. Within the loop, we utilize the encode_plus method from the pre-initialized tokenizer to perform tokenization and convert the text into a numerical format. All tokenization results are then merged into a tensor. The results of the tokenization input_ids and attention_masks which are numerical representations of the text—are subsequently used as input for the BERT model.

The BERT model architecture was enhanced by adding a classification layer on top. This included a dropout layer with a dropout rate of 0.1, followed by a fully connected linear layer for producing the classification results. The model was fine-tuned using the AdamW optimizer with a learning rate of 1e-5, over five epochs, to adapt the pre-trained BERT model to our multi-class event classification task. Next, we initialized a LabelEncoder object. The LabelEncoder is used to convert categorical labels (in this case, the 'prediction_label' column in the dataset) into integer values for model training purposes. The LabelEncoder was then applied to convert the 'prediction_label' column into integers. The dataset was split into two parts: 80% for training and 20% for testing. After preparing the data, we built the classification model. The model architecture consists of multiple layers, including the BERT model, a dropout layer, and a fully connected linear layer for classification output. The BERT model leverages the pre-trained architecture, and the dropout layer, with a dropout rate of 0.1, helps prevent overfitting by randomly disabling units during training. To ensure the robustness and reliability of our model, we evaluated its performance using metrics such as precision and recall, particularly given the imbalanced nature of certain event classes. These evaluation metrics are critical in contexts like ours, where accuracy alone may not fully capture the model's performance. This approach aligns with the findings of Khalid *et al.* [54], which emphasize the importance of multi-objective evaluation for determining model usefulness, particularly in academic search applications.

The next step is to train the classification model using the pre-processed data. First, we create a TensorDataset object using 'train_inputs', 'train_masks', and 'train_labels'. The TensorDataset is used to combine input data and labels into a single dataset. Next, we utilize 'DataLoader' to create a dataloader from 'train_data'.The dataloader manages the dataset during training, including dividing it into batches of size 'batch_size_train' and randomizing the data ('shuffle=True'). After setting up the dataloader, we begin the model training process. We specify the number of epochs (full iterations through the dataset) with epochs = 5. Next, we iterate through each epoch using a loop. At each iteration within an epoch, we set the model to training mode with 'classifier.train()'. We then fetch one batch from the dataloader ('train_dataloader') ccontaining the input IDs, attention masks, and labels.

Subsequently, a forward pass through the BERT model is performed. The output from the BERT model is taken from the last layer and further processed through the dropout and linear layers to obtain logits, which are the model's output values. We then calculate the loss using the logits and the actual labels with the loss function ('nn.CrossEntropyLoss()'). The calculated loss is used to compute the gradients via a backward pass. Finally, we optimize the model by updating its parameters based on the previously computed gradients using the AdamW optimizer, a variant of the Adam algorithm that incorporates L2 regularization (weight decay). AdamW helps maintain stability and reduces overfitting. Once the BERT model is trained, we evaluate its performance using the Classification Report. This report provides key evaluation metrics, such as precision, recall, F1-score, and accuracy. These metrics are crucial for assessing how well the model classifies each class, as well as its overall performance.

We chose the stacking ensemble method due to its ability to leverage the strengths of multiple models, thereby improving overall accuracy and robustness in predictions. By combining machine learning models such as Random Forest and Support Vector Machine, the stacking ensemble reduces the weaknesses of individual models and enhances performance through collective learning. On the other hand, fine-tuning the BERT model enables precise event detection by utilizing its strong language understanding capabilities, specifically adapted to the context of Indonesian Twitter data. This combination ensures that our dynamic pricing model is both accurate and adaptive to real-time social media insights. The stacking ensemble method combines several base classifiers, such as Random Forest and Support Vector Classifier, to improve performance. This method allows for the easy integration of additional classifiers or retraining with new datasets from different regions or event types. Fine-tuning the BERT model involves adapting the pre-trained model to specific datasets, allowing it to learn contextually relevant information from various applications. This process requires minimal customization, primarily retraining the final layer with new datasets. By combining the stacking ensemble method with a fine-tuned BERT model, our approach effectively integrates the strengths of machine learning and deep learning techniques. The stacking ensemble method improves predictive accuracy through model diversity, while the BERT model provides a deep contextual understanding of language, particularly in the context of Indonesian Twitter data. This dual approach ensures robust event detection and dynamic pricing strategies based on real-time social media insights.

Given the complexity of language and context in tweets, our model employs several strategies to handle linguistic nuances and variations across different event types. We fine-tuned the BERT model specifically for Indonesian Twitter data, which involved training it on a diverse dataset of tweets covering various events, such as sports, concerts, and natural disasters. The BERT model's architecture allows it to understand the global context by processing all words in a sentence simultaneously, making it well-suited for capturing subtle linguistic nuances and contextual variations. We also incorporated the Term Frequency-Inverse Document Frequency (TF-IDF) technique in our feature extraction process to emphasize words that are unique and significant to specific events. This method enhances the model's ability to differentiate between contexts by assigning higher importance to words that are particularly relevant to a certain type of event.

IV. RESULTS AND DISCUSSION

A. RESULT

The results of the experiments to create a Twitter (X) data labeling model using the stacking ensemble method, as well as the fine-tuning of the BERT model for event detection, are presented in this section.

1) Results of stacking ensemble model

First, we evaluate the performance comparison between individual models and the stacking ensemble model. In the comparison of Random Forest (RF), Support Vector Machine (SVM), and the ensemble models on the test data (as shown in Table 3), it is evident that the ensemble model outperforms both RF and SVM, achieving an accuracy rate of 0.99. The Random Forest model performed very well with an accuracy of 0.98, while the Support Vector Machine model achieved an accuracy of 0.97. These results indicate that the use of ensemble models, particularly with a voting classifier approach, enhances classification capabilities by integrating the strengths of each base model (RF and SVM), leading to more optimal performance on the test data. The ensemble model delivers consistent and superior results, highlighting its potential for implementation in classifying event-related Twitter (X) data across specific categories, including timing and other event-related factors.

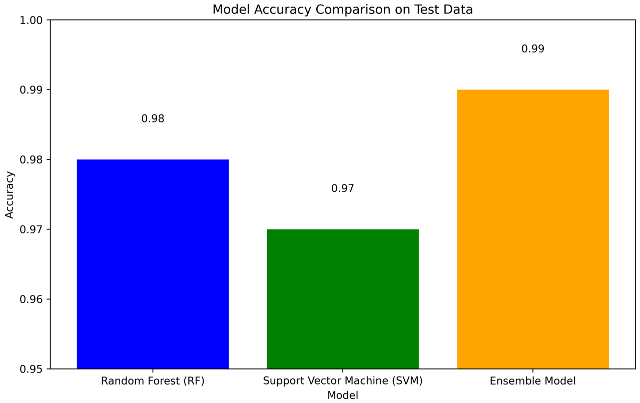

FIGURE 6. Classification Model Accuracy Comparison on Test Data

Each class demonstrates high precision, recall, and F1-score, reflecting the model's ability to classify different types of events in the Twitter (X) data accurately. For instance, the "motorcycle racing" class achieved a precision and recall rate of approximately 0.99, with F1-score and support values reaching 1.00, indicating the model's consistent ability to recognize and classify these events. The overall evaluation results, both at the macro and weighted levels, confirm the ensemble model's capability to accurately label the test data, which consisted of 690 samples—a portion of the total 3,461 data points used during the training and testing process. The 80/20 split between training and testing data ensures a balanced representation, enabling the model to generalize well to previously unseen data. More detailed evaluation results are presented in Table 3.

TABLE 3. Ensemble Stacking Model Evaluation

Label	Precision	Recall	F1-Score	Support
motorcycle_race	1.00	0.99	1.00	118
flood	1.00	0.96	0.98	106
earthquake	0.99	1.00	0.99	91
volcanoes	0.99	1.00	0.99	74
riot's	0.98	1.00	0.99	93
concert	0.99	0.99	0.99	81
other	0.97	0.97	0.97	32
football	0.98	0.99	0.98	95
Accuracy			0.99	690
Macro Avg	0.99	0.99	0.99	690
Weighted Avg	0.99	0.99	0.99	690

We also evaluated the stacking ensemble model using a learning curve. The results show that the model performs exceptionally well on the training data, with the training score (represented by the red line) remaining stable at around 0.99. However, on the cross-validation data, the model's performance steadily improves as more data is added. Initially, the cross-validation score (CV score) was approximately 0.44, then increased to 0.58, 0.85, and eventually peaked at 0.96. These results indicate that the model has a strong ability to generalize to unseen data, demonstrating its robustness and effectiveness in handling new inputs.

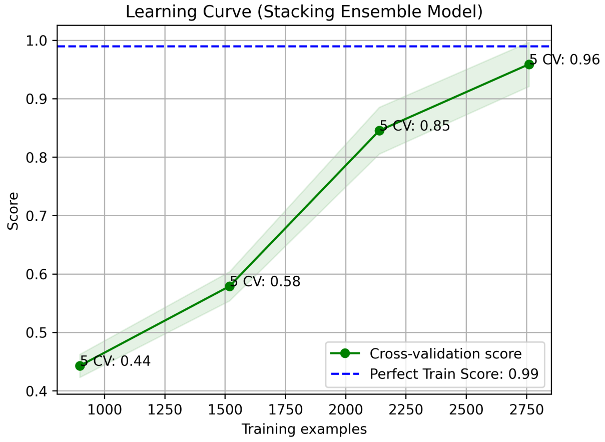

FIGURE 7. Learning Curve (Stacking Ensemble Model)

We then tested the ensemble stacking model with independent data. A new dataset was retrieved, consisting of 3,689 data points using the same keywords as the previous dataset, including music concert, soccer, riot, flood, volcano, motorcycle racing, earthquake, and others. The model was evaluated on this independent data using cross-validation to assess its performance.

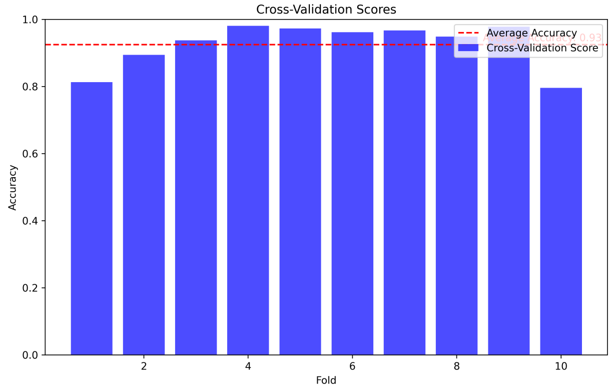

FIGURE 8. Cross Validation Evaluation Results with Independent Data

The results of the cross-validation show a variation in accuracy values across each fold: [0.813, 0.894, 0.938, 0.981, 0.973, 0.962, 0.967, 0.949, 0.978, 0.796]. The average accuracy across all folds is approximately 0.92. These values provide insight into how well the ensemble model generalizes to independent data that was not seen during training. Despite some variations, the high average accuracy suggests strong performance in classifying independent data.

Additionally, we validated the model using an independent dataset that had not been included in the training process. This independent data was preprocessed using the same TF-IDF vectorizer and label encoder as the training data. The ensemble model, consisting of a RandomForestClassifier and an SVM combined with a VotingClassifier using soft voting, was then used to predict the labels of the independent data. The predicted labels were compared to the true labels, and the model's performance was evaluated using metrics such as accuracy, precision, recall, and F1-score. The cross-validation scores and the independent data validation demonstrate the model's robustness and generalizability, highlighting its effectiveness in real-world scenarios.

2) BERT Model Fine Tuning Evaluation Results for Event Detection

The evaluation of the BERT model for event detection yielded satisfactory results, achieving an accuracy rate of 0.94. These results were derived from measuring various performance metrics such as precision, recall, and F1-score for each class. In particular, the model demonstrated high accuracy in recognizing specific events such as 'football' and 'flood', with precision and recall scores nearing 1.00. However, certain classes, such as 'other', exhibited lower precision and recall values, indicating challenges in accurately recognizing these categories. The primary reason for this is the smaller amount of data available for the 'other' class, limiting the model's ability to learn from enough examples and make accurate predictions. This lack of data representation makes it difficult for the model to distinguish the 'other' class from other event types.

To further validate the BERT model's performance, we utilized an independent dataset that had been labeled using the previously trained stacking ensemble model. The independent data was preprocessed using the same tokenizer and label encoder as the training data. The fine-tuned IndoBERT model was then applied to predict the labels of the independent data. Cross-validation was employed on the independent dataset to assess the model's robustness. The cross-validation scores demonstrated high accuracy, further confirming the model's effectiveness in real-world scenarios.

TABLE 4. Evaluation Metrics BERT Model

Class	Precision	Recall	F1-Score	Support
football	1.00	0.99	0.99	83
concert	0.95	0.98	0.97	107
volcanoes	0.96	0.98	0.97	102
earthquake	1.00	0.93	0.96	56
riots	0.89	0.96	0.92	120
flood	1.00	1.00	1.00	102
motorcycle_race	0.90	0.87	0.88	141
other	0.77	0.63	0.69	27
Accuracy			0.94	738
Macro Avg	0.93	0.92	0.92	738
Weighted Avg	0.94	0.94	0.94	738

To achieve these results, the independent dataset was tokenized using the indobert-base-p1 tokenizer and transformed into numerical representations. The dataset was split into training and testing sets in an 80:20 ratio. The BERT model was fine-tuned with a learning rate of 0.00001 and a batch size of 32. A dropout layer was utilized to prevent overfitting, and the AdamW optimizer with a cross-entropy loss function was employed during training. The evaluation was conducted using the hold-out test set, with metrics such as precision, recall, and F1-score computed to ensure robustness. The overall accuracy achieved on the hold-out test set was 94.31%, demonstrating the model's strong performance in event detection tasks. Table 4 shows that the BERT model performs well across most classes, with balanced F1-score values. Despite some variations in performance between classes, the weighted average remains high at 0.94. This indicates that the model is effective for event detection on Indonesian Twitter (X) data, though improvements in certain classes are needed to further optimize overall detection.

B. DISCUSSION

1) Introduction to the Discussion

The experimental results obtained in this study show better results compared to some other similar studies. For example, in a study conducted by Bhardwaj. *et al.* [24], which combined human and artificial intelligence contributions in event detection with accuracies between 0.71 and 0.86, the approach in this study, which involves ensemble stacking models in the labeling process and fine-tuning of the BERT model, managed to surpass that accuracy by reaching 0.94. This confirms that the combination of labeling techniques using ensemble stacking models and fine-tuning the BERT model can provide more reliable event detection results.

2) Comparison with Related Studies

Another study by Yavari. *et al.* [27], with a research focus on event prediction using Twitter (X) data analysis, the result of this study is that the precision of event prediction will increase as the time of an event approaches. Average precision results of around 0.71, which can be increased to 0.81 and 0.85 closer to the event. However, the results of our study managed to overcome these challenges by achieving a higher accuracy of 0.94, using the ensemble stacking model in the labeling process and fine-tuning the BERT model. This success indicates that the combination of techniques can significantly improve the accuracy and generalization of the model.

An additional challenge comes from other research such as that conducted by Goyal. *et al.* [26], which does not specifically present its results but claims effectiveness in event detection and storyline generation through social media. Our research has a significant impact on our efforts to continuously refine and improve our methodology, creating a superior contribution in the domain of event detection on social media platforms.

Furthermore, there is research conducted by Ramachandran *et al.* [23], which aims to improve event detection on Twitter (X) through feature analysis using Naive Bayes and Decision Tree algorithms. The results of our study show a precision level of up to 0.97, recall of 0.95, and F1-Score of 0.96. When looking at the results of the classification report, such as precision, recall, and F1-Score on average, this study managed to achieve a higher level of performance compared to the average results of our research. However, when looking at the results in detail for each class or event, our method shows superiority in certain classes. For example, in class '0' (soccer), our detection results show a precision of 1.00, recall of 0.99, and F1-Score of 0.99. This indicates that, although the average of our results may be slightly below, the method we applied is able to provide excellent performance, even exceeding the results of other studies on certain classes. Thus, this research not only offers a more effective solution in the context of Indonesian Twitter (X), but also provides a foundation for the development of more advanced and adaptive event detection methods on social media platforms.

In this study, we utilized evaluation metrics including precision, recall, F1-score, and accuracy to assess the performance of our models. Precision measures the accuracy of positive predictions, indicating the proportion of true positives among all positive predictions. Recall assesses the model's ability to identify all actual positive instances, representing the proportion of true positives detected by the model. The F1-score, which is the harmonic mean of precision and recall, provides a balanced evaluation, particularly useful when there is a trade-off between precision and recall. Accuracy offers an overall measure of how often the model's predictions—both positive and negative—are correct.

We also conducted a comparison between our stacking ensemble model and several baseline models, including Random Forest (RF), Support Vector Machine (SVM), and other commonly used deep learning models. The results showed that our stacking ensemble model, which combines RFC and SVC with a Voting Classifier as the meta-classifier, significantly outperformed these baseline models. For instance, the RF and SVM models achieved accuracies of 0.98 and 0.97, respectively, while our stacking ensemble model reached an accuracy of 0.99. This indicates that the stacking ensemble approach effectively mitigates the weaknesses of the individual base models, resulting in more accurate and consistent predictions. Therefore, we assert that our stacking ensemble model can be considered state-of-the-art in the context of event detection from social media data, particularly for applications in dynamic pricing of airline tickets.

3) Implications for Airline Revenue and Dynamic Pricing

The relation between event detection and dynamic pricing in our study is explained by examining the influence of various events such as music concerts, sporting events, riots, and natural disasters on airfare pricing dynamics. When important events attract large numbers of people to a certain area, the demand for airfare to that location increases, which often results in higher prices due to increased occupancy rates. Conversely, negative events such as riots or natural disasters can lead to a decrease in demand, as travelers avoid the affected areas, potentially resulting in lower ticket prices to stimulate demand. Accurate event detection is achieved through real-time analysis of social media platforms such as Twitter (X), where the volume and sentiment of tweets related to specific events are monitored. This real-time data allows airlines to predict changes in travel demand and adjust pricing strategies dynamically. For example, in the case of positive events, airlines can raise ticket prices to capitalize on the surge in demand, while for negative events, they can lower prices to maintain occupancy levels by encouraging travel despite adverse conditions.

The integration of event detection into dynamic pricing models involves sophisticated machine learning algorithms to process and interpret large amounts of social media data. This process creates a predictive model that informs real-time price adjustments, optimizing revenue and occupancy. By incorporating real-time event detection into dynamic pricing models, airlines can make data-driven decisions, improving their pricing strategies to better respond to market changes. This detailed explanation aims to make the relationship between event detection and dynamic pricing more explicit and easy to understand. Moreover, the events detected through our method are not just signals for general demand shifts, but are integrated directly into the dynamic pricing model as new features, along with the sentiment associated with each event. This integration allows the pricing model to react not only to the occurrence of events but also to the public's sentiment towards these events, providing a more nuanced and accurate adjustment to pricing strategies. For example, a positive sentiment surrounding a major concert in a destination city could lead to an anticipatory increase in ticket prices, whereas negative sentiment due to an event like a natural disaster could trigger price reductions to encourage travel or manage occupancy.

This approach offers a significant improvement over traditional dynamic pricing mechanisms, which typically rely on historical data and generalized demand trends. By incorporating real-time event detection and sentiment analysis, our model allows airlines to adapt more rapidly to market changes, making pricing decisions that are more aligned with current conditions. This not only enhances revenue optimization but also improves customer satisfaction by aligning prices more closely with market realities.

4) Further Discussion and Impact

Similarly, other related studies have contributed to the development of event detection on social media. For example, the research conducted by Chen *et al.* [28], aims to improve the detection of adverse events related to the COVID-19 vaccine. They used a multi-label classification method with diverse label selection strategies, achieving optimal accuracy results of up to 0.98. Based on the comparison with these studies, we can see that our study is able to compete and even surpass these results in some aspects. For example, in one of the event classes in our study, the optimal accuracy rate reached 0.99. This shows that our method has the potential to provide more precise detection results, especially in the context of specific events on Indonesian-language Twitter (X).

This study reveals a fundamental problem in the existing airfare pricing practice, where conventional approaches have not fully utilized the potential of event detection to achieve the desired level of responsiveness and accuracy. Through the application of more sophisticated event detection methods, this research is able to overcome these obstacles and strengthen our knowledge base regarding event detection for dynamic pricing of airfare. By identifying existing weaknesses and presenting innovative solutions, this research makes a positive contribution in shaping a new framework that is more agile and adaptive, which can ultimately improve the effectiveness and competitiveness of the airline industry in the face of changing market dynamics.

The scalability of the proposed approach is further shown by its potential applicability in different regions and for different types of events. By retraining the model with relevant data sets, the method can be adapted to specific needs without significant modifications. For example, the approach can be adapted to monitor events in different languages or cultural contexts by fine-tuning the BERT model with the respective regional data sets. The flexibility of the stacking ensemble method in integrating diverse classifiers also supports its applicability in various domains.

In our evaluation, the fine-tuned BERT model demonstrated strong performance across various event types, accurately capturing the nuances of language used in different contexts. For example, the model showed high precision and recall for events like 'football' and 'flood,' where linguistic patterns and contextual cues are distinct. However, it faced challenges in the 'other' category, which consists of more ambiguous or less frequent event types. This indicates the model's reliance on the richness of the training data and its ability to adapt to specific linguistic contexts when such data is sufficiently represented.

V. CONCLUSION

The results of this event detection research have made a significant contribution in two main aspects. Firstly, we successfully developed a stacking ensemble model for data labeling. By employing a two-layer approach, where the base layer consists of Random Forest Classifier (RFC) and Support Vector Classifier (SVC), and the main layer utilizes Voting Classifier, our model achieves an impressive maximum accuracy rate of 0.99. Additionally, evaluation of the ensemble stacking model was conducted on independently unlabeled data, with the evaluation results using cross-validation method reaching a commendable score of 0.92, indicating excellent evaluation performance. Secondly, our research introduces the use of the fine-tuning method for the BERT model in event detection. With an overall accuracy rate of 0.94, the model proficiently identifies eight distinct event types: 'football', 'concert', 'volcanoes', 'earthquake', 'riot', 'flood', 'motorcycle_racing', and 'other'. The combination of these two contributions provides a robust framework for event detection in Indonesian Twitter (X) data.

Moving forward, addressing imbalanced data remains a primary focus for improving model performance. Strategies such as rearranging class distribution or applying class-specific sampling techniques may enhance the model's generalizability, especially for classes with fewer samples. Furthermore, for future event detection endeavors, exploring the integration of image data presents an intriguing direction. As research in image detection progresses, integrating visual data from social media platforms like Twitter (X) may offer additional insights, complementing text analysis for a more comprehensive understanding of events. It is noteworthy that there is a wide range of image detection-related research that can provide inspiration for further exploration and development.

Additionally, expanding event detection to include variables relevant to dynamic pricing of airline tickets presents an intriguing prospect. Integrating insights from event detection into dynamic pricing models holds promise for enhancing pricing strategies, optimizing revenue, and ensuring adaptability in dynamic market landscapes. This research can be extended to develop strategies for observing the dynamics of airline ticket prices by incorporating event features derived from social media. This extension would provide a more comprehensive understanding of price fluctuations influenced by social events.

This study demonstrates the effectiveness of using a stacking ensemble method and fine-tuning BERT for dynamic pricing of airline tickets based on event detection. While the results are promising, there are several areas for further research and potential improvements. Future research could explore integrating real-time data sources, such as social media trends, weather forecasts, and economic indicators, to enhance the dynamic pricing model. This integration could make the pricing model more responsive to sudden changes in demand and supply conditions. Additionally, expanding the feature set to include more variables, such as customer demographics, booking time, and competitor pricing, could improve the accuracy and robustness of the model. Researchers could investigate the impact of these additional features on the model's performance.

Applying transfer learning techniques to adapt the BERT model fine-tuned on airline ticket data to other industries could also be a fruitful area of exploration. This approach could reduce the amount of training data required for new applications, such as hotel booking, car rentals, or event ticket sales, where dynamic pricing is also relevant. Furthermore, while this study focused on stacking ensemble and BERT, other advanced machine learning models, such as reinforcement learning and neural networks, could be explored to optimize dynamic pricing strategies further. Comparative studies could provide insights into the most effective models for different scenarios.

The findings from this study could be applied to other industries that use dynamic pricing. For instance, the retail industry could benefit from dynamic pricing models to adjust prices based on customer behavior and market trends. Similarly, the hospitality industry could use these models to optimize room rates based on occupancy rates and seasonal demand. By addressing these areas for further research, we can continue to enhance dynamic pricing models and expand their applicability to various industries, thereby maximizing their potential benefits. Furthermore, this research can be extended to develop strategies for observing the dynamics of airline ticket prices by incorporating event features derived from social media. This extension would provide a more comprehensive understanding of price fluctuations influenced by social events.

REFERENCES

[1] S.-H. Chung, H.-L. Ma, M. Hansen, and T.-M. Choi, "Data science and analytics in aviation," 2020.

[2] A. K. Choudhary, R. Jagadeesh, E. Girija, M. Madhuri, and N. Shravani, "Flyhigh: Machine learning based airline fare prediction model," in 2023 6th International Conference on Information Systems and Computer Networks (ISCON), pp. 1–8, IEEE, 2023.

[3] A. Dadoun, M. Defoin-Plate, T. Fiig, C. Landra, and R. Troncy, "How recommender systems can transform airline offer construction and retailing," in Artificial Intelligence and Machine Learning in the Travel Industry: Simplifying Complex Decision Making, pp. 93–107, Springer, 2023.

[4] A. Ahmed and A. M. Abdulkareem, "Big data analytics in the entertainment industry: Audience behavior analysis, content recommendation, and revenue maximization," Reviews of Contemporary Business Analytics, vol. 6, no. 1, pp. 88–102, 2023.

[5] J. A. Abdella, N. M. Zaki, K. Shuaib, and F. Khan, "Airline ticket price and demand prediction: A survey," Journal of King Saud University-Computer and Information Sciences, vol. 33, no. 4, pp. 375–391, 2021.

[6] S. Thirumuruganathan, N. Al Emadi, S.-g. Jung, J. Salminen, D. R. Robillos, and B. J. Jansen, "Will they take this offer? a machine learning price elasticity model for predicting upselling acceptance of premium airline seating," Information & Management, vol. 60, no. 3, p. 103759, 2023.

[7] Q. Hou, M. Han, and Z. Cai, "Survey on data analysis in social media: A practical application aspect," Big Data Mining and Analytics, vol. 3, no. 4, pp. 259–279, 2020.

[8] M. Asgari-Chenaghlu, M.-R. Feizi-Derakhshi, L. Farzinvash, M.-A. Balafar, and C. Motamed, "Topic detection and tracking techniques on twitter: a systematic review," Complexity, vol. 2021, pp. 1–15, 2021.

[9] E. Alomari, I. Katib, A. Albeshri, T. Yigitcanlar, and R. Mehmood, "Iktishaf+: a big data tool with automatic labeling for road traffic social sensing and event detection using distributed machine learning," Sensors, vol. 21, no. 9, p. 2993, 2021.

[10] S. Alotaibi, R. Mehmood, I. Katib, O. Rana, and A. Albeshri, "Sehaa: A big data analytics tool for healthcare symptoms and diseases detection using twitter, apache spark, and machine learning," Applied Sciences, vol. 10, no. 4, p. 1398, 2020.

[11] S. Hinduja, M. Afrin, S. Mistry, and A. Krishna, "Machine learning-based proactive social-sensor service for mental health monitoring using twitter data," International Journal of Information Management Data Insights, vol. 2, no. 2, p. 100113, 2022.

[12] C. H. Mendhe, N. Henderson, G. Srivastava, and V. Mago, "A scalable platform to collect, store, visualize, and analyze big data in real time," IEEE Transactions on Computational Social Systems, vol. 8, no. 1, pp. 260–269, 2020.

[13] H. Peng, R. Zhang, S. Li, Y. Cao, S. Pan, and S. Y. Philip, "Reinforced, incremental and cross-lingual event detection from social messages," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 980–998, 2022.

[14] K. Chattu and D. Sumathi, "Corpus creation in telugu: Sentiment classification using ensemble approaches," SN Computer Science, vol. 4, no. 6, p. 860, 2023.

[15] A. H. Hossny, L. Mitchell, N. Lothian, and G. Osborne, “Feature selection methods for event detection in twitter: a text mining approach,” Social Network Analysis and Mining, vol. 10, pp. 1–15, 2020.
[16] A. Dhiman and D. Toshniwal, “An approximate model for event detection from twitter data,” IEEE Access, vol. 8, pp. 122168–122184, 2020.
[17] K. M. Azharul Hasan, S. D. Shovon, N. H. Joy, and M. S. Islam, “Automatic labeling of twitter data for developing covid-19 sentiment dataset,” in 2021 5th International Conference on Electrical Information and Communication Technology (EICT), pp. 1–6, 2021.
[18] M. Erdmann, E. Ward, K. Ikeda, G. Hattori, C. Ono, and Y. Takishima, “Automatic labeling of training data for collecting tweets for ambiguous tv program titles,” in 2013 International Conference on Social Computing, pp. 796–802, 2013.
[19] M. A. W. Nirbhaya and L. H. Suadaa, “Traffic incident detection in jakarta on twitter texts using a multi-label classification approach,” in 2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA), pp. 290–295, 2023.
[20] A. Bhardwaj, A. Blarer, P. Cudré-Mauroux, V. Lenders, B. Motik, A. Tanner, and A. Tonon, “Event detection on microposts: A comparison of four approaches,” IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, pp. 1467–1478, 2021.
[21] J. C. Chamby-Diaz and A. Bazzan, “Identifying traffic event types from twitter by multi-label classification,” in 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), pp. 806–811, 2019.
[22] K. Alfalqi and M. Bellaiche, “Emergency events detection based on integration of federated learning and active learning,” International Journal of Information Technology, vol. 15, no. 6, pp. 2863–2876, 2023.
[23] D. Ramachandran and R. Parvathi, “Enhanced event detection in twitter through feature analysis,” in Research Anthology on Social Media Advertising and Building Consumer Relationships, pp. 442–457, IGI Global, 2022.
[24] A. Bhardwaj, J. Yang, and P. Cudre-Mauroux, “Human-in-the-loop rule discovery for micropost event detection,” IEEE Transactions on Knowledge and Data Engineering, 2022.
[25] Z. Rezaei, B. Eslami, M. A. Amini, and M. Eslami, “Event detection in twitter by deep learning classification and multi label clustering virtual backbone formation,” Evolutionary Intelligence, vol. 16, no. 3, pp. 833–847, 2023.
[26] P. Goyal, P. Kaushik, P. Gupta, D. Vashisth, S. Agarwal, and N. Goyal, “Multilevel event detection, storyline generation, and summarization for tweet streams,” IEEE Transactions on Computational Social Systems, vol. 7, no. 1, pp. 8–23, 2019.
[27] A. Yavari, H. Hassanpour, B. Rahimpour Cami, and M. Mahdavi, “Event prediction in social network through twitter messages analysis,” Social Network Analysis and Mining, vol. 12, no. 1, p. 78, 2022.
[28] D. Chen and R. Zhang, “Covid-19 vaccine adverse event detection based on multi-label classification with various label selection strategies,” IEEE Journal of Biomedical and Health Informatics, 2023.
[29] A. S. Alammary, “Arabic questions classification using modified tf-idf,” IEEE Access, vol. 9, pp. 95109–95122, 2021.
[30] 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.
[31] M. G. Meharie, W. J. Mengesha, Z. A. Gariy, and R. N. Mutuku, “Application of stacking ensemble machine learning algorithm in predicting the cost of highway construction projects,” Engineering, Construction and Architectural Management, vol. 29, no. 7, pp. 2836–2853, 2022.
[32] X. Yin, Q. Liu, Y. Pan, X. Huang, J. Wu, and X. Wang, “Strength of stacking technique of ensemble learning in rockburst prediction with imbalanced data: Comparison of eight single and ensemble models,” Natural Resources Research, vol. 30, pp. 1795–1815, 2021.
[33] Z. Jin, X. Lai, and J. Cao, “Multi-label sentiment analysis base on bert with modified tf-idf,” in 2020 IEEE International Symposium on Product Compliance Engineering-Asia (ISPCE-CN), pp. 1–6, 2020.
[34] Y. Chen, X. Kou, J. Bai, and Y. Tong, “Improving bert with self-supervised attention,” IEEE Access, vol. 9, pp. 144129–144139, 2021.
[35] A. G. Gozal, H. Pranoto, and M. F. Hasani, “Sentiment analysis of the indonesian community toward face-to-face learning during the covid-19 pandemic,” Procedia Computer Science, vol. 227, pp. 398–405, 2023.
[36] K. Bok, I. Kim, J. Lim, and J. Yoo, “Efficient graph-based event detection scheme on social media,” Information Sciences, vol. 646, p. 119415, 2023.
[37] P. Jha, D. Dembla, and W. Dubey, “Deep learning models for enhancing potato leaf disease prediction: Implementation of transfer learning based stacking ensemble model,” Multimedia Tools and Applications, vol. 83, no. 13, pp. 37839–37858, 2024.
[38] Y. Sun, H. Zhang, T. Zhao, Z. Zou, B. Shen, and L. Yang, “A new convolutional neural network with random forest method for hydrogen sensor fault diagnosis,” IEEE Access, vol. 8, pp. 85421–85430, 2020.
[39] W. Han, “Analyzing the scale dependent effect of urban building morphology on land surface temperature using random forest algorithm,” Scientific Reports, vol. 13, no. 1, p. 19312, 2023.
[40] W. Zhou, Z. Yan, and L. Zhang, “A comparative study of 11 non-linear regression models highlighting autoencoder, dbn, and svr, enhanced by shap importance analysis in soybean branching prediction,” Scientific Reports, vol. 14, no. 1, p. 5905, 2024.
[41] M. Najafzadeh, S. Basirian, and Z. Li, “Vulnerability of the rip current phenomenon in marine environments using machine learning models,” Results in Engineering, vol. 21, p. 101704, 2024.
[42] J. Yao, Z. Wang, L. Wang, M. Liu, H. Jiang, and Y. Chen, “Novel hybrid ensemble credit scoring model with stacking-based noise detection and weight assignment,” Expert Systems with Applications, vol. 198, p. 116913, 2022.
[43] A. Dumakude and A. E. Ezugwu, “Automated covid-19 detection with convolutional neural networks,” Scientific Reports, vol. 13, no. 1, p. 10607, 2023.
[44] M. A. Khan, N. Iqbal, H. Jamil, D.-H. Kim, et al., “An optimized ensemble prediction model using automl based on soft voting classifier for network intrusion detection,” Journal of Network and Computer Applications, vol. 212, p. 103560, 2023.
[45] A. Batool and Y.-C. Byun, “Towards improving breast cancer classification using an adaptive voting ensemble learning algorithm,” IEEE Access, 2024.
[46] H. Chen, D. Jiang, and H. Sahli, “Transformer encoder with multi-modal multi-head attention for continuous affect recognition,” IEEE Transactions on Multimedia, vol. 23, pp. 4171–4183, 2020.
[47] J. Li, B. Chiu, S. Shang, and L. Shao, “Neural text segmentation and its application to sentiment analysis,” IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 2, pp. 828–842, 2020.
[48] R. Qasim, W. H. Bangyal, M. A. Alqarni, A. Ali Almazroi, et al., “A fine-tuned bert-based transfer learning approach for text classification,” Journal of healthcare engineering, vol. 2022, 2022.
[49] S. Bano, S. Khalid, N. M. Tairan, H. Shah, and H. A. Khattak, “Summarization of scholarly articles using bert and bigru: Deep learning-based extractive approach,” Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 9, p. 101739, 2023.
[50] S. Bano and S. Khalid, “Bert-based extractive text summarization of scholarly articles: A novel architecture,” in 2022 International Conference on Artificial Intelligence of Things (ICAIoT), pp. 1–5, IEEE, 2022.
[51] W. Khan, A. Daud, K. Khan, S. Muhammad, and R. Haq, “Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends,” Natural Language Processing Journal, p. 100026, 2023.
[52] F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn, “A systematic evaluation of large language models of code,” in Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming, pp. 1–10, 2022.
[53] J. Hou, X. Li, H. Yao, H. Sun, T. Mai, and R. Zhu, “Bert-based chinese relation extraction for public security,” IEEE Access, vol. 8, pp. 132367–132375, 2020.
[54] S. Khalid, S. Wu, and F. Zhang, “A multi-objective approach to determining the usefulness of papers in academic search,” Data Technologies and Applications, vol. 55, no. 5, pp. 734–748, 2021.

NUR ALAMSYAH received his bachelor's degree in informatics engineering from Sekolah Tinggi Manajemen Informatika (STMIK) Bandung, Bandung, Indonesia, in 2003, and his master's degree in information systems from Sekolah Tinggi Manajemen (STMIK) LIKMI, Bandung, in 2013.

He is currently pursuing his PhD at the Doctor of Computer Science Program, Telkom University, Bandung. His dissertation topic is "A New Model of Dynamic Pricing of Airline Tickets with Event Sentiment Factor from Social Media Twitter (X)". Currently, I am a lecturer and researcher at Universitas Informatika Dan Bisnis Indonesia in the information systems study program.

He is the author of 4 books and more than 50 articles. He has participated in more than 20 international seminars published in IEEE Xplore. His research field is data science.

SAPARUDIN received his bachelor's degree in 1993 at Sriwijaya University in the mathematics education study program. Master's degree was obtained in 2000 from ITB Bandung. Ph.D. degree from Universiti Teknologi Malaysia in 2012. Currently he is a Professor in the field of computer science at Telkom University, and he is also a full-time lecturer and researcher at Telkom University. His current research interests include Computer Vision and Data Science.

Previously, he has received institutional and national research grants from Sriwijaya University and the Ministry of Higher Education in the last five years. Grants on topics like "Prototype of Endemic Disease Detection System Based on Image Texture Using Method Development", "Multi-Object Face Detection System Using Internet of Things Technology", "Design of Disease Classification System Based on Retinal Image Using Convolutional Neural Network", and others received from the Sriwijaya University DIPA grant.

He has authored more than 40 articles published in international journals and proceedings. His articles include "Face Detection Using the Viola-Jones Method with Skin Color Segmentation", "Hybrid Multilevel Thresholding-Otsu and Morphology Operation for Retinal Blood Vessel Segmentation", and "Multiple Face Image Feature Extraction Using Geometric Moment Invariants Method".

ANGELINA PRIMA KURNIATI She was born in Kudus, Central Java, Indonesia, on July 1983. She gained a bachelor's degree in Informatics (Telkom University, Indonesia, 2005), a master's degree in Informatics (Bandung Institute of Technology, Indonesia, 2010), and a Ph.D. in Computer Science (University of Leeds, UK, 2020). She is an Associate Professor at the School of Computing at Telkom University. She teaches regular and international classes on many subjects, including process mining and advanced data mining. Her current research interest is process mining, where she applied to healthcare and academics, among other domains. She was involved in many projects, including the analysis of patient treatments in BPJS Kesehatan dataset using process mining.

She has published more than 20 articles in international and national journals. Some of them are "Process mining for healthcare: Characteristics and challenges", "Process mining of disease trajectories: A literature review", "The assessment of data quality issues for process mining in healthcare using Medical Information Mart for Intensive Care-III, a freely available e-health record database", and many more have been written and published in journals and proceedings.