IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 August 2022, accepted 16 September 2022, date of publication 26 September 2022,
date of current version 30 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3209662

RESEARCH ARTICLE

HyVADRF: Hybrid VADER–Random Forest and GWO for Bitcoin Tweet Sentiment Analysis

ANNY MARDJO[1] AND CHIDCHANOK CHOKSUCHAT[2], (Member, IEEE)
[1]College of Digital Science, Prince of Songkla University, Hat Yai, Songkhla 90110, Thailand
[2]Division of Computational Science, Faculty of Science, Prince of Songkla University, Hat Yai, Songkhla 90110, Thailand

Corresponding author: Chidchanok Choksuchat (chidchanok.ch@psu.ac.th)

This work was supported in part by the Digital Science for Economy, Society, Human Resources Innovative Development and Environment Project through Reinventing Universities and Research Institutes under Grant 2046735 of the Ministry of Higher Education, Science, Research and Innovation, Thailand; and in part by APNIC (Asia Pacific Network Information Centre) Foundation through SWITCH SEA (Supporting Women in Internet Research Leaders in Southeast Asia) Project-TH-03, for data science and applied AI professional knowledge, and research support.

ABSTRACT In recent years, Bitcoin and other cryptocurrencies have been increasingly considered investment options for emerging markets. However, Bitcoin's erratic behavior has discouraged some potential investors. To get insights into its behavior and price fluctuation, past studies have discovered the correlation between Twitter sentiments and Bitcoin behavior. Most of them have exclusively focused on their relationships, instead of the Twitter sentiment analysis itself. Finding the most suitable classification algorithms for sentiment analysis for this kind of data is challenging. For the enormous data in Twitter, the supervised sentiment analysis approach of unlabeled data can be time-consuming and expensive, which has been studied to be superior to unsupervised ones. As such, we propose the HyVADRF (hybrid valence aware dictionary and sentiment reasoner (VADER)–random forest) and gray wolf optimizer (GWO) model. A semantic and rule-based VADER was used to calculate polarity scores and classify sentiments, which overcame the weakness of manual labeling, while a random forest was utilized as its supervised classifier. Furthermore, considering Twitter's massive size, we collected over 3.6 million tweets and analyzed various dataset sizes as these are related to the model's learning process. Lastly, GWO parameter tuning was conducted to optimize the classifier's performance. The results show that 1) the HyVADRF model had an accuracy of 75.29%, precision of 70.22%, recall of 87.70%, and F1-score of 78%. 2) The most ideal dataset size percentage is 90% of the total collected tweets ($n = 1{,}249{,}060$). 3) The standard deviations are 0.0008 for accuracy and F1-score and 0.0011 for precision and recall. Hence, the HyVADRF model consistently delivers stable results.

INDEX TERMS Hyperparameter, random forest, bitcoin, gray wolf optimization, tweet sentiment analysis, VADER.

I. INTRODUCTION

As one of the interesting topics in the present world, cryptocurrency has changed the way people think about money. It is a digital currency governed by a cryptographic protocol that uses Blockchain technology [1]. Its continuous adoption and widespread usage have added value in its real-world applications by a substantial amount. The first cryptocurrency is Bitcoin, which was developed in 2009 [2]. It is a type of electronic cash without central governing and can be used as a medium for online transactions between any two parties. Bitcoin is a very volatile currency, and its price is influenced by socially constructed opinions. Past studies discovered that some of the extreme price increases and decreases in Bitcoin coincided with dramatic events in China [3]. The rise of the Internet technology has played an unprecedented role in increasing the number of users' opinions and emotions shared on social media and e-commerce platforms either by text or multimedia data [4], [5], [6], [7]. This phenomenon has resulted in the production and generation of a large variety of data, which can be analyzed to assess sentiments. The analysis of sentiments is beneficial for individuals and

The associate editor coordinating the review of this manuscript and approving it for publication was Rosalia Maglietta.

organizations, especially given the immense production of data [8]. Although several sentiment analysis approaches for opinion mining have been developed, such as machine learning, lexicon-based approach, and hybrid approaches, supervised machine learning has been proven to be more accurate than unsupervised ones. However, to build and evaluate a classifier model, this approach needs labeled data [9], which can be tedious, expensive, and error prone [10]. This can be problematic for typically scarce labeled and enormous data, such as a microblogging system like Twitter. Thus, the algorithm used for this study, HyVADRF (hybrid valence aware dictionary and sentiment reasoner (VADER)–random forest (RF)) and gray wolf optimizer (GWO), aids in overcoming the manual labeling problem by performing non-manual labeling using the semantic lexicon-based VADER algorithm.

Compared with texts in traditional media, texts in microblogging can be noisy, arbitrary, and ambiguous [11], [12], making it difficult for supervised machine learning classification to infer knowledge from them. Text representation models, such as term frequency–inverse document frequency (''*TF–IDF*'') or ''*n-gram*,'' often lead to a high-dimensional feature space because of the large-scale size of the dataset and vocabulary. Furthermore, short and noisy texts make the data representation very sparse. This high-dimension sparse representation poses significant challenges in building an interpretable model with a high prediction accuracy. Meanwhile, microblogging's large-scale size can provide more raw data to extract features for model complexity, which makes the machine learning model more robust and accurate [13]. Although the dataset size can control the learning process and determine the values of model parameters that a learning algorithm ends up learning, only a few studies have explored this factor in sentiment analysis. Thus, another important factor of sentiment analysis is the ideal choice of dataset size to capture all necessary features to create a performance classifier model when raw data are large because labeling and processing all raw data is extremely time-consuming.

In sum, the contributions of this paper are threefold: (1) We show how semantic lexicon-based VADER can be used to label tweets. (2) We add knowledge on the machine learning algorithm for the sentiment analysis of tweets. (3) We reveal the impact of the dataset size on the machine learning performance.

The remainder of this paper is presented as follows: Section II presents related works. Section III discusses the study's methodology. Section IV gives the Results. Section V presents the discussion. Finally, Section VI presents the conclusions of the study.

II. RELATED WORKS

A. TWITTER SENTIMENT ANALYSIS

The rising popularity of cryptocurrency has increased the spread of its information through online media and social online platforms [14]. By analyzing sentiments on socioeconomic phenomena and public opinions, social media can be used to predict future events and changes [15]. The correlation between Twitter and the price prediction of cryptocurrency has been validated in previous studies [16], [17].

In recent years, hybrid sentiment analysis combining a semantic lexicon and supervised machine learning has been increasingly studied [18], [19], [20]. One of the most popular lexical semantic approaches to calculate sentiment polarity scores is VADER. Introduced in 2014, VADER is a lexicon and rule-based sentiment analysis model that calculates the polarities (positive/negative) and intensity (strength) of emotions to obtain the sentiment score. The advantages of VADER include the following: (i) It is an open-source tool; (ii) it is a human-centric approach; and (iii) it is particularly designed for social media content [21]. Furthermore, supervised machine learning algorithms, such as support vector machine (SVM) and naive bayes (NB), are the most frequently used algorithms for sentiment analysis either in combination with VADER or on their own. Supervised learning has been found to provide more accurate sentiment analysis than unsupervised learning, such as sentiment lexicons [22].

Saif *et al.* [12] showed that Twitter data are sparser than other types of data (e.g., movie review data) due to the large number of infrequent words present within tweets. Such a feature can be due to spelling mistakes and the usage of slang words. Furthermore, Twitter contains a large amount of noisy data, such as URLs, punctuation, and special symbols. Thus, irrelevant words and data, which are merely present due to some coincidence or do not influence the current text, may affect the average polarity or entropy of the text as these are outliers to the text in focus. The automated identification of relevant information from these data is imperative due to the immense volume of raw data, which have prompted many researchers [23], [24], [25], [26] to explore various feature selection methods and classifier models. Due to its simplicity and computation efficiency, a very popular structured text representation method is the bag-of-words model in which documents or sentences are represented as a list of words using a document-term matrix (DTM) [27]. The association of words in the matrix is formed based on the distances between them. This approach has been successfully applied for text classification, text clustering, and information retrieval. Most DTMs tend to be high dimensional and sparse [28] because any given document will contain only a subset of unique terms that appear throughout the corpus. This condition will result in any corresponding document row having zeros for terms that were not used in that specific document. Therefore, we need an approach to reduce dimensionality. TF–IDF is a popular method of evaluating the word weight value in a collection of documents [29], [30]. It represents the distribution of each word in a document across the entire document or corpus. Each word is assigned a TF–IDF score by multiplying the word's TF by its IDF. The steps to get a TF–IDF score is 1) to calculate the TF value with ((1), 2) calculate the IDF value with (2), and 3) calculate the TF–IDF weight value with (3).

$$\boldsymbol{tf}\,(t,d) = \frac{f_d(t)}{\max\limits_{w \in d} f_d(w)} \tag{1}$$

$$\boldsymbol{idf}\ (t, D) = \ln\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \quad (2)$$

$$\boldsymbol{tfidf}\,(t, d, D) = \boldsymbol{tf}\,(t, d)\ x\ \boldsymbol{idf}\,(t, D) \quad (3)$$

where **tf***(t,d)* represents the number of times that a word appears in a document and **idf***(t,D)* is the number of documents that contain that word [29]. *TF–IDF (t,d,D)* is the natural logarithm of the total number of documents divided by the word's *DF*. In text mining, the *TF–IDF* approach of the DTM is similar to the mean, instead of the median, and these outliers will be included in its calculation if they are not removed. Therefore, the dispersed terms of the matrix should be removed to preserve only the most frequent words, aid generalization, and prevent overfitting. This method generally reduces the matrix without losing significant relations inherent to the matrix. It can be performed using the function *RemoveSparseTerm()* of R. For its advantage, we decided to use the *TF–IDF* approach for this study.

As previously mentioned, this study also aims to analyze the impact of the dataset size on the performance of sentiment analysis algorithms as the training dataset size is related to the model's learning process. The best suitable selection gives the optimum performance for the developed model. In a recent study [31], the importance of obtaining adequately sized, unbiased validation and training sets was identified as a crucial factor in the assessment and development of robust machine learning models. The dataset size can be considered the model hyperparameter, in which an ideal configuration is an external part of machine learning algorithms and cannot be estimated from the observed data.

B. GWO

Bio-inspired computing (BIC) are algorithms based on the natural behavior of animals, birds, insects, and other natures. These algorithms require several algorithm-dependent parameters and a certain number of iterations to gain the optimized value of the objective function. Hence, it is time and resource consuming. Nevertheless, these algorithms can uncover unknown patterns and have a lower reliance on mathematical modeling or exhaustive training [32].

According to Tang and Wu [33], BIC can be classified into three categories: evolutionary algorithms (EAs), swarm intelligence (SI), and bacterial foraging algorithms (BFAs). Inspired from the genetic evolution process, the most popular EA is the genetic algorithm (GA). GA is based on Charles Darwin's theory of survival of the fittest that uses crossover and mutation as two operators [34]. The second category, SI, draws inspiration from animal behaviors. The most popular algorithms from this category are particle swarm optimization (PSO) and ant colony optimization (ACO). The last category, BFA, is a novel SI algorithm based on the foraging behavior of *E. coli* [35].

Past studies have adopted various BIC categories to optimize the hyperparameters of machine learning algorithms in various domains. In their studies on malaria risk prediction, Tai and Dhaliwal [36] applied a GA to optimize the hyperparameter value of three machine learning algorithms (LightGBM, ridge regression, and support vector regression). Hu *et al.* [37] compared PSO with other SI models, GWO [38], and GA to optimize the SVM rock mass classifier model. The results showed that the GWO-optimized SVM performed the best. ACO was adopted by Koyhomayoon *et al.* [39] to optimize adaptive neuro-fuzzy inference systems to predict the groundwater level.

In this study, we used GWO. Introduced in 2014, this algorithm is inspired by the leadership hierarchy and hunting mechanism of gray wolves in nature. There are four types of wolves in the gray wolf hierarchy. The oldest and leader of the pack is the alpha (α), with the main responsibility of deciding for the pack. The next rank is the beta (β), which is an advisor of the alpha and discipliner of the pack. The lowest rank in the hierarchy is the omega (Ω), which is required to yield to other dominant wolves. The delta (δ) wolf dominates the omega and reports to the alpha and beta. According to Kayhomayoon *et al.* [39], this algorithm uses the following steps: 1) a wolf calculates its distance from α, β, and δ using Equations 4–9 and 2) update its position with Equation 10.

$$D_\alpha = |2r_2 \cdot X_\alpha - X_i|, \quad (4)$$

$$D_\beta = \left|2r_2 \cdot X_\beta - X_i\right|, \quad (5)$$

$$D_\delta = |2r_2 \cdot X_\delta - X_i|, \quad (6)$$

$$X_1 = X_\alpha - (2a \cdot r_1 - a) \cdot D_\alpha, \quad (7)$$

$$X_2 = X_\beta - (2a \cdot r_1 - a) \cdot D_\beta, \quad (8)$$

$$X_3 = X_\delta - (2a \cdot r_1 - a) \cdot D_\delta, \quad (9)$$

$$X_1\,(t+1) = \frac{X_1 + X_2 + X_3}{3}, \quad (10)$$

where X_α, X_β, and X_δ are the positions of α, β, and δ, respectively. D_α, D_β, and D_δ represent the distances between i and other wolves (α, β, δ). With the iteration process, a decreases linearly from 2 to 0. *r1* and *r2* are two random numbers between range parameters for the boundary search space. Fig. 1 depicts the flow chart of GWO.

In their study on email detection, Batra *et al.* [40] found that k-NN classification combined with GWO had 100% recall and the least computational times among the Bayesian information criterion algorithms.

III. METHODOLOGY

In this section, we propose the HyVADRF and GWO model for bitcoin tweet sentiment analysis research framework due to its benefits. First, this algorithm uses the VADER algorithm to calculate a compound polarity score for labeling raw data, which is less expensive, error prone, and faster compared to manual labeling. Second, as supervised machine learning was known to be better than unsupervised ones, we decided to use RF, NB, L2-SVM, and DT as machine learning algorithms. Third, the GWO algorithm and *tuneRanger* were used to tune the parameters for machine learning optimization. Fig. 2 presents the proposed sentiment analysis of Twitter tweets related to the Bitcoin framework.

A. DATA EXTRACTION

Data were collected between January 1, 2021 and December 31, 2021. Tweets were crawled by employing Twitter API.

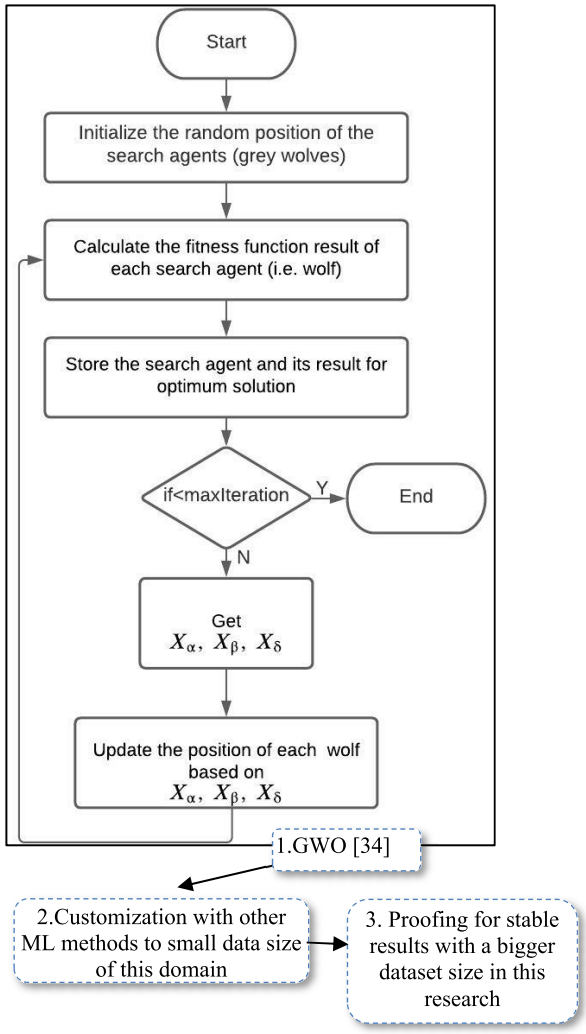


FIGURE 1. Flowchart of GWO adapted from [34].

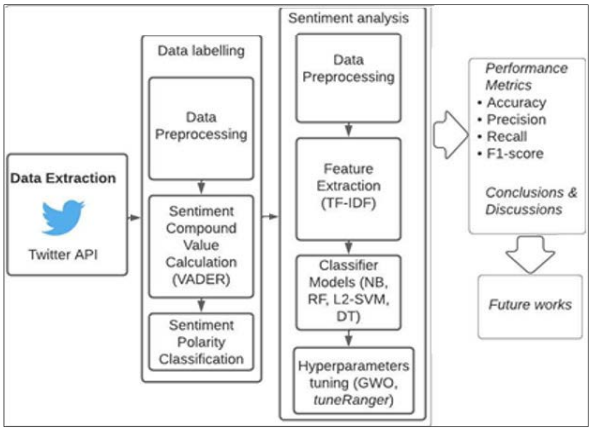


FIGURE 2. HyVADRF: Hybrid VADER–random forest and GWO for Bitcoin tweet sentiment analysis research framework.

During crawling, all the tweets with the keyword "Bitcoin" in either the content or hashtags were collected. We restricted our collection to English tweets only to avoid a mixed-language dataset. The total dataset collected was 3,625,091 tweets.

B. DATA PREPROCESSING AND LABELING USING VADER

The tweet dataset does not enclose a labeled output. Tags consisting of positive or negative are labeled to train a supervised classifier. Thus, VADER, a rule-based lexicon method, was applied to label the dataset. Before VADER was applied to tweets, "noise" removal was performed to raw data. Manual cleaning of raw data and the use of the regular expression (*RegEx*) in natural language processing (i.e., removal of URL links, hashtags symbols, and irrelevant tweets) were used very carefully to avoid decreased accuracy.

Afterward, VADER was used to produce the score values of negative, neutral, positive, and compound polarities for each tweet. Following Pano and Kashef [41], a compound value below or equal to -0.05 is considered to have a negative polarity, whereas that greater or equal to 0.05 has a positive polarity. The values between 0.05 and -0.05 have a neutral polarity. The pseudocode is presented in Algorithm 1.

Algorithm 1: VADER Labeling Algorith

Input: Twitter dataset T
Output: Labeled Twitter dataset L
Process:
To generate data labeling:
for $t \in T$
    Clean t from URL links, hashtags symbols, and irrelevant tweets using RegEx.
    Calculate the sentiment compound value (cv) using the VADER library
    if cv is greater or equal to 0.05
        class = "positive"
    elseif cv is less than 0.05 and cv is greater than -0.05
        class = "neutral"
    else
        class = "negative"
    end if
end for

C. SENTIMENT CLASSIFICATION

Tweets were preprocessed before the machine learning algorithms were applied. Neutral-value comments are detached. Only tweets with positive and negative labels were preprocessed and used for machine learning algorithms, following a prior study [18].

The preprocessing steps started with creating corpus documents for this dataset. Then, "noise" removal steps, such as eliminating punctuations and numbers, were performed. The next step is removing stop words in English (e.g., "are," "as," "is," "of," and "the"), which are unnecessary words in classifying the documents. Afterward, stemming is performed, which is a process of transforming different tenses of words to their root form (e.g., fishing, fish, and fisher to fish). This step aids in the removal of unwanted computation of words and therefore reduces the time consumed by the algorithm in training all the tenses of words. The unnecessary white spaces were also removed. A DTM using the TF–IDF feature extraction method was applied to convert the documents into feature (i.e., term) vectors. These vectors can easily be understood by a machine learning algorithm.

Training each algorithm to classify text data using an entire document or a sentence is an important text data classification step, but it is very hard. Thus, tokenization is necessary to transform a sentence into terms and use them in classifier training.

Algorithm 2: Machine Learning Trainin

Input: Labelled Twitter dataset L,
resulting from Algorithm 1
Output: classifying the machine learning model
Process:
To select the best machine learning model:
- Create a vector corpus (V) of L
- Clean V by removing punctuations, numbers, stopping words, and stemming
- Create a document term matrix (M) using TF–IDF
- Remove sparse terms from the M using term sparsity threshold
- Split M into 70% for the training set (N) and 30% for the test set (D)

for $j \in$ (NB, DT, RF, L2-SVM)
train N using the j model
test the trained model using D
end for
Choose a trained model with the best performance (accuracy, precision, recall, and F-score).

The dataset was divided data to 7:3, i.e., 70% for training and 30% for testing. Using five-fold cross-validation, four supervised machine learning algorithms, namely, NB, DT, L2-SVM, and RF, were employed to train the models. As the dataset was in a large quantity, we used R packages that could efficiently process the data: the package *ranger* for RF [42], *LiblineaR* for L2-SVM [43], *fastNaiveBayes* for NB [44], and *caret* for DT [45].

In the five-fold cross-validation, five nearly identical-sized divisions were randomly divided from the dataset, where a division was used for the testing set and four divisions for the training set of classification. This process was repeated five times, and the final result is the average of the five evaluations. The performance of each model was calculated using the ''flat'' performance measure of the confusion matrix, such as accuracy, precision, recall, and F-score, as this study performed a binary classification [46]. In detail, a confusion matrix has a true positive (TP), which is correctly classified as negative tweets, whereas a true negative (TN) is correctly classified as positive tweets. Meanwhile, a false positive (FP) is the positive tweets that are misclassified as negative tweets, and a false negative (FN) is the negative tweets that are misclassified as positive tweets.

D. HYPERPARAMETER TUNING

In the current study, the RF has the highest performance among the machine learning algorithms. To obtain the optimized performance of RF, its hyperparameters were tuned using GWO and *tuneRanger* [47]. As both tuning approaches required multiple iterations, we randomly subset 100,000 records from the full cohort to keep the computational time reasonable. For GWO, four important hyperparameters were tuned, as suggested in the literature [47], [48]. Table 1 summarizes the tuned hyperparameters, the definition, and their tuning ranges. The population was set to 30 with the max iteration of 100 in the GWO, as in the past study [49].

TABLE 1. Hyperparameters' description and their tuning range.

No	Hyperparameters	Explanation	Range
1	*min.node.size*	Minimum number of observations in terminal node	[1–10]
2	*num.trees*	Maximum depth of decision trees	[500–3000]
3	*mtry*	Number of variables to possibly split in each node.	[1–92]
4	*sample.fraction*	Fraction of observations to sample	[0.2–1]

During the hyperparameter tuning, the training performance from the fivefold CV was used as the fitness function of the GWO. Each hyperparameter was represented by a wolf in the GWO. With each iteration of GWO, wolf positions were updated to maximize the fitness value, and the hyperparameters were optimized accordingly. The pseudo-code is depicted in Algorithm 3.

Another hyperparameter tuning method, *tuneRanger*, allows simultaneously tuning RF parameters using an automatic model-based optimization process [47]. Arguments for this method were set to defaults based on the provided example of the literature [47]. Finally, to explore the effect of hyperparameter tuning, we compared the performance of the standard RF and tuned RF (GWO-tuned RF and *tuneRanger*-tuned RF) with the dataset size that gave the highest performance metrics of the standard RF. The standard RF used the default hyperparameter values specified in the *ranger* R package.

Algorithm 3: GWO Optimization Algorith

Input: classifying the machine learning model *ML*, training set (N), and test set (D) resulting from Algorithm 2
Output: optimized classifying machine learning model
Process:
To perform model optimization using GWO:
- Initialize the random position of the gray wolf population X_i ($I = 1, 2, \ldots, n$)
- Calculate the fitness function of each search agent using the *ML* train model with N and D
- Store the search agent and its fitness function

While $t <$ *maxIteration*
For each search agent
Update the position of the current search agent
End for
$t = t+1$
end while
Get the search agent with an optimum fitness function

E. MACHINE SPECIFICATION

The machine specifications are as follows: Lenovo IdeaPad S1145-14IIL, Processor Intel^{TM} Core^{TM} i5-1035G1 CPU @ 1.00 GHz 1.19 GHz, installed RAM 20.0 GB (19.8 GB usable), and system type 64-bit operating system, and x64-based processor. The installed software is RStudio version 1.4.1717. R's libraries were used in this set: {*readr, tm, caret, Metrics, caret, fastNaiveBayes, LiblineaR, metaheuristicOpt, vader, dplyr*}.

IV. RESULTS

Prior to the current study, we conducted a preliminary experiment using the current research method and framework for a sample dataset of 1,000 positive and 1,000 negative tweets spanning 12 days from June 4, 2021 to June 15, 2021. The preliminary finding found that the model accuracy was 86.12% and the F1-score was 86.18%. Based on this promising result, we expanded the timeline of the data collection to one year as yearly data would be a reliable representation of an entire boom and bust cycle of Bitcoin prices.

The labeled data obtained from the polarity score of VADER for the three classes (positive, negative, and neutral) are graphically represented in Fig. 3. The final labeled dataset contains a total of 3,625,091 tweets with 1,879,669 positive tweets; 1,120,892 neutral tweets; and 624,530 negative tweets.

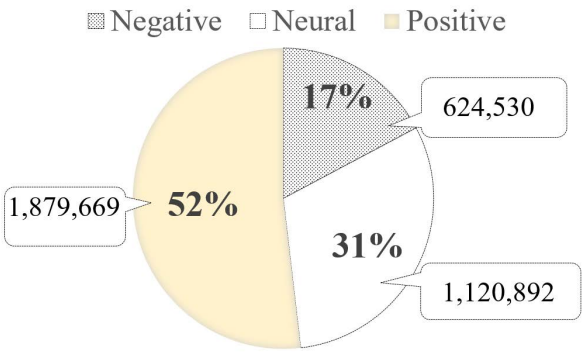
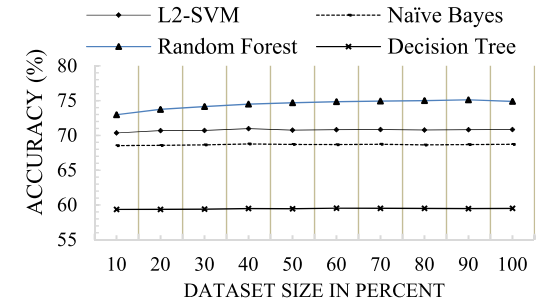
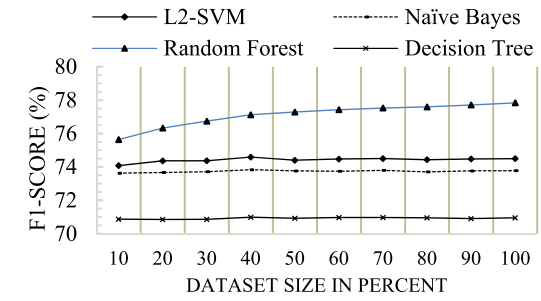


FIGURE 3. Percentage of the sentiment VADER results.

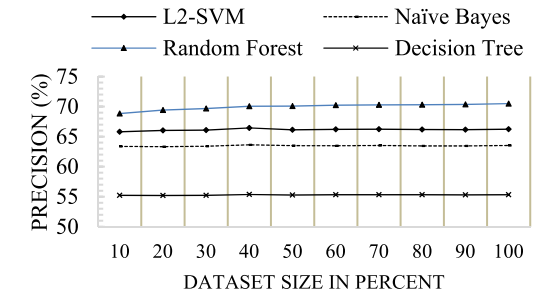
As we intended to perform binary classifications (positive and negative), class imbalance clearly occurred as the positive class had a much larger number of observations than the negative class. This factor could cause a machine learning bias toward the minority class and thus the poor performance of the classifier [50]. To improve our class imbalance, we under-sampled positive tweets, and we used the same number of observations (624,500 tweets) for each positive and negative tweet, which resulted in a total of 1,249,000 tweets. We adopted undersampling of the majority class as the oversampling approach duplicates the sample of the minority class, which can cause overfitting [51]. The undersampling approach was also more effective for our study because our minority class has a sufficient number of samples despite the severe imbalance. The results of using various dataset size percentages and machine learning algorithms are depicted in Fig. 3.



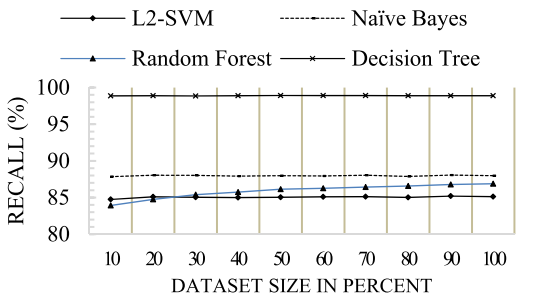
(a) ***Accuracy*** *(y-axis) vs. dataset size (x-axis)*



(b) ***F1-score*** *(y-axis) vs. dataset size (x-axis)*



(c) ***Precision*** *(y-axis) vs. dataset size (x-axis)*



(d) ***Recall*** *(y-axis) vs. dataset size (x-axis)*

FIGURE 4. Performance results using various ML algorithms.

The evaluation of the performance of the machine learning algorithms is shown in Fig. 4. We gradually increased the percentage of the dataset size for the training data and test data. We performed a baseline random inference implementation by re-shuffling, re-sampling, and running each algorithm for five times with different seeds and used the average accuracy, precision, recall, and F1-score. For all the dataset

size percentages, RF gave the best results in terms of accuracies in the range of 72%–75%, precisions of 68%–70%, and F1-scores of 75%–77%. DT achieved the highest recall scores with values above 98%. However, these were compensated with low precision scores in the range of 55% and F1-scores of 70%, which made DT an unsuitable algorithm for these data. Meanwhile, RF did not have the highest recall scores, but they were within the range of 83%–86%. Thus, RF is the most suitable algorithm for this dataset.

Using RF, we analyzed the most ideal dataset size percentage. As depicted in Fig. 5, 90% gave the smallest difference of accuracy between training set and test set, deeming it as the most ideal dataset size percentage.

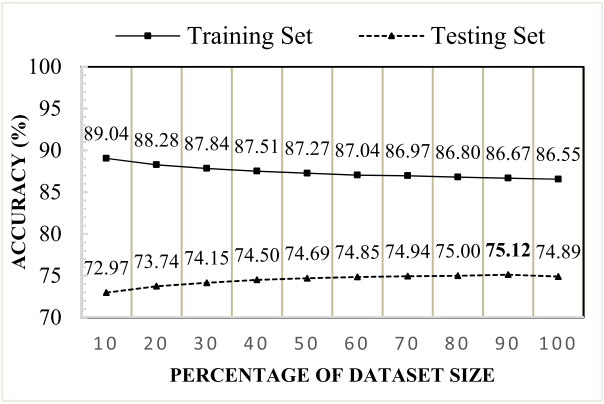


FIGURE 5. Learning curve of RF.

Although 90% of the whole 1,249,060 tweets (from $n = 1{,}124{,}154$) is the most ideal size for these data, we only used a random subset of 100,000 tweets for GWO and *tuneRanger* optimizations as they required multiple iterations, which consumed the computing time and resources. Using GWO, the optimum RF hyperparameters were determined to be *min.node.size* = 2, *num.trees* = 2500, *mtry* = 6 and *sample.fraction* = 1 Meanwhile, *tuneRanger* returned *num.threads* = 2, *mtry* = 5, *min.node.size* = 2, *sample.fraction* = 0.648, and *num.trees* = 1000. These RF hyperparameters were then used to train 1,124,154 tweets using a 70% training set and 30% test, which are shown in Fig. 5.

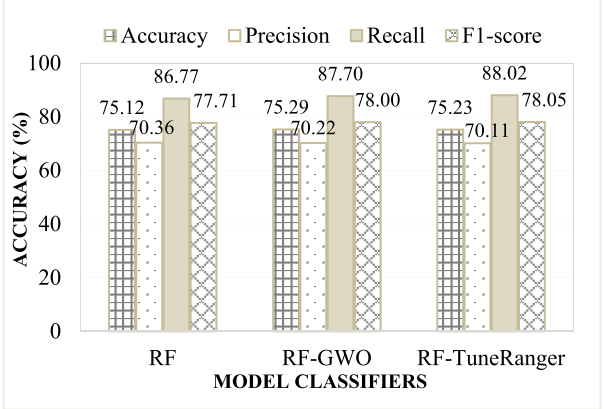


FIGURE 6. Model classifiers with different evaluation metrics.

Fig. 6 demonstrates a performance comparison of the RF models with default parameters and the optimum parameters from GWO and *tuneRanger*. Accuracy was increased from 75.12% to 75.29% (RF–GWO) and 75.23% (RF–*tuneRanger*). The F1-score increased from 77.71% to 78% (RF–GWO) and 78.05% (RF–*tuneRanger*). Similarly, the recall was improved from 86.77% to 87.70% (RF–GWO) and 88.02% (RF–*tuneRanger*). At the same time, the precision decreased from 70.36% to 70.22% (RF–GWO) and 70.11% (RF–*tuneRanger*).

In general, the hybrid RF–GWO and hybrid RF–*tuneRanger* slightly outperformed the single RF model. This slight improvement is not surprising as the improvement through tuning tends to be less obvious where RF performs satisfactorily [52]. Furthermore, the impact of RF tuning is much smaller compared to that of other machine learning algorithms, such as SVM [53].

To obtain more representative results, we also compared the standard deviation (SD) of each model. In terms of the accuracy, the SD decreased from 0.0015 to 0.0008 (RF–GWO) and 0.0014 (RF–*tuneRanger*). The SD of precision was reduced from 0.0020 to 0.0011 (RF–GWO) and 0.0016 (RF–*tuneRanger*). Moreover, the SD of recall increased from 0.0007 to 0.0011 (RF–GWO) and 0.0019 (RF–*tuneRanger*). The SD of the F1-score decreased from 0.0011 to 0.0008 for RF–GWO but increased to 0.0015 for RF–*tuneRanger*. These results confirmed that the RF–GWO is more stable compared to either a single RF or RF–*tuneRanger*. In addition, they showed the feasibility of GWO to improve the classifier model.

Although our hybrid VADER RF–GWO model has a lower accuracy (75.29%) compared to those proposed in similar past studies [18], [20], the dataset we used was much larger than their studies. In their studies of evaluating the performance of Indonesian politicians based on YouTube comments using a hybrid lexicon and SVM, Tanseba *et al.* [20] achieved an accuracy of 84%, precision of 91%, and recall of 80%. However, their dataset is limited to 1000 comments. Similarly, Chaitra [18] used 2,586 comments to analyze opinions toward mobile phone use using hybrid VADER and naïve Bayes, resulting in an accuracy of 79.78% and an F1-score of 83.72%. In our case, we used 1,124,154 tweets with a 70% training set and 30% test set. The hybrid VADER RF–GWO model of these data gave low SDs for accuracy, precision, recall, and F1-score. This result supports the finding of a prior study that large training sets appear to be the most accurate and consistently deliver robust results.

V. DISCUSSIONS AND CONCLUSION

To some extent, past studies lack studies comparing the behaviors and performances of machine learning algorithms using different dataset sizes and hyperparameter tuning methods. This condition is regrettable given the importance of the dataset size on the massive quantity of data, such as social media data. From a theoretical perspective, this study contributes to the existing literature by exploring the role of

dataset sizes and hyperparameter tuning methods on machine learning performances.

In this experiential study, different machine learnings and dataset sizes were compared. The results reveal the non-trivial effects of the dataset size on the performances of classifier models. Regardless of the algorithm, repeat training using different dataset sizes will significantly benefit to gain a better understanding of data and select the trained model. Out of the existing machine learning algorithms, we suggest using RF with a dataset size of 1,124,154 tweets. Moreover, GWO can be used to tune the RF model's parameters, i.e., *mtry*, *sample.fraction*, *min.node.size*, and *num.trees*, which make the model more accurate and robust than the standard RF.

The main outcome of our study is the development of a sentiment classifier that can arbiter the sentiment type of tweets. To translate it into practical implication in the context of our study where cryptocurrency is found to be influenced by social opinions, the reliable classifier can identify the correct patterns to guide investors' decision to buy or sell cryptocurrency, leading to less risk and uncertainty to the fullest extent, along with maximizing returns.

Although our proposed study has given a valuable novel algorithm of sentiment classification, it has some limitations. First, this study only used two optimization methods, GWO, and *tuneRanger*, for hyperparameter tuning. Future studies could compare other methods, such as grid search. Second, this study used only four performance evaluation metrics (accuracy, precision, recall, and F1-score). Hence, there is a need to further extend the use of other metrics that can estimate and compare error rates, such as receiver operating characteristic (ROC) and area under the ROC curve, in the future. Third, the data were only Twitter tweets related to Bitcoin. The past study has found that investors were considering alternative cryptocurrencies, such as Ethereum, Ripple, Litecoin, Stellar, and Dash [54]. Therefore, this sentiment classifier algorithm can be applied with Twitter data related to other cryptocurrencies to explore its robustness.

TABLE 2. Acronyms.

Symbol	Description
ACO	Ant colony optimization
BIC	Bio-inspired computing
BFA	Bacterial foraging algorithm
DT	Decision tree
EA	Evolutionary algorithms
IDF	Inverse document frequency
GA	Genetic algorithm
GWO	Gray wolf optimizer
L2-SVM	L2-loss support vector classification
NB	Naïve Bayes
PSO	Particle swarm optimization
R	A programming language and open-source environment for statistical computing and graphics supported by the R Core Team and R Foundation for Statistical Computing
RF	Random forest
SI	Swarm intelligence
SVM	Support vector machine
Shiny	A web application used to deploy R programs
TF	Term frequency
TF–IDF	Term frequency–inverse document frequency

APPENDIX

See Table 2.

REFERENCES

[1] U. W. Chohan, "Cryptocurrencies: A brief thematic review," IRPN: Innovation Finance, Tech. Rep., 2017. [Online]. Available: https://ssrn.com/abstract=3024330, doi: 10.2139/ssrn.3024330.

[2] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[3] L. Kristoufek, "What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis," *PLoS ONE*, vol. 10, no. 4, Apr. 2015, Art. no. e0123923, doi: 10.1371/journal.pone.0123923.

[4] R. K. Behera, D. Naik, S. K. Rath, and R. Dharavath, "Genetic algorithm-based community detection in large-scale social networks," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9649–9665, Jul. 2020, doi: 10.1007/s00521-019-04487-0.

[5] S. Gupta and B. B. Gupta, "XSS-secure as a service for the platforms of online social network-based multimedia web applications in cloud," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 4829–4861, Feb. 2018, doi: 10.1007/s11042-016-3735-1.

[6] H. Wang, Z. Li, Y. Li, B. B. Gupta, and C. Choi, "Visual saliency guided complex image retrieval," *Pattern Recognit. Lett.*, vol. 130, pp. 64–72, Feb. 2020, doi: 10.1016/j.patrec.2018.08.010.

[7] Z. Zhang, R. Sun, C. Zhao, J. Wang, C. K. Chang, and B. B. Gupta, "CyVOD: A novel Trinity multimedia social network scheme," *Multimedia Tools Appl.*, vol. 76, no. 18, pp. 18513–18529, Sep. 2017, doi: 10.1007/s11042-016-4162-z.

[8] B. Liu, "Text sentiment analysis based on CBOW model and deep learning in big data environment," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 2, pp. 451–458, Feb. 2020, doi: 10.1007/s12652-018-1095-6.

[9] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surveys*, vol. 34, no. 1, pp. 1–47, 2002, doi: 10.1145/505282.505283.

[10] X. Zhu. (2005). *Semi-Supervised Learning Literature Survey Computer Sciences*. University of Wisconsin, Madison. [Online]. Available: http://pages.cs.wisc.edu/ jerryzhu/pub/ssl_survey.pdf

[11] S. Feng and A. Kirkley, "Integrating online and offline data for crisis management: Online geolocalized emotion, policy response, and local mobility during the COVID crisis," *Sci. Rep.*, vol. 11, no. 1, pp. 1–14, Dec. 2021, doi: 10.1038/s41598-021-88010-3.

[12] H. Saif, Y. He, and H. Alani, "Alleviating data sparsity for Twitter sentiment analysis," in *Proc. 2nd Workshop Making Sense Microposts (MSM), Big Things Come Small Packages 21st Int. Conf. TheWorld Wide Web (WWW)*, 2012, pp. 2–9. [Online]. Available: http://oro.open.ac.uk/38501/

[13] N. F. F. D. Silva, L. F. S. Coletta, and E. R. Hruschka, "A survey and comparative study of tweet sentiment analysis via semi-supervised learning," *ACM Comput. Surveys*, vol. 49, no. 1, pp. 1–26, Jul. 2016, doi: 10.1145/2932708.

[14] Y. B. Kim, J. G. Kim, W. Kim, J. H. Im, T. H. Kim, S. J. Kang, and C. H. Kim, "Predicting fluctuations in cryptocurrency transactions based on user comments and replies," *PLoS ONE*, vol. 11, no. 8, Aug. 2016, Art. no. e0161197, doi: 10.1371/journal.pone.0161197.

[15] H. Schoen, D. Gayo-Avello, T. M. Panagiotis, M. Eni, S. Markus, and G. Peter, "The power of prediction with social media," *Internet Res.*, vol. 23, no. 5, pp. 528–543, 2013.

[16] D. Garcia and F. Schweitzer, "Social signals and algorithmic trading of bitcoin," *Roy. Soc. Open Sci.*, vol. 2, no. 9, Sep. 2015, Art. no. 150288.

[17] R. C. Phillips and D. Gorse, "Predicting cryptocurrency price bubbles using social media data and epidemic modelling," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–7.

[18] V. D. Chaithra, "Hybrid approach: Naive Bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 5, p. 4452, Oct. 2019, doi: 10.11591/ijece.v9i5.pp4452-4459.

[19] X. Fan, X. Li, F. Du, X. Li, and M. Wei, "Apply word vectors for sentiment analysis of APP reviews," in *Proc. 3rd Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2016, pp. 1062–1066, doi: 10.1109/ICSAI.2016.7811108.

[20] F. I. Tanesab, I. Sembiring, and H. D. Purnomo, "Sentiment analysis model based on YouTube comment using support vector machine," *Int. J. Comput. Sci. Softw. Eng.*, vol. 6, no. 8, pp. 180–185, 2017.
[21] F. Valencia, A. Gómez-Espinosa, and B. Valdés-Aguirre, "Price movement prediction of cryptocurrencies using sentiment analysis and machine learning," *Entropy*, vol. 21, no. 6, p. 589, Jun. 2019, doi: 10.3390/e21060589.
[22] A. Bermingham and A. F. Smeaton, "Classifying sentiment in microblogs: Is brevity an advantage?" in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2010, pp. 1833–1866.
[23] C. Huang, J. Zhu, Y. Liang, M. Yang, G. P. C. Fung, and J. Luo, "An efficient automatic multiple objectives optimization feature selection strategy for internet text classification," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 5, pp. 1151–1163, May 2019, doi: 10.1007/s13042-018-0793-x.
[24] S. Kübler, C. Liu, and Z. A. Sayyed, "To use or not to use: Feature selection for sentiment analysis of highly imbalanced data," *Natural Lang. Eng.*, vol. 24, no. 1, pp. 3–37, Jan. 2018.
[25] A. Kumar and R. Khorwal, "Firefly algorithm for feature selection in sentiment analysis," in *Computational Intelligence in Data Mining* (Advances in Intelligent Systems and Computing). Singapore: Springer, 2017, pp. 693–703.
[26] A. Tommasel and D. Godoy, "A social-aware online short-text feature selection technique for social media," *Inf. Fusion*, vol. 40, pp. 1–17, Mar. 2018.
[27] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," in *Mining Text Data*, C. Aggarwal and C. Zhai, Eds. Boston, MA, USA: Springer, 2012, doi: 10.1007/978-1-4614-3223-4_6.
[28] D. Buenaño-Fernández, S. Luján-Mora, and W. Villegas-Ch, "Improvement of massive open online courses by text mining of students emails: A case study," in *Proc. 5th Int. Conf. Technol. Ecosyst. Enhancing Multiculturality*, Oct. 2017, pp. 1–7, doi: 10.1145/3144826.3145393.
[29] L. Havrlant and V. Kreinovich, "A simple probabilistic explanation of term frequency-inverse document frequency (TF-IDF) heuristic (and variations motivated by this explanation)," *Int. J. Gen. Syst.*, vol. 46, no. 1, pp. 27–36, Mar. 2017, doi: 10.1080/03081079.2017.1291635.
[30] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Document.*, vol. 60, no. 5, pp. 493–502, 2004.
[31] S. H. Park and K. Han, "Methodologic guide for evaluating clinical performance and effect of artificial intelligence technology for medical diagnosis and prediction," *Radiology*, vol. 286, no. 3, pp. 800–809, 2018, doi: 10.1148/radiol.2017171920.
[32] A. A. Sekh, D. P. Dogra, S. Kar, P. P. Roy, and D. K. Prasad, "ELM-HTM guided bio-inspired unsupervised learning for anomalous trajectory classification," *Cogn. Syst. Res.*, vol. 63, pp. 30–41, Oct. 2020, doi: 10.1016/j.cogsys.2020.04.003.
[33] W. J. Tang and Q. H. Wu, "Biologically inspired optimization: A review," *Trans. Inst. Meas. Control*, vol. 31, no. 6, pp. 495–515, Dec. 2009, doi: 10.1177/0142331208094044.
[34] J. Lu, T. Zhao, and Y. Zhang, "Feature selection based-on genetic algorithm for image annotation," *Knowl.-Based Syst.*, vol. 21, no. 8, pp. 887–891, Dec. 2008, doi: 10.1016/j.knosys.2008.03.051.
[35] K. M. Passino, "Bacterial foraging optimization," *Int. J. Swarm Intell. Res.*, vol. 1, no. 1, pp. 1–16, Jan. 2010, doi: 10.4018/jsir.2010010101.
[36] K. Y. Tai and J. Dhaliwal, "Machine learning model for malaria risk prediction based on mutation location of large-scale genetic variation data," *J. Big Data*, vol. 9, no. 1, pp. 1–22, Dec. 2022, doi: 10.1186/s40537-022-00635-x.
[37] J. Hu, T. Zhou, S. Ma, D. Yang, M. Guo, and P. Huang, "Rock mass classification prediction model using heuristic algorithms and support vector machines: A case study of Chambishi copper mine," *Sci. Rep.*, vol. 12, no. 1, pp. 1–20, Dec. 2022, doi: 10.1038/s41598-022-05027-y.
[38] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[39] Z. Kayhomayoon, F. Babaeian, S. G. Milan, N. A. Azar, and R. Berndtsson, "A combination of metaheuristic optimization algorithms and machine learning methods improves the prediction of groundwater level," *Water*, vol. 14, no. 5, p. 751, Feb. 2022, doi: 10.3390/w14050751.
[40] J. Batra, R. Jain, V. A. Tikkiwal, and A. Chakraborty, "A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, Apr. 2021, Art. no. 100006, doi: 10.1016/j.jjimei.2020.100006.
[41] T. Pano and R. Kashef, "A complete vader-based sentiment analysis of bitcoin (BTC) tweets during the ERA of COVID-19," *Big Data Cogn. Comput.*, vol. 4, no. 4, pp. 1–17, 2020, doi: 10.3390/bdcc4040033.
[42] M. N. Wright and A. Ziegler, "Ranger: A fast implementation of random forests for high dimensional data in C++ and R," *J. Stat. Softw.*, vol. 77, no. 1, pp. 1–17, 2017, doi: 10.18637/jss.v077.i01.
[43] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
[44] M. Skogholt. (2020). *FastNaiveBayes: Extremely fast Implementation of a Naive Bayes Classifier*. [Online]. Available: https://CRAN.R-project.org/package=fastNaiveBayes
[45] M. Kuhn. (2021). *Caret: Classification and Regression Training*. [Online]. Available: https://CRAN.R-project.org/package=caret
[46] E. Costa, A. Lorena, A. Carvalho, and A. Freitas, "A review of performance evaluation measures for hierarchical classifiers," in *Proc. AAAI Workshop*, 2007, pp. 1–6.
[47] P. Probst, A.-L. Boulesteix, and M. Wright, "Hyperparameters and tuning strategies for random forest," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol 9, no. 3, p. e1301, 2018.
[48] B. T. Pham, C. Qi, L. S. Ho, T. Nguyen-Thoi, N. Al-Ansari, M. D. Nguyen, H. D. Nguyen, H.-B. Ly, H. V. Le, and I. Prakash, "A novel hybrid soft computing model using random forest and particle swarm optimization for estimation of undrained shear strength of soil," *Sustainability*, vol. 12, no. 6, p. 2218, Mar. 2020, doi: 10.3390/su12062218.
[49] R. D. Sudhakara, M. D. Reddy, and S. Moupuri, "Network reconfiguration of distribution system for loss reduction using GWO algorithm," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3226–3234, 2017, doi: 10.11591/ijece.v7i6.
[50] N. Rout, D. Mishra, and M. K. Mallick, "Ensemble learning for handling imbalanced datasets with the combination of bagging and sampling methods," *Indian J. Public Health Res. Develop.*, vol. 9, no. 9, p. 1412, 2018, doi: 10.5958/0976-5506.2018.01189.0.
[51] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-sensitive learning vs. Sampling: Which is best for handling unbalanced classes with unequal error costs?" in *Proc. DMIN*, vol. 7, Jun. 2007, pp. 35–41.
[52] R. Couronné, P. Probst, and A.-L. Boulesteix, "Random forest versus logistic regression: A large-scale benchmark experiment," *BMC Bioinf.*, vol. 19, no. 1, pp. 1–14, Dec. 2018, doi: 10.1186/s12859-018-2264-5.
[53] R. G. Mantovani, A. L. D. Rossi, J. Vanschoren, B. Bischl, and A. C. de Carvalho, "Effectiveness of random search in SVM hyper-parameter tuning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8, doi: 10.1109/IJCNN.2015.7280664.
[54] Q. Ji, E. Bouri, C. K. M. Lau, and D. Roubaud, "Dynamic connectedness and integration in cryptocurrency markets," *Int. Rev. Financial Anal.*, vol. 63, pp. 257–272, May 2019, doi: 10.1016/j.irfa.2018.12.002.

ANNY MARDJO received the bachelor's degree in business administration from RMIT University, Melbourne, Australia, and the master's degree in information technology from Swinburne University, Melbourne. She is currently pursuing the Doctoral degree with the College of Digital Science, Prince of Songkla University (Hat Yai Campus), Thailand. She is also working as a Lecturer at the Faculty of Commerce and Management, Prince of Songkla University (Trang Campus). Prior to that, she worked at Deutsche Bank, Singapore, as a Software Engineer, focusing on corporate internet banking. Her research interests include internet-based applications and data analysis.

CHIDCHANOK CHOKSUCHAT (Member, IEEE) received the Ph.D. degree in computer and information science from Silpakorn University. She is currently an Assistant Professor at the Division of Computational Science, Faculty of Science, Prince of Songkla University (Hat Yai Campus), Thailand. She is also with the Office of Digital Innovation and Intelligent Systems, Prince of Songkla University. Her research interests include parallel processing, ontology engineering, web services, linked open data, data science toolkits, and the Internet of Things.