Improved performance of fake account classifiers with percentage overlap features selection

Aris Tjahyanto[1], Rivanda Putra Pratama[1], Ary Mazharuddin Shiddiqi[2]
[1]Department of Information Systems, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2]Department of Informatic, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Aug 10, 2023
Revised Sep 20, 2023
Accepted Sep 29, 2023

Keywords:

Bots
Fake accounts
Fake users
Feature selection
Internet

ABSTRACT

Feature selection plays a crucial role in the development of high-performance classification models. We propose an innovative method for detecting fake accounts. This method leverages the percentage overlap technique to refine feature selection. We introduce our technique upon earlier work that showcased the enhanced efficacy of the Naïve Bayesian classifier through dataset normalization. Our study employs a dataset of account profiles sourced from Twitter, which we normalize using the Min-Max method. We analyze the results through a series of comprehensive experiments involving diverse classification algorithms—such as Naïve Bayes, decision tree, k-nearest neighbors (KNN), deep learning, and support vector machines (SVM). Our experimental results demonstrate a 100% accuracy achieved by the SVM and deep learning classifiers. The results are attributed to the percentage overlap technique, which facilitates the identification of four highly informative features. These findings outperform models with more extensive feature sets, underscoring the efficacy of our approach.

Corresponding Author:

Aris Tjahyanto
Information Systems Department, Institut Teknologi Sepuluh Nopember
Surabaya, East Java, Indonesia
Email: aristj@its.ac.id

1. INTRODUCTION

Indonesia's rise in Internet users signifies a growing era of information openness, particularly through social media platforms. According to a study conducted by HootSuite and we are social between January 2022 and January 2023, there has been a 5.2 percent increase in Internet users in Indonesia compared to the previous year. This situation brings the number of active Internet users to 212.9 million, accounting for 77 percent of the population, with 23 percent still offline at the beginning of the year. Furthermore, the study reveals that 167 million people, approximately 60.4 percent of the total population, are active social media users [1].

However, there is a downside alongside these technological advancements, as the prevalence of fake accounts or bots on social media is rising [2]. These fake accounts undermine the credibility of genuine account owners and pose risks by disseminating fake news, deceptive web ratings, and spam. Such bots are also used to construct misleading opinions or spread false information. As the reliance on internet sources for information grows, including in Indonesia, the impact of fake accounts becomes more significant and concerning. The increasing number of internet users further highlights the urgency of addressing this issue. Unfortunately, these sources are frequently used to disseminate false information, rumors, politically biased comments, and more, potentially harming society [3]. Fake news creators can be categorized into genuine users and bots. Genuine users are real individuals who generate or share fake news. Among these, news readers and active social media users play a significant role in propagating fake news on online social networks, intentionally or unintentionally, as they often pass it on to others. On the other hand, non-human generators of fake news are typically represented by network chatbots and replicators. In this context, bots refer to computer programs specifically designed to mimic human behavior on social networking sites [4]. Furthermore, fraudulent activities associated with fake accounts may include sharing unsafe links, excessive following, creating multiple duplicate accounts, and sending unrelated or irrelevant links [5].

Multiple techniques are employed to detect fake accounts on social media, encompassing filtering methods, rule-based approaches, and the application of machine learning. Among these methods, machine learning has emerged as the most effective in identifying fraudulent accounts. Akyon *et al*. [5] conducted research where they leveraged various features such as the number of followers, the count of accounts followed, and others. They applied a range of classifiers, including Naïve Bayes, support vector machines, logistic regression, neural networks, and deep learning, to analyze and categorize account datasets [6]–[8].

Kupershtein's approach focuses on static and dynamic parameters extracted from social media profiles for fake account identification. Static parameters include relatively unchanging data like profile pictures, names, and dates of birth. In contrast, dynamic parameters can fluctuate over time, such as the number of connections or friends [9]. On the other hand, Velayutham *et al*. considered features like the "friend-to-follow" ratio (reputation) and other attributes associated with fake accounts. They employed the Naïve Bayes algorithm for classification [10]. The effectiveness of classification hinges on several factors, including the choice of classification algorithm, dataset pre-processing techniques, and the selection of relevant features.

Numerous studies have dedicated their efforts to enhancing classification performance by exploring various algorithms and feature extraction or selection techniques. Feature extraction endeavors to derive lower-dimensional subspaces from the original data, whereas feature selection aims to identify an optimal subset of features while preserving the existing ones [11]. There are four primary approaches to feature selection techniques: filter, wrapper, embedded, and ensemble methods. In this context, the InfoGain method operates as a filter, employing information entropy to rank variables and selecting those with an entropy value greater than 0 [12]. Yuanchao conducted a study that delved into data augmentation and normalization effectiveness for classification purposes [13]. On another front, Li *et al*. introduced an innovative approach employing a heterogeneous graph, harnessing text information features from Twitter, and applying a rapid learning method to detect Sybil accounts. This alternative model incorporates textual information alongside structural data [14]. In previous research related to fake account classification, Akyon *et al*. obtained an F1-Score value of 86% using SVM and neural networks [5], while Pratama *et al*. obtained an F1-Score value of 77% which also used SVM as a classifier [15].

In this paper, we employ the InfoGain technique for feature selection and compare its performance with a new approach called percentage overlap. We used four of the common metrics for classification performance evaluation: accuracy, precision, recall, and F1-Score. Subsequently, we evaluate the effectiveness of various classification algorithms, including Naïve Bayes (NB), logistic regression (LR), decision tree (DT), k-nearest neighbors (KNN), support vector machines (SVM), and deep learning (DL), in the context of fake account identification. Section 2 outlines our methodology, while Section 3 discusses the results and performance of the classification. Section 4 provides the concluding remarks, followed by the list of references.

2. METHOD

We conducted a comprehensive study encompassing data collection, data pre-processing, feature selection, and the application of classification processes to assess the performance of our proposed method. Our investigation commenced with an in-depth comparison of fake account classification algorithms. This comparative analysis involved classifying fake accounts through three distinct scenarios. Firstly, we employed the Naïve Bayes (NB) algorithm. Subsequently, we utilized the support vector machine (SVM) algorithm and applied the deep learning algorithm. To facilitate this analysis, we partitioned the account profile dataset into two sections: training and testing datasets. This division used a 75:25 ratio, ensuring a robust evaluation framework.

2.1. Features extraction

We initiated the data collection process by employing a web crawling technique to retrieve the account profile dataset directly from the Twitter platform. This task was efficiently carried out using a Python program package known as Twint. The information gathered encompassed essential details such as usernames, follower counts, following counts, post counts, biographies, profile pictures, and verification status [15]. Following the data collection phase, our next critical step involved dataset refinement, specifically labeling accounts as authentic or fake. The dataset comprises nine distinct features, as outlined in Table 1. Prior to normalization, it is noteworthy that each feature exhibited a unique range of values. These value ranges varied from tens to hundreds, thousands, or even millions. The disparities in these value ranges can significantly impact the dataset's positioning and, consequently, could influence the classifier's performance.

Table 1. Features for identifying fake accounts and their descriptions

No	Features	Descriptions
1	Profile Tweets	Number of postings
2	Profile Follower	Number of followers
3	Profile Following	Number of followings
4	Reputation	Reputation of the account, calculated using Number of followers divided by (Number of followers + Number of followings)
5	Char Username	Length of user names
6	Char Number	Number of digits in user names
7	Char Bio	Number of characters in bio
8	Profile Picture	Absence of a profile picture
9	Verified	Account verification status

We collected the data twitter using crawling techniques on the account of the Health Ministry Republic of Indonesia, the COVID-19 Task Force, Provincial Health Service East Java, and the Surabaya City Health Service. This process was done using the mention feature in Twitter. After collecting profile data for the account, the next step is giving a label for each account profile data. This process is carried out by labeling the data Twitter account with information about whether an account is authentic or fake. This labeling process is carried out manually and assisted by a tool called Tweetbotornot to avoid bias. Tweetbotornot works by entering an ID number or username; then the program will give a probability score whether the account is a social bot or not. The score from this tool starts from zero 0 to 1. The higher the score, the account is considered a social bot. There are two modes of choice when using this program: default and fast. The default mode analyzes parameters from account information and tweets from an account. Meanwhile, the fast mode takes parameters from account information only [15]. Finally, we collected 1097 authentic and 903 fake accounts, as shown in Figure 1.

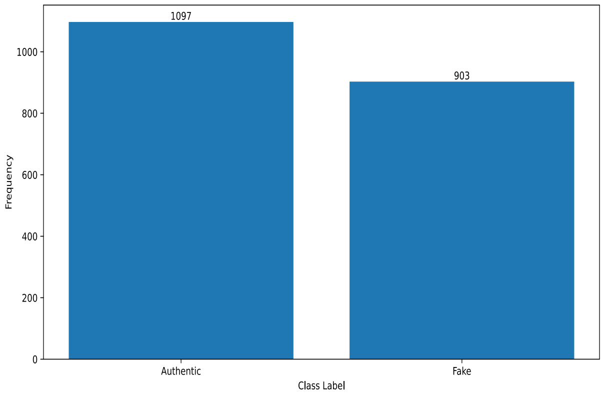


Figure 1. Distribution of fake and authentic accounts

Before carrying out the training and testing process, we normalize the dataset using the Min-Max technique to perform linear transformations of the original data to balance values, as shown in (1) [16], [17]. The results of normalization yield a value range between 0 and 1.

$$X' = \frac{X - \min\{X_j\}}{\max\{X_j\} - \min\{X_j\}} \quad (1)$$

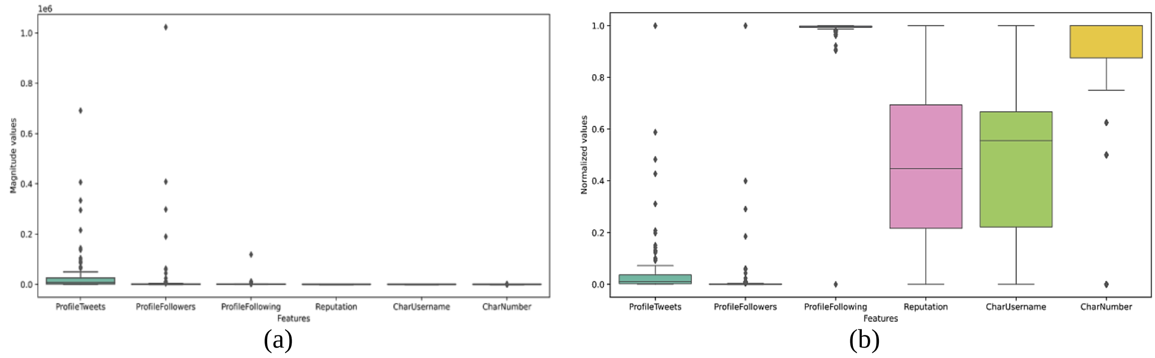
Let X represent the current value of the data and X' be the result of normalizing the data. The maximum value of {Xj} corresponds to the highest value found in feature or data column j, while {Xj} 's minimum value corresponds to that column's lowest value. After the data labeling process, the feature scaling or data normalization process aims to standardize numeric column values to the same scale without altering the range of values. This is conducted to mitigate the dominant influence of varying units among feature variables, which could lead to biased results, as depicted in Figure 2. Before normalization, the first two features, ProfileTweets and ProfileFollowers, appear to dominate the other features, as shown in Figure 2(a).

Normalization is applied selectively, primarily when substantial disparities in feature ranges or units exist within the dataset. Once normalization is executed, feature values are confined within the standardized range of 0 to 1. To illustrate, after normalization, features like ProfileFollowing, Reputation, CharUserName, and CharNumber exhibit more pronounced proximity to 1, as visually represented in Figure 2(b). This standardized scale effectively mitigates the influence of varying data ranges or units, consequently enhancing the overall performance of the dataset. Following the normalization procedure, the subsequent step involves a visual assessment of the effectiveness of these features in differentiating between fake and authentic accounts. This evaluation is achieved by introducing a variable that identifies shared elements between the two groups. A smaller proportion of shared elements signifies a more distinctive variable for class differentiation. Thus, a reduced overlap between elements enhances the suitability of the variable as a feature for classification.

We utilize a boxplot graph, as depicted in Figure 3, to visualize the overlapping elements and categorize features based on classes or categories. In Figure 3(a), it becomes apparent that fake accounts tend to have fewer followers than authentic accounts. The number of followers clusters closer to zero for fake accounts, while for authentic accounts, this metric exhibits a more dispersed distribution. A parallel trend is observed in Figure 3(b) concerning the number of posts. Here, authentic accounts display a broader spectrum of post numbers, distinguishing them from fake ones.

$$Reputation = \frac{Follower}{Follower+Following} \tag{2}$$

Furthermore, in Figure 3(c), we present the reputation score, which is calculated using in (2). The utilization of boxplots to visualize these features offers invaluable insights into their capacity to effectively discriminate between the two account classes, facilitating the selection of discriminative features for classification purposes. Conversely, as shown in Figure 3(d), the CharNumber feature exhibits limited effectiveness in distinguishing between the two account classes. Figure 3 provides compelling evidence that fake accounts' average reputation scores significantly lag behind authentic accounts' average reputation scores. Consequently, these features can be leveraged to ascertain whether an account is genuine or fraudulent.



(a) (b)

Figure 2. Feature scaling or data normalization: (a) before normalization and (b) after normalization

2.2. Features selection

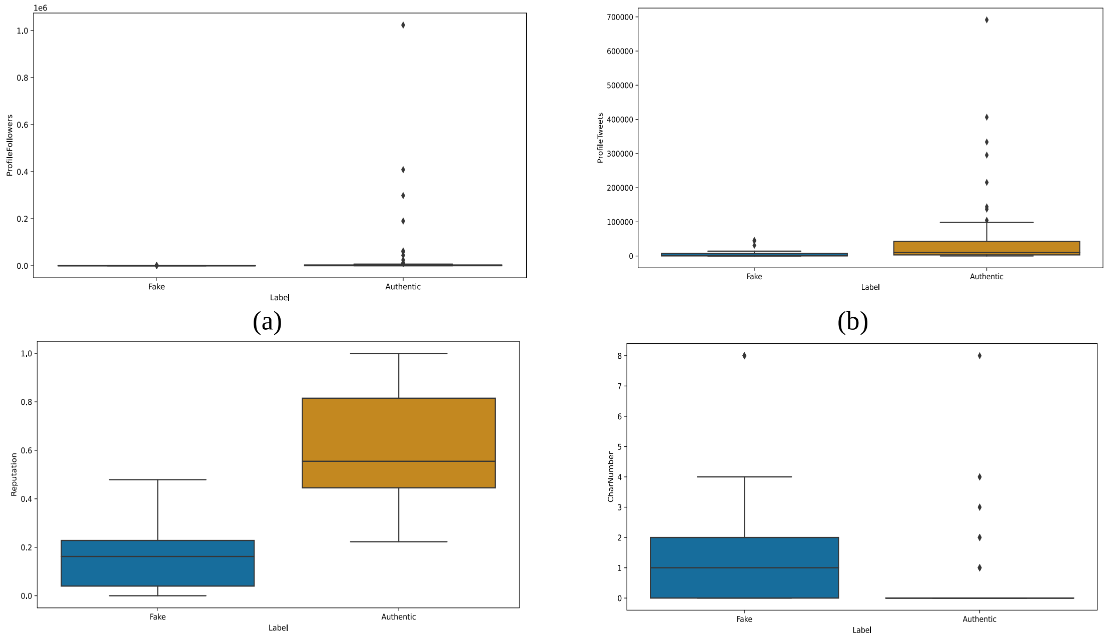
The feature selection process holds significant importance in classification tasks, serving as a pivotal step toward achieving optimal outcomes. Its primary aim is enhancing classifier performance, mainly when dealing with a limited number of available features, necessitating identifying and eliminating irrelevant ones. Focusing on the pertinent features empowers us to make targeted efforts to elevate overall classification performance. We draw upon the fake account features previously employed in [5], which exhibited exceptional efficacy in classifying fake accounts. Table 1 overviews the nine features harnessed for identifying fake accounts.

To gain deeper insights into these features' characteristics, we employ a ranking technique founded on information gain (IG), a widely recognized algorithm in machine learning research for generating feature rankings. IG quantifies feature rankings based on the system's entropy [18], [19]. Additionally, we introduce a quantitative measure known as the percentage overlap (PO) to evaluate a feature's capacity to distinguish between different classes. Denoted as PO (x, y), this metric represents the sum of shared proportions between two classes or categories, as outlined in (3). By combining the IG-based ranking approach with the PO metric, our goal is to select features that significantly contribute to the discrimination between fake accounts and authentic ones. This feature selection strategy provides valuable insights and elevates the classification model's accuracy and performance.

$$PO(x,y) = \frac{2*(\max\{0,\min\{\max\{x\},\max\{y\}-\max\{\min\{x\},\min\{y\}\}\})}{\max\{x\}-\min\{x\}+\max\{y\}-\min\{y\}} \quad (3)$$

Where *x* contains a subset of the values of the individual features or data column that have the class label c_k, dan *y* contains values that are a subset of individual features with a non-class label c_k. The methodology for calculating the percentage overlap for all features or data columns is depicted in Figure 4. We first extract the corresponding values of *x* and *y* to compute the percentage overlap for each feature. Subsequently, the specific percentage overlap value is computed for each feature individually. This sequential procedure is then repeated for all the features or data columns within the dataset. Once the entirety of the percentage overlap (PO) values has been computed, they are utilized to identify the candidate features that will be employed in the subsequent classification process.

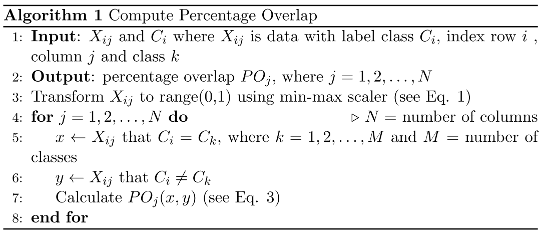
We use percentage overlap to quantify the discrimination of the classes or categories by the score of *PO (x, y)*. The smaller the PO score, the greater the ability of the variable to differentiate between the classes. For example, the follower profile is a variable related to how many followers an account has. The PO value of the ProfileFollower feature is close to zero (Figure 5). For those variables, the fake authentic classes have a minimal percentage overlap. Thus, this variable is expected to be one of the candidate variables that can distinguish between the two classes.



(a) (b)

(c) (d)

Figure 3. Comparison of the overlapping elements of the four features: (a) profile follower, (b) profile tweets, (c) reputation, and (d) char number features



```
Algorithm 1 Compute Percentage Overlap
Input: $X_{ij}$ and $C_i$ where $X_{ij}$ is data with label class $C_i$, index row $i$ , column $j$ and class $k$
Output: percentage overlap $PO_j$, where $j = 1, 2, \ldots, N$
Transform $X_{ij}$ to range(0,1) using min-max scaler (see Eq. 1)
for $j = 1, 2, \ldots, N$ do                    ▷ $N$ = number of columns
    $x \leftarrow X_{ij}$ that $C_i = C_k$, where $k = 1, 2, \ldots, M$ and $M$ = number of classes
    $y \leftarrow X_{ij}$ that $C_i \neq C_k$
    Calculate $PO_j(x, y)$ (see Eq. 3)
end for
```

Figure 4. Computing the percentage overlap for data X_{ij} and class label C_j

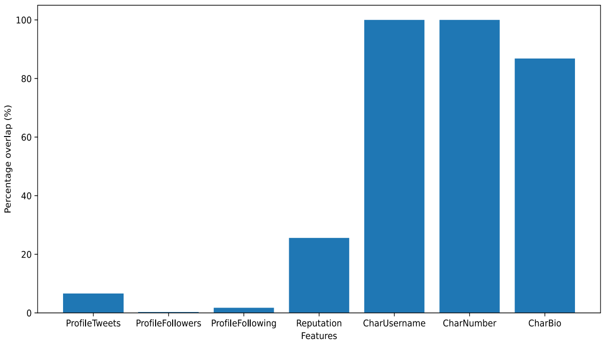


Figure 5. Percentage overlap between fake and authentic for several variables

2.3. Classification

According to Akyon and Kalfaoglu, Twitter harbors two types of fake social engagement: fake accounts and bot accounts [5]. Fake accounts encompass profiles managed by humans and social media accounts operated under specific instructions. In contrast, bot accounts are entirely automated and controlled by machine or computer programs, executing tasks according to predefined instructions. Both fake accounts and bots exhibit similar characteristics, such as a limited number of posts and a smaller follower count than authentic accounts.

Our approach to identifying fake accounts involved the application of various machine-learning techniques. We employed supervised machine learning algorithms that utilized multiple datasets containing features and corresponding labels to classify fake accounts. These features were employed as inputs for the model learning process and were derived from account attributes, including metrics like the number of followers, posts, and more. To assess the effectiveness of our classification algorithm, we considered key performance metrics, including precision, Recall, F1-Score, Accuracy, and ROC/AUC. Accuracy provides insight into the proportion of correct classifications achieved by the classifier. Precision measures the number of true positive answers the classifier provides, with higher precision values indicating fewer misclassifications. However, precision alone does not evaluate whether the classifier successfully captures all correct answers, which is where the Recall value becomes crucial. Recall assesses the classifier's ability to identify potential positive answers as expected correctly.

The higher the Precision and Recall values, the better the classification performance is. However, achieving exceedingly high Precision and Recall simultaneously is often challenging in practical scenarios. Hence, it becomes crucial to strike a balance between these two metrics. The F1-Score, a composite measurement, provides an average of both Precision and Recall values [20]. In addition to the previously mentioned performance metrics, this research paper incorporates ROC/AUC. Receiver operating characteristics (ROC) serves as an alternative performance evaluation tool for classification problems, with AUC representing the area under the ROC curve [21], [22]. Furthermore, selecting the classifier model with the most significant AUC is imperative, as it signifies a higher true positive rate or a lower false positive rate.

The classification process encompasses the utilization of several algorithms, including Naïve Bayes (NB), logistic regression (LR), decision tree (DT), k-nearest neighbors (KNN), support vector machines (SVM), and deep learning (DL). Initially, training is conducted to generate the classifier model for NB, LR, DT, KNN, and SVM algorithms. Subsequently, testing is executed, and essential performance indicators such as Precision, Recall, Accuracy, F1-Score, and ROC/AUC are recorded. It's worth noting that for the Decision Tree (DT) algorithm, as the initial determination of weight values is system-generated, running it multiple times may yield varying models. To mitigate this variability, the training process is repeated several times, and the model with the highest accuracy is selected and retained. This chosen model is then employed for the subsequent testing phase.

3. RESULTS AND DISCUSSION

The experiment results revealed the top five features based on information gain (IG) with a threshold of IG > 0, namely Reputation, ProfileFollowers, ProfilesTweets, CharBio, and CharNumber. Interestingly, the IG results did not deviate significantly from the four features with a minor PO score: ProfileTweets, ProfileFollowers, ProfileFollowing, and Reputation. We evaluated three feature sets to enable a comprehensive comparison (as presented in Table 2). The first group incorporated all available features, offering a complete dataset for analysis. The second group narrowed the feature selection to five key attributes based on the ranking of information gain (IG). Finally, the last group employed a subset of four features, selected based on their minor percentage overlap (PO) scores.

Table 2. Three dataset groups with different number of features

Number of Features	Features	Features Selection
9	ProfileTweets, ProfileFollower, ProfileFollowing, Reputation, CharUsername, CharNumber, CharBio, ProfilePicture, Verified	All features
5	Reputation, ProfileFollowers, Profiles Tweets, CharBio, Char Number	Info Gain
4	ProfileTweets, ProfileFollowers, ProfileFollowing, Reputation	Percentage overlap

In the classification process targeting fake accounts, we employed a machine learning approach that encompassed various algorithms, including Naïve Bayes (NB), logistic regression (LR), decision tree (DT), k-nearest neighbors (KNN), support vector machines (SVM), and deep learning (DL). We considered two versions of the support vector machines, denoted as SVM1 with parameters C = 1 and gamma = 0.1 and SVM2 with parameters C = 100 and gamma = 0.1. The selection of the first parameter combination was arbitrary, while the second was determined optimally through a grid search process. Specific hyperparameters were fine-tuned for each classifier via grid search, as detailed in Table 3.

Table 3. Hyperparameters for tuning the classifiers

Classifier	Parameters	Values	Description
NB	var_smoothing	[1e-8, 1e-9, 1e-10]	Prevent division by zero and avoid undefined results in probability calculations
LR	solver	['newton-cg', 'lbfgs', 'liblinear']	Affect the training speed and convergence
	penalty	['l2']	Ridge regularization
	C	[100, 10, 1.0, 0.1]	Regularization strength to prevent overfitting
DT	max_depth	[1,3,5,7,9]	Prevent overfitting
	min_samples_leaf	[1,2,3,4,5]	Preventing unnecessary splits
	criterion	["gini", "entropy"]	Selects the best feature to split on
	max_leaf_nodes	[None,10,20,30,40,50]	Control the size of the tree
KNN	n_neighbors	[1,3,7,9,11,13]	Impact the model's performance
	weights	['uniform', 'distance']	The weight of neighbors' contributions
	metric	['euclidean', 'manhattan', 'minkowski']	The distance metric between data points.
DL	model__neurons	[6, 9, 13, 17]	The number of neurons in each hidden layer
	optimizer__learning_rate	[0.0001, 0.001, 0.01]	Affects the speed of convergence and weights
	kernel_initializer	['uniform', 'lecun_uniform', 'normal']	Helps the model converge faster
	random_state	[1,2,5,9,42]	Sets the random seed for reproducibility
SVM	C	[0.1, 1, 10, 100, 1000]	Trade-off between maximizing margin and minimizing the classification error
	gamma	[1, 0.1, 0.01, 0.001, 0.0001]	Controls the shape of the decision boundary

The deep learning (DL) approach involved utilizing a multilayer perceptron (MLP) architecture featuring nine neurons in the input layer, two hidden layers comprising six neurons, and two in the output layer. The activation functions employed were 'Relu' for the hidden layers and 'Softmax' for the output layer [23], [24]. In partitioning the fake accounts for training and testing purposes, we maintained a consistent 75:25 ratio in all our experiments. The experiments offered valuable insights by examining the performance of all classifiers across different feature groups. Additionally, we conducted specific experiments with varying comparison ratios, particularly for SVM2, to gauge the robustness of each feature group under different dataset sizes. These analyses collectively contribute to a comprehensive understanding of the classifiers' performance and shed light on the influence of feature selection in effectively detecting fake accounts.

Our comprehensive examination revolved around the classification process employing three distinct feature groups: one comprising all nine features, another with five selected features, and a third with four features chosen based on their PO scores. The primary objective was to discern the impact of varying the number of features and identify the specific attributes that significantly influenced the classification performance outcomes. To evaluate the accuracy of each test, we employed a range of metrics, including accuracy, precision, recall, and F1 score, with the confusion matrix serving as a crucial component in these calculations. Based the results obtained in the preceding experiment, we determined the most effective classification algorithm and analyzed the effect of the data normalization process.

The account profile dataset underwent a two-step process involving labeling and normalization, where we employed the Min-Max method. Subsequently, we divided the dataset into two segments for training and testing. The classification process entailed the utilization of Naïve Bayes (NB), logistic regression (LR), decision tree (DT), k-nearest neighbors (KNN), support vector machines (SVM), and deep learning (DL) algorithms. Each classifier was applied to the three dataset groups comprising 9, 5, and 4 features, as detailed in Table 2. Following this, meticulous calculations were performed to determine the accuracy, precision, recall, and F1-score values presented in Table 4.

Table 4. Accuracy of several classifiers for three dataset groups with different number of features

Classifier	Accuracy			Precision			Recall			F1-Score		
	9	5	4	9	5	4	9	5	4	9	5	4
NB	0.67	0.70	0.67	0.74	0.75	0.74	0.76	0.79	0.76	0.67	0.70	0.67
LR	0.93	0.93	1.00	0.96	0.96	1.00	0.89	0.89	1.00	0.91	0.91	1.00
DT	0.87	0.90	0.96	0.84	0.88	0.98	0.87	0.90	0.94	0.85	0.88	0.96
KNN	0.93	1.00	0.97	0.91	1.00	0.98	0.95	1.00	0.94	0.93	1.00	0.96
DL	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SVM1	0.87	0.83	0.70	0.92	0.90	0.35	0.78	0.72	0.50	0.81	0.75	0.41
SVM2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

The performance evaluation unveiled that the DL and SVM2 classifiers consistently delivered outstanding results across all three feature groups, as depicted in Table 4. The SVM2 and DL provide an F1-Score of 100%, and this outperform previous research conducted by Akyon *et al*. by 86% [5] and Pratama and Tjahyanto which got an F1-Score of 77% [15]. Notably, the dataset with four features, selected using the percentage overlap method, outperformed both the 9 and 5 feature groups for most classifiers. These outcomes underscore the pivotal role of feature selection in enhancing classifier performance. Thoughtful choices of features can significantly boost classification accuracy for all classifiers. However, it's worth noting that SVM1, when employing default hyperparameter values, struggled to effectively differentiate between fake and authentic accounts, resulting in a significant decline in classification performance in this particular scenario. Conversely, the DL and SVM2 classifiers, with hyperparameters optimized through the grid-search process, consistently outperformed others across all evaluation metrics.

Notably, the performance trends for the logistic regression (LR) and decision tree (DT) classifiers exhibited contrasting patterns: a reduction in the number of features used correlated with increased performance. These observations provide valuable insights into the impact of feature selection and classifier hyperparameters on classification performance, contributing to the refinement of fake account detection methods. This understanding contributes significantly to refining fake account detection methods. However, our experiments with Naïve Bayes (NB) yielded less satisfactory results across all three feature groups, with accuracy, precision, recall, and F1 scores ranging from 0.67 to 0.79. This outcome suggests that the dataset size employed in these experiments may be inadequate and requires further enhancement, such as exploring the potential correlations between features. This limitation arises because Naïve Bayes assumes that the presence of one feature is entirely independent of the presence of any other feature [25].

Table 5 provides an overview of the accuracy achieved by support vector machines with optimal hyperparameters (SVM2) for the dataset groups featuring 9, 5, and 4 features while considering various test data sizes. A test data size of 0.1 or 10% indicates that 90% of the data was allocated for training, leaving 10% for testing. The results indicate that all three dataset groups performed well when provided with more training data than testing data. However, their performance exhibited a noticeable decline when a smaller portion of the dataset was allocated for training. The dataset group comprising four features (ProfileTweets, ProfileFollowers, ProfileFollowing, and Reputation) consistently outperformed the other dataset groups, showcasing superior accuracy levels across various test data sizes.

Table 5. Accuracy of SVM2 for the three datasets

Data Training Size (%)	Accuracy		
	9 features	5 features	4 features
0.10	1.00	1.00	1.00
0.15	1.00	1.00	1.00
0.20	1.00	1.00	1.00
0.25	1.00	1.00	1.00
0.30	0.97	0.97	0.97
0.40	0.96	0.96	0.96
0.50	0.97	0.97	0.97
0.60	0.89	0.93	0.94
0.70	0.89	0.93	0.93
0.80	0.92	0.89	0.92
0.90	0.80	0.89	0.93

Figure 6 represents the ROC and AUC curves for all classifiers utilizing the complete set of available features. These ROC and AUC values are critical for evaluating the classifiers' discriminatory ability. An AUC value of 1.0 signifies flawless differentiation between fake and authentic accounts, while a value of approximately 0.5 indicates an inability to distinguish between the two classes. As discerned from the figure, it becomes evident that the most potent classifier distinguishing between fake and authentic accounts is Deep Learning (DL), boasting an impressive AUC score of 0.96. Conversely, the classifier with the least discriminatory capability is the decision tree (DT), as indicated by its AUC value.

Figure 7 shows the ROC and AUC curves obtained by the SVM2 classifier using four features. Notably, the figure illustrates that SVM2 achieves an impressive AUC score of 0.98, signifying that the dataset containing four features outperforms both the nine and five feature datasets. Moreover, with an AUC of 0.98, it unequivocally confirms the robust discriminatory capability of the classification model based on SVM2 when employing the four selected features. Based on the experimental results across various classifiers and dataset groups, it becomes evident that the dataset containing four features, selected using the percentage overlap (PO) method, consistently delivers exceptional results, outperforming both the nine and five-feature datasets. Generally, the dataset comprising four features exhibits higher accuracy and AUC scores than those containing nine or five features.

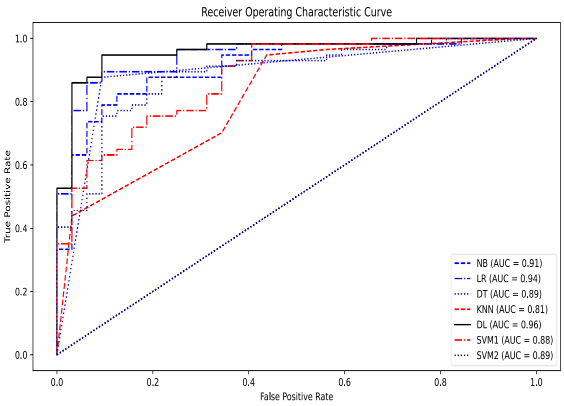


Figure 6. ROC curve using all features

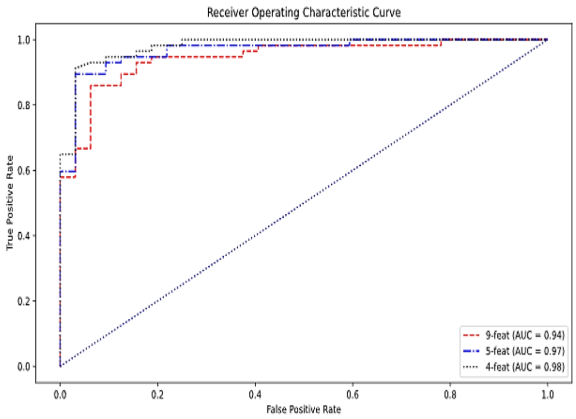


Figure 7. ROC curves for 9, 5, and 4 features of the SVM2 classifier

Figure 8 provides a visual representation of the accuracy achieved by the SVM2 classifier for different test data sizes, considering varying numbers of features. The features are organized and sorted based on their respective percentage overlap (PO) values. Accuracies remain relatively stable at around 65% for datasets featuring three or fewer features. However, as the number of features increases to four, there is an observable upward trend in accuracy. Notably, employing test data sizes less than or equal to 25% consistently yields stable accuracies, hovering around 100% for all the features utilized. Nevertheless, it is important to acknowledge that the accuracy tends to decline as the number of features employed increases.

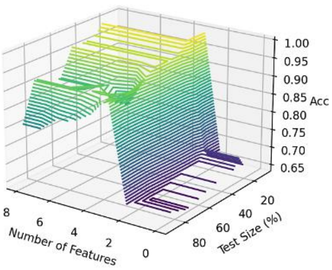


Figure 8. Accuracy 3D illustration of the SVM2 classifier for several features and test data sizes

The percentage overlap technique has shown superior performance in feature selection compared to InfoGain. However, this technique requires manual determination of the number of features to be selected, necessitating further investigation to explore alternative methods for more precise feature limitation. Furthermore, it is essential to acknowledge that the experiments were restricted to binary classification and focused on a single dataset. This limitation stems from the specific nature of the problem, which revolves around distinguishing between fake and authentic Twitter accounts based on extracted features. Consequently, it becomes imperative to explore the reliability and adaptability of the percentage overlap technique for other datasets, including multiclass classification scenarios. Extending the analysis to encompass diverse datasets and multiclass problems would enhance the generalizability and applicability of the feature selection approach, thereby enriching the understanding of its effectiveness across various contexts.

4. CONCLUSION

This research investigates the efficacy of the percentage overlap technique for sorting and selecting crucial features, determining the minimum number required for classifying fake accounts. The investigation revealed that the selected features, namely ProfileTweets, ProfileFollowers, ProfileFollowing, and Reputation, outperformed other feature selection methods, including those with five and nine features. Among the experimented classifiers, the deep learning (DL) and support vector machines (SVM) classifiers demonstrated remarkable performance, achieving 100% accuracy and 100% F1-Score. This result indicates their high precision and effectiveness in distinguishing between fake and authentic accounts compared to other classifiers. We intend to extend our research by applying the percentage overlap technique to binary or multiclass classification scenarios and exploring its applicability to general classification problems.

ACKNOWLEDGEMENTS

This study was supported in part by the TA-upgrading 2023 and Research Grant (Rector Decree number 1039/IT2/T/HK.00.01/2023).

REFERENCES

[1] S. Chimphlee and W. Chimphlee, “Machine learning to improve the performance of anomaly-based network intrusion detection in big data,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 2, p. 1106, May 2023, doi: 10.11591/ijeecs.v30.i2.pp1106-1119.

[2] S. R. Ramya, R. Priyanka, S. S. Priya, M. Srinivashini, and A. Yasodha, “SVM Based Fake Account Sign-In Detection,” in *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2023, pp. 509–514. doi: 10.1109/ICOEI56765.2023.10125850.

[3] M. R. Kondamudi, S. R. Sahoo, L. Chouhan, and N. Yadav, “A comprehensive survey of fake news in social networks: Attributes, features, and detection approaches,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 6, p. 101571, Jun. 2023, doi: 10.1016/j.jksuci.2023.101571.

[4] X. Zhang and A. A. Ghorbani, “An overview of online fake news: Characterization, detection, and discussion,” *Information Processing & Management*, vol. 57, no. 2, p. 102025, Mar. 2020, doi: 10.1016/j.ipm.2019.03.004.

[5] F. C. Akyon and M. Esat Kalfaoglu, “Instagram Fake and Automated Account Detection,” in *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2019, pp. 1–7. doi: 10.1109/ASYU48272.2019.8946437.

[6] S. Khaled, N. El-Tazi, and H. M. O. Mokhtar, “Detecting Fake Accounts on Social Media,” in *2018 IEEE International Conference on Big Data (Big Data)*, Dec. 2018, pp. 3672–3681. doi: 10.1109/BigData.2018.8621913.

[7] A. Jaiswal, H. Verma, and N. Sachdeva, “Swarm optimized Fake News Detection on Social-media textual content using Deep Learning,” in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, May 2023, pp. 1–8. doi: 10.1109/ACCAI58221.2023.10201229.

[8] A. P. Rodrigues *et al.*, “Real-Time Twitter Spam Detection and Sentiment Analysis using Machine Learning and Deep Learning Techniques,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–14, Apr. 2022, doi: 10.1155/2022/5211949.

[9] O. Voitovych, L. Leonid Kupershtein, L. Kupershtein, and V. Holovenko, “DETECTION OF FAKE ACCOUNTS IN SOCIAL MEDIA,” *Cybersecurity: Education, Science, Technique*, vol. 2, no. 18, pp. 86–98, 2022, doi: 10.28925/2663-4023.2022.18.8698.

[10] T. Velayutham and P. K. Tiwari, “Bot identification: Helping analysts for right data in twitter,” in *2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall)*, Sep. 2017, pp. 1–5. doi: 10.1109/ICACCAF.2017.8344722.

[11] S. Karami, F. Saberi-Movahed, P. Tiwari, P. Marttinen, and S. Vahdati, “Unsupervised feature selection based on variance–covariance subspace distance,” *Neural Networks*, vol. 166, pp. 188–203, Sep. 2023, doi: 10.1016/j.neunet.2023.06.018.

[12] Y. Chen *et al.*, “Comparison of feature selection methods for mapping soil organic matter in subtropical restored forests,” *Ecological Indicators*, vol. 135, p. 108545, Feb. 2022, doi: 10.1016/j.ecolind.2022.108545.

[13] X. Yuanchao, C. Zhiming, and K. Xiaopeng, “Improved pitch shifting data augmentation for ship-radiated noise classification,” *Applied Acoustics*, vol. 211, p. 109468, Aug. 2023, doi: 10.1016/j.apacoust.2023.109468.

[14] S. Li, J. Yang, G. Liang, T. Li, and K. Zhao, “SybilFlyover: Heterogeneous graph-based fake account detection model on social networks,” *Knowledge-Based Systems*, vol. 258, p. 110038, Dec. 2022, doi: 10.1016/j.knosys.2022.110038.

[15] R. P. Pratama and A. Tjahyanto, “The influence of fake accounts on sentiment analysis related to COVID-19 in Indonesia,” *Procedia Computer Science*, vol. 197, pp. 143–150, 2022, doi: 10.1016/j.procs.2021.12.128.

[16] N. A. Husin, N. Salim, and A. R. Ahmad, “Modeling of dengue outbreak prediction in Malaysia: A comparison of Neural Network and Nonlinear Regression Model,” in *2008 International Symposium on Information Technology*, Aug. 2008, pp. 1–4. doi: 10.1109/ITSIM.2008.4632022.

[17] Z. Mustaffa and Y. Yusof, “A comparison of normalization techniques in predicting dengue outbreak,” *International Conference on Business and Economics Research*, vol. 1, pp. 345–349, 2011, [Online]. Available: http://www.ipedr.com/vol1/74-G10007.pdf

[18] M. T. Martín-Valdivia, M. C. Díaz-Galiano, A. Montejo-Raez, and L. A. Ureña-López, “Using information gain to improve multi-modal information retrieval systems,” *Information Processing & Management*, vol. 44, no. 3, pp. 1146–1158, May 2008, doi: 10.1016/j.ipm.2007.09.014.

[19] A. Tjahyanto, Y. K. Suprapto, and D. P. Wulandari, “Spectral-based Features Ranking for Gamelan Instruments Identification using Filter Techniques,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 11, no. 1, p. 95, Mar. 2013, doi: 10.12928/telkomnika.v11i1.895.

[20] D. M. W. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” *Mach. Learn. Technol*, no. 2, 2008, [Online]. Available: http://arxiv.org/abs/2010.16061

[21] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” in *Proceedings of the 23rd international conference on Machine learning - ICML ’06*, 2006, pp. 233–240. doi: 10.1145/1143844.1143874.

[22] Z. DeVries *et al*., “Using a national surgical database to predict complications following posterior lumbar surgery and comparing the area under the curve and F1-score for the assessment of prognostic capability,” *The Spine Journal*, vol. 21, no. 7, pp. 1135–1142, Jul. 2021, doi: 10.1016/j.spinee.2021.02.007.

[23] Q. Gong, W. Kang, and F. Fahroo, “Approximation of compositional functions with ReLU neural networks,” *Systems & Control Letters*, vol. 175, p. 105508, May 2023, doi: 10.1016/j.sysconle.2023.105508.

[24] X. Liu, D. Wang, and S.-B. Lin, “Construction of Deep ReLU Nets for Spatially Sparse Learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7746–7760, Oct. 2023, doi: 10.1109/TNNLS.2022.3146062.

[25] U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. García-Nieto, “Efficient Water Quality Prediction Using Supervised Machine Learning,” *Water*, vol. 11, no. 11, p. 2210, Oct. 2019, doi: 10.3390/w11112210.

BIOGRAPHIES OF AUTHORS

Aris Tjahyanto received his Bachelor's degree from the Department of Electrical Engineering, Sepuluh Nopember Institute of Technology (ITS) in 1989 and completed his Master's in Computer Studies (M.Kom) at the Faculty of Computer Science, University of Indonesia (UI) in 1995. Then, in 2015, I completed the Doctoral program in Electrical Engineering ITS with a research topic related to Artificial Intelligence for sound recognition. Starting in 1991, he worked at ITS as a lecturer, currently focusing on research related to artificial intelligence at the Data Acquisition and Information Dissemination Laboratory (ADDI) of the ITS Information Systems Department. Teaching courses including Business Analytics and System Development and Implementation. He can be contacted at email: aristj@its.ac.id.

Rivanda Putra Pratama has been working in the software development industry since 2016. Rivanda began their career as an Intern at Universitas Airlangga in 2016. He joined Indadi Group as an IT Programmer and then moved to Vascomm as a Core Developer in 2019. He has worked at Komunal Indonesia as a Back End Developer since 2021. Rivanda Putra Pratama obtained a Bachelor's in Information Systems from Universitas Airlangga in 2017. Rivanda then completed a Masters in Information Systems at Institut Teknologi Sepuluh Nopember (ITS) in 2021. Email: rivanda.19052@mhs.its.ac.id.

Ary Mazharuddin Shiddiqi received a Ph.D. degree in Informatics from the Institut Teknologi Sepuluh Nopember, Indonesia, in 2004, a master's degree in Information Technology from Monash University, Australia, in 2009, and the Ph.D. degree in computer science and software engineering from the University of Western Australia, Australia, in 2019. He is an Associate Professor at the Institut Teknologi Sepuluh Nopember, Indonesia. His research interests include data stream mining, wireless sensor networks, graph theory and applications, and leak detection systems. He can be contacted at email: ary.shiddiqi@if.its.ac.id.