Identification of Spam on Social Media by Semi-supervised Learning Approach

Sudana Shashi Kiran, Sure Uday Kiran, Elluru Amrutha, E. Yasaswi, Annam Kumar Sai, Kusam Sravya, and S. R. Reeja(✉)

School of Computer Science and Engineering, VIT-AP University, Amaravati, India
{shashi.21bce8644,udaykiran.21bce9527,amrutha.21bce8405, yasaswi.21bce7221,kumarsai.21bce9432, sravya.21bce8209}@vitapstudent.ac.in, reeja.sr@vitap.ac.in

Abstract. This study examines how machine learning methods can be used to identify Twitter spammers. Due to spammers' increased use of social media platforms, it is crucial to combat their fraudulent operations. This study uses extensive Twitter data, such as user profiles, tweet content, and network connections, to create algorithms that accurately differentiate between genuine users and spammers. By successfully identifying and filtering out spammers utilizing a range of machine learning algorithms, deep learning techniques, feature extraction methodologies, and even categorizing tweets based on their emotional tone, the ultimate goal is to improve platform trust and user experience. In order to efficiently identify spammers and fake users, the article provides a solution that makes use of a unique machine learning algorithm that meticulously analyses user personal information and account history. Because of its popularity, Twitter has drawn spammers and bogus users, which has caused confusion. The study seeks to address "Twitter Spam Drift" by applying a Semi-Supervised Learning Approach (SSLA), which adjusts to changes in spam behavior and has demonstrated promising results on English datasets. To identify spam behaviors on Twitter, the project uses deep-learning classifier algorithms like LSTM, and GRU. Here, as can be seen, we use various methods to achieve accuracy ranging from 87% to 97%.

Keywords: Semi-Supervised Learning · Social Media · Ham And Spam Messages

1 Introduction

Social networking platforms like Twitter have seen a sharp rise in popularity recently. As a result, there has been an increase in spammers trying to monetize the website by indulging in spamming activities and creating fictitious profiles in order to gain more views and followers. Twitter is mostly used to disseminate news and updates, although some users abuse the service to spread spam and untrue information. Global losses have resulted from the failure of current machine learning techniques to combat the expanding spam problem. Several spam-detecting techniques have failed in the fight

O. Castillo et al. (Eds.): PerSOM 2023, LNICST 517, pp. 134–153, 2024.
https://doi.org/10.1007/978-3-031-66044-3_10

against spammers because to the complicated spammers' software [2]. The purpose of this research is to examine how machine learning techniques are applied to the detection of Twitter spam. We can create models that reliably differentiate between real users and spammers by utilizing large-scale data from Twitter, such as user profiles, tweet content, and network connections. A large dataset of tagged Twitter accounts will be gathered for the project, which will also make use of feature extraction methods and look at different machine learning algorithms. The ultimate goal is to assist in the development of efficient procedures that improve user experience by successfully identifying and filtering out spammers. Spammers have become a major worry as social media platforms like Twitter have grown so quickly. These spammers, whether human or automated, take advantage of the platform to disseminate false information, advertise frauds, and influence public opinion. The integrity and credibility of the platform must be preserved by identifying and combatting Twitter spammers. Due to the fact that spamming tactics are constantly evolving and conventional approaches based on rules have been shown to be ineffectual, interest in applying machine learning techniques has developed [1].

As the use of social media platforms grows in popularity, spammers looking to take advantage of the community also increase. They take part in a variety of nefarious activities, including the promotion of phony goods, phishing, the transmission of malware, and breaking community rules. With 450 million active users each month, Twitter has emerged as a top destination for spammers. Spam tweets are posted by roughly 10% of active Twitter users, posing serious hazards to user privacy and data security. Twitter uses several spam detection technologies to fight this problem, suspending tens of thousands of spam accounts per day. To avoid discovery, spammers vary their tactics frequently, leading to the problem known as "Twitter Spam Drift." The Semi-Supervised Learning Approach (SSLA), a potential technique put forth by researchers, promises to address the problem of Twitter Spam Drift and increase spam detection accuracy. SSLA uses unlabeled data to better understand domain structure and improve the accuracy of spam detection. Researchers can develop a powerful system for differentiating between real users and spammers using SSLA and other machine learning algorithms, which would improve user experience on social media sites [4]. Social media use is prevalent and has a big impact on people's lives as they exchange news about politics, entertainment, and other topics. However, a big problem has emerged with the increase in spammers submitting unrelated content. The sheer number of tweets—billions are sent each year—makes it difficult to spot spam. To create a secure and reliable environment for users on Twitter, the main objective is to find and filter out spammers [3]. By applying the Random forest technique and using existing models. Our ability to identify and filter spammers will improve. The main objective is to significantly improve the current models using our technique, which is based on the random forest algorithm and use RF-Algorithm to improve the model even further.

2 Literature Survey

In order to combat the spam drift, we developed a semi-supervised learning strategy in this article. When there is a greater amount of unlabeled data than labelled data, SSLA is utilized. It enhances the model's learning process by combining labelled and

unlabeled input. Numerous generic or non-generic semi-supervised algorithms as well as wrapper strategies are available. We employed general methods based on supervised learning algorithms. The closest neighbor algorithm and any supervised approach can be merged with the semi-supervised method known as YATSI. The YATSI is divided into two phases. An initial prediction model is developed using a supervised classifier and predictions are generated using this model to pre-label unlabeled occurrences in the first step. This model is trained on labelled data. Pre-labeled and labeled data are combined into one set. According to the algorithm parameter, unlabeled data is given higher weight and labeled data is given a weight of 1.0. In the second stage, K nearest neighbors of the combined data are selected for the instance that needs to be classified, and they forecast the class based on the highest sum of weights relating to that class. In this study, we made predictions using the filtered closest neighbor technique and a random forest base classifier. Because it performs better than other methods in predicting spam tweets, random forest is utilized [4].

Here, we offered a way to identify phony users and spam. The system that we suggested is depicted here. The model comprises the methods used in processing, designing the model, and testing it in real-time on Twitter. It adheres to the norms used in natural language processing jobs. Our system was used to identify these tweets. We classified communications that were already stored in our database as spam and real using a variety of techniques. Using this information, we chose a model or technique that provides high accuracy and implemented it in our model. This approach aids in distinguishing between fake and real traits [2]. Various algorithm comparison study is shown in Table 1.

Bag of Words: Determines how frequently a word appears in a manuscript. Each word's frequency within the corpus of a document is contained in the vector of that document.

TF-IDF: In order to employ TF-IDF, you must first determine the TF-IDF score for each word in your data set is in relation to the document, and then you must input the results into a vector. Each word in the text is given a score using this method based on the likelihood that it will appear in texts from other categories as well as the number of items where it does [1].

Research: The team gathers data and information about the problem domain during this stage, including knowledge of different spam subtypes, current spam patterns, and potential features that could be applied to spam detection.

Concept Generation: In this stage, the team generates a variety of concepts and ideas for designing the spam detection system. These ideas could encompass several models, approaches, or algorithms.

Analysis: After coming up with concepts, the team analyzes each one to determine its advantages, disadvantages, and viability. The most promising strategies are found using this study.

Requirements and Constraints: The team determines the precise needs and restrictions for the system after choosing the best concept or concepts. These specifications may include user experience criteria, resource constraints, response times, and accuracy levels.

Prototyping and Implementation: The selected concept(s) are then put into practice as early iterations of the system or prototypes. The design is tested and validated using these prototypes.

Testing and Evaluation: To make sure they adhere to the established requirements and constraints, the prototypes go through comprehensive testing and review. Changing parameters, fine-tuning models, and iterating on the design may all be part of this phase.

Feedback and Iteration: The team iterates on the design, making adjustments and enhancements as necessary based on the findings of the testing, user input, and any unforeseen problems.

Deployment and Monitoring: The system is put into production once it satisfies the required standards. However, spam detection is a never-ending task, therefore regular surveillance and modifications are necessary to keep up with evolving spamming methods [3].

Table 1. Table shows advantage, disadvantage and gap of different algorithms

Algorithms	Advantage	Disadvantage	GAP
Decision Tree [5]	Simple to grasp and visualize	Susceptible to overfitting if no trimming	Decision trees may struggle to identify complicated links and patterns for accurate spam categorization in unstructured and high-dimensional Twitter data
Logistic Regression [6]	Simple and comprehensible	Possibly perform poorly with complex data	The complex and nonlinear patterns found in the various and dynamic Twitter spam data may be too much for logistic regression to handle, resulting in poor detection accuracy

(*continued*)

Table 1. (*continued*)

Algorithms	Advantage	Disadvantage	GAP
Multinomial Naive Bayes [7]	Effective and efficient for classifying texts	A belief in the independence of traits	Simple linguistic features in Twitter data can be made easier to grasp by naive multinomial Bayes, which can make it more difficult to identify minor patterns and potentially lead to less accurate spam identification
Support Vector Machine [8]	High-dimensional data effective	Sensitive to kernel and parameter selection	The high dimensionality and noise in textual data may limit SVM's ability to discern complex spam patterns, which may reduce its effectiveness in spotting Twitter spam
Bernoulli Naive Bayes [9]	Quick & easy	Feature dependency sensitivity	Because Bernoulli Naive Bayes assumes binary feature independence, which makes it challenging for it to understand complex relationships in tweet content, its accuracy may decrease in difficult spam detection conditions

(*continued*)

Table 1. (*continued*)

Algorithms	Advantage	Disadvantage	GAP
Semi-Supervised Algorithm [10]	Combines tagged and unlabeled data	May not operate properly if the labeling is insufficient	The YATSI algorithm's drawback is that it depends on a predetermined set of traits, which may make it difficult for it to adapt and successfully capture changing and unique Twitter spam trends
1D convolutional neural Network (CNN) model/algorithm [11]	Learning features automatically	Limited Understanding of Semantics	1D CNN must be able to capture long-range relationships in sequential data in order to detect Twitter spam
GRU (Gated Recurrent Unit) [12]	Decreased vanishing gradient issue	A short-term memory deficit	GRU falls short of LSTM when it comes to gathering intricate sequential patterns over longer sequences
LSTM (Long Short-Term Memory) [13]	Properly manages sequential data	Processes sequential data appropriately	Because it overfits on short datasets and has trouble processing extremely long sequences, LSTM is not ideal for recognizing Twitter spam

3 SPAM Detection Algorithms

3.1 Decision Tree

It is a form of supervised classifier used to understand classification and regression problems. The central nodes of this classifier's tree-like structure describe the characteristics of the data set, while the branches denote the rules or decisions, and the leaf nodes stand in for the outcome. Decision nodes and leaf nodes are two additional types of nodes found in decision trees. Various branch nodes are connected to the decision node in this situation, which is used to make decisions. The result of these branches is represented by a leaf node, which has no branches. Entropy is utilized in this case to help the decision tree partition the data. It is important because it can alter the structure of the decision

tree by establishing new rules. You can write it as Eq. (1).

$$H(S) = -p_+ \log_2(p_+) - p_- \log_2 p_- \tag{1}$$

In Eq. (1) where, (p_+) indicates the % of + ve class, (p_-) indicates the % of -ve class.

Drawbacks: Decision trees may struggle to identify complicated links and patterns for accurate spam categorization in unstructured and high-dimensional Twitter data.

3.2 Logistic Regression

The binary functions in this algorithm are separated using the sigmoid function. Let x be the initial feature of the vector, which is used to determine the likelihood that the given text would be output. It is provided as

$$p = \frac{e^{a+bx}}{1 + e^{a+bx}} \tag{2}$$

Here, in Eq. (2), p stands for the probability of a, i.e., it represents the fundamental logarithm's foundation, while a and b represent the parameters of the representation. Where $x = 0$, p returns the value, and b is used to regulate the rapid changes in probability that occur when x is changed by one bit.

Drawbacks: Logistic regression may not be able to manage the intricate and non-linear patterns present in the diverse and ever-changing Twitter spam data, leading to insufficient accuracy in detection.

3.3 Multinomial Naive Bayes

It is employed to find differences in counts between several categories. It goes by the names count rate and reflect cont. The text provided indicates the word count and assigns distinct categories to each. The multinomial naive bayes algorithm is only utilized in this particular section. Its formal name is Eq. (3).

$$p(X|C_k) = \frac{(\sum_{i=1}^{n} x_i)!}{\prod_{i=1}^{n} x_i!} \prod_{i=1}^{n} Pk_i^{x_i} \tag{3}$$

Here, in Eq. (3), we find the likelihood that a feature of class 'Ck' will have the criterion 'X'. The count rate, which defines frequency, can then be found by using the value of feature x, which can either be 0 or 1. It must be produced by a Pi multinomial. The number of instances of the provided event I is now displayed as 'xi'.

Drawbacks: Naïve multinomial Bayes can make complex linguistic features in Twitter data easier to understand, making it more difficult to spot minor trends and potentially resulting in less accurate spam detection.

3.4 Support Vector Machine

An efficient supervised method of teaching for regression and classification of Support Vector Machine (SVM) is an application. It reveals a hyperplane that optimizes the

margin between distinct classes to provide the best separation by grouping data points into several categories.

Drawbacks: SVM's efficacy in identifying Twitter spam may be impacted by the high-dimensionality and noise present in textual data, which may limit its capacity to recognize intricate spam patterns.

3.5 Random Forest

Step1: In random forest model there are n number of random records that are having k number of records taken from data sets.
Step2: There will be decision trees constructed for each every experiment.
Step3: And these individual decision trees all generate an outcome separately.
Step4: And all outcomes are compared using accuracy, majority voting, and the final outcome will be taken.

Drawbacks: A restriction in the detection of Twitter spam is the Random Forest possibility to overfitting on noisy and high-dimensional text input, which could lead to lower generalization performance.

3.6 Bernoulli Naive Bayes

Bernoulli Naive Bayes is a type of algorithm known as Naive Bayes that is used for binary task classification. When working with binary features (i.e., only accepting values of 0 or 1), it is extremely useful because it evaluates the likelihood of a class given the presence or absence of each feature separately [1, 2]. Formula for computing the probability in Bernoulli Navie Bayes:

$$p(X|C_k) = \prod_{i=1}^{n} p_{k_i}^{x_i}\left(1 - pk_i\right)^{(1-x_i)} \tag{4}$$

In Eq. (4), xi is the value of the feature xi (either 0 or 1), where P(y) is the previous probability of class y. The probability of feature xi given class y is P(xi | y). The projected class for the given input features is then supplied as the class with most chances.

Drawbacks: The accuracy of Bernoulli Naive Bayes in challenging spam detection situations may suffer as a result of its assumption of binary feature independence, which makes it difficult for it to comprehend complicated relationships in tweet content.

3.7 Semi Supervised Algorithm

After preprocessing, I separated the resulting dataset into labelled and unlabelled data with a 30% and 70% division, respectively. The labeled data was then once more divided into a training set and a testing set for utilizing a partition of 60% and 40%. I used the tagged data to train Random Forest, a classifier model that performs better at spam identification than others. For additional semi-supervised learning, I prelabelled the unlabeled data using the model developed from the labelled data. Combine the data that

has been labeled and pre-labeled. Assign weights for labelled data as 1.0 and unlabelled data as W.

$$W = \frac{\pi r^2 F^*(\text{len}(x_{\text{train}})}{\text{len}(Du)} \tag{5}$$

According to the linked paper, F value at 0.1 will produce better results. F is a YATSI parameter in Eq. (5). In step 2, I applied the YATSI algorithm using the K nearest Neighbor search. Take the argmax of the weights of the appropriate class of nearest neighbors for each prediction query [4].

Drawbacks: The YATSI algorithm's disadvantage is that it is dependent on a predetermined set of characteristics, which may hinder it from adjusting and effectively capturing evolving and different Twitter spam trends.

3.8 1D Convolutional Neural Network (CNN) Model/Algorithm

We discovered that this method or algorithm is more effective when testing live data and messages on Twitter utilizing the tweep api after comparing it to other models and algorithms. Tweepy api can be used by utilizing a Python module. Due to the fact that this Tweety api offers a number of built-in functions and sends data in JSON format. And we employ the dedication technique to interact with the API. It also includes creating new accounts and using current user accounts after the creation of this fresh developer account.

We receive 4 keys from Twitter, of which 2 are private keys. These keys are utilized to access data in Json format. Because of this, we can also discover a variety of information about a tweet or message, such as the user name, the login device, the time the message was sent or received, etc. These are additionally utilized to thoroughly check spammers and individuals with phony accounts. We identify spammers using Twitter poly and the terms and conditions. There are numerous accounts that use bots to send spam on Twitter. As a result, it became challenging to examine the live communications. As a result, our team was able to come up with a solution by comparing an SMS data set that includes both spam and non-spam messages. We use this sms dataset as a reference because text messages and tweets have the same format. After comparing the results, we can decide where our model needs improvement in order to be more accurate. This is also used in Twitter to categorize spam and non-spam individuals.

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total number of predictions}} \tag{6}$$

$$\text{Specificity} = \frac{\text{No. of true negatives}}{\text{No. of true negatives} + \text{No. of false positives}} \tag{7}$$

$$\text{precision} = \frac{\text{Number of True Positive}}{\text{No. of True Positive} \ + \ \text{No. of False Positive}} \tag{8}$$

By Eq. (7) and (8) we will find $\text{F1} - \text{Score}$,

$$\text{F1} - \text{Score} = 2 * \frac{\text{Precision} * \text{Sensitivity}}{\text{Precision} \ + \ \text{Sensitivity}} \tag{9}$$

Here, we discover precision through Eq. (6), specificity from Eq. (7), and accuracy from Eq. (8). Now we'll determine the F1-Score value using Eqs. (7) and (8).

3.9 Gated Recurrent Unit (GRU)

An architecture of recurrent neural network (RNN) is known as the GRU (Gated Recurrent Unit) algorithm is utilized for sequential data processing tasks as well as for natural language processing applications including text classification and spam detection. The more conventional RNN has some drawbacks, including the vanishing gradient issue and the challenge of capturing long-term dependencies, which are addressed by the GRU variation. In order to regulate the information above the recurrent units, the GRU method incorporates gating mechanisms, allowing the model to specifically update and forget about the information. The following are the GRU's essential elements:

Determines how much of the previous data should be kept and how much needs to be updated with the most recent input.

Reset Gate: Regulates how much historical data should be disregarded in order to avoid interference from unrelated history data.

Candidate Activation: The current input and the prior activation are combined and weighted by the reset gate to create a new candidate activation.

Final Activation: Creates the updated activation for the current time step by combining the previous activation and the candidate activation, weighted by the update gate.

Drawbacks: When it comes to collecting complex sequential patterns over longer sequences, GRU falls short of LSTM.

3.10 Long Short-Term Memory (LSTM)

The LSTM (Long Short-Term Memory) algorithm is another sort of recurrent neural network (RNN) architecture used to store data in sequential processing applications, as text classification and spam detection. Traditional RNN drawbacks including the vanishing gradient issue and the condition in capturing long-term province are addressed by LSTM. LSTM introduces specialized memory cells and gating processes enable the model to store, read, and delete information across a long period of time. The key components of the LSTM include:

Cell State: The memory element that covers the whole sequence and transports data throughout various time steps.

Input Gate: Identifies a rough estimate of the amount of current that should be stored in the designated cell state.

Forget Gate: This gate should clear the preceding cell state.

Output Gate: How much of the cell state is transmitted to the following time step is determined by the output gate.

Tweets are often represented as collections of words or characters when using the LSTM algorithm for Twitter spam detection. The LSTM network processes these sequences and learns to extract pertinent information and predict whether a given tweet is spam or not. The LSTM algorithm discovers the ideal weights and parameters during training by minimizing a loss function via methods including back propagation across

time. On a different validation or test set, the model's performance can be assessed using measures like precision, recall, F1 score, or accuracy.

Drawbacks: The disadvantage of LSTM for identifying Twitter spam is that it overfits on small datasets and struggles to handle very long sequences.

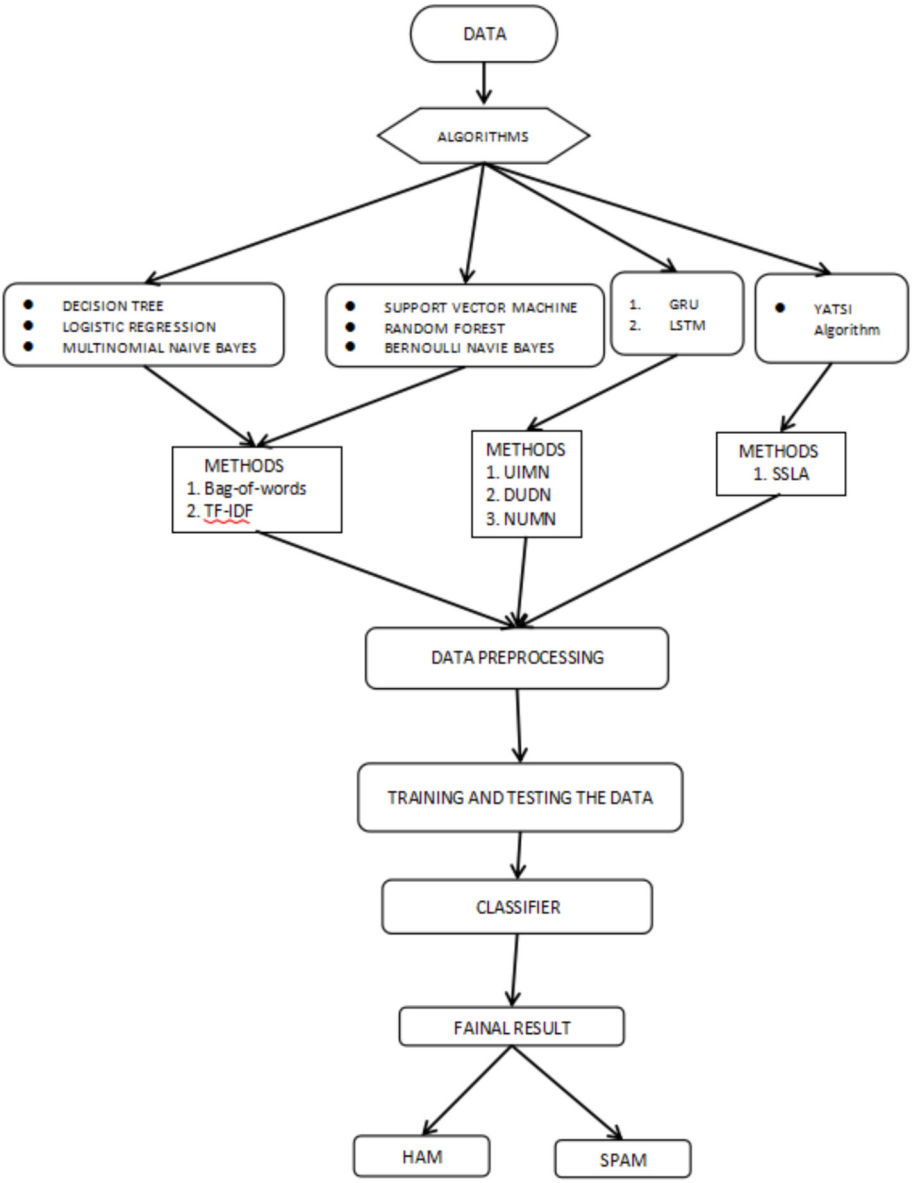

Fig. 1. Various algorithms flow

The process for determining the differences between utilizing various algorithms to determine accuracy and the various approaches employed are shown in Fig. 1. Then, how does processing work when we have varying degrees of accuracy? With that, we can even determine whether messages are ham and spam.

4 Comparisons of Data Analysis

4.1 Data Preprocessing

The provided code creates a bar plot in the Fig. 2 graph using matplotlib.py plot to display the distribution of the 'ham' and 'spam' categories in the Data Frame df's 'target' column. The height of each bar in the picture displays the count of each category, making it easy to compare the frequency of the 'ham' and 'spam' categories in the dataset. For ease of understanding, the figure has a title and axis labels. The x-axis and y-axis represent the categories and count, respectively. The number of spam messages is shown in violet on the graph below, while the number of ham messages are shown in red. A red bar indicates 4516 ham communications, while a violet bar indicates 653 spam messages.

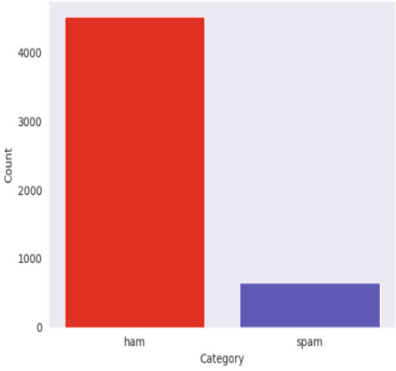

Fig. 2. Showing ham and spam messages count (Color figure online)

The x-axis and y-axis in the graph in Fig. 3 serve as placeholders for the class and count, respectively, and the plot has a title and axis names for clarity. When both positive and negative classes are equally counted, the blue and orange graphs reflect them.

The total number of Twitter accounts is shown in Fig. 4, and it is again broken down into bogus accounts and tweets with spam content. The count of the total twitter accounts are 36, the count of fake accounts are 18 and the count of spam content tweets are 25.

The histograms in Fig. 5 depict the distribution of message lengths for the two categories ('ham' and 'spam'). The legend indicates which histogram corresponds to which category, while the x-axis label provides information about the data being displayed. Here, the lengths of ham messages and spam messages are displayed; the graphs in blue and orange respectively illustrate the lengths of ham messages and spam messages.

Spam message length is represented by the yellow graph in Fig. 6 and ham message length by the green graph. By this graph we say that ham messages are majority classes

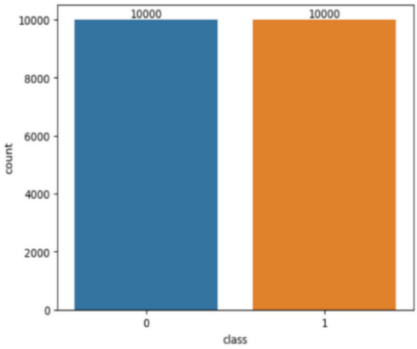

Fig. 3. Graph showing positive class and negative class (Color figure online)

Fig. 4. Count of fake and spam accounts (Color figure online)

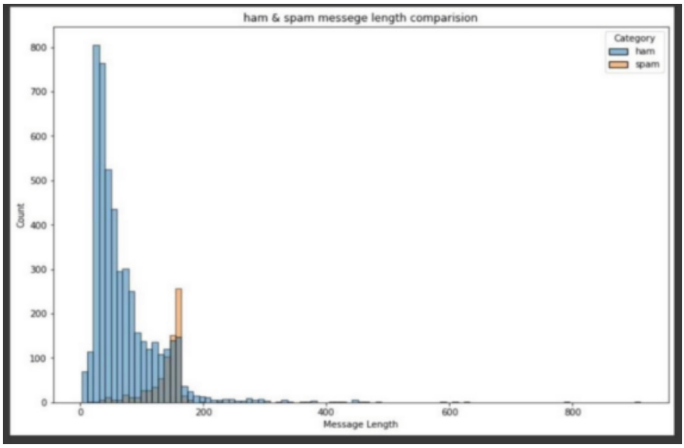

Fig. 5. Ham and spam message length comparison (Color figure online)

as they are of highest in number and spam messages are minority classes as they are of smallest in number. Now we can say that length of ham is highest as compared to spam.

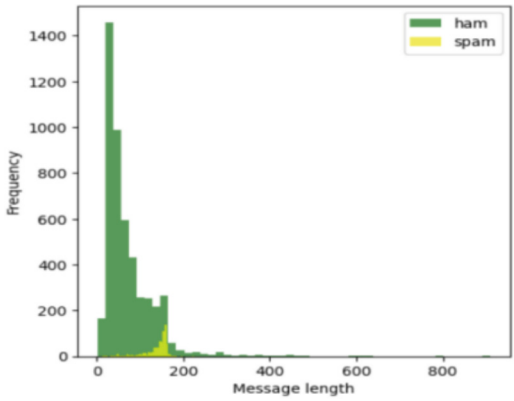

Fig. 6. Distribution of message lengths (Color figure online)

The code shown in Fig. 7 uses Seaborn to build a count plot that displays the distribution of characters ('num_chars') in a dataset. Each 'num_chars' category is shaded based on the 'target' variable it is linked to. This narrative helps us identify any patterns or trends in the data and shows how the character count varies depending on the target demographic.

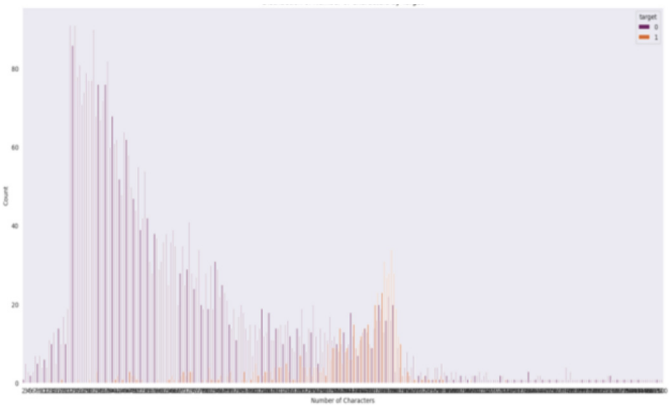

Fig. 7. Distribution of number of characters by target

The code in Fig. 8 uses Seaborn's Pair Grid to generate a grid of scatter plots, kernel density plots, and histograms to display the correlations and distributions between the variables in the Data Frame (df). When the data points are colored using the 'target' variable, it is easier to identify patterns and variations between groups.

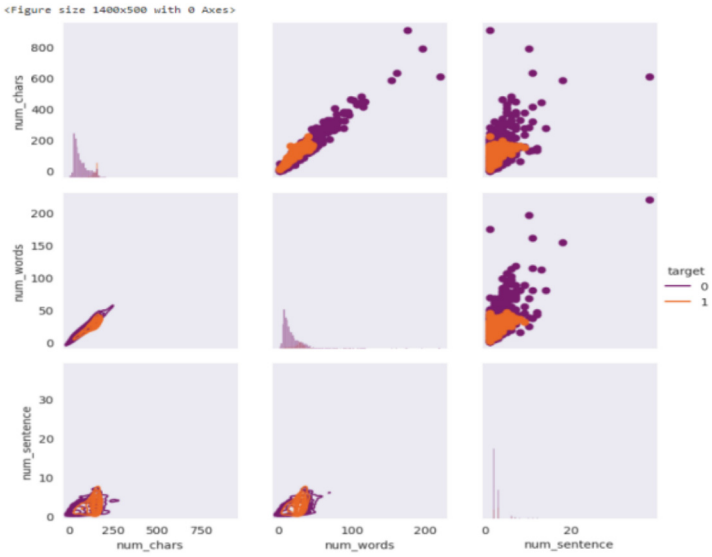

Fig. 8. Showing relationships between the variables in Data frames

The following code creates a long-format Data Frame from the Data Frame df, calculates its correlation matrix, and uses Seaborn to create a bar plot in Fig. 9 that displays the correlations between the variable pairs. The variable pairings are displayed on x-axis, while the correlation coefficients are displayed on y-axis. The correlation between a pair of variables is shown by each bar. The image allows us to quickly assess the magnitude and direction of correlations between the variables in the Data Frame.

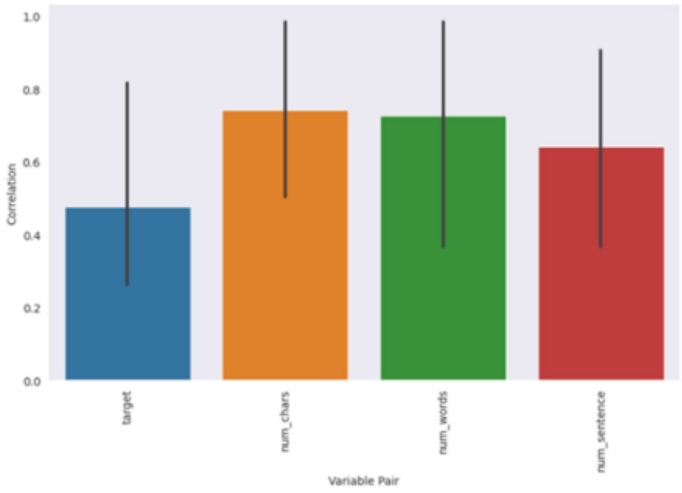

Fig. 9. Correlations between variables

The figure in Fig. 10 provides a graphic representation of the confusion matrix that makes it easier to grasp and learn more about how effectively a classification model is working.

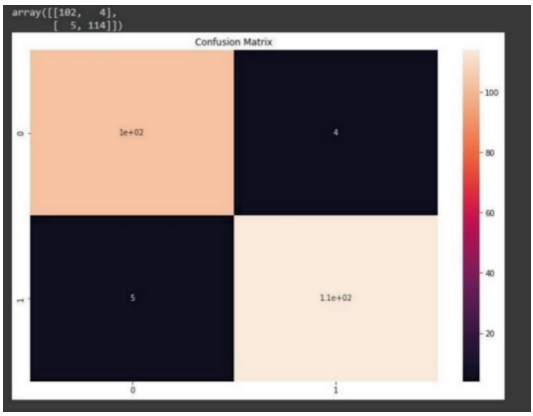

Fig. 10. Confusion matrix

The confusion matrix for a classification model is shown in Fig. 11 by contrasting the actual labels with predicted labels.

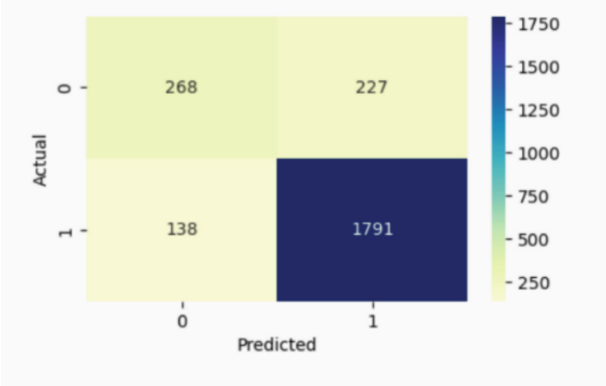

Fig. 11. Comparison of actual and predicted labels

5 Comparisons of the Accuracy of Twitter Spam Detection Using Different Algorithms

In the Table 2, various algorithms are used to compare each other's predictions on the accuracy of the dataset that was used. Here, we can observe that a couple models have accuracy levels around 97% shown in Table 3 and Fig. 12. If we compare the remaining models, the maximum accuracy is 97.58%, and it is decreasing. The highest accuracy we have compared so far is 97.58%, and the lowest accuracy is 85%. As a result of our comparison, Random Forest and SVM are the top algorithms because to its high accuracy. The YATSI algorithm offers a reliable and consistent solution to the Twitter spam drift issue.

Table 2. Table contains different f1 score, precision, recall, support of different algorithms

Algorithm	F1 score	Precision	Recall	Support
Logistic regression	0.82	0.97	0.71	138
Decision tree	0.69	0.82	0.59	138
SVM	0.90	0.97	0.84	138
KNN				
Random forest	0.90	0.98	0.83	138
Multinomial naïve bayes	0.88	1.00	0.78	138
Bernoulli Naïve Bayes	0.93	0.99	0.88	138
GRU	0.71	0.82	0.62	649
LSTM	0.73	0.66	0.81	649
YATSI	0.91	0.93	0.89	2018

Table 3. Table contains different accuracy of using different algorithms

Algorithm	Accuracy
Logistic Regression	95.84%
Decision Tree	92.84%
SVM	97.58%
KNN	90.52%
Random Forest	97.58%
GRU	93%
LSTM	94%
YATSI	85%

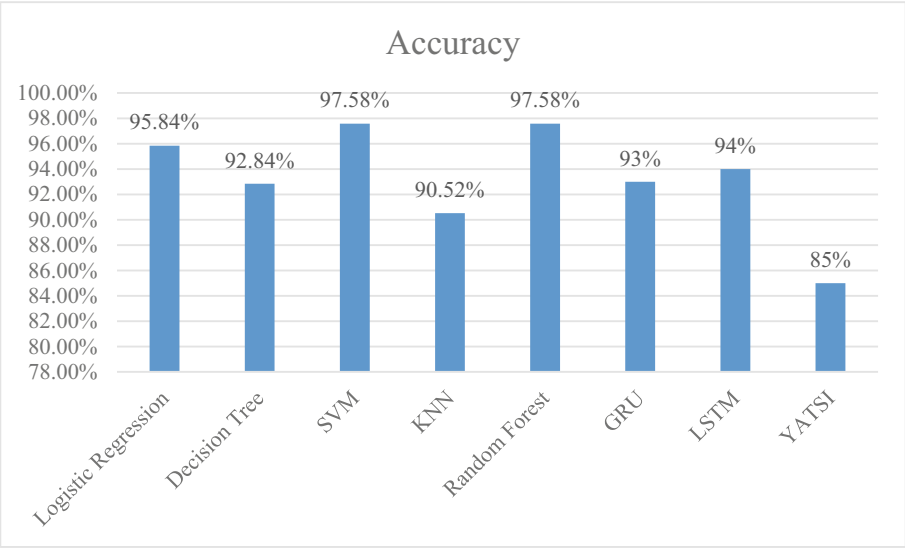

Fig. 12. Accuracy Graph

By identifying spammers and fake users on Twitter, the study also brought attention to the inadequacies of the ML and DL algorithms caused by the complexity of spammers. According to the study, an algorithm based on personal information like followers, user name, date of birth, and account creation date would be more accurate. The study also addressed the problem of Twitter's pervasive spam by introducing the YATSI algorithm as a consistent and dependable solution to the Twitter spam drift problem. While the accuracy of our study model is increased by employing the Random Forest approach, doing so requires more computing resources and is more difficult due to its subfunctions. It sometimes fails to contain all variables, such as age, gender, and the date of account creation, etc. It took more time and people to develop the model using the RF-Algorithm, which takes more resources than usual.

6 Conclusion

In order to build a sophisticated spam detection system, this study assessed a range of traits and machine learning methods. Textual, user-based, and network-based features were all extensively used in earlier spam detection research. The report addressed a wide range of feature extraction methodologies. Important traits including tweet content, user behavior, and relationships between users were successfully captured by these features. The research also examined a variety of machine learning algorithms, including supervised and unsupervised methods, in order to create accurate categorization models. This paper makes a substantial contribution to the field of Twitter spam detection as a whole by its thorough analysis of existing methodologies and recommendation of a strong spam detection system. The conclusions reached can act as a starting point for further research, stimulating the development of more robust and potent spam detection systems, hence

enhancing user experience and safety on platforms like Twitter. The methodology for real-time Twitter spam tweet identification utilized the SMS Spam Collection dataset. Different outputs were produced by the various machine learning algorithms used in the study, which included Multinomial Naive Bayes (0.97098), Bernoulli Naive Bayes (0.98355), Logistic Regression (0.95841), KNN (0.90522), Decision Tree (0.92843), and Random Forest (0.97582). Future study is encouraged to apply real-data analysis to detect new spamming techniques and to enhance feature techniques with cutting-edge methods.

7 Future Scope

Future potential for using a semi-supervised learning strategy to identify spam on social media is quite promising. This method can improve the accuracy of spam detection and adjust to changing spamming strategies by utilizing machine learning and merging labeled and unlabeled data. The semi-supervised learning model's scalability and capacity for handling big datasets make it an excellent choice for real-time spam identification and mitigation as social media platforms expand, thereby enhancing user experience and security. Furthermore, continuous improvements in data analytics and artificial intelligence are probably going to enhance and optimize this strategy's efficacy in battling the always shifting terrain of social media spam.

8 Limitations

Some of the limitations of the semi-supervised learning approach for social media spam identification include the requirement for high-quality labeled data, the possibility of difficulty in adjusting to quickly evolving spam patterns, the tendency to overfit when working with a small number of labeled instances, difficulties deciphering complex models, privacy concerns related to sensitive data, and the resource-intensive nature of training and maintaining the model over time. It is imperative to tackle these limitations in order to implement this strategy in practical situations.

References

1. Hemalatha, P., Ragapriya, N., Sanjana, V., Shiva Bhavani, K.: Extreme learning machine for spammer detection and fake user identification from Twitter. J. Crit. Rev. **10**(03), 184–192 (2023)
2. Alom, Z., Carminati, B., Ferrari, E.: A deep learning model for Twitter spam detection. Online Soc. Netw. Media **18**, 100079 (2020)
3. Rodrigues, A.P., Fernandes, R., Shetty, A., Lakshmanna, K., Mahammad Shafi, R.: Real-time Twitter spam detection and sentiment analysis using machine learning and deep learning techniques. Comput. Intell. Neurosci. **2022**, 1–15 (2022)
4. Meda, C., Bisio, F., Gastaldo, P., Zunino, R.: A machine learning approach for Twitter spammers detection. In: 2014 International Carnahan Conference on Security Technology (ICCST), pp. 1–6. IEEE (2014)

5. Imam, N., Issac, B., Jacob, S.M.: A semi-supervised learning approach for tackling Twitter spam drift. Int. J. Comput. Intell. Appl.Comput. Intell. Appl. **18**(02), 1950010 (2019)
6. El-Mawass, N., Honeine, P., Vercouter, L.: Simil Catch: enhanced social spammers detection on twitter using markov random fields. Inf. Process. Manag. **57**(6), 102317 (2020)
7. Dewang, R.K., Singh, A.K.: State-of-art approaches for review spammer detection: a survey. J. Intell. Inf. Syst. **50**(2), 231–264 (2018)
8. Meda, C., et al.: Spam detection of Twitter traffic: a framework based on random forests and non-uniform feature sampling. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 811–817. IEEE (2016)
9. Json.org. (n.d.). JSON. http://www.json.org/. Accessed 22 Aug 2018
10. Lin, G., Sun, N., Nepal, S., Zhang, J., Xiang, Y., Hassan, H.: Statistical Twitter spam detection demystified: performance, stability and scalability (2017)
11. Hu, X., Tang, J., Gao, H., Liu, H.: Social spammer detection with sentiment information. In: 2014 IEEE International Conference on Data Mining, pp. 180–189 (2014). https://doi.org/10.1109/ICDM.2014.141
12. Srivastava, G., Maddikunta, P.K.R., Gadekallu, T.R.: A two-stage text feature selection algorithm for improving text classification. ACM Trans. Asian Low-Resource Lang. Inf. Process. **20**, 1–19 (2021)
13. Ghelani, P.H., Bhalodia, T.M.: Opinion mining and opinion spam detection. Int. Res. J. Eng. Technol. (IRJET) **4**, 11 (2017)