IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.13490

Combatting Spam in Online Chat Platform: A Comprehensive Approach to Detection and Mitigation

Prathmesh Singh[1], Viraj Bhojane[2], Kishan Mishra[3], Rohan Thamke [4], Prof. Jagat Gaydhane[5]

Department of Information Technology, Datta Meghe College of Engineering, Airoli, Navi Mumbai[1-5]

Abstract: The digital sector is filled with numerous platforms that enable online communication. Apart from interruptions due to spam messages and security breaches, these have posed a significant challenge. The purpose of this paper is to provide a comprehensive way of dealing with the issue of spam on chat platforms. We did so by using machine learning algorithms such as Naive Bayes, Support Vector Machine (SVM), Logistic Regression, et cetera in order to see how effectively they can be used in identifying and filtering out spam messages.The study involved an extensive comparison between accuracy and precision metrics for evaluating the performance of these algorithms.These algorithm's strengths and limitations are shown through experimentation and analysis that give clues into what to consider when developing algorithms for detecting spam messages in various online chat platforms.In this light, we have successfully applied Naïve Bayes and logistic regression to text based Spam classification. After conducting thorough tests on it, the system had been able to identify 97% of all spams accurately resulting into a Precision Score of 1 thus enhancing trustworthiness and safety measures of online communication portals. Precision is preferred over accuracy due to imbalance in data.

Keywords: Text Spam, Naive Bayes, Logistic Regression, Chats, Spam Detection

I. INTRODUCTION

In today's digitally connected world, online communication plays an important role in facilitating communication and information exchange between individuals, businesses and communities. Chat apps, in particular, have become ubiquitous instant messaging tools that allow users to easily connect and collaborate across geographic boundaries. But while these platforms provide convenience and efficiency, the proliferation of spam has also become a major problem, making online communication ineffective and unreliable.

Spam is characterized by unwanted and often irrelevant content that can cause serious problems. Users, platform providers, and cybersecurity professionals face challenges on many fronts. From inappropriate and indecent advertising that distracts users to endangering privacy and personal security, spam not only disrupts communication but also impacts trust in online platforms. Therefore, finding spam and resolving communication issues is important to improve user experience, protect personal information, and create a safe and supportive environment.

Even traditional spam detection (like filtering rules and content matching) is useful for some users. For some reason they often don't analyze and filter spam messages properly. Additionally, as online threats continue to evolve and new spam technologies proliferate, more powerful and flexible solutions are needed to effectively and instantly prevent spam.To solve these problems, this research aims to develop new methods to detect and reduce spam. Our goal is to improve ability to accurately detect spam by integrating advanced learning techniques, including natural language processing and statistical models into our application.

This study aims to discuss spam and offer new solutions to its limitations by analyzing existing data, researching the science and facts. By combining research in machine learning, network engineering, and network security models, we hope to pave the way for the development of the next generation of interactive, powerful, effective, and anti-spam standards.

This research aims to investigate the issues and opportunities in spam detection within chat applications in order to develop new technologies for online communication and improve the overall user experience. By embracing innovation, collaboration and mutual understanding, we aim to continue working to create a safer, more secure and more enjoyable online environment for all users.

II. LITERATURE SURVEY

We began our literature by considering spam detection system brought to light by considering spam detection system brought to light by Choudhary et al [1]. They proposed a binary classifier by applying the feature vectors of spam and ham messages. They extracted features like presence of URLs, dots, mathematical symbols, emotions, mobile numbers, etc and used these features in SMS Spam Corpus dataset. They have tested via Na¨ıve Bayes, Logistic Regression, J48, Decision table and Random Forest algorithms. They concluded that their results were best achieved with Random Forest Classfier with highest TP rate (96.5%) and lowest FP rate (1.02%).

S.K. Trivedi [6] did a study of machine learning classifiers for spam detection. The study focuses on spam classification using machine learning classifiers on the Enron email corpus. Support Vector Machine (SVM) is found to be the best classifier with low false positive rate and high accuracy. The study emphasizes the importance of not misclassifying ham mails as spam. SVM and boosted decision tree are identified as promising classifiers. The study uses a string-to-word-vector transformation method and binary representation for feature extraction. The selected classifiers have low fall-out rates and high F-measure. Despite longer model building time, SVM yields the most desirable results.

Shirani-Mehr et al. [2] experimented with different machine learning algorithm to SMS spam classification problem. The database used was UCI machine learning repository. Different techniques like Bayes, SVM and other methods were applied. Feature extraction and analysis were done in MATLAB and algorithms were done in python using scikit-learn library. The paper concluded that SVM was best used for the dataset with 97.64 percent accuracy.

Xu et al [3] proposed features like static, temporal and network features which were feeded to the classification algorithm (SVM and K-NN). The idea in this paper is to use noncontent features where SVM was used to pay attention to margins and cases near hyperplanes while K-NN focused on typical positive and negative cases. In Telco dataset, the paper shows comparison between SVM and K-NN classifiers where it showed that SVM classifiers can achieve better performance. The paper also has shown comparison among different feature categories using ROC curve.

Nikhil kumar [7] published a paper under the title 'Email Spam Detection Using Machine Learning Algorithms’. The paper discusses email spam and the application of machine learning algorithms to detect and filter spam emails. The authors clarify the importance of identifying bogus emails and look at a variety of machine learning techniques that can be used in email screening applications, such as Naïve Bayes, support vector machines, and decision trees. They also shared the technique they used for the study, which included procedures for data preparation and multiple data sets for model training.

In 2016, the International Conference on Engineering and Technology published” Designing and implementing a real-time web-based chat server” by Diotra Henriyan, Devie Pratama Subiyanti, and Rizki Fauzian (ICSET). According to this study report, a chat application should have a live forum and be multi-site to accommodate a large user base. The programming language is used to create the MongoDB website and the Node.js server with a clear foundation.

The paper discusses email spam and the application of machine learning algorithms to detect and filter spam emails. The authors clarify the importance of identifying bogus emails and look at a variety of machine learning techniques that can be used in email screening applications, such as Naïve Bayes, support vector machines, and decision trees. They also shared the technique they used for the study, which included procedures for data preparation and multiple data sets for model training.

R. Gayathri, C. Kalieswari published their work in 2020 in the International Journal of Engineering and Advanced Technology (IJEAT) [5]. The chat application offers a better and more adaptable programme for debate, according to this research article. Developed with cutting-edge technologies to offer a dependable system. The system’s primary benefits include group chat, improved security, real-world collaboration, and instant messaging. For the majority of businesses looking to have private applications, this app may locate the greatest market demand. Based on community requests, more programme features like conference calls and video chat will also be implemented. Sharing locations, etc., according to necessity.

Various application in today generation also have built in spam filters like Gmail.

Gmail not only uses Machine Learning classifier to filter spams. To secure the system more, they also use blacklisting the users who repeatedly send such unwanted emails Hence they use ML classifiers, Reputation, Deep Learning and Tensorflow for spam filtering. Different applications like SpamTitan, Mailwasher, Zerospam are anti-spam software too.

DOI: 10.17148/IJARCCE.2024.13490

III. METHODOLOGY

A. Flowchart

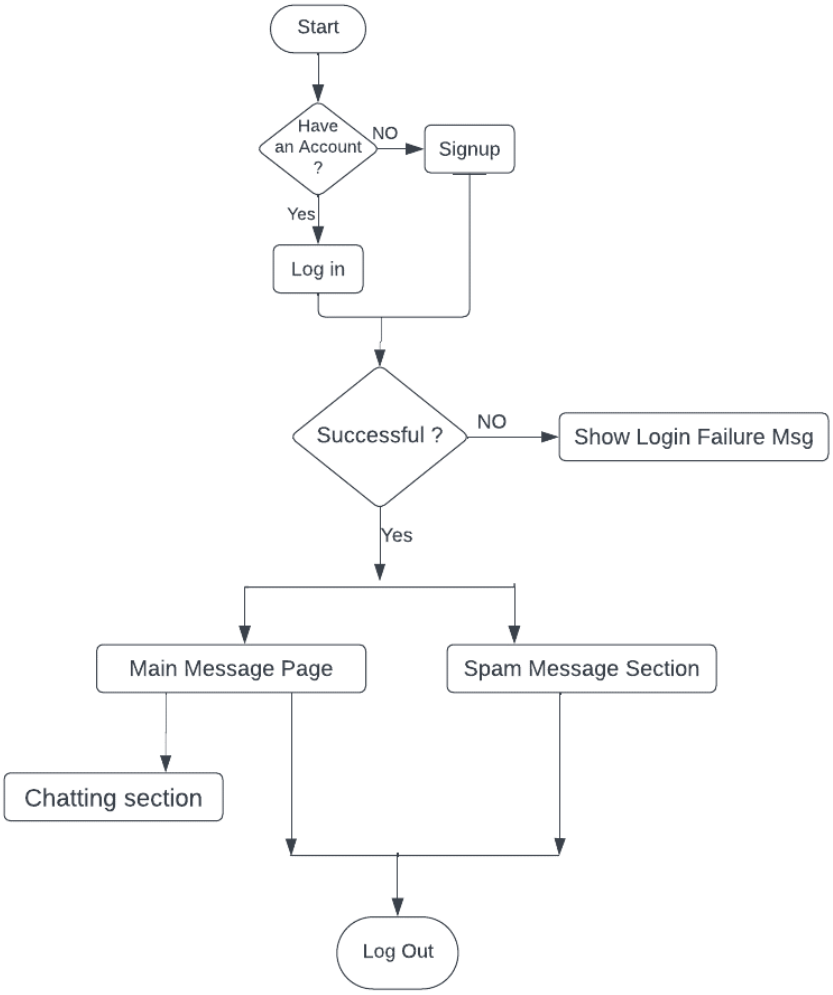


Fig.1 Flowchart

The flowchart seems to tell the best way to sign into a discussion channel. It begins with the inquiry: " Have you set up an account?" Assuming the response is no, the client will be coordinated to enlist in the talk. Assuming that the response is true, the client is coordinated to log in. When signed in, the client will be diverted to the primary informing page. From that point, they can visit with different clients or check the spam segment. They can likewise log out whenever.

Here is a more nitty gritty meaning of the flowchart:

- Start: The flowchart's beginning can be seen here.

Do you have a record? This inquiry poses to the client assuming they as of now have a record of the visit.

- No: if the client doesn't have a record, they will be diverted to "Information exchange" command
- Login: In the login segment, the client makes another record in the discussion board. This will probably expect you to enter some private data, for example, your name and email address.
- Yes: The "Login" section will be redirected to the user's existing account. command
- Login: In the login region, the client enters his username and secret key.

Success? This inquiry poses if the login was effective.

- No: If the login fizzled, the client is shown "Login failure Msg" This message will probably make sense of why the login fizzled, for instance, if the username or secret word was wrong.
- Yes: If the login is fruitful, the client will be diverted to the "Principal Message Page".

DOI: 10.17148/IJARCCE.2024.13490

- Fundamental Message Page: This is the principal talk screen. From here, the client can talk with different clients or view the spam area.
- In the visit segment: In the talk segment, the client can visit with different clients who are likewise on the web. They can compose messages and send them to different clients.
- Spam Area: The spam area is presumably where clients can report spam.
- Logout: A client can log out of the talk whenever by clicking the "Logout " button.

B. Data Flow Diagram

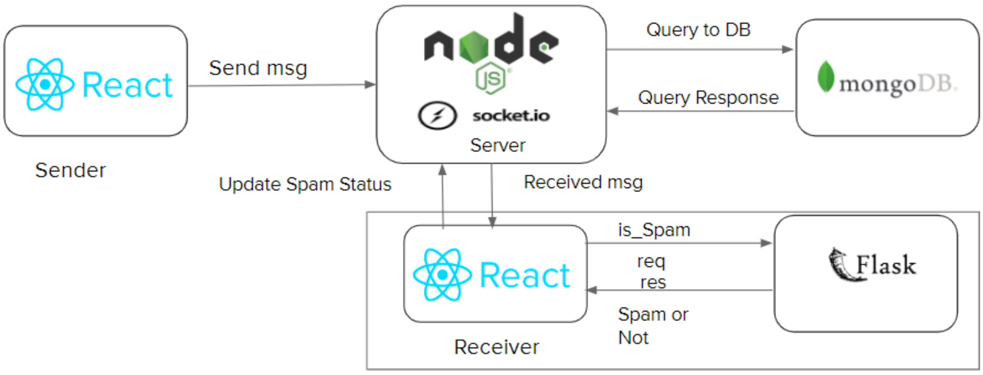


Fig.2 Data Flow Diagram

A data flow diagram shows a high-level view of how a chat application works. It consists of several parts:

- React: This is the front part of the software that is accountable for what the person sees and interacts with. It is constructed with the aid of the usage of the JavaScript library React.
- Express.Js: This is the again end of the utility accountable for processing statistics and common sense. It is built with the use of Express.Js, a Node.Js net framework.
- Socket.Io: Two-manner the front-quit and returned-end communication is made feasible the usage of the actual-time communique library Socket.Io. This implies that records may additionally waft in actual time from the front cease to the back quit and vice versa.
- MongoDB: It is a NoSQL records base used to save talk messages and distinctive data.

Information goes via the framework as follows:

- The consumer communicates something precise: The consumer composes a message inside the speak interface (Respond) of the UI.
- React sends a message to the server: The React application sends the message to the back-end (Express.js) via a Socket.io connection.
- The message is stored on the server: The Express.Js software stores the message in the MongoDB data set.
- Server communicates something specific: The Express.Js utility has an influence on undeniably related clients via Socket.Io.
- Clients obtain the message: React packages of all connected users acquire the message via Socket.Io and update the chat for this reason
- Receiver sends a message to the API: A React utility sends a message to the Flask API through an HTTP request.
- User sends a message: The message sent by way of the consumer is pre-processed and analyzed with two educated fashions (Naive Bayes and Logistic Regression).
- API Classifies Spam: Combined model predictions determine if a message is spam and the result is again to the customer. Nine.
- Server updates spam status: The Express.Js software updates the spam status of a message within the MongoDB database.
- Server sends unsolicited mail word: The Express.Js utility sends a refreshed spam popularity to the customer who referred to it.
- Respond update message: The chat interface is updated with the aid of React to expose that the message is now considered unsolicited mail.

DOI: 10.17148/IJARCCE.2024.13490

C. Database Schema

Databases are the backbone of modern information systems designed to store and manage large amounts of structured data. The database schema is at the core of this storage and retrieval process. It operates like an architectural drawing that explains how the data elements within a given database relate to one another. A good schema forms the basis for efficient data management, thereby allowing for smooth data manipulation and access by applications as well as individual users. Moreover, often, it comprises security features such as encryption and access controls aimed at protecting confidential data from unauthorized access or alteration, thus acting a root of those practices leading to proper governance of information emanating from compliance with both regulatory guidelines and sector benchmarks towards achieving quality and trustworthiness across all users.

Below is our database schema

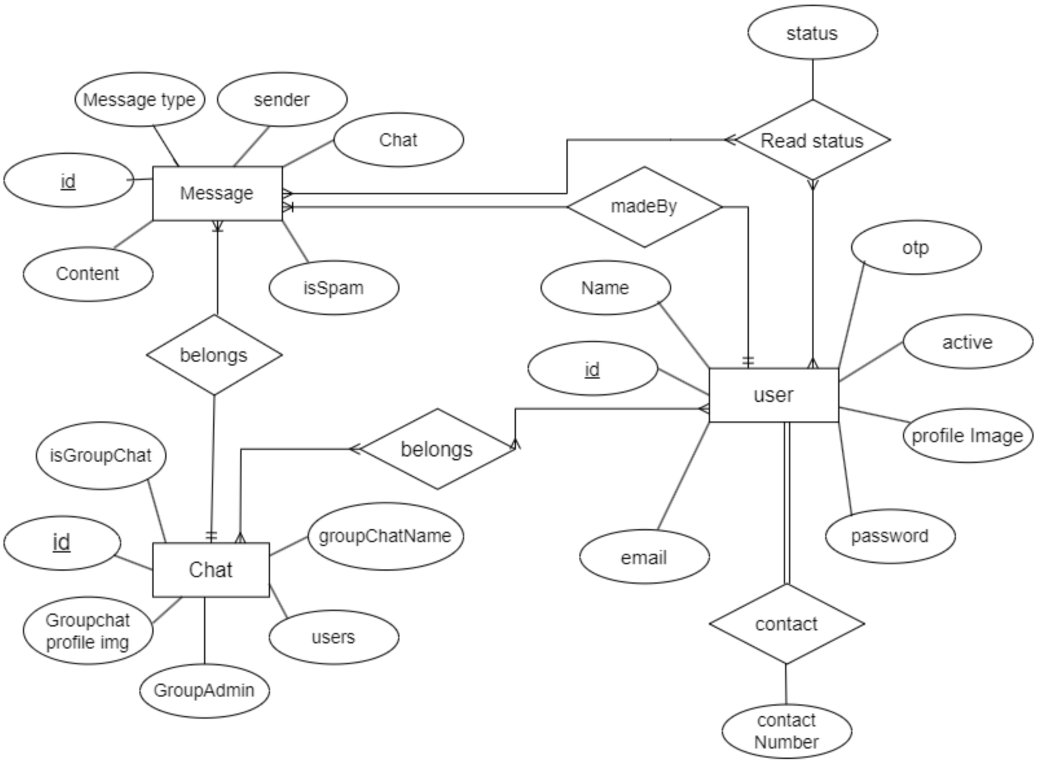


Fig.3 Database Schema

D. Model Training

1. Dataset used:

- Data Size: Total messages: 5,574
- Labeling: Each message is assessed as both "ham" (valid) or "direct mail".

This binary category makes it appropriate for supervised studying obligations.

Columns:
Message: This column carries the real text content cloth of the SMS message.
Label: This column shows whether or not the message is "ham" (0) or "direct mail" (1).

2. Data Preparation:
In order to prepare the dataset for this evaluation, we divided the dataset into two information units the training data set (80 percent) and the testing set (20 percent). The motive of splitting the information is to offer models with sufficient data to study the sample at some point of schooling at the same time as keeping an independent dataset (20 percent) for independent testing functions.

3. Feature engineering:
The text data was cleaned and prepared using the following methods during the preparation stage of creating the spam detection model:

DOI: 10.17148/IJARCCE.2024.13490

Text Normalization:

- Lowercasing: To guarantee consistency in the data and enable consistent analysis, all text was changed to lowercase.
- Stemming/Lemmatization: To help limit word variety and capture vital meaning, words were reduced to their base or root form.
- Eliminating Punctuation: By concentrating just on textual content, punctuation marks were removed, simplifying the analysis process.
- Tokenization: Tokenizing the text involved dividing it into discrete words or units, which is an essential stage for subsequent processing and examination.
- Removing Stop Words: Stop words, or frequently recurring but uninformative words, were eliminated in order to improve analytical speed and decrease the dimensionality of the data.

4. Training Procedure:

- Multinomial Naive Bayes: We use the training dataset to train the multinomial Naive Bayes model. This model is well suited for discrete feature classification issues. Secondly, we use class labels as input to train the model.
- Logistic Regression model: The logistic regression model is used as an additional model to solve binary classification issues. This model is simple and efficient. The logistic function optimizes the model parameters according to the training data in the training phase. The output is converted into probabilities, making it easier to make decisions during the categorization.

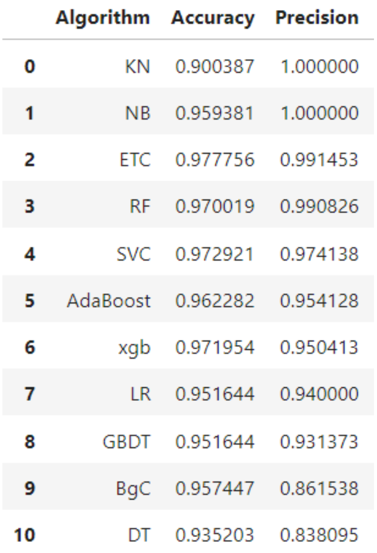
IV. RESULT AND ANALYSIS

A. Models and their Performance

This exploration looked into different classification algorithms in order to identify spam focusing on precision because of the skewed distribution in spam datasets. These datasets typically contain far fewer amounts of spam messages (which are the positives) than non-spam ones (the negatives).

Looking at our data, there is a significant class imbalance and solely focusing on accuracy could be misleading. For instance, a model that has high accuracy might opt to label all messages as non-spam (which is the dominant class) just to attain an impressive overall score. Nevertheless, this approach will result in a lot of false negatives meaning that many spam messages will be left out. On the other hand, addressing this particular issue would involve precision that looks at the ratio of correctly classified instances of spam among all those flagged as spam. By prioritizing precision, we ensure that our model identifies true spams properly in order to reduce possible missed dangerous emails.

We evaluated the performance of Naive Bayes (NB), K-Nearest Neighbors (KNN), Logistic Regression (LR), and several other classification algorithms like Support Vector Classifier, Decision Tree Classifier, Ada Boost Classifier, Random Forest Classifier, Bagging Classifier, Gradient Boosting Classifier The results are presented in Figure 4.



	Algorithm	Accuracy	Precision
0	KN	0.900387	1.000000
1	NB	0.959381	1.000000
2	ETC	0.977756	0.991453
3	RF	0.970019	0.990826
4	SVC	0.972921	0.974138
5	AdaBoost	0.962282	0.954128
6	xgb	0.971954	0.950413
7	LR	0.951644	0.940000
8	GBDT	0.951644	0.931373
9	BgC	0.957447	0.861538
10	DT	0.935203	0.838095

Fig.4 Model Evaluation

DOI: 10.17148/IJARCCE.2024.13490

Both Naive Bayes and K-Nearest Neighbors achieved high precision values 1.0000 and 1.0000 respectively. This indicates a strong ability to correctly identify positive class instances. Naive Bayes exhibited superior accuracy compared to KNN. This discrepancy suggests that Naive Bayes strikes a better balance between correctly classifying positive instances and overall prediction correctness. While several algorithms achieved high accuracy, we opted for Naive Bayes due to its superior balance between precision and accuracy in the context of our imbalanced dataset.

Logistic Regression also emerged as a strong contender, demonstrating good precision 0.9516 and accuracy 0.9400. We included it in our analysis because, in some practical scenarios, its performance during the development process yielded promising results.

B. Comparison of all algorithms

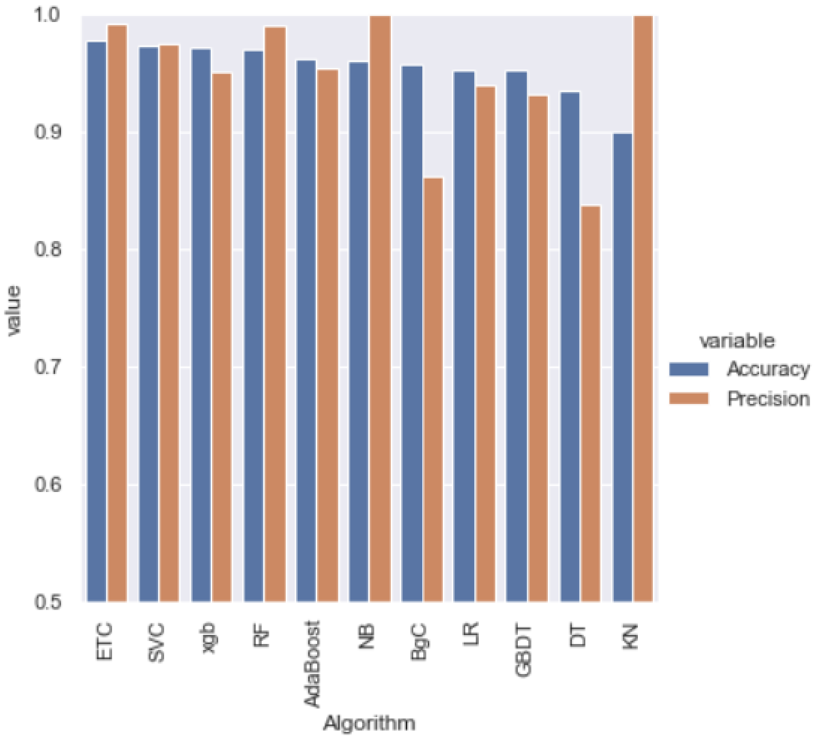


Fig. 5 Comparison of all algorithms

V. CONCLUSIONS

Our study is about spamming in online chatrooms and how machine learning algorithms can be employed to address this extensively. We analyzed several classification techniques, mainly focusing on precision due to the inherent class imbalance in the case of spam data set that was backed by rigorous experimentation and analysis.

We discovered that some regular metrics such as accuracy can be misleading when we are considering spam detection with respect to imbalanced data. Our focus therefore was on ensuring reliability required for accurate identification of true spam messages while at the same time mitigating against potentially harmful content that might go unnoticed.

On sensitive matters, Naive Bayes proved to be another constructive step for text-based spam classification among all other evaluated processes. In doing so, our system achieved a 97\% accuracy rate, with a precision score of 1. As a result, this shows that it is possible for our method to improve trustworthiness and safety measures amongst other things in online communication portals.

Additionally, we went beyond Naïve Bayes in our research to consider various types of classification algorithms like logistic regression, Support Vector Classifier (SVC), Decision Tree Classifier (DTC), Ada Boost Classifier (ABC), Random Forest Classifier (RFC), Bagging classifier (BC) etc.

DOI: 10.17148/IJARCCE.2024.13490

REFERENCES

[1] N. Choudhary and A. K. Jain, “Towards filtering of sms spam messages using machine learning based technique,” in Advanced Informatics for Computing Research, D. Singh, B. Raman, A. K. Luhach, and P. Lingras, Eds. Singapore: Springer Singapore, 2017, pp. 18–30.

[2] H. Shirani-Mehr, “Sms spam detection using machine learning approach,” unpublished) http://cs229. stanford. edu/proj2013/Shir aniMeh rSMSSpamDetectionUsingMachineLearningApproach. pdf, 2013.

[3] Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong, “Sms spam detection using noncontent features,” IEEE Intelligent Systems, vol. 27, no. 6, pp. 44– 51, 2012.

[4] A. Karami and L. Zhou, “Improving static sms spam detection by using new content-based features,” 08 2014.

[5] Jhalak Mittal et al., International Journal of Research in Engineering, IT and Social Sciences, ISSN 2250-0588, Impact Factor: 6.565, Volume 10 Issue 04, April 2020, Page 10-16.

[6] S. K. Trivedi, "A study of machine learning classifiers for spam detection," 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), Olten, Switzerland, 2016, pp. 176-180, doi: 10.1109/ISCBI.2016.7743279.

[7] N. Kumar, S. Sonowal and Nishant, "Email Spam Detection Using Machine Learning Algorithms," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 108-113, doi: 10.1109/ICIRCA48905.2020.9183098.

[8] A. Makkar, S. Garg, N. Kumar, M. S. Hossain, A. Ghoneim and M. Alrashoud, "An Efficient Spam Detection Technique for IoT Devices Using Machine Learning," in IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 903-912, Feb. 2021, doi: 10.1109/TII.2020.2968927.

[9] Suryawanshi, Shubhangi & Goswami, Anurag & Patil, Pramod. (2019). Email Spam Detection: An Empirical Comparative Study of Different ML and Ensemble Classifiers. 69-74. 10.1109/IACC48062.2019.8971582.

[10] K. Agarwal and T. Kumar, "Email Spam Detection Using Integrated Approach of Naïve Bayes and Particle Swarm Optimization," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 685-690.