Grenze International Journal of Engineering and Technology, January Issue

Email Spam Detection and Malware Filtering using Machine Learning

Kavitha H
Associate Professor, Dept. of ISE Siddaganga Institute of Technology, Tumakuru, India
Email: hkavitha@sit.ac.in

***Abstract*—Spam and malware pose a threat to Internet users as they can lead to the loss of sensitive information, financial losses, and privacy breaches. Modern spam filtering and malware detection methods are becoming less and less effective as spammers and malware authors continue to develop new techniques to bypass them. Machine learning holds great promise for these problems. This paper proposes a machine learning-based approach to detect spam and filter out malware. This method is evaluated using information about spam and non-spam emails.By comparing the results of various classification algorithms, the Naive Bayesmethod has been shown to be very effective in detecting spam and malware.**

***Index Terms*— spam, malware, machine learning, privacy breaches, classification algorithms.**

I. INTRODUCTION

An email has become an important communication tool for individuals and businesses. The proliferation of email as a means of communication has led to the proliferation of spam and malware. Spam refers to junk or unwanted email, and malware refers to software designed to harm your computer. Spam and malware can cause many problems, including loss of sensitive data, financial losses, and privacy issues. Modern methods of spam filtering and malware detection rely on rules and signatures.

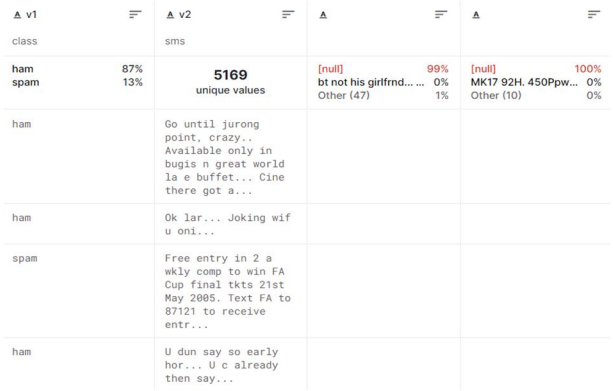
This method is becoming more ineffective as spammers and malware authors continue to develop new ways to remove them. In recent years, machine learning techniques have been used to improve the accuracy of this method. He is concerned with the problems he faces in his daily life. The purpose of this article is to present search engine-based spam detection and malware filtering as a training method for spam detection and malware filtering.

Email spam detection is the process of relating and filtering out unwanted emails. Traditional approaches to email spam discovery include rule-grounded ways and content-grounded filtering. Rule-grounded ways involve the creation of rules that identify spam emails grounded on specific characteristics, similar to the sender's address, subject line, and content. Content-grounded filtering involves as saying the content of the email to determine if it's spam. Machine learning approaches to email spam detection have been developed to ameliorate the delicacy of these ways. Supervised learning algorithms, similar to decision trees, Naive Bayes, and Support Vector Machines (SVM), are generally used for email spam detection. These algorithms are trained on a dataset of labelled emails that are moreover spam or not spam. Once the algorithm is trained, it can be used to classify new emails as either spam or not spam.

Grenze ID: 01.GIJET.11.1.539_2

II. LITERATURE SURVEY

Various related works to detect spam using machine-learning approaches have been conducted in the past. In [1], the author presents a comparative analysis of various machine learning algorithms for the classification of email spam. The authors experiment with different feature selection techniques and classifiers such as Naive Bayes, Decision Tree, Random Forest, K-Nearest Neighbour(KNN),and Support Vector Machines(SVM) to determine the best approach for spam classification. The study concludes that the Random Forest classifier outperforms other algorithms in terms of accuracy and performance in classifying spam emails. In [2], the authors combined the Term Frequency-Inverse Document Frequency (TF-IDF) technique with random forest classification and achieved 97.5% accuracy. TF-IDF is a method for measuring words in the text that analyses data using two metrics, time frequency, and text frequency variation. In [3] the author uses four machine learning methods decision tree, pure Bayesian, logistic regression, and random forest to filter spam and achieve 97% accuracy using a random forest classifier. This study concludes that although many machine learning algorithms have been used for spam classification, no single algorithm provides the best results in all situations and the choice of algorithm depends on the specific characteristics of the information and problems at hand.
In [4], the authors noted that different classifications may perform worse or better depending on the data type and task. This is done using different language processing techniques, starting with the inclusion of deep learning to understand language. Convolutional Neural Network (CNN) for sentence classification and Bidirectional Long-Term Memory (BiLSTM) for sequence recording. In [5], the authors present a representation of two transformation methods with the Transformers (BERT) language model showing the best results compared to other studies. BERT is used for word embedding, taking into account the left and right semantic content of the words, and can create different words for the same word depending on the context. In [6], the author stated that the Naive Bayesian algorithm is a fine-grained filtering that can use features to identify spam. It is an analysis method used to classify articles. The algorithm analyzes the data and then uses Naive Bayes to determine if the email is spam. This is the best way to check if an email is spam. This is a clever way to save a lot of time by specifying the entire message as a template rather than individual words. In [7], the authors talk about machine learning as a method of intelligence that allows machines to learn and evolve through experience. After using the Naive Bayes algorithm, emails are used as input for the feedback model. Results are classified as 0 for not spam and 1 for spam. In [8], the authors provide an overview of different feature selection methods, such as content-based and header-based approaches, and compare the performance of various classifiers, including Naive Bayes, Decision Trees, Support Vector Machines, and Deep Learning models. The paper also discusses the impact of different factors such as feature selection, dataset size, and imbalance on the performance of the models. In [9], the authors reviewed various techniques for detecting spam. The authors discuss various spam detection methods, including domain-based, path-based, and hybrid methods, and compare the performance of various machine learning methods. The report also addresses spam detection issues such as data inconsistencies, strategy deviation, and bias. The study concludes that AI can be effective at detecting spam with high accuracy and the choice of algorithm depends on the characteristics of the data and the problem at hand. In [10], the authors discuss the use of KNN, Naive Bayes, and Inverse (Density-Based Noise Applied Spatial Clustering) DBSCAN to classify spam based on text and images. The results show that combining text and image quality can improve classification accuracy compared to using only a single processing mode. KNN has the highest accuracy, while Naive Bayes has the lowest error rate. Reverse DBSCAN has been shown to be useful for identifying anomalies in data.



A v1 class	A v2 sms	A	A
ham 87% spam 13%	5169 unique values	[null] 99% bt not his girlfrnd... ... 0% Other (47) 1%	[null] 100% MK17 92H. 450Ppw... 0% Other (10) 0%
ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got a...		
ham	Ok lar... Joking wif u oni...		
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entr...		
ham	U dun say so early hor... U c already then say...		

Fig 1: Email corpus

A. Data Set

The spam dataset from Kaggle was used to inform our model. The "Spam.csv" file contains a set of SMS tagged timestamps for SMS spam detection. Contains 5,574 SMS messages in English that are considered malicious (legal) or spam. All lines have communication. Each column has two columns, v1 contains flags (spam or not spam) and v2 contains the original text. The corpus was collected via free Internet searches. Fig. 1 shows several sample emails in the dataset.

III. Methodology

The methodology for email spam detection using machine learning typically involves several steps which begin with collecting the data set till deploying the trained modelfor real-time email spam detection. When a new email is received, it is processed using the same pre-processing steps as the training data, and the machine learning model is used to classify the email as either spam or non-spam.

A. Data Cleaning

Data cleaning is an important step in the spam detection process, the quality of the data used to train the machine learning model directly affects its accuracy and efficiency in spam detection. Therefore, it is important to ensure that the data is properly cleaned before using it to train the model. The first step in data cleaning is to remove unnecessary or unrecognizable data. This includes deleting all emails not related to spam detection. Next is data normalization, which involves transforming data into a consistent format that machine learning models can handle effectively. Another important aspect of data cleaning is the analysis of missing or missing data. This can be done by removing data or using estimates that are consistent with other data.

B. Exploratory Data Analysis (EDA)

EDA involves exploring data to better understand its structures, patterns, and properties. An important aspect of EDA is to analyze the distribution of text in the database, ie the ratio of spam to non-spam, to block spam. Fig. 2 shows the percentage of spam and non-spam in the email corpus, showing that there are more legitimate emails than spam. This helps identify gaps in data that need to be addressed before training machine learning models. Another part of the EDA is identifying email features that best predict whether an email is a spam. This can be done by analyzing the frequency of certain words or phrases in the email, or by analyzing the email's metadata such as sender, recipient, and subject line. Fig. 3 shows a comparison of characters, words, and phrases encountered in spam and ham.

C. Data Preprocessing

Data pre-processing plays an important role in spam detection as it helps prepare data before it is fed into machine learning models. It includes tokenization, which converts text to lowercase numbers and splits the string into phrases such as words, keywords, phrases, symbols, and other things called tokens. The next is to remove special characters and drop words that don't add much information to the sentence, such as clauses, prepositions, pronouns, conjunctions, and others. Rooting is reducing a word to its root, which is added to suffixes and prefixes or roots called a lemma. Rooting is important in language comprehension (NLU) and language processing (NLP).

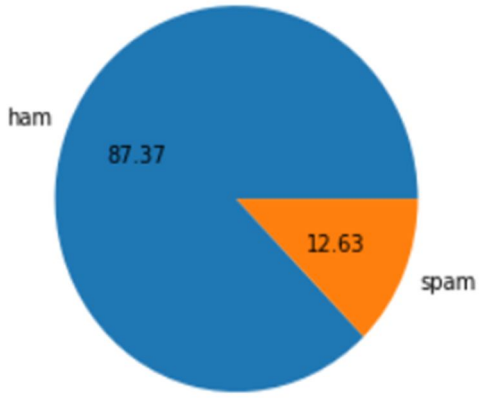


Fig 2: Pie chart for email classification

Fig 3: Comparison heatmap

D. Classic Classifiers

A classical classifier is an algorithm used to classify data into predefined groups based on features or characteristics. These are often preferred when the number of features is small and the dataset is relatively balanced. They are generally easy to use and can achieve high performance in many situations, making them an essential part of the machine learning toolbox. Some of the classifications used in this approach are given below.

Linear Regression

Linear regression is one of the simplest methods used to model the relationship between a variable and one or more variables. Fig. 4 shows the best-fit line connecting most data points. The best-fit line is found by minimizing the distance between the points and the best-fit line - this is called least squares.

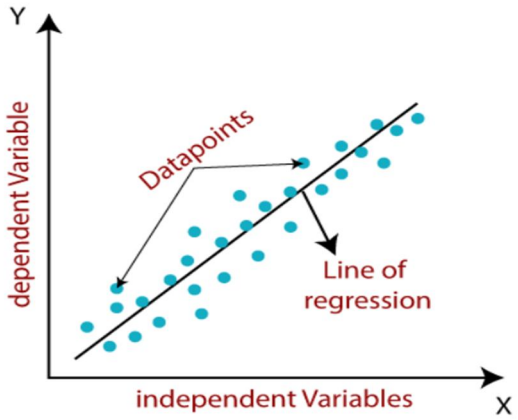


Fig 4: Linear Regression

Logistic Regression

Logistic regression is used to model the probability of independent outcomes (usually two). First, the score is calculated using an equation similar to the regression equation. The previous score is put into a sigmoid that returns the result. This result can be converted to binary output.

K-Nearest Neighbour

It is a supervised classification algorithm that divides some points and data vectors into several classes to predict the distribution of new samples. K-Neighbors is a lazy algorithm. It does not decide on its own but divides new points according to the measure of similarity which may be Euclidean distance. Euclidean distance measures Euclidean distance between the points (a, b) and (x,y) and defines who its neighbors are.

$$Dist((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2} \quad (1)$$

Naive Bayes

It is a supervised, parametric, and probabilistic learning and classification algorithm widely used for the classification of high-dimensional learning datasets. It assumes that the probability of one feature is independent of the probability of other features and is based on the principle of Bayes' theorem.

$$P(A|B) = P(B|A) * P(A) / P(B) \quad (2)$$

where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: The probability of the evidence given that the probability of a hypothesis is true.

P(A) is the Prior Probability ofthe hypothesis before observing the evidence.

P(B) is the Marginal Probability of Evidence

Support Vector Machine

A popular supervised learning algorithm for classification problems. The main solution of the support vector machine algorithm is to generate lines or decision boundaries based on the concept of decision points. Fig. 5 shows the hyperplane as the output of the new distribution model.

In two dimensions, "a plane is a line that bisects the plane, where each class occurs on one side, there may be more than one plane separating the two classes, but only one plane forms the distance between the class marginor distance.

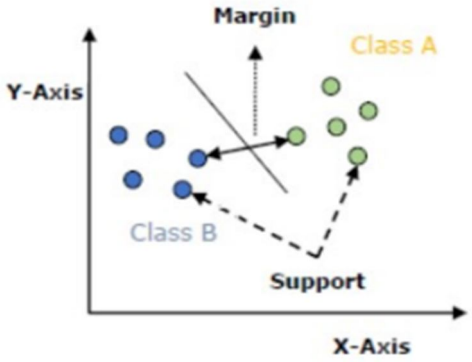


Fig 5: Support Vector Machine

Decision Tree

It is a decision support tool that uses a tree-like model of decisions and their consequences, including the consequences of events, resource costs, and energy consumption. This is a way of presenting algorithms that contain only control statements. Decision trees are often used in decision analysis to help identify the strategies most likely to achieve a goal, but they are also a popular tool in machine learning. Fig. 6 shows a simple decision tree for estimating the cost of a house, taking into account the characteristics of bathrooms and bedrooms.

Random Forest

Ensemble learning is a method that uses multiple learning algorithms together. This is to enable better performance from a single algorithm, while Bootstrap sampling is a resampling method that uses random sampling and replacement. Bagging is used to gather boot data to see if that makes any sense. A random forest classifier is a tree classifier that has different types of decision trees of different shapes and sizes.

IV. IMPLEMENTATION

The dataset is split into train and test data – 80% of the total dataset is used for training and 20% is used for testing. The implementation is carried out using Jupyter Notebook editor, PyCharm, and Streamlit for integration of the model into the website. Various Python libraries including NumPy, Pandas, Matplotlib, etc. have been used for the implementation. The cleaned email corpus after being analyzed using visualizations from Matplotlib is processed in a single function as shown in Fig. 7. Further,Countvectorizer is used to convert text to numerical data and counts the frequency of words in a sentence. Since, TF-IDF is better than Countvectorizer because it not only focuses on the frequency of words present in the corpus but also provides the importance of the words,Tfidf- MultinomialNB is chosen, and its accuracy and precision are observed.Fig. 8 and Fig. 9 show the output for sample spam and ham email.

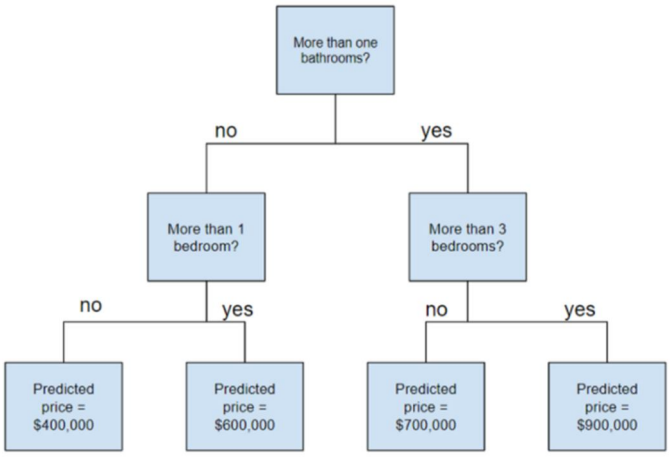


Fig 6: Decision Tree

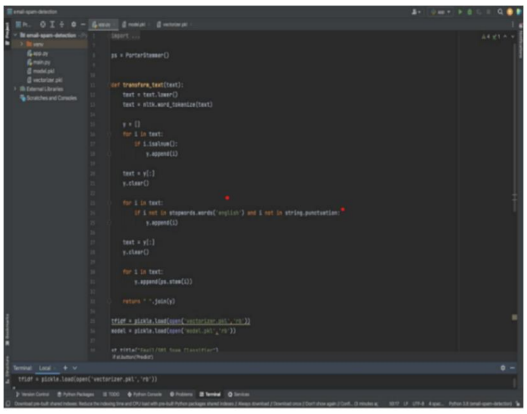


Fig 7: Transform function

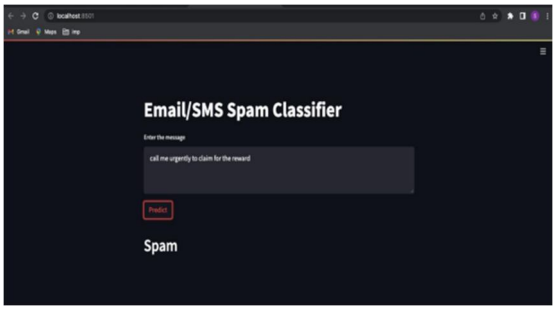


Fig 8: Spam email

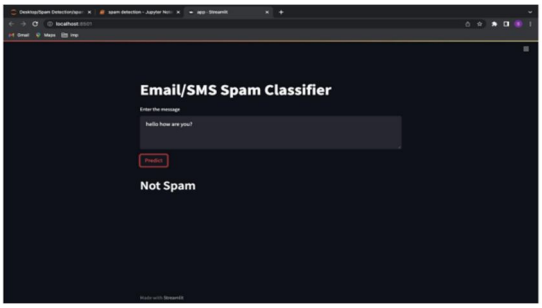


Fig 9: Ham email

V. RESULT

The results of the spam detection comparison using different machine learning algorithms are presented below. The data from emails labelled as spam or non-spam to train and test the performance of various machine learning algorithms, including random forests, pure Bayesian, support vector machines (SVM), decision tree, and K-Nearest Neighbors algorithm (KNN). Supervised machine learning can distinguish good words and classify them into the right categories. He also scored the standard and gained weight. For example, Gmail's interface uses machine learning-based algorithms to make sure users' inboxes are free of spam.

During the application process, only text (message) can be classified and graded, domain name and e-mail address cannot be classified and graded. The results show that the KNN algorithm has 90.1% accuracy, followed by Naive Bayesian with 97.1%, Random Forest with 97.5%, and Decision Tree with 93%. KNN and Naive Bayes also scored highest with 100% accuracy, followed by Random Forest with 98.2% accuracy. Comparative studies show that machine learning algorithms can detect spam effectively, and Bayes is the best algorithm in this case, considering the accuracy and accuracy metrics for the distribution of many negative names. Our findings also highlight the importance of considering different components and using multiple metrics to evaluate algorithm performance. Fig. 10 shows the metric scores for each classification algorithm in order from highest to lowest, and Fig. 11 shows the same comparison chart.

	Algorithm	Accuracy	Precision
1	KN	0.905222	1.000000
2	NB	0.971954	1.000000
5	RF	0.975822	0.982906
8	ETC	0.979691	0.975610
0	SVC	0.974855	0.974576
4	LR	0.956480	0.969697
6	AdaBoost	0.961315	0.945455
10	xgb	0.968085	0.933884
9	GBDT	0.946809	0.927835
7	BgC	0.959381	0.869231
3	DT	0.930368	0.830000

Fig 10: Result from Comparison Table

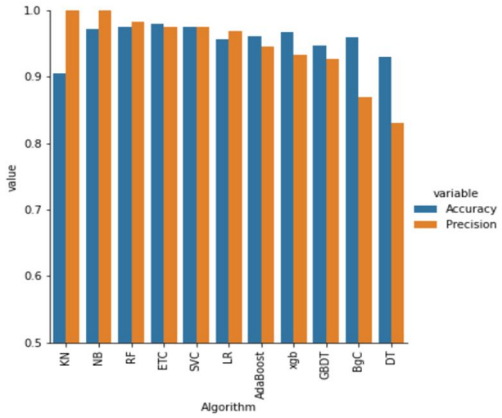


Fig 11: Result Comparison Graph

VI. Conclusion

From these results, it can be concluded that the performance of the classification system is affected by the quality of the data. Inconsistent and repetitive data may not only increase elapsed time but also reduce the accuracy of findings. As mentioned earlier, each algorithm has advantages and disadvantages. From this project, it can be concluded that machine learning algorithms are an important part of creating spam apps. To make it useful, future improvements should be made to improve the scoring of the data.This project is focused solely on filtering, analysing and distributing emails without interference and can only check and calculate the accuracy of spam. Therefore, the plan will be used to overcome the shortcomings of the current spam search.

References

[1] Vinitha, V. Sri, and D. KarthikaRenuka. "Performance Analysis of E-Mail Spam Classification using Different Machine Learning Techniques." 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE). IEEE, 2019.

[2] Sjarif, NilamNur Amir, NurulhudaFirdausMohdAzmi, SuriayatiChuprat, HaslinaMdSarkan, YazriwatiYahya, and SurianiMohd Sam. "SMS spam message detection using term frequency-inverse document frequency and random forest algorithm." Procedia Computer Science 161 (2019): 509-515.

[3] Lakshmanarao, A., K. Chandra Sekhar, and Y. Swathi. "An efficient spam classification system using ensemble machine learning algorithm." Journal of Applied Science and Computations 5, no. 9, 2018.

[4] Wang, Xiao-Lin. "Learning to classify email: a survey." In 2005 International conference on machine learning and cybernetics, vol. 9, pp. 5716-5719. IEEE, 2005.

[5] Chen, Yahui. "Convolutional neural network for sentence classification." Master's thesis, University of Waterloo, 2015.

[6] Cormack, Gordon V. "Email spam filtering: A systematic review." Foundations and Trends® in Information Retrieval 1, no. 4 (2008): 335-455.

[7] Manaranjan Pradhan and U Dinesh Kumar, "Machine Learning using Python", First Ed., Wiley, IIM Bangalore, 2019.

[8] Mansoor, R. A. Z. A., NathaliDilshaniJayasinghe, and MuhanaMagboul Ali Muslim. "A comprehensive review on email spam classification using machine learning algorithms." 2021 International Conference on Information Networking (ICOIN). IEEE, 2021.

[9] Karim, Asif, et al. "A comprehensive survey for intelligent spam email detection." IEEE Access 7 (2019): 168261-168295.

[10] Harisinghaney, Anirudh, et al. "Text and image-based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm." 2014 International Conference on ReliabilityOptimization and Information Technology (ICROIT). IEEE, 2014.