SpamML: An Efficient Framework for Detecting Spam Emails Using Machine Learning

Maged Farouk[a] , Nashwa S Ragab[a], Diaa Salama [b,c], Omnia Elrshidy[a],Yassin Ehab [a] Hana Hazem [a,] Merna Ashraf [a], Mohamed Tarek [a], Mohab Tamer [a], Abdelrahman Ahmed [a], Reda Elazab[a]

[a]Department of Business Information Systems, Faculty of Business, Alamein International University, Alamein, Egypt

[b] *Faculty of Computers Science, Misr International University, Cairo, Egypt*

[c], *Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt*

*Corresponding Author: Diaa Salama [diaa.salama@miuegypt.edu.eg]

ARTICLE DATA

Article history:
Received 08 Jan 2024
Revised 24 July 2024
Accepted 24 July 2024
Available online

Keywords:
Spam Email Prediction; Machine Learning; Classification; Naïve Bayes; Gradient Boosting; Linear Regression; K-Nearest Neighbor.

ABSTRACT

Spam detection or anti-spam techniques are methods to identify and filter out unwanted, unsolicited, or malicious emails, commonly known as spam. These techniques aim to enhance email security, reduce the risk of phishing attacks, and improve the overall user experience. The prediction of spam emails falls under the broader email filtering or classification category. Specifically, it is a part of the field of machine learning and data mining, where techniques are employed to automatically categorize emails into different classes, such as "spam" or "non-spam" (ham). This process involves using various algorithms and features to analyze emails' content, structure, and metadata to determine whether they will likely be spam or legitimate messages. Our objective is to use Machine Learning to predict and identify simplistically whether the Email is Spam Or Not. It was concluded and considered that the two datasets we can use have many Machine Learning algorithms. The proposed algorithms were tested: k-nearest Neighbor, Gradient Boosting, Random Forest, Naïve Bayes, Decision Tree, and Logistic Regression. After rigorous testing, the only algorithm, Gradiant boosting, stayed dominant in most of the testing, achieving accuracies of 98.5%; also, the other dataset with the best algorithm was Gradiant boosting, which scored the highest accuracy in all the testing, which was 98.6%. As shown in this paper, Machine Learning algorithms, such as supervised or unsupervised models, are trained on datasets containing examples of both spam and legitimate emails. These models then use the learned patterns to classify incoming emails. Can adapt to new spam patterns, effectively handling complex relationships in data.

1. Introduction

Email spamming refers to the act of distributing unsolicited messagess emails of the opposite nature are known as ham, or useful emails The word "spam" came into existence from "Shoulder Pork HAM", a canned precooked meat marketed in 19371 [1].

Machine Learning is one of the most important and valuable applications of artificial intelligence (AI), The primary purpose of machine learning algorithms is to build automated tools to access and use the data for training[2].

we considered 4 parts in the email's structure that can be used for intelligent analysis: (A) Headers Provide Routing Information, contain mail transfer agents (MTA) that provide information like email and IP address of each sender and recipient of where the email originated and what stopovers, and final destination. (B) The SMTP Envelope, containing mail exchangers' identification, originating source and destination domains\users. (C) First part of SMTP Data, containing information like from, to, date, subject appearing in most email clients (D) Second part of SMTP Data, containing email body including text content, and attachment. Based on the number the relevance of an emerging intelligent method, papers representing each method were identified, read, and summarized. Insightful findings, challenges and research problems are disclosed in this paper.

This comprehensive survey paves the way for future research endeavors addressing theoretical and empirical aspects related to intelligent spam email detection [3]. Machine learning facilitates the processing of vast quantities of data. It typically provides faster and more accurate results to detect spam or ham emails by using datasets with several algorithms(k-nearest Neighbor, Gradient Boosting, Random Forest, Naïve Bayes, Decision Tree, and Logistic Regression) [2].

Reported that in 2019, the first quarter, the threats caused by unsolicited emails used 55.97% of traffic, which is 0.07% more than the percentage in the 2018 fourth quarter. Spamming email messages is increasing by sending spoofing, phishing, and junk emails by 60%-70%[4].

The Main Contribution of this paper can be summarized as follows: Implementing an Efficient Spam Email detection program based on Machine Learning and Deep Learning models.
The two Datasets implemented in this paper Contain About 10 algorithms. In the process of using these algorithms, Cross-Validation with a number of folds =10 was used, and the two datasets were split into two partitions: 70% for training and 30% for testing.

The remaining sections in this paper are ordered as the following; related work is discussed in the Third section. Moreover, The forth section clarifies the proposed methodology of the research; it consists of dataset description and used algorithms. The results of the proposed algorithms can be found in the fifth section and their analysis. The conclusion is located in the sixth section. An acknowledgement towards all the supporting figures of this research is present in the seventh section.

3. Related Work

Spam emails detection problem has already drawn researchers' attention. Several significant works to detect spam emails have been proposed. In this section; prior related works that focus on the spam classification using ML and deep learning techniques are discussed .

In [5] aims to propose a machine learning based hybrid bagging approach by implementing the two machine learning algorithms: Naïve Bayes and J48 (decision tree) for the spam email detection. In this process, dataset is divided into different sets and given as input to each algorithm. Total three experiments are performed and the results obtained are compared in terms of precision, recall, accuracy, f-measure, true negative rate, false positive rate and false negative rate. The two experiments are performed using individual Naïve Bayes & J48 algorithms. Third experiment is the proposed SMD system implemented using hybrid bagged approach. The overall accuracy of 87.5% achieved by the hybrid bagged approach based SMD system.

In [6] getting its pace as a medium for communications used in social media platforms, websites, and emails. Spam emails are inappropriate and unwanted messages usually sent to breach security. The proposed study utilizes the existing machine learning algorithms including Naive Bayes, CNN, SVM, and LSTM to detect and categorize email content. Spam can be sent from anywhere on the planet from users having deceptive intentions that has access to the Internet. we gathered dataset from online available resources such as "kaggle" and "UCI repository" and then converted it into Urdu using Google Trans Ajax API in CSV format.

In [7] authores find that machine Learning Methods for Spam Email Classification" provides a comprehensive review of various machine learning algorithms used for spam email classification. The increasing volume of spam emails has necessitated the development of reliable anti-spam filters, and machine learning techniques have proven to be highly successful in automatically filtering spam emails.

The paper discusses popular machine learning methods such as Bayesian classification, k-NN, ANNs, SVMs, Artificial immune system, and Rough sets. These algorithms are evaluated and compared based on their performance using the Spam Assassin spam corpus.

In [8] authors find that email faces challenges like spam, phishing, and malware-infected messages. Spam, the most prevalent, involves sending similar unwanted emails to numerous users, often containing invalid data or deceptive links. The percentage of spam in email traffic reached an alarming 70.17%, with an increase in malicious attachments. Detecting spam involves various techniques, including classification algorithms like Naïve Bayes and Decision Trees, distinguishing legitimate from spam emails using collected dataset features..

In [9] The authors study introduces a pre-trained transformer model called BERT, which uses attention layers to detect spam emails from non-spam ones.The model achieves a 98.67% accuracy rate and 98.66% F1 score when compared to a baseline DNN model and classic classifiers like k-NN and NB.Natural language processing enhances the model's accuracy, and the model's persistence and robustness are tested against unseen data.Online communication, particularly through email, has become a crucial part of daily life.the rise of spam emails, unsolicited advertising, has led to the need for automatic spam detection.This technology not only improves user experience but also protects machines from potential damage and saves network resources.

In [10] author internet's widespread adoption introduced email as a primary mode of communication, but it also brought forth security challenges, notably the issue of spam – unsolicited, bulk messages exploiting the vulnerabilities of email systems. Spamming poses threats due to the ease with which senders can falsify identities, leading to widespread fraud and personal information theft. While various solutions like spam filters and anti-spam software have been developed, the persistent.

4. Methodology

4.1 Dataset description

Here, we demonstrate how we will predict and identify if there are spam emails or not through machine learning. We used Two datasets with several Algorithms.

The first and second dataset consists of 10 features. The dataset was split into two partitions: 70% for training and 30% for testing. A detailed description of the features can be found below:

TABLE 1

Description of Spam Emails

Features	Type	Value
word_freq_make	Numerical	from 0 to 4.54
word_freq_address	Numerical	from 0 to 14.3
word_freq_all	Numerical	from 0 to 5.1
word_freq_3d	Numerical	from 0 to 42.8
word_freq_our	Numerical	from 0 to 10
word_freq_over	Numerical	from 0 to 5.88
word_freq_remove	Numerical	0 to 7.27
word_freq_internet	Numerical	0 to 11.1
word_freq_order	Numerical	0 to 5.26
word_freq_mail	Numerical	0 to 18.2
word_freq_receive	Numerical	0 to 2.61
word_freq_will	Numerical	0 to 9.67
word_freq_people	Numerical	0 to 5.55
word_freq_report	Numerical	0 to 10
word_freq_addresses	Numerical	0 to 4.41
word_freq_free	Numerical	0 to 20
word_freq_business	Numerical	0 to 7.14
word_freq_email	Numerical	0 to 9.09
word_freq_you	Numerical	0 to 18.8
word_freq_credit	Numerical	0 to 18.2
word_freq_your	Numerical	0 to 11.1

word_freq_font	Numerical	0 to 17.1
word_freq_000	Numerical	0 to 5.45
word_freq_money	Numerical	0 to 12.5
word_freq_hp	Numerical	0 to 20.8
word_freq_hpl	Numerical	0 to 16.7
word_freq_george	Numerical	0 to 33.3
word_freq_650	Numerical	0 to 9.09
word_freq_lab	Numerical	0 to 14.3
word_freq_labs	Numerical	0 to 5.88
word_freq_telnet	Numerical	0 to 12.5
word_freq_857	Numerical	0 to 4.76
word_freq_data word	Numerical	0 to 18.2
_freq_415	Numerical	0 to 4.76
word_freq_85	Numerical	0 to 20
word_freq_technology	Numerical	0 to 7.69
word_freq_1999	Numerical	0 to 6.89
word_freq_parts	Numerical	0 to 8.33
word_freq_pm	Numerical	0 to 11.1
word_freq_direct	Numerical	0 to 4.76
word_freq_cs	Numerical	0 to 7
word_freq_meeting	Numerical	0 to 14.3
word_freq_original	Numerical	0 to 3.57
word_freq_project	Numerical	0 to 20
word_freq_re	Numerical	0 to 21.4
word_freq_edu	Numerical	0 to 22.1
word_freq_table	Numerical	0 to 2.17
word_freq_conference	Numerical	0 to 10
char_freq_;	Numerical	0 to 4.38
char_freq_(	Numerical	0 to 9.75
char_freq_[	Numerical	0 to 4.08
char_freq_!	Numerical	0 to 32.5
char_freq_$	Numerical	0 to 6
char_freq_hash	Numerical	0 to 19.8
capital_run_length_average	Numerical	1 to 1.1K
capital_run_length_longest	Numerical	1 to 9989
capital_run_length_total	Numerical	1 to 15.8K
Spam	classfication	0 or 1

TABLE 2
Description of Spam Mails Classification

Features	Type	Value
word_freq_make	Numerical	from 0 to 4.54
word_freq_address	Numerical	from 0 to 14.3
word_freq_all	Numerical	from 0 to 5.1
word_freq_3d	Numerical	from 0 to 42.8
word_freq_our	Numerical	from 0 to 10
word_freq_over	Numerical	from 0 to 5.88
word_freq_remove	Numerical	0 to 7.27
word_freq_internet	Numerical	0 to 11.1
word_freq_order	Numerical	0 to 5.26
word_freq_mail	Numerical	0 to 18.2
word_freq_receive	Numerical	0 to 2.61
word_freq_will	Numerical	0 to 9.67
word_freq_people	Numerical	0 to 5.55
word_freq_report	Numerical	0 to 10
word_freq_addresses	Numerical	0 to 4.41
word_freq_free	Numerical	0 to 20
word_freq_business	Numerical	0 to 7.14
word_freq_email	Numerical	0 to 9.09
word_freq_you	Numerical	0 to 18.8
word_freq_credit	Numerical	0 to 18.2
word_freq_your	Numerical	0 to 11.1
word_freq_font	Numerical	0 to 17.1
word_freq_000	Numerical	0 to 5.45
word_freq_money	Numerical	0 to 12.5
word_freq_hp	Numerical	0 to 20.8
word_freq_hpl	Numerical	0 to 16.7
word_freq_george	Numerical	0 to 33.3
word_freq_650	Numerical	0 to 9.09
word_freq_lab	Numerical	0 to 14.3
word_freq_labs	Numerical	0 to 5.88
word_freq_telnet	Numerical	0 to 12.5
word_freq_857	Numerical	0 to 4.76
word_freq_data word	Numerical	0 to 18.2
_freq_415	Numerical	0 to 4.76
word_freq_85	Numerical	0 to 20
word_freq_technology	Numerical	0 to 7.69
word_freq_1999	Numerical	0 to 6.89
word_freq_parts	Numerical	0 to 8.33
word_freq_pm	Numerical	0 to 11.1
word_freq_direct	Numerical	0 to 4.76
word_freq_cs	Numerical	0 to 7
word_freq_meeting	Numerical	0 to 14.3
word_freq_original	Numerical	0 to 3.57
word_freq_project	Numerical	0 to 20
word_freq_re	Numerical	0 to 21.4
word_freq_edu	Numerical	0 to 22.1
word_freq_table	Numerical	0 to 2.17
word_freq_conference	Numerical	0 to 10

char_freq_;	Numerical	0 to 4.38
char_freq_(	Numerical	0 to 9.75
char_freq_[	Numerical	0 to 4.08
char_freq_!	Numerical	0 to 32.5
char_freq_$	Numerical	0 to 6
char_freq_hash	Numerical	0 to 19.8
capital_run_length_average	Numerical	1 to 1.1K
capital_run_length_longest	Numerical	1 to 9989
capital_run_length_total	Numerical	1 to 15.8K
Spam	classfication	0 or 1

4.2 About Algorithms

1-Gradient Bposting

Gradient boosting is a machine learning technique that operates within a functional space based on boosting; unlike traditional boosting, which focuses on residuals, gradient boosting centers around pseudo-residuals. This method generates a prediction model comprising an ensemble of weak prediction models, commonly simple decision trees, that have minimal assumptions about the data.

2-Random Forest

It can manage datasets that include continuous variables, seen in regression scenarios, and categorical variables encountered in classification scenarios. Random Forest excels in both classification and regression tasks. In this tutorial, we will delve into the mechanics of Random Forest and apply it to a classification assignment.

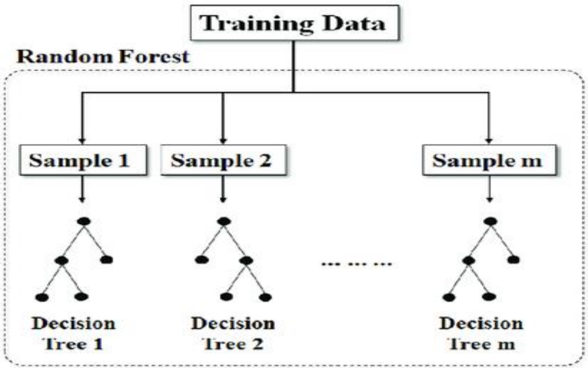


Fig1. Random Forest Algorithm

3-Neural Network

A sequence of algorithms designed to identify inherent patterns within a dataset by emulating the processes of the human brain. Neural networks, in this context, denote configurations of neurons, which can be either organic or artificial.

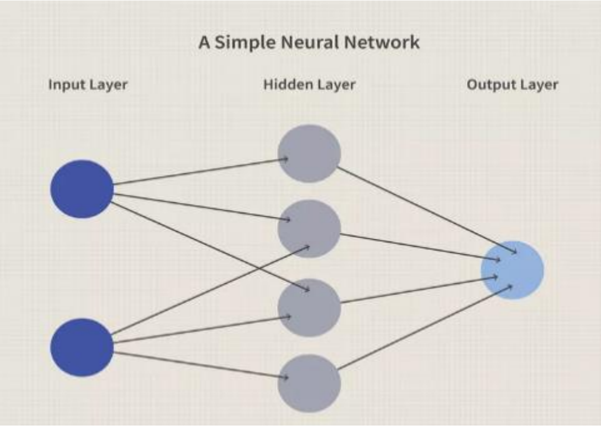


Fig2. Neural Network Algorithm

4-AdaBoost

Short for adaptive boosting, it is a versatile ensemble machine learning algorithm applicable to diverse classification and regression tasks. It operates as a supervised learning method, classifying data by amalgamating multiple weak or base learners (such as decision trees) into a robust learner.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

5-Naive Bayes

The Naive Bayes classifier is a supervised machine learning technique based on Bayes' Theorem, assuming independence among predictors. It is commonly used for tasks like text classification and is part of the generative learning algorithm family. Its key advantage is its conditional independence assumption, enabling quick and accurate predictions.

6- Stochastic Gradient Descent (SGD) is an iterative method for optimizing an objective function with suitable smoothness properties. It can be regarded as a stochastic approximation of gradient descent optimization since it replaces the actual gradient with an estimate calculated from a randomly selected subset of the data.

7-Tree: In computer science, a tree is a widely used abstract data type representing a hierarchical tree structure with a set of connected nodes. Each node in the tree can be connected to many children but must be connected to exactly one parent, except for the root node, which has no parent.

8-k-Nearest Neighbors (kNN): The kNN algorithm is a robust and intuitive machine learning method to tackle classification and regression problems. By capitalizing on similarity, kNN predicts the label or value of a new data point by considering its K closest neighbors in the training dataset.

9-Support Vector Machine (SVM): SVM is a supervised learning algorithm for classification and regression. Its main objective is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space.

10-Constant Time Algorithm: An algorithm is said to run in constant time if it requires the same amount of time regardless of the input size. For example, accessing any single element in an array takes continuous time as only one operation has to be performed to locate it.

4.3 Performance matrix

1- F1 Score: The F1 score measures a model's accuracy that considers both precision and recall, where the goal is to classify instances correctly as positive or negative. Precision measures how many predicted positive cases were actually positive, while recall measures how many actual positive instances were correctly predicted. A high precision score means the model has a low rate of false positives, while a high recall score means the model has a low rate of false negatives.

Mathematically speaking, the F1 score is a weighted harmonic mean of precision and recall. It ranges from 0 to 1, with 1 being the best possible score. The formula for the F1 score is:

F1 = 2 * (precision * recall) / (precision + recall)

$$F_{beta} = (1+\beta^2)\frac{precision * recall}{\beta^2 * precision + recall}$$

2- The harmonic mean gives more weight to low values. This means that if either precision or recall is low, the F1 score will also be low, even if the other value is high. For example, if a model has high precision but low recall, it will have a low F1 score because it is not correctly identifying all of the positive instances.

3- Accuracy is an ML metric that measures the proportion of correct predictions made by a model over the total number of predictions made. It is one of the most widely used metrics to evaluate the performance of a classification model.

Accuracy can be calculated using the following formula:

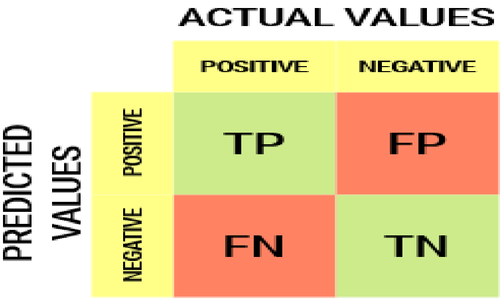
Accuracy = (number of correct predictions) / (total number of predictions)

$$ACC = \frac{Tp + Tn}{Tp + Fp + Tn + Fn}$$

4.4 Confusion Matrix

An N x N matrix is used for evaluating a classification model's performance, where N is the total number of target classes. The matrix compares the actual target values with those the machine learning model predicted. This gives us a holistic view of how well our classification model performs and what kinds of errors it makes.

For a binary classification problem, we would have a 2 x 2 matrix, as shown below, with 4 values:



		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Fig 3. Show Confusion Matrix

5. Results

Talking about results here, the results will be shown related to the two data sets we have

First Dataset

We used the following algorithms in the first table: Gradient Boosting, Random Forest, Neutral Network, AdaBoost, Naive Bayes, Stochastic Gradient Descent, Tree, kNN, SVM, and Constant. Data Shown by charts in (Figure 1) presents the difference in (AUC, CA, F1, and MCC)

Gradient Boosting and Random Forest have the top Accuracy (0.985 and 0.984), respectively. Gradient Boosting And Random Forest algorithms were the best Algorithms for Predicting spam emails; radiant Boosting and Neural networks share the same CA(Cluster Analysis), which is 0.946; here, we divide the data into similar groups with similar features to maximize the heterogeneity between clusters (groups) and the similarities between in-cluster samples.

Table 3
Cross Validation With Number of folds-10

Model	AUC	CA	F1	PRICE	RECALL	MCC
Gradient Boosting	0.985	0.946	0.946	0.946	0.946	0.886
Random Forest	0.984	0.949	0.949	0.949	0.949	0.894
Neural Network	0.982	0.946	0.946	0.946	0.946	0.887
AdaBoost	0.968	0.941	0.941	0.941	0.941	0.876
Naive Bayes	0.960	0.892	0.892	0.892	0.892	0.773
Stochastic Gradient Descent	0.920	0.925	0.925	0.925	0.925	0.843
Tree	0.921	0.927	0.927	0.927	0.927	0.847
kNN	0.873	0.810	0.809	0.809	0.810	0.600
SVM	0.689	0.606	0.607	0.657	0.606	0.260
Constant	0.499	0.606	0.457	0.367	0.606	0.000

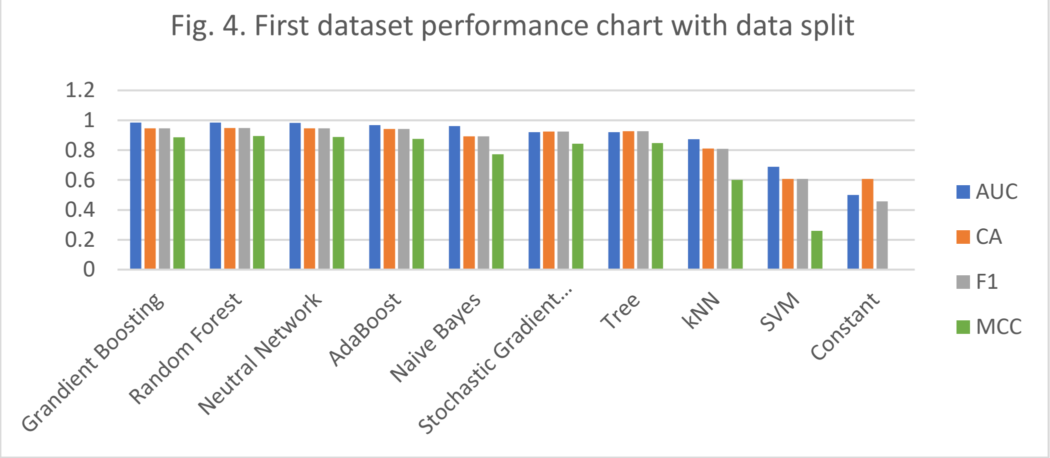


Fig4. First dataset performance chart with 10 k-fold

Second Dataset

Tree and Stochastic Gradient Descent models perform well with AUCs of 0.921 and 0.919, respectively. Random Forest and Neural Network models excel with high AUCs of 0.983 and 0.982 respectively, indicating strong predictive abilities. Naive Bayes has a decent AUC of 0.960, but it's lower than Random Forest and Neural Network. Logistic Regression and AdaBoost models perform well with AUCs of 0.972 and 0.968, respectively, and balanced precision and recall values. kNN (k-Nearest Neighbors) shows suboptimal performance with lower AUC and other metrics than the different models. The gradient Boosting model performs exceptionally well with a high AUC of 0.985, similar to Random Forest and Neural Networks. The Constant model, possibly a baseline model, has a low AUC of 0.499, indicating it's not learning anything useful. [11-12]

Table 4
Cross Validation With Number of folds-10

Model	AUC	CA	F1	PRICE	RECALL	MCC
TREE	0.921	0.927	0.927	0.927	0.927	0.847
Stochastic Gradient Descent	0.919	0.925	0.925	0.925	0.925	0.843
Random forest	0.983	0.945	0.945	0.945	0.945	0.885
Neural Network	0.982	0.946	0.946	0.946	0.946	0.887
Naive Bayes	0.960	0.892	0.892	0.892	0.892	0.773
Logistic Regression	0.972	0.928	0.928	0.928	0.928	0.848
CNN	0.873	0.810	0.809	0.809	0.810	0.600
Gradient Boosting	0.985	0.946	0.946	0.946	0.946	0.886
Constant	0.499	0.606	0.457	0.367	0.606	0.000
AdaBoost	0.968	0.941	0.941	0.941	0.941	0.876

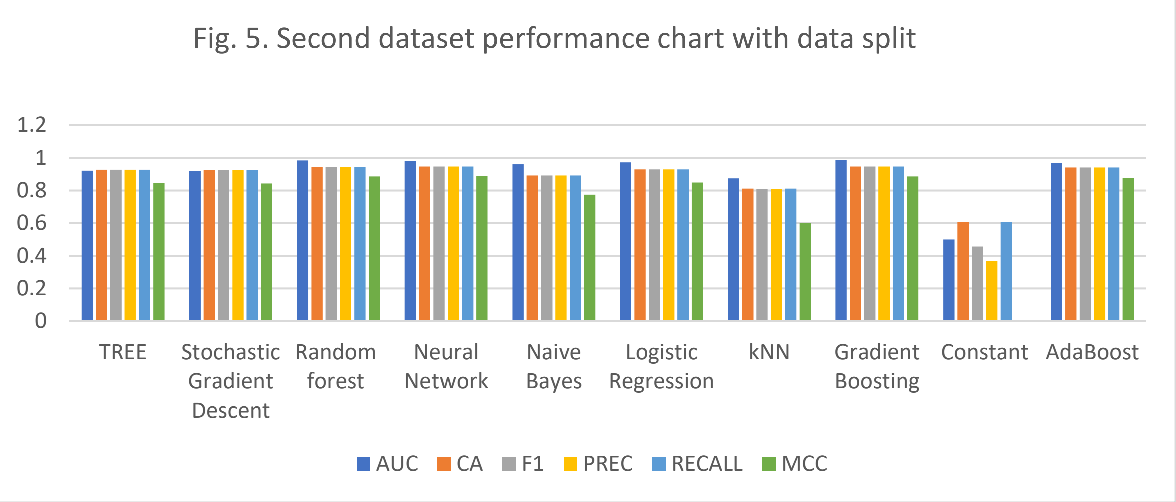
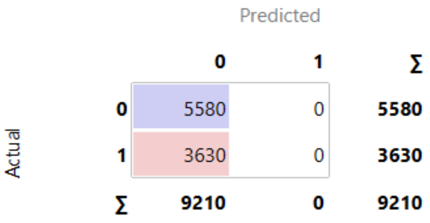


Fig5. Second dataset performance chart with 10 k-fold

5.1 Confusion matrix

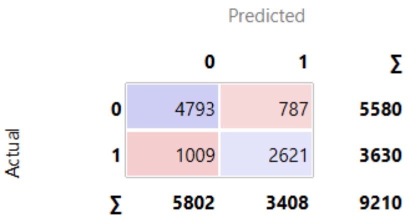
TABLE 2: Confusion Matrix

CONSTANT



		Predicted		
		0	1	Σ
Actual	0	5580	0	5580
	1	3630	0	3630
	Σ	9210	0	9210

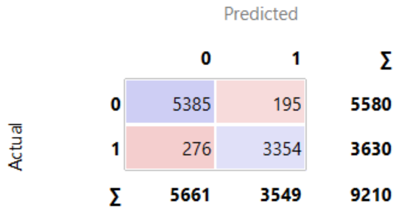
KNN



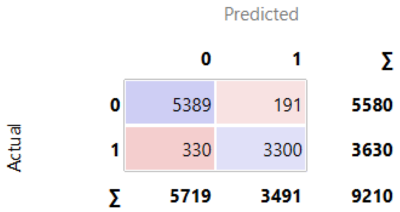
		Predicted		
		0	1	Σ
Actual	0	4793	787	5580
	1	1009	2621	3630
	Σ	5802	3408	9210

ADA Boost

Gradient Boosting

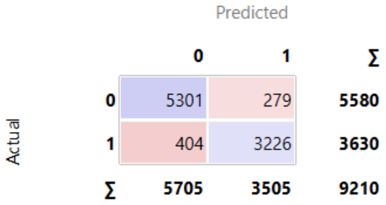


		Predicted		
		0	1	∑
Actual	0	5385	195	5580
	1	276	3354	3630
	∑	5661	3549	9210



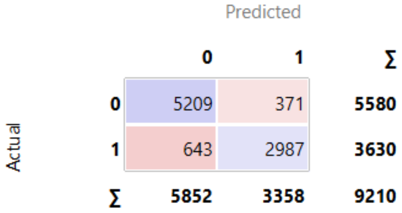
		Predicted		
		0	1	∑
Actual	0	5389	191	5580
	1	330	3300	3630
	∑	5719	3491	9210

LOGISTICS



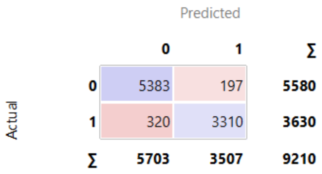
		Predicted		
		0	1	∑
Actual	0	5301	279	5580
	1	404	3226	3630
	∑	5705	3505	9210

NAÏVE BAYES



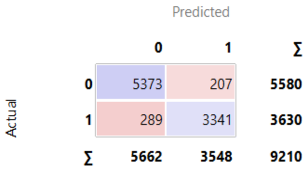
		Predicted		
		0	1	∑
Actual	0	5209	371	5580
	1	643	2987	3630
	∑	5852	3358	9210

Random Forest



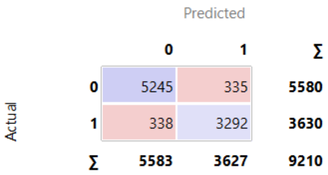
		Predicted		
		0	1	∑
Actual	0	5383	197	5580
	1	320	3310	3630
	∑	5703	3507	9210

Neural Network



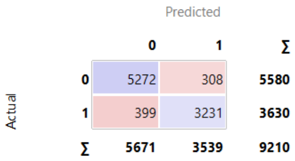
		Predicted		
		0	1	∑
Actual	0	5373	207	5580
	1	289	3341	3630
	∑	5662	3548	9210

TREE



		Predicted		
		0	1	∑
Actual	0	5245	335	5580
	1	338	3292	3630
	∑	5583	3627	9210

SGD



		Predicted		
		0	1	∑
Actual	0	5272	308	5580
	1	399	3231	3630
	∑	5671	3539	9210

6. Conclusion

This paper proposes a method for detecting and predicting the spam emails that Phone users receive. We proposed a machine learning sequence method that helps us to predict those spammed emails. We used Orange, an open-source data visualization and machine learning tool known for its user-friendly interface and versatility. In orange, we used algorithms that helped us in prediction processes and facilitate it, which are(k-nearest Neighbor, Gradient Boosting, Random Forest, Naïve Bayes, Decision Tree, Logistic Regression, Stochastic Gradient Descent, Adaboost, Support Vector Machine, Constant Time Algorithm). the only algorithm Gradient boosting stayed dominant in most of the testing achieving accuracies of 98.5%, also the other dataset the best algorithm was Gradiant boosting which scored the highest accuracy in all the testing which was 98.6%. The importance of email spam detection has grown in recent years due to the rise in spam emails and the ever-increasing issues that come with it. Differentiating spam emails from required ones is a crucial task.

References

[1] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. IEEE Access, 7, 168261-168295.Olusanya, B. O., & Newton, V. E. (2007). Global burden of childhood hearing impairment and disease control priorities for developing countries. The Lancet, 369(9569), 1314-1317.

[2] Delany, S. J., Buckley, M., & Greene, D. (2012). SMS spam filtering: Methods and data. Expert Systems with Applications, 39(10), 9899-9908.Stokoe Jr, W. C. (2005). Sign language structure: An outline of the visual communication systems of the American deaf. Journal of deaf studies and education, 10(1), 3-37.

[3] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. IEEE Access, 7, 168261-168295.Bungeroth, J., & Ney, H. (2004, May). Statistical sign language translation. In sign-lang@ LREC 2004 (pp. 105-108). European Language Resources Association (ELRA).

[4] Rayan, A. (2022). Analysis of Email Spam Detection Using a Novel Machine Learning-Based Hybrid Bagging Technique. Computational Intelligence and Neuroscience, 2022.San-Segundo, R., Barra, R., Córdoba, R., D'Haro, L. F., Fernández, F., Ferreiros, J., ... & Pardo, J. M. (2008). Speech-to-sign language translation system for Spanish. Speech Communication, 50(11-12), 1009-1020.

[5] Sharma, P., & Bhardwaj, U. (2018). Machine Learning-based Spam Email Detection—International Journal of Intelligent Engineering & Systems, 11(3).

[6] Siddique, Z. B., Khan, M. A., Din, I. U., Almogren, A., Mohiuddin, I., & Nazir, S. (2021). Machine learning-based detection of spam emails. Scientific Programming, 2021, 1-11Arvanitis, N., Constantinopoulos, C., & Kosmopoulos, D. (2019, November). Translation of sign language glosses to text using sequence-to-sequence attention models. In 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 296-302). IEEE.

[7] Awad, W. A., & ELseuofi, S. M. (2011). Machine learning methods for spam email classification. International Journal of Computer Science & Information Technology (IJCSIT), 3(1), 173-184.Stoll, S., Camgoz, N. C., Hadfield, S., & Bowden, R. (2020). Text2Sign: towards sign language production using neural machine translation and generative adversarial networks. International Journal of Computer Vision, 128(4), 891-908.

[8] Sarju, S., & Thomas, R. (2014). Spam email detection using structural features. International Journal of Computer Applications, 89(3).

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[10] Yüksel, A., Çankaya, Ş., & Üncü, I. (2017). Design of a machine learning-based predictive analytics system for spam problems. Acta Physica Polonica A, 132(3), 500-504.

[11] Hair Jr, J., Black, W., Babin, B. and Anderson, R., Multivariate data analysis, Seventh Ed., Pearson, Harlow, UK, 2014.Klima, E. S., & Bellugi, U. (1979). The signs of language. Harvard University Press.

[12] Fraley, C. and Raftery, A., How many clusters?. Which clustering method? Answers via model-based cluster analysis. The Computer Journal, 41(8), pp. 578-588, 1998. DOI: 10.1093/comjnl/41.8.578Moryossef, A., Yin, K., Neubig, G., & Goldberg, Y. (2021). Data augmentation for sign language gloss translation. arXiv preprint arXiv:2105.07476.

[13] Li, X. M., & Kim, U. M. (2012, June). A hierarchical framework for content-based image spam filtering. In 8th International Conference on Information Science and Digital Content Technology (ICIDT) (pp. 149–155). Jeju

[14] Dada, E. G., Bassi, J. S., Chiroma, H., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. Heliyon, 5(6).

[15] Christina, V., Karpagavalli, S., & Suganya, G. (2010). Email spam filtering using supervised machine learning techniques. International Journal on Computer Science and Engineering (IJCSE), 2(09), 3126-3129.

[16] El Naqa, I., & Murphy, M. J. (2015). What is machine learning? (pp. 3-11). Springer International Publishing.

[17] Cruz, J. A., & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. Cancer informatics, 2, 117693510600200030.