Detection and Classification of Legitimate and Spam Emails using K-Nearest Neighbor Augmented with Quadratic Sieve Algorithm

Jumoke Soyemi
Dept of Computer Science
The Federal Polytechnic, Ilaro

Mudasiru Hammed
Dept of Computer Science
The Federal Polytechnic, Ilaro

ABSTRACT

Spam in emails is a major challenge that is inherent in today's internet as it endangers financial institutions and poses a threat to individual users. Various techniques have been proposed by different studies to prevent spam in emails; however, classification and filtering technique using machine intelligence methods are the most efficient among the several methods. This study employed a K-Nearest Neighbor (KNN) augmented with the Quadratic Sieve algorithm to detect and classify legitimate emails and spam. The sieve algorithm revealed all the prime numbers for all the dataset used, starting from the input dataset to reduce the errors that may cause an imbalance in the classification. The result from this study shows that implementation of KNN augmented with Quadratic Sieve algorithm detects and properly classifies legitimate e-mail as well as spam much better.

Keywords

Spam email, K-Nearest Neighbor, Cross-Validation, Quadratic Sieve Algorithm

1. INTRODUCTION

Electronic email is one of the most efficient and powerful modes of communication [1] because it is effective and inexpensive. Many companies and government parastatals engage E-mail as a dominant form of inter and intra-organizational written communication forming an essential part of life in the same manner as mobile phones [1]. The enormous popularity of email, its simplicity and ease of use, however, attracted spammers with lots of unwanted emails [2]. Spam emails are unsolicited messages found in emails without the consent of the receiver [3]. Majority of spams are sent with the mission to sell products and services, and these spams work because many people respond to them; meanwhile, it costs the sender virtually nothing [3]. The spams are becoming big trouble for the recipients because storages spaces are being wasted.

Filtering and classification are the most commonly adopted methods by some of these machine intelligence approaches where the systems identify whether a message is a spam or non-spam based on the message content and some other characteristics of the message [4]. Some of these techniques are neural network (NN), Optimization Techniques such as Genetic Algorithm (GA), Support Vector Machine (SVM), K-means (KM), K-nearest neighbor (KNN), and Naïve Bayes (NB). However, "in the presence of a significant overlapping, the task of learning from imbalanced data can be a very difficult problem" [5]. The class imbalance problem is found in areas such as machine learning and pattern recognition [5]. Therefore, due to the increase in spams, detection techniques to reduce or minimize the spams are required.

Data mining with meta-classifiers stacking, using different classifiers for training, testing, and filtering for data preprocessing and feature selection was carried out in study [6]. Although the study reported improvement in email spam detection compared to the single classifier approach, there was no laid down procedure on how the system detects imbalance in a two-class data set, that is, a minority class and a majority class. Study [7] used K- nearest neighbor for data classification, and the study presented various output with various distances which were used in the algorithm. Nevertheless, the system was inefficient in defining the clusters at initialization time. Study [1] used clustering and association rules generated by the Apriori algorithm for detecting spam in email messages. The results obtained was okay as a result of the K-means techniques implemented, however, the system could not perform maximally because clusters were declared at initialization time and a small value of K may lead to a large variance in classifications while setting K to a large value may lead to errors in the model.

Study [8], utilized content-based features algorithm to extract both "link features and content for spam filtering pages with colony optimization method for detection and the topology of the web-graph". However, this study used traditional indexing methods that quickened the value of K, and the right value of K is essential for good classification. A small value of K will lead to a large variance in classifications while setting K to a large value may lead to errors in the model, as mentioned earlier. Also, [9] worked on "spam detection using clustering, random forests, and active learning". The method allowed the labelling of sample messages for learning, a spam detection model using the random forest for classification and active learning for refining the classification model. However, error in the classification occurred each time the test dataset and training dataset have some features in common.

This study employed a k-nearest neighbor (KNN) augmented with integer factorization algorithm (Quadratic sieve) to drastically reduce spam in email messages. An integer factorization algorithm is a self-checking algorithm that can verify that the product of the outputted factors equals the original integer input. This algorithm will detect any errors or overlapping that may cause imbalance and reduces failure that may occur in the system when detecting spam in the email messages.

2. MATERIALS AND METHODS

This study used an average number of characters per length of the email message, which is limited to 78 characters features to identify legitimate email messages. Other features are insertion of a shortened URL in the email messages, number of URLs that contain IP addresses and if dots are used appropriately in the domain of URL. Feature extractor scans through the email to identify the above features to classify whether the email is legitimate or spam. The modified K-NN algorithm used in this study is depicted in algorithm 1.

2.1 Algorithm: A Modified K-NN Algorithm for E-mail Spam Detection and Classification

Stage A. Training

Step 1: If the average number of characters per word in the email text is more than 78 character

Step 2: If number of URLs contain IP addresses

Step 3: If number of dots in the domain of any URL is more than the maximum number required.

Step 4: If total number of words in the text is more than the number specified by legitimate e-mail

Step 5: If number of hyperlinks is more than the required number in standard e-mail

Step 6: If number of words in the title of the pages exceed specified number of pages

Step 7: If the content of the email is found to be redundant

Stage B. Filtering

Step 1: Determine values for the above training features set. Any values outside the training features values, classify given message as spam. Otherwise classify it as legitimate mail

Step 2: If values of messages are outside of the training features, classify as spam, otherwise classify it as legitimate mail.

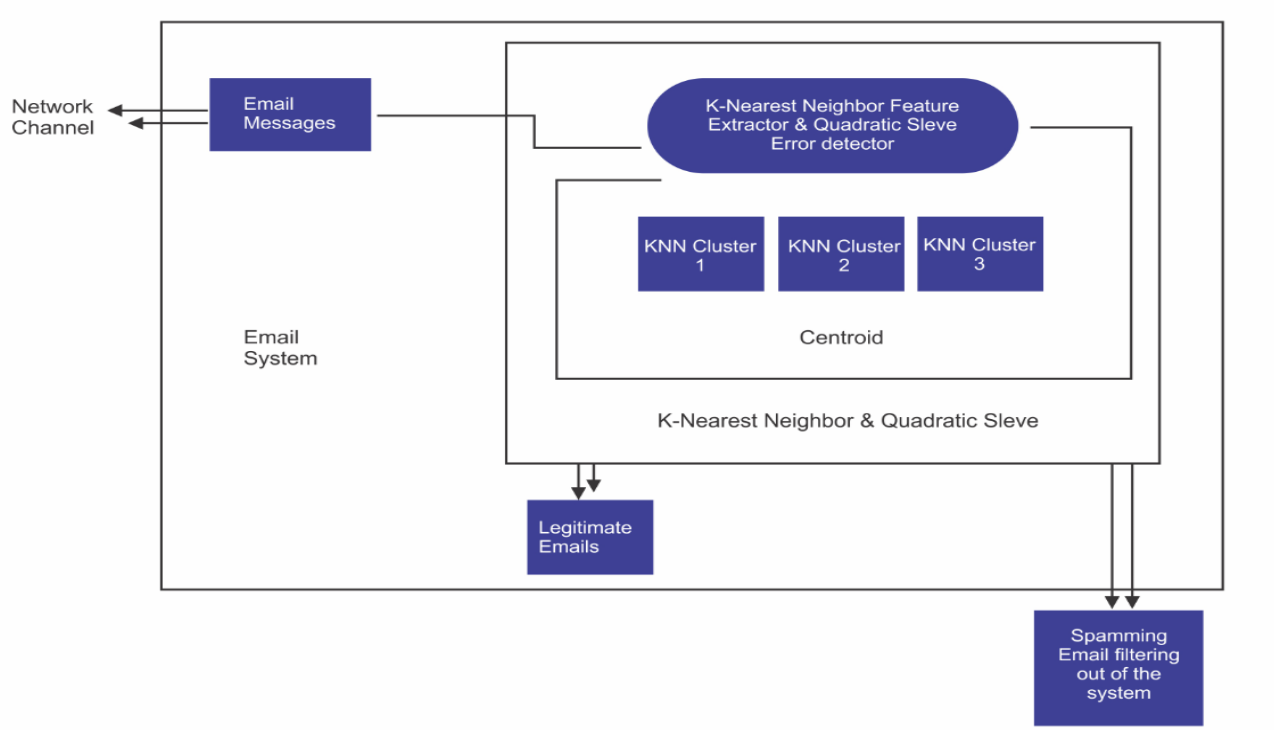

Network Channel
Email Messages
K-Nearest Neighbor Feature Extractor & Quadratic Sleve Error detector
KNN Cluster 1
KNN Cluster 2
KNN Cluster 3
Centroid
Email System
K-Nearest Neighbor & Quadratic Sleve
Legitimate Emails
Spamming Email filtering out of the system

Fig 1: Architecture of Spam Detection System

2.2 E-mail spam detection and classification with K-nearest neighbor

K-nearest neighbor method search the set of given training features and determine the query point to be X. In each feature, it locates the closest points $(x_1, x_2 x_3, ..., x_n)$ to the query point X. The outcome of $(x_1, x_2 x_3, ..., x_n)$ which is the nearest to the query point X will be taken as $(y_1, y_2, y_3, ..., y_n)$) and the outcome of X then taken to be Y. The KNN predicts outcome Y of the query point X to be the average of the outcomes of its KNN. That is, in the study e-mail messages are taken to be X, the KNN algorithm determines its k nearest neighbors among the messages in the training set. If there are more spams among these neighbors, such messages are classified as spam, otherwise classified as legitimate e-mail messages. The proposed system used Euclidean distance to calculate similarities between the features of the test features and corresponding features of each instance in training set to get spam messages in each neighbor. The Euclidian distance is a metrics distance that is expressed in equation 1.

2.3 Distance Matric

Accurate predictions with KNN are made to determine a metric to measure the distance between the query point and cases from the e-mail training samples using Euclidean distance in equation 1.

$$d = \sum_{i=1}^{x} (x - p)^2 \qquad (1)$$

Where x is the query point

and p are the cases from the email training samples.

Standardization of the e-mail features for distance calculation was done in this study so that KNN could make an accurate prediction. Since the features are greatly varying in value ranges, then using the features directly in distance metrics would effectively give more weight to features with larger values.

2.4 Distance Weight

K-nearest neighbor predictions are based on an intuitive approach of the objects that are close in the distance, which is potentially similar. Each e-mail feature is evaluated, and a weight is assigned to it based on how useful the feature is to distinguish the classes of the dataset. The identified KNN takes into account the distance of each neighbor and forms a structural density when a score is produced for each of the neighbors. The scores for each class are averaged to produce one classification score for each one of the classes. The class yielding the highest classification score is selected for the classification; equation 2 was used to calculate the weight distance.

$$W = \sum_{i=1}^{k} w(\mathrm{x}, \mathrm{p}) = 1 \qquad (2)$$

A test feature is compared to the training set; once the test features are found in the training set, the algorithm will, therefore, predict legitimate e-mail message. But if they are not found, the prediction is made to be illegitimate e-mail message, and those test features that are nearest or closest to the training set are likely also to be legitimate e-mail message. Mathematically, K-nearest neighbor predictions is the average of the K-nearest neighbor outcome, which shown in equation 3

2.5 Cross Validation

This study evaluates the performance stability of the KNN, by dividing the data into training and testing data and calculation was done to check the e-mail messages that were correctly classified. It was discovered that 10.5% of the dataset was not accurately classified because of over-fitting and overlapping in the dataset. That is, a two-class data set is imbalanced because one of the classes, which in the minority, one is heavily underrepresented in comparison to the other class. This error may lead to misclassification of the KNN, which is an imbalance in the proposed system. Quadratic Sieves algorithm that has the capability of self-checking was used to detect errors that may cause imbalances in the KNN for proper detection and classification. This study also computed the nearest neighbors of a particular test email and discarded all neighbors that belong to the same account, which is grouped as the test features.

Furthermore, for any given problem, a small value of K will lead to a large variance in classifications. Alternatively, setting K to a large value may lead to errors in the model. Although, cross-validation technique can estimate the optimal value of K that may likely reduce error in the model. However, this study drastically reduces the errors that cause an imbalance in the KNN classifications using additional techniques which is Quadratic Sieves algorithm that has the capability of self-checking to detect errors that cause imbalances in the classification. This algorithm is capable of handling a very large volume of dataset, and it can also estimate a large number of K which yields accurate classifications of KNN.

2.6 Quadratic Sieves algorithm for detecting errors in e-mail spam detection and classification

The study [10] used Quadratic Sieves algorithm as a factorization algorithm to detects errors that cause imbalances in the model. This algorithm arrives at final classification by filtering out unwanted input data from a larger starting set of the input dataset. This reduced the two-class data set, that is one of the classes, which is a minority and is heavily underrepresented in comparison to the other class that is the majority one (overlapping). The principle is to treat the given integer, N, as the product of two numbers X and Y, where N=X*Y, or Y=N/X. The algorithm tracks the Y=X/N curve in unity steps and stops when X*Y=N. The algorithm uses the square root of N as a starting point to check email dataset.

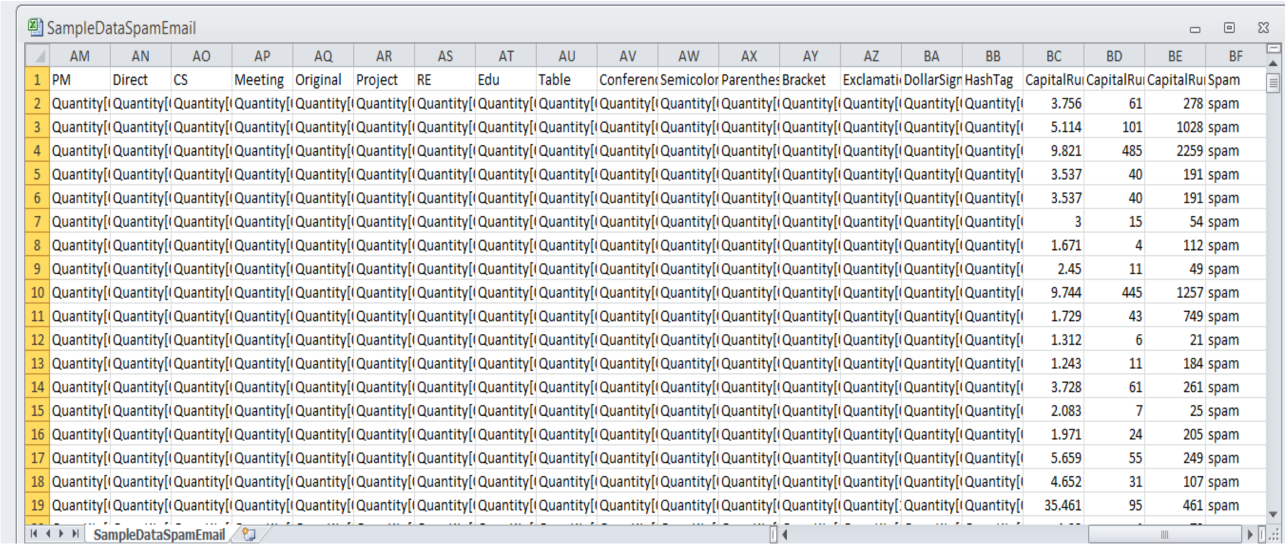
The sieve algorithm 4, was developed for prime numbers which effectively cross out all composite numbers leaving only the primes. The sieve found all of the prime numbers for all the dataset used starting from the input dataset. Finding the prime numbers also eliminated variance in the classification when K is set to a large value. The algorithm ensures that input dataset yield correct output. This strongly allows the system to detect errors that may cause an imbalance in the system detection and classification.

3. RESULT AND DISCUSSION

This study used python 3.5.0 version with natural language processing techniques to implement the e-mail spam detection and classification system. The natural language processing techniques which are a machine learning technique allow the system to interpret emails' features used to be able to predict whether email message contains spam or not and the system was tested on Windows 8.1 operating system.

In the proposed system, the network channel allowed the passage of any e-mail messages sent by several users and the proposed system was installed on different users' machine rather than e-mail server for proper classification. Whenever e-mail messages are sent, and immediately they were delivered by the mail transport agent to the user's machine. The proposed system used intelligence based on training samples to mark legitimate messages and those that are spam and the unwanted messages are filtered out from the system before they are finally delivered to the user's machine. It was revealed in the literature that overlapping has a nagging problem to some of the studies that used KNN to classify e-mail spam. The overlapping did not occur in this study when implemented and the system adequately detected and classified spam and spam in an e-mail. In testing the system developed, 5000 email dataset were pulled from UCI repository to test the system and the total time taken the system to predict spam or non-spam (total runtime) were recorded in cell column (BE), and the predictions spam or non-spam can be found in the cell (BF) as shown in Table 1 and Table 1.

Table 1. Classification of Spam Email



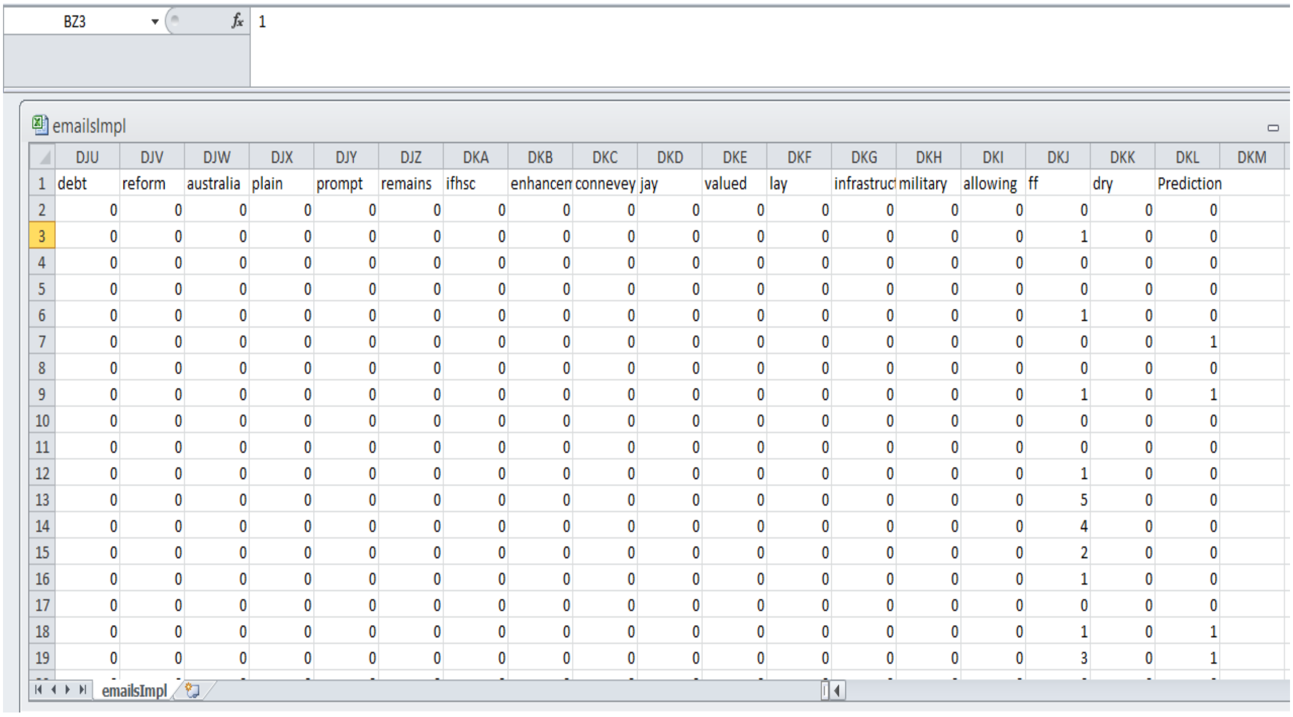
SampleDataSpamEmail

	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF
1	PM	Direct	CS	Meeting	Original	Project	RE	Edu	Table	Conferen	Semicolor	Parenthes	Bracket	Exclamati	DollarSign	HashTag	CapitalRu	CapitalRu	CapitalRu	Spam
2	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	3.756	61	278	spam
3	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	5.114	101	1028	spam
4	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	9.821	485	2259	spam
5	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	3.537	40	191	spam
6	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	3.537	40	191	spam
7	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	3	15	54	spam
8	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	1.671	4	112	spam
9	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	2.45	11	49	spam
10	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	9.744	445	1257	spam
11	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	1.729	43	749	spam
12	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	1.312	6	21	spam
13	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	1.243	11	184	spam
14	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	3.728	61	261	spam
15	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	2.083	7	25	spam
16	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	1.971	24	205	spam
17	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	5.659	55	249	spam
18	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	4.652	31	107	spam
19	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	Quantity[	35.461	95	461	spam

SampleDataSpamEmail

For experimental testing, a secondary dataset of email messages was downloaded from UCI machine learning repository which contains legitimate and email spam. The dataset was trained and normalized; the test features of dataset were extracted during the training which was used to model the classification of spam and non-spam email messages. The test features were tested with email messages that are spam and they were predicted as a spam as shown in Table 1, the average time and the total time taken the system to predict that a message contains spam were recorded. Table 1 depicts email messages that contain spam and they were captured where spam predictions were densely populated. Set of test features were also tested with email messages that are non-spam and they were predicted as a non-spam message. The system implementation results were converted to CSV file format to better organize the large amounts of dataset used. Quadratic sieve used the keyword "Quality" in each cell as it is shown in table 1 to shows that two-class data set were not misrepresented. The numbers of spams in each cell were recorded. KNN searches for numbers of spams in each cell and the average was determined and the highest score is selected for further predictions whether the email message is a spam or non-spam. This study used zero (0) to denote email messages that do not contain spam while one (1) denote email messages that contain spam as it is shown in cell column DKL in Table 2. Misrepresentations were not possible when dataset were refining, and the system was accurately classified as spam and non-spam.

Table 2. Screenshot of E-mail Spam Predictions



BZ3 *fx* 1

emailsImpl

	DJU	DJV	DJW	DJX	DJY	DJZ	DKA	DKB	DKC	DKD	DKE	DKF	DKG	DKH	DKI	DKJ	DKK	DKL	DKM
1	debt	reform	australia	plain	prompt	remains	ifhsc	enhancem	connevey	jay	valued	lay	infrastruct	military	allowing	ff	dry	Prediction	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	

emailsImpl

4. CONCLUSION

This study concluded that e-mail is a powerful tool to share information and serves as an attractive platform. But the major challenge which is spam can be detected and classified with machine intelligence which is considered as the most efficient. This study adopted a KNN algorithm augmented with Quadratic Sieve algorithm that could detect errors, especially errors that can lead to misclassification when the proposed system is classifying legitimate and spam messages. The technique used in this study enhances detection and classification capabilities of machine learning to accurately detect, identify and filter spam in e-mail messages. The system is efficient with a high degree of accuracy in email spam detection, and classification with KNN augmented with Quadratic Sieve algorithm.

5. REFERENCES

[1] Basavaraju, M. Prabhakar, R.A. 2010. Novel Method of Spam Mail Detection using Text Based Clustering Approach. International Journal of Computer Applications. 5(4), 15 – 25.

[2] Sahil, P. Dishant, G. Mehak, A. Ishita, K. and Nishtha, J. 2013. Comparison and Analysis of Spam Detection Algorithms. International Journal of Application or Innovation in Engineering & Management (IJAIEM). 2 (4), 1 – 7.

[3] Garcia, V. Mollineda, R.A. and Sa´nchez, J.S. 2008. On the k-NN performance in a challenging scenario of imbalance and overlapping, Springer-Verlag London Limited, United Kingdon, 269–280.

[4] Man, Q. and Mousoli. R. 2010. Semantic analysis for spam filtering. In: Proceedings of the Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD).

[5] Qi, M. and Mousoli, R. 2010. Semantic analysis for spam filtering. In 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, pp. 2914-2917, IEEE.

[6] ZhiWei M. Singh M.M.and Zaaba, Z.F. 2017. Email spam detection: a method of metaclassifiers stacking. In The 6th international conference on computing and informatics, pp. 750-757.

[7] Aman, K. and Singh, M.D. 2013. A review of data classification using K-Nearest Neighbour algorithm. International Journal of Emerging Technology and Advanced Engineering. 3(6), 354 – 360.

[8] Rungsawang, A. Taweesiriwate, A. and Manaskasemsak, B. 2011. Spam Host Detection using Ant Colony Optimization, in IT Convergence and Services. Springer, pp. 13-21.

[9] Dave, D. and Harry, W. 2009. Spam detection using clustering, random forests, and active learning. CEAS 2009 – Sixth Conference on Email and Anti-Spam, July 16-17, 2009, Mountain View, California USA.

[10] ZhiWei M. Singh M.M. and Zaaba Z.F. 2017. Email spam detection: a method of metaclassifiers stacking. In The 6th international conference on computing and informatics, pp. 750-757.