ARTICLE TEMPLATE

Content-based Spam Email Detection Using N-gram Machine Learning Approach

Nusrat Jahan Euna.[a] and Syed Md. Minhaz Hossain[a,b] and Md. Musfique Anwar[c] and Iqbal H. Sarker[a]

[a]Department of Computer Science & Engineering, Chittagong University of Engineering & Technology , Chittagong 4349, Bangladesh; [b]Premier University, Chittagong-4000, Bangladesh; [c]Jahangirnagar University, Dhaka, Bangladesh

ARTICLE HISTORY
Compiled September 13, 2021

ABSTRACT

Recently, spam emails have become a significant problem with the expanding usage of the Internet. It is to some extend obvious to filter emails. A spam filter is a system that detects undesired and malicious emails and blocks them from getting into the users' inboxes. Spam filters check emails for something "suspicious" in terms of text, email address, header, attachments, and language. However, we have used different features such as word2vec, word n-grams, character n-grams, and a combination of variable length n-grams for comparative analysis in our proposed approach. Different machine learning models such as support vector machine (SVM), decision tree (DT), logistic regression (LR), and multinomial naïve bayes (MNB) are applied to train the extracted features. We use different evaluation metrics such as precision, recall, f1-score, and accuracy to evaluate the experimental results. Among them, SVM provides 97.6 % of accuracy, 98.8% of precision, and 94.9% of f1-score using a combination of n-gram features.

KEYWORDS
Spam Detection; Feature extraction; N-grams; Machine Learning.

1. Introduction

In recent years, web security is becoming one of the most critical issues. Most of our daily services start using the internet, mobile computing, and electronic media. Email is one of the mediums to communicate and increases in volume with the increasing use of the internet. Spamming is one of the most straightforward attacks in email messaging. Besides, users frequently receive annoying spam messages and malicious phishing messages by subscribing to different websites, products, services, catalogs, newsletters, and other types of electronic communications (1; 8). In some cases, spam email is produced by mass-mailing viruses or Trojan horses. According to the China Anti-Spam Alliance's new survey on data, a typical Internet user receives 35 emails per week on average among which, 41% of emails are spam emails. The presence of such spam messages wastes time as well as bandwidth on internet connections. Furthermore, they are often associated with offensive content and spread computer viruses. Due to

Iqbal H. Sarker. Email: iqbal@cuet.ac.bd

these obligations, cyber specialists are devoted to developing accurate spam detection in digital communication.

Moreover, there are many solutions to filter spam, e.g., the blacklist and white-list filtering techniques, decision tree based approaches, email address based approaches, and machine learning based methods. The majority of them rely heavily on text analysis of the content of an email. As a result, there is a growing demand for effective anti-spam filters that automatically identify and remove spam messages or alert users to possible spam messages. However, spammers always investigate the loopholes of existing spam filtering techniques. They have introduced a new design for spreading spam emails in a wide range. Therefore, the existing system does not function against them. Tokenization attack sometimes misleads spam filtering by adding extra spaces. Therefore, email contents are needed to be structured (14). Moreover, inspite of having the highest accuracy in machine learning based spam email detection(3; 17), false positive (precision of 92.9%) is an issue due to one-shot detection of email threats. Addressing the false positive issues and changes in various attack design, the stop words and other unwanted information are removed from the texts for further analysis in our proposed approach. After pre-processing, these texts go through numerous feature extraction methods, such as word2vec, word n-gram, character n-gram, and a combination of variable length n-gram. Different machine learning techniques such as support vector machine (SVM), decision tree (DT), logistic regression (LR), and multinomial naïve bayes (MNB) (15) are applied on these matrices to perform the classification of the emails. The primary contributions of this paper are the following:

(i) to create a content-based spam filter that can classify spam and ham e-mails.
(ii) to analyze numerous feature extraction methods, such as word2vec, word n-gram, character n-gram, and combination of variable length n-gram.
(iii) to evaluate the performances of numerous experiments and achieve the best performance using support vector machine (SVM), decision tree (DT),logistic regression (LR) and multinomial naive bayes (MNB) with proper features.

The rest of the paper is organised as follows. Section 2 discusses the related works; proposed approach for detecting e-mail spam is presented in Section 3; results and analysis are demonstrated in Section 4; and finally, the paper is concluded in Section 5.

2. Related works

Liu et al. (2) proposed a spam filtering method for emails based on their content. Their proposed technique is divided into two phases: training and classification. The extracted keywords from individual users' emails are compared against a spam and ham keywords corpus and achieved an overall accuracy of 92.8% and precision of 84.6% . Gaurav et al. (3) suggested a spam mail detection system for detecting spam emails based on the document labeling concept and applied three algorithms such as Multinomial naïve bayes, Decision Tree, and Random Forest for email classification. The Random Forest technique achieved the highest accuracy of 92.97% and precision of 92.9% among these classifiers. Ioannis Kanar et al. (4) introduced a low-level data-based spam detection approach. Instead of using the 'bag of words' approach to extract features, they employed character n-grams to create a 'bag of character n-grams'. Kiliroor et al. (5) suggested a model for detecting unwanted or unsolicited messages from the users' walls on online social networking sites, which had an accuracy of

91.18%. Weimiao Feng et al. (6) proposed a method based on SVM-NB. The SVM method is used to split training samples into different groups and to find dependent training samples. Moon et al. (7) proposed a spam mail filtering system based on n-gram indexing to support vector machines. They practiced with emails obtained from various users and performed the filtering procedure with SVM classifier. Kaur et al. (8) proposed a spam detection technique using N-gram analysis and machine learning techniques. The N-grams that are built are used to predict unlabeled data.

Ahmad et al. (9) proposed a method achieving 96% of accuracy, in which an optimal subset of features is chosen for the learning process and support vector classifier is used to classify. Sarker et al. (16) performed an effectiveness analysis of machine learning security modeling with optimal features on a broad scale. Nayak et al. (10) proposed a method for spam email detection, which employs a hybrid bagging approach as feature and combined the Naive Bayes and Decision Tree machine learning algorithms as classifiers which achieve overall 88.12% of accuracy. Sheu et al.(11) proposed a method concentrating on email header analysis using a decision tree classifier to search for spam association guidelines at first. Next, an effective systematic filtering process is generated based on these association laws. Chen et al. (12) proposed a systematized spam filtering method based on decision tree data mining methodology to evaluate spam association rules and apply these rules to create an effective spam filtering method. This method provides a precision of 96%. Kumar et al. (13) proposed an approach that verifies the email header and URL as well as analyzes the body texts using different rules. They also employed Bayesian classifier and Apriori algorithm to classify files and attachments. Khamis S.A et al.(17) proposed a framework working with email header features for email spam detection by analyzing two email datasets. Support Vector Machine was used to classify email, which provides 88.80% of accuracy.

3. Methodology

The proposed system for spam e-mail detection is depicted in Figure 1. It has four phases: preprocessing, feature extraction, training, and prediction. Several preprocessing steps are performed before extract features from the text. Feature extraction techniques are used to extract features from the preprocessed texts. In the training phase, these extracted features are used to train machine learning classifiers (such as support vector machine (SVM), decision tree (DT), logistic regression (LR), and multinomial naïve bayes (MNB). Finally, in the prediction stage, the contents of the emails are predicted as spam or ham.

3.1. Preprocessing

The *preprocessing* step involves eliminating inconsistencies and mistakes from raw data to make it more understandable. As a result, we must preprocess our data before feeding it into our model. Consider the following email:

"hello! We want to make localized version of the software...."

This email can be preprocessed in the following manner:

- Special character removal: Each text is stripped of special characters such as (,=,>), numbers, and punctuation. After removal of those special characters

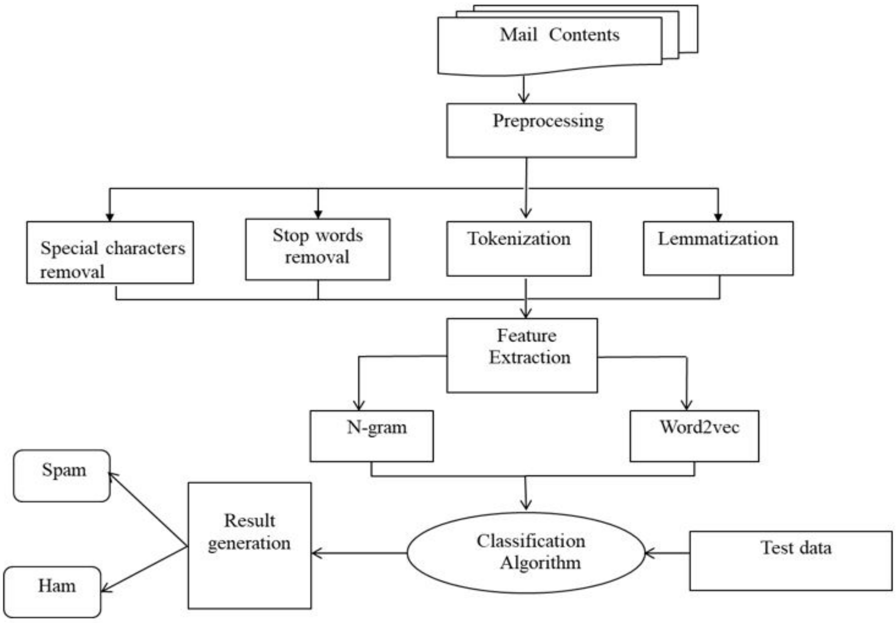

Figure 1. steps of proposed methodology for content-based spam email detection.

from the content of the above email, the text would become *"hello we want to make localized version of the software"*.

- Stop words removal: Words like "the," "an," "this," "a" etc. that aren't needed for recognizing spam or ham emails and hence those words are excluded. After removing the stop words, the text of the given e-mail becomes - *"want make localized version software"*.
- Tokenization: Tokenization is the process of breaking down a large text into smaller tokens. Tokenization provides a list of words like such as *("want", "make", "localized", "version", "software")* for the above given e-mail.
- Lemmatization: Lemmatization generally aims to eliminate only inflectional endings and restore the lemma, which is the base or dictionary form of a phrase. After lemmatizing the text of the e-mail, we get words like *("want", "make", "localize", "version","software")*.

3.2. Feature extraction

In feature extraction phase converts raw data into useful knowledge by reformatting, merging, and converting primary features into new ones. We have used the following feature extraction techniques:

3.2.1. N-gram:

An n-gram is an n-tuple or set of n words or characters those follow one another. The number of consecutive terms that can be treated as one gram is indicated by the letter *'n'*. As machine learning algorithms cannot access raw text. We have then applied the word n-grams, character n-grams, and a combination of variable length n-grams to gain a better understanding of the sentences. The texts are not in structured forms.

So, we convert text into numerical vectors using TF-IDF [1].

Word n-gram: Word n-grams deal with tuples or group of words. The word n-gram representation of an e-mail is shown in Table 1

Table 1. Word n-gram representation of an e-mail.

Sentence	“We want to make localized version of the software”
Uni-gram	‘we’, ‘want’, ‘to’, ‘make’, ‘localized’, ‘version’, ’of’,’the,’software’
Bi-gram	‘we want’, ‘want to’, ‘to make’, ‘make localized’, ’localized version’, ‘version of’, ’of the’, ‘the software ’
Tri-gram	‘we want to’, ‘want to make’, ‘to make localized’, ‘make localized version’, ‘localized version of’,’version of the’,’of the software’

Character n-gram: A text that is represented by a sequence of characters is known as a character n-gram. Unlike word n-grams, character n-grams can detect a word’s identification and possible neighbors and the word’s morphological makeup. Table 2 shows the character n-gram representation of an email.

Table 2. Character n-gram representation of an email.

Sentence	“We want to make localized version of the software”
Uni-gram	‘w’, ‘e’, ‘w’, ‘a’, ‘n’, ‘t’, ‘t’, ‘m’, ‘a’, ‘k’, ‘e’, ‘l’, ‘o’, ‘c’, ‘a’, ‘l’, ‘i’, ‘z’,’e’,’d’
Bi-gram	‘we’, ‘ew’, ‘wa’, ‘an’, ‘nt’, ‘tt’, ‘tm’, ‘ma’, ‘ak’, ‘ke’, ‘el’, ‘lo’, ‘oc’
Tri-gram	‘wew’, ‘ewa’, ‘wan’, ‘ant’, ‘ntt’, ‘ttm’, ‘tma’, ‘mak’, ‘ake’, ‘kel’, ‘elo’, ‘loc’, ‘oca’, ‘cal’

Combination of variable length n-gram: The variable length is not pre-defined by a combination of variable length n-grams. It can merge unigrams and bigrams, or trigrams and fivegrams. Table 3 shows the combination of variable length n-gram representation of an e-mail.

Table 3. combination of variable length n-gram representation of an e-mail.

Sentence	“We want to make localized version of the software”
uni-gram+bi-gram	‘we’,‘want’,‘to’,‘make’,‘localized’,‘version’,’of’, ’the’,’software’, ‘we want’, ‘want to’, ‘to make’, ‘make localized’, ‘localized version’, ‘version of’, ‘of the’,’the software’
bi-gram+tri-gram	‘we want’, ‘want to’,‘to make’,‘make localized’, ‘localized version’, ‘version of’, ‘of the’,’the software’‘we want to’, ‘want to make’, ‘to make localized’,‘make localized version’, ‘localized version of’,’version of the’,’of the software’

[1] https://towardsdatascience.com/text-vectorization-term-frequency-inverse-document-frequency-tfidf-5a3f9604da6d

3.2.2. Word2vec:

Word2vec uses a neural network model to learn word associations from a large corpus of text. This type of model can identify synonyms and suggest new terms for a sentence. Word2vec correlates each different word with a specific set of integers known as a vector, as the name indicates. We generate a *bag of words* model out of the entire corpus, with each word being a vector.

3.3. Training

The features gathered in the previous phase are used to train a machine learning model. Support vector machine (SVM), decision tree (DT), logistic regression (LR), and multinomial naïve bayes (MNB) are used to train the extracted features in our proposed approach. In the following subsections, we discuss these algorithms.

3.3.1. Support Vector Machine:

Support vector machine is a supervised machine learning model. It generalizes between two classes. The first goal of the SVM is to seek out a hyperplane that will distinguish between the 2 classes. The equation of hyper-plane is shown in Eq. 1:

$$w.x_i + b = 0 \tag{1}$$

where, w is the weight factor and b is that the bias, and x is the feature vector of sample i.

3.3.2. Logistic Regression:

Logistic regression works effectively with binary classification problems. The activation *sigmoid* function's mathematical equation that results to binary classification is shown below in Eq. 2:

$$F(z) = \frac{1}{1 - e^{-}z} \tag{2}$$

Now, in the above equation,

$$z = w_0 + w_1.x_1 + w_2.x_2 + ... + w_n.x_n \tag{3}$$

The model's co-efficient produced using *Maximum Likelihood Estimation* is $w0$, $w1$, $w2$,..., wn, and the features or independent variables are $x0$, $x1$, $x2$,..., xn in the preceding equation. Finally, the binary outcome likelihood is calculated using z in the previous equation, where the possibilities are separated into two categories based on the given information (x).

3.3.3. Decision Tree:

The decision tree has two types of nodes: external and internal nodes. External nodes represent the decision class, while internal nodes have the features required for categorization. A top-down strategy was used to examine the decision tree, which split homogeneous data into subsets. Its entropy is calculated using Eq.4 which defines sample homogeneity.

$$E(s) = \sum_{l=1}^{n} p_i log_2 p_i \tag{4}$$

The entropy of a sample in the training class is $E(S)$, and the probability of a sample in the training class is pi. Entropy was used to determine the splitting consistency. During the split, all of the features are considered to identify the appropriate split for each node. Random state 0 controls the recombination of the features.

3.3.4. Multinomial naïve bayes:

In Natural Language Processing(NLP), the multinomial naive bayes algorithm is a common probabilistic learning method. It assesses the probability of each tag for each sample and returns the tag with the highest probability. The Bayes theorem, as established by Bayes, calculates the likelihood of an event occurring based on prior knowledge of the conditions involved. In Eq5 the formula is shown.

$$P(A|B) = P(A) \times P(B|A)/P(A) \tag{5}$$

When a predictor B is already available, we evaluate the likelihood of sophistication A. $P(B)$ denotes the probability distribution of B, $P(A)$ denotes the prior probability of sophistication A, and $P(B|A)$ denotes the probability of predictor B given the probability of class A.

The trained classifier models are then used for predicting the text contents as spam or ham.

4. Evaluation Results and Analysis

We have implemented all the experiments in Intel Core i5 processor with GPU 8 GB RAM on Python 3.7 on jupyter notebook.

The 'spam or ham e-mail' dataset is collected from Kaggle [2], an online data publishing source. The dataset contains 5731 e-mails of which 1369 e-mails are spam and 4362 e-mails are ham.The training set contains 80% of the total data, while the testing set only contains 20% of the total data.

We use various machine learning algorithms to assess the system's performance, including SVM, MNB, DT, and LR. Again, various feature extraction approaches are used to test a variety of models.Table 4 shows the performance evaluation of different feature extraction methods. In the word n-gram feature, the SVM classifier provides the highest accuracy of 95.4% and precision of 98.2% considering bi-gram. In contrast, the logistic regression classifier achieves the best performance of 95.7% accuracy and 98.2% precision considering tri-gram. Again, for character n-gram, naive bayes classifier provides the highest 93.8% accuracy and 100% precision considering bi-gram and SVM achieves highest 95.9% accuracy and 97.5% precision considering tri-gram. We also combine variable-length n-gram features and find that the combination

[2] https://www.kaggle.com/balakishan77/spam-or-ham-email-classification

of (uni-gram+bi-gram) provides the highest accuracy of 97.6% and precision of 98.8% using the SVM classifier. For the Word2vec method, the logistic regression classifier provides the highest 83.1% of accuracy and 83.6% of precision.

We consider the uni-gram, bi-gram, and five-gram to investigate the features from email content. Using uni-gram, we alleviate the unwanted tokenization of white spaces. It decreases the possibility of spam considered to be ham (FN) or vice versa (FP) and provides the highest accuracy as shown in Figure 2. In conclusion, SVM has proven to be the best classifier and is effective in recognizing capabilities for our dataset due to robustness in high dimensions of features. With the combination of uni-gram and bi-gram, SVM achieves the highest accuracy of 97.6%, the precision of 98.8%, and f1-score of 94.9%.

Table 4. Performance comparison among different feature extraction methods.

Word-n gram	Classifier	Precision(%)	Recall(%)	f1 score(%)	Accuracy(%)
bi-gram	SVM	98.2	82.6	89.7	95.4
	Logistic regression	97.1	72.8	83.2	92.9
	Decision tree	79.0	94.2	85.9	92.5
	Naïve bayes	100	74.2	85.2	93.8
tri-gram	SVM	96.7	74.6	84.2	93.2
	Logistic regression	98.2	83.6	90.4	95.7
	Decision tree	70.4	91.6	79.6	88.7
	Naïve bayes	100	74.2	85.2	93.8
Character- n gram	**Classifier**	**Precision(%)**	**Recall(%)**	**f1 score(%)**	**Accuracy(%)**
bi-gram	SVM	88.1	51.0	64.6	86.5
	Logistic regression	86.1	47.4	61.2	85.5
	Decision tree	71.4	57.2	63.5	84.2
	Naïve bayes	100	74.2	85.2	93.8
tri-gram	SVM	97.5	85.5	91.1	95.9
	Logistic regression	96.8	78.6	86.7	94.2
	Decision tree	81.5	80.0	80.8	90.8
	Naïve bayes	100	74.2	85.2	93.8
Combination of variable length n-gram	**Classifier**	**Precision(%)**	**Recall(%)**	**f1 score(%)**	**Accuracy(%)**
uni-gram+ bi-gram	SVM	98.8	91.3	94.9	97.6
	Logistic regression	98.7	82.9	90.1	95.6
	Decision tree	87.0	88.0	87.0	93.9
	Naïve bayes	100	78.2	87.8	94.7
bi-gram+five-gram	SVM	99.2	90.9	94.8	97.6
	Logistic regression	98.7	85.1	91.4	96.1
	Decision tree	87.9	87.3	87.6	94.0
	Naïve bayes	56.5	99.3	72.0	89.4
Word2vec	**Classifier**	**Precision(%)**	**Recall(%)**	**f1 score(%)**	**Accuracy(%)**
	SVM	91.3	7.6	14.0	77.6
	Logistic regression	83.6	37.0	53.1	83.1
	Decision tree	55.0	55.6	55.3	78.4
	Naïve bayes	68.9	69.4	51.7	68.9

A receiver operating characteristic (ROC) curve shows how well a classification

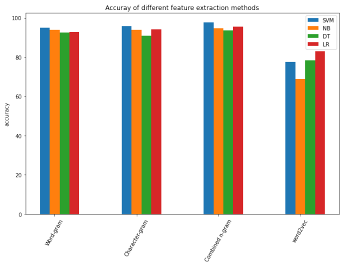

Figure 2. Accuracy comparison among different ML classifiers.

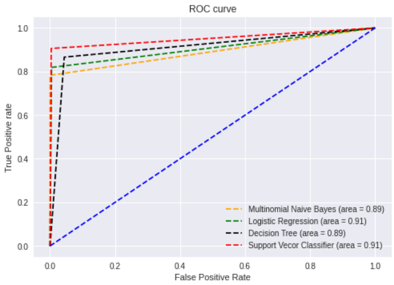

Figure 3. ROC curve for combination of variable length n-gram using various classification models.

model performs over various classification thresholds. We have drawn the ROC curve for detecting spam e-mails for the combination of uni-gram and bi-gram using different classifiers as shown in Figure 3. Logistic regression and SVM classifier performs better in perspective of the area under the ROC curve, and it is 0.91 in both cases.

Table 5. Comparison of our work with other benchmark works.

Reference	Feature Extraction	Classifier	Accuracy	Precision
[3]	document labeling	RF	92.97%	92.90%
[17]	header features	SVM	88.80%	-
Our method	combination of variable length n-gram	SVM	97.6%	98.8%

Further, we also compare our proposed method with some benchmark works of (3) and (17) as shown in Table 5. Our proposed method outperforms the two benchmark methods with an accuracy of 97.6% and precision of 98.8% using SVM classifier.

5. Conclusion

Accurate spam detection is an integral part of email communication. Despite accurate detection of spam (3; 17), false positive rate is also an issue. To do so, we present a content-based spam email detection approach. We use multinomial naïve bayes, logistic regression, support vector machine, and decision tree classifiers for learning the various features from the contents of emails. For comparative research, we use word n-gram (bi-gram, tri-gram), character n-gram (bi-gram, tri-gram), the combination of variable length n-grams (uni-gram and bi-gram, bi-gram and five-gram), and word2vec features. Among them, SVM achieves the best performance of 97.6% accuracy, 98.8% precision, and 94.9% f1-score for the combination of variable length n-gram (uni-gram and bi-gram). In the future, we can extend our work by analyzing the features using context-based machine learning.

References

[1] Sarker, Iqbal H., Md Hasan Furhad, and Raza Nowrozy. AI-driven cybersecurity: an overview, security intelligence modeling and research directions, SN Computer Science 2.3, 1-18 (2021)

[2] P. Liu and T. Moh, :Content Based Spam E-mail Filtering, 10.1109/CTS.2016.0052,218-224(2016)

[3] Gaurav, Devottam and Tiwari, Sanju Mishra and Goyal, Ayush and Gandhi, Niketa and Abraham, Ajith,:Machine intelligence-based algorithms for spam filtering on document labeling,Springer,9625-9638(2020)

[4] Kanaris, Ioannis and Kanaris, Konstantinos and Stamatatos, Efstathios,:Spam detection using character n-grams,Springer,95-104(2006)

[5] Kiliroor, Cinu C and Valliyammai, C, :Social context based Naive Bayes filtering of spam messages from online social networks,Springer,699-706(2019)

[6] Feng, Weimiao and Sun, Jianguo and Zhang, Liguo and Cao, Cuiling and Yang, Qing,:A support vector machine based naive Bayes algorithm for spam filtering,IEEE,1-8(2016)

[7] Moon, Jongsub and Shon, Taeshik and Seo, Jungtaek and Kim, Jongho and Seo, Jungwoo,:An approach for spam e-mail detection with support vector machine and n-gram indexing,Springer,351-362(2004)

[8] Kaur, Simran,:Spam Detection using N-gram Analysis and Machine Learning Techniques,(2019)

[9] Ahmad, Saleh Beyt Sheikh and Rafie, Mahnaz and Ghorabie, Seyed Mojtaba,:Spam detection on Twitter using a support vector machine and users' features by identifying their interactions,Springer,1-23(2021)

[10] Nayak, Rakesh and Jiwani, Salim Amirali and Rajitha, B,:Spam email detection using machine learning algorithm, Elsevier,(2021)

[11] Sheu, Jyh-Jian and Chu, Ko-Tsung and Li, Nien-Feng and Lee, Cheng-Chi,:An efficient incremental learning mechanism for tracking concept drift in spam filtering,Public Library of Science San Francisco, CA USA,(2017)

[12] Sheu, Jyh-Jian and Chen, Yin-Kai and Chu, Ko-Tsung and Tang, Jih-Hsin and Yang, Wei-Pang,:An intelligent three-phase spam filtering method based on decision tree data mining,Wiley Online Library,4013-4026(2016)

[13] Kumar, Santosh and Gao, Xiaoying and Welch, Ian and Mansoori, Masood,:A machine learning based web spam filtering approach,IEEE,93-980(2016)

[14] Sarker, Iqbal H. Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective, SN Computer Science, 1-22 (2021)

[15] Sarker, Iqbal H. Machine learning: Algorithms, real-world applications and research directions, SN Computer Science 2.3, 1-21 (2021)

[16] Sarker, Iqbal H. CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks, Internet of Things 14, 100393 (2021)

[17] Khamis, Siti Aqilah and Foozy, Cik Feresa Mohd and Ab Aziz, Mohd Firdaus and Rahim, Nordiana,Header based email spam detection framework using Support Vector Machine (SVM) Technique,Springer, 57-65 (2020)