doi:10.20944/preprints202109.0251.v1

ARTICLE

TF-IDF Feature-based Spam Filtering of Mobile SMS using Machine Learning Approach

Syed Md. Minhaz Hossain[a,b], Khaleque Md. Aashiq Kamal[b], Anik Sen[a,b] and Iqbal H. Sarker[a,⋆]

[a]Department of Computer Science & Engineering, Chittagong University of Engineering & Technology, Chattogram–4349, Bangladesh; [b]Department of Computer Science & Engineering, Premier University, Chattogram–4000, Bangladesh

ABSTRACT

Short Message Service (SMS) is becoming the secure medium of communication due to large-scale global coverage, reliability, and power efficiency. As person–to–person (P2P) messaging is less secure than application-to-person (A2P) messaging, anyone can send a message, leading to the attack. Attackers mistreat this opportunity to spread malicious content, perform harmful activities, and abuse other people, commonly known as spam. Moreover, such messages can waste a lot of time, and important messages are sometimes overlooked. As a result, accurate spam detection in SMS and its computational time are burning issues. In this paper, we conduct six different experiments to detect SMS spam from the dataset of 5574 messages using machine learning classifiers such as Multinomial Naïve Bayes (MNB) and Support Vector Machine (SVM), considering variations of *Term Frequency– Inverse Document Frequency (TF–IDF)* features for exploring the trade-off among accuracy, F1-score and computational time. The experiments achieve the best result of the accuracy of 98.50%, F1–score of 98%, and area under roc curve (AUC) of 0.97 for multinomial naïve bayes classifier with TF–IDF after stemming.

KEYWORDS
Spam detection, SMS, Security, Machine learning

1. Introduction

The rising of 5G and cloud technology introduces a new ecosystem that incorporates the connectivity of devices and technologies. SMS is becoming the secure medium of communication for machine–to–machine (M2M) or machine–to–person (M2P), for large-scale global coverage, reliability, power efficiency, and reduction of SMS cost (2). Besides, SMS has a vital role in making a decision based on the received data in Internet of Things (IoT) (5). IoT users can get notifications and other alarms from IoT devices through SMS (6). As the smart devices used in our everyday life activities are mostly directed by internet connectivity, the risk of data privacy or cyber-attacks is increasing day by day (16). Cyber-attack causes vital infrastructural destruction with massive losses of $345 per incident (1). Spamming is an effective way to spread

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

malware through the internet and mobile network. Usually, a mobile user faces a crisis of spamming through SMS. This fact led attackers to use SMS spam for spreading payload of cyber-attacks (3). Moreover, SMS spam is considered the most straightforward technique to deploy phishing attacks (4). As a result, security specialists are very devoted to developing an efficient SMS spam detection or classification method.

Machine learning techniques such as SVM, decision tree (DT), logistic regression (LR), and MNB play an essential role in detecting anomalies or classifying SMS spam (15; 17; 18; 19; 24). Moreover, these techniques have a vital contribution to detecting email spam messages (20; 21). Several studies have been done on the different types of proposed methods for the filtering of mobile SMS spam (25; 26; 27; 28). Different feature extraction methods such as word2vec, word n-gram, character n-gram, and combination of variable length n-gram are used to extract features in several works (22; 23; 31). Moreover, in spite of having the highest accuracy in machine learning based spam SMS detection(29; 30), the ratio of false positive (precision) and false negative (recall) is an issue.

Moreover, in Gupta et al. (2018), a deep learning algorithm is used to detect spam SMS and achieves 99.1% of accuracy (32). Using deep learning in detecting spam SMS is computationally expensive and time-consuming. Besides, deep learning models require a large amount of data. Despite having better accuracy in the deep learning method, it is a burning question of high accuracy with less time. Machine learning techniques consume less time than deep learning based spam SMS detection. Bagging and boosting algorithms achieve better results than traditional machine learning methods (31). However, there are still limitations of computational complexity and loss of interpretability. Moreover, TF–IDF has limitation in finding the similarity of words for each document. Despite this limitation, TF–IDF is significant for its improvement in the ratio of false positive (precision) and false negative (recall) (31) cause of its formulation.

As a result, for exploring the trade-off among accuracy, F1-score and time consuming, we conduct six different experiments to detect SMS spam from the dataset using machine learning classifiers such as MNB and SVM. The primary contributions of this paper:

(i) to develop an accurate and less time consuming spam filter for SMS .
(ii) to investigate different feature extraction process using TF-IDF and evaluate the impact of stemming before determining TF-IDF score. We investigate three process: TF-IDF with stemming, IF-IDF without stemming, and TF-IDF with message length.
(iii) to investigate the effect of message length in feature extraction.
(iv) to determine the best classification model between SVM and MNB with the most appropriate feature extraction process.

The rest of the paper is organised as follows. Section 2 discusses the previous benchmark works; proposed method for detecting spam is presented in Section 3; experiments, results and observations are illustrated in Section 4; and finally, the paper is concluded in Section 5.

2. Related works

In this section, we will look at comparative studies that other researchers have conducted. Joe et al. (7) developed a mobile SMS spam filtering system. They used 460

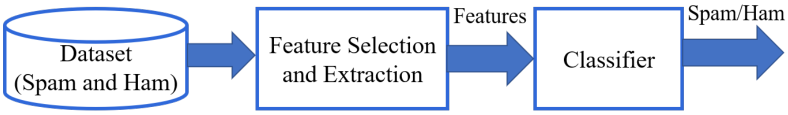

Figure 1. The overall procedure of the TF-IDF.

messages as a dataset, with 300 messages to train the SVM model and 160 for testing. Their model produced the best results with a feature vector value of 150 and a gamma value of 0.01 gamma. In (8), researchers worked on 875 messages as a data source, consisting of Turkish and English languages. They showed that BoW and Sfs together give the F1 score effectively. KNN and SVM are used as classifier models in their experiments. On the other hand, Foozy et al. (9) used Naïve Bayes and J48 as classifier models to conduct their experiments. They select the Malay language to generate their data set. Both Naïve Bayes and J48 models have given promising results. The text processing approach has been deployed in the work of (10). They used a public dataset with 5,574 SMS of the English language. Different classification models, including B.C4.5, KNN, SVM, Naïve Bayes, PART, have been used in this work.

Moreover, Reaves et al. (11) used the only SVM as a classifier model on the labeled dataset of SMS, which was gathered over 400 days. Their proposed classification model achieved 96.8% of recall with a high precision value. In (12), researchers have used a probabilistic data structure (PDS) for ensemble-based spam classification. They also showed that the number of spam messages could hamper the social networks in the field of IoT. Arifin et al. (13) worked on two different datasets consisting of 5,574 and 1,324 messages. Naïve Bayes classifier model is used in their work. To improve performance, they used FP-Growth as an association rule. The collaboration of both of them has given the highest accuracy value, with 98%. In (29), authors used TF−IDF as feature extraction and random forest as classifiers. This work achieves 97.50% accuracy. In (30), machine learning classifiers such as Logistic regression (LR), K-nearest neighbor (K-NN), and decision tree (DT) are used for classification spam in mobile device communication. The performance of LR achieved a high accuracy of 99%.

3. Proposed Methodology

At first, a dataset of spam and ham has been collected. Then, we applied the pre–processing and feature extraction method to the dataset. The extracted features have been used to create a feature vector for training and testing the classifier model. The classifier model finds an SMS is spam or ham. The general structure of our model is shown in Fig. 1.

3.1. Pre–processing

Pre–processing is one of the significant phases in text classification. Pre–processing eliminates noises and unwanted words from raw text and makes the text structured. Sometimes, unwanted words and features can affect the performance of various probabilistic classifiers. In this section, we depict all the pre-processing steps done in our work with an example.

Consider an SMS : (“hi! we bring a software....”)

3.1.1. Redundant character removal

For each SMS, special characters such as (,=,¿), numbers, and punctuation are removed. So, SMS become like (“hi we bring a software”).

3.1.2. Removal of stop words

Then from each SMS, we remove various words like ”the,” ”an,” ”this,” ”a,” and others that are not have meaning in extracting the features of spam or ham. At last, output would be (“bring software”).

3.1.3. Tokenization

In this phase, we break a large text in to a list of words such as (“bring”, “software”).

3.1.4. Lemmatization

Lemmatization eliminates inflectional morpheme and stores the lemma. Lemma is basically a base or a dictionary form of a phrase or word. After this phase, it would be (“bring”, “software”).

3.2. Feature Extraction

Our method explores features based on the three different ways: (1) Length of the message, (2) TF-IDF, (3) Stemming.

(1) Length of the messages: The length of the message is a crucial feature to determine the nature of any message. In this extraction phase, the length of all messages has been calculated using the number of remaining words after the pre–processing step.

(2) TF–IDF: TF–IDF is another vital extraction process to evaluate the nature of any SMS. Pre-processing is performed before starting the process of the TF–IDF algorithm. (i) After the pre–processing step, tokenization is performed on the sentences instead of words. Then the weight value is assigned. ii) Then, the calculation of word frequency is done. iii) In this step, TF (Term Frequency) is calculated using the formula in Eq.1.

$$TF(x) = \frac{(\text{No.of times words } x \text{ in a doc})}{(\text{Total no.of words in that doc})} \tag{1}$$

iv) After that, a table is created for the frequency of every word in every sentence. v) Then, IDF (Inverse document frequency) is calculated by Eq. 2.

$$IDF(x) = \log_e \frac{\text{Total no.of docs}}{\text{No.of docs with } x \text{ in it}} \tag{2}$$

vi) TF-IDF is calculated by multiplying the value of equation (1) and equation (2). vii) In these steps, the average score of all words is calculated to find the threshold value. Finally, any words are selected if the corresponding score is higher than the threshold value. Figure 2 concludes the overall procedure of the TF-IDF process.

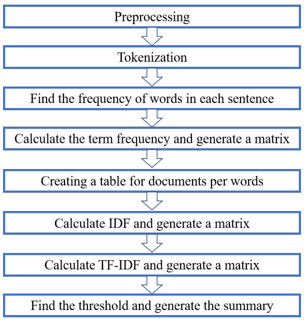
Preprocessing
Tokenization
Find the frequency of words in each sentence
Calculate the term frequency and generate a matrix
Creating a table for documents per words
Calculate IDF and generate a matrix
Calculate TF-IDF and generate a matrix
Find the threshold and generate the summary

Figure 2. The overall procedure of the TF–IDF process.

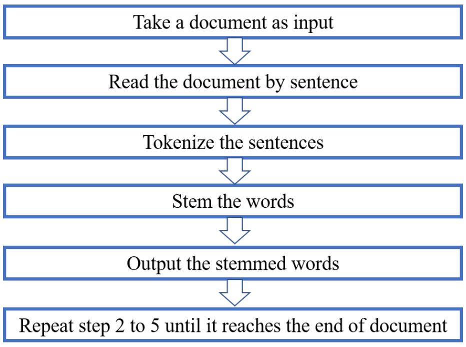
Take a document as input
Read the document by sentence
Tokenize the sentences
Stem the words
Output the stemmed words
Repeat step 2 to 5 until it reaches the end of document

Figure 3. The overall procedure of Stemming process.

(3) Stemming: Finally, we have used the stemming process before extracting TF-IDF scores. It is a way to find the root of any word. By eliminating prefixes, suffixes, we can get the root of a word. Figure 3 concludes the overall procedure of the stemming process.

3.3. Classifiers

After selecting three different feature extraction process, we have conducted six individual experiments using two classifier models: support vector machine (SVM) and multinomial naive bayes (MNB).

3.3.1. Support Vector Machine

Support vector machine is a supervised machine learning model used in various machine learning applications due to its higher accuracy. SVM classifies data in a binary classification problem by determining the best hyperplane that separates all data points of one class from those of the other. The goal of SVM is to maximize the separated hyperplane using formula as shown in Eq.3 to differentiate the classes:

$$w.x_i + b = 0 \tag{3}$$

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

3.3.2. Multinomial Naïve Bayes:

Naive Bayes algorithm is computationally efficient used in text analysis. Naive Bayes gives all words equal importance to track every single phrase in text. Naive Bayes uses Bayes' theorem, where features are mutually independent. That means the probability of one feature does not depend on the other feature. The probability model is formulated in Eq.4.

$$P(A|B) = P(A) \times P(B|A)/P(A) \tag{4}$$

where, P(B) is prior probability of B, P(A) = prior probability of A and P(B|A) = occurrence of predictor B given class A probability.

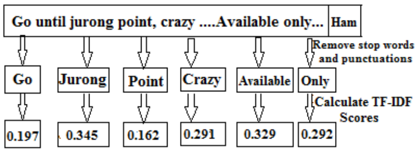

Figure 4. An example for TF-IDF score calculation.

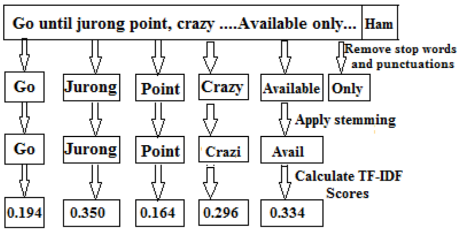

Figure 5. An example for TF–IDF score calculation after applying stemming process.

4. Results and Analysis

All of the experiments were carried out on a computer with an Intel Core i5 processor and 4 GB of RAM. We have implemented the experiments using Python 3.7 on Spider IDE.

4.1. Dataset

We have used an open dataset of SMS messages from the kaggle (14). The dataset is labeled with two classes: spam and ham. It contains a total of 5,572 messages, of which 4,825 are ham, and 747 are spam (4516 unique ham messages and 653 spam messages). We divide the dataset into 70-30, 67-33, 80-20 proportion for training purpose. As the 70-30 proportions attain a better accuracy, we consider 70-30% of training and test messages by different classifiers.

4.2. Classification using SVM and Multinomial Naïve Bayes

The calculation of TF–IDF score and TF–IDF after stemming for a sample SMS (labeled as ham) from our examined dataset are as shown in Figure 4 and Figure 5. SVM and MNB classifiers are used to incorporate all three features of messages. At first, to select the best hyper-parameters, grid search method is used. In SVM, the best Gamma is defined as 1 with the sigmoid kernel. The value of Alpha is set to 0.2 in MNB. Six different experiments are debated with critical analysis based on these hyper-parameters.

4.2.1. Performance Measure

As our ham and spam samples are imbalanced, we consider accuracy and F1-score for evaluating our detection model as formulated in Equations (5)–(11).

$$\text{Accuracy of a model } = \frac{\sum_k(\text{Recognition rate of each class} \times N_k)}{N} \tag{5}$$

$$\text{Recognition rate of a class} = \frac{\text{True Positive} + \text{True Negative}}{\text{Number of all samples of that class}} \tag{6}$$

doi:10.20944/preprints202109.0251.v1

$$\text{Precision of a model} = \frac{\sum_k(\text{ Precision of each class} \times N_k)}{N} \tag{7}$$

$$\text{Precision of a class} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{8}$$

$$\text{Recall of a model} = \frac{\sum_k(\text{Recall of each class} \times N_k)}{\text{N}} \tag{9}$$

$$\text{Recall of a class} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{10}$$

$$F_1 \text{ score } = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

where k represents each of the class, N_k indicates the number of samples in class k, and N is the total number of samples used to test the model.

4.2.2. Performance Evaluation for Different Feature Extraction Methods using Various Classifiers

The accuracy and F1 score of six different experiments are summarized in Table 1. With only the TF–IDF score as feature extraction, both SVM and MNB achieve remarkable accuracy of 97.85% and 98.45%, respectively. SVM and MNB models are 97.85% and 98.50% accurate for TF–IDF score with stemming, respectively. In this case, we can say that using the TF-IDF score with stemming improves the accuracy of the MNB classifier model by 0.05%. Finally, to compare the results, we add another feature (message length). However, in this case, the SVM model's accuracy has dropped to 86.12%, whereas MNB's accuracy has remained consistent at 98.27%. As a result, we can deduce that factoring message length into the experiment adds no value.

doi:10.20944/preprints202109.0251.v1

Table 1. Performance evaluations of all the experiments

Exp. No.	Feature Extraction Process	Classifier	Accuracy	F1 Score	Computational Time (sec)
1.	Pre-processing and TF-IDF	SVM	97.85%	98%	0.545
2.	Pre-processing and TF-IDF with stemming	SVM	97.85%	98%	0.484
3.	Pre-processing,stemming TF-IDF and Length of the messages	SVM	86.12%	80%	0.567
4.	Pre-processing and TF-IDF	Multinomial Naïve Bayes	98.45%	98%	0.021
5.	Pre-processing and TF-IDF with stemming	Multinomial Naïve Bayes	98.50%	98%	0.023
6.	Pre-processing,stemming TF-IDF and Length of the messages	Multinomial Naïve Bayes	98.27%	98%	0.128

However, SVM with the stemming, TF–IDF scores, and length of the messages give the lowest f1 score, which is 80%. The other two cases give a better f1 score with a value of 98%. F1 scores are also consistent for the MNB classifier model for all three cases with a value of greater or equal to 98%.

From Table 1, it is obvious that with stemming TF–IDF scores are better for both SVM and MNB. MNB achieves better performance for TF–IDF integrated with stemming as stemming traces the different terms of the same meaning significantly.

4.2.3. Performance Representation for the best classifier using AUC and confusion matrix

We test 1115 messages, including 1003 ham messages and 112 spam messages using both the classifiers SVM and MNB with variations of TF–IDF. As we achieve the best accuracy using MNB with TF–IDF features (extracted after stemming), it is shown in Table 2 for individual classes. MNB achieves better accuracy in every case than SVM due to the performance loss for imbalances in positive and negative support vectors in SVM.

Table 2. Performances of each classes in detecting spam messages using MNB with TF-IDF features(extracted after stemming).

Classes	**Precision**	**Recall**	**F1-score**
Ham	100%	**96%**	98%
Spam	96%	**100%**	98%

To evaluate the results further, we have drawn the ROC curves for both cases. The micro average area under the ROC curve (AUC) for SVM is 0.93, and for MNB is 0.97. Figure 6 shows the area under the ROC curve (AUC)for MNB classifier. It implies that MNB is more robust to filter the spam in SMS with stemming and the TF–IDF process.

4.2.4. Computational time analysis for classifying spam

The impact of feature selection on computational time is reflected in Table 1. The computational time for SVM drops from 0.5445 seconds to 0.484 seconds after stemming. In MNB, however, it is almost the same as stemming. However, considering all

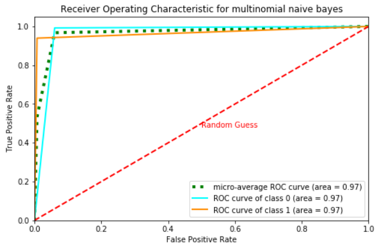

Figure 6. ROC curve for multinomial naïve bayes classifier model using TF-IDF integrated with stemming.

Table 3. Comparison of our work with other benchmark works.

Reference	Feature Extraction	Classifier	Accuracy	F1-score
[29]	TF-IDF	Random Forest	94.5%	96.24%
[30]	TF-IDF	Logistic Regression	96.8%	97.00%
Our method	TF-IDF	Multinomial Naïve Bayes	98.50%	98.00%

the factors (accuracy, F1 score, and computational time), MNB is the optimal model for detecting spam SMS using TF-IDF integrated with stemming.

4.2.5. Comparison among the benchmark spam detection method

We compare our work with the other benchmark methods and are represented in Table 3. We implement their feature extraction technique and classification models on our dataset to compare with other benchmark works. Our proposed method achieves better accuracy of 98.50% and F1 score of 98% than the other two works.

4.2.6. Critical Evaluation

Despite the fact that our proposed method provides better accuracy and F1 score, it still has some misclassifications. Consider the following example of a ham misclassified message: “Ok lar... Joking wif u oni.” The majority of the words in this message are removed during the pre-processing stage. In the feature extraction process, only ”Joke” is counted. Despite the fact that it is a ham message, it is flagged as spam. In this case, the message’s context is critical. We’ll also use context-based machine learning to look into the features of the messages.

5. Conclusion

For security in message communication, detection of spam plays an important role. Accuracy and efficiency in detecting spam are significant issues to allow users to read only relevant messages. However, to solve these issues, we investigate all the possibilities of improving SMS classification accuracy using variations of TF–IDF. Best scores of accuracy 98.50% and F1–score 98% have been found by incorporating stemming and TF–IDF using MNB. Besides, conclude that message length does not matter to detection spam in SMS. When we use stemming, it also increases the accuracy in MNB; however, execution time remains constant for filtering SMS. Finally, we can conclude that MNB shows better results than SVM in all cases, whether the size of words decreases or the length of message increases or decreases. We will try to extend our work by detecting text embedded in an image (stego image) and developing a method using ontology and semantic web for scalable SMS spam filtering.

References

[1] Hu, X.: Large-Scale Malware Analysis, Detection, and Signature Generation. [online] Available at: https://https://deepblue.lib.umich.edu/handle/2027.42/89760 [Accessed 28 Feb. 2020]

[2] Lau, Charles Q., et al. ”In search of the optimal mode for mobile phone surveys in developing countries. A comparison of IVR, SMS, and CATI in Nigeria.” Survey Research Methods. Vol. 13. No. 3. 2019.

[3] Dang.That wasn't supposed to happen. [online] Available at: https://mobilecommons.com/blog/2016/01/how-text-messaging-will-change-for-the-better-in-2016 [Accessed 28 Feb. 2020]

[4] Gupta, B.B., Tewari, A., Jain, A.K., Agrawal, D.P.: Fighting against phishing attacks: state of the art and future challenges. Neural Computing and Applications. 28, 3629–3654 (2016)

[5] Atzori, L., Iera, A., Morabito, G., Nitti, M.: The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization. Computer Networks. 56, 3594–3608 (2012)

[6] M&M Research Group, 2012. Internet of Things (IoT) & M2M communication market-advanced technologies, future cities & adoption trends, roadmaps & worldwide forecasts 2012- 2017. [online] Available at: https://www.prnewswire.com/news-releases/internet-of-things-iot–machine-to-machine-m2m-communication-market—advanced-technologies-future-cities–adoption-trends-roadmaps–worldwide-forecasts-2012—2017-216448061.html [Accessed 28 Feb. 2020]

[7] Joe, I., Shim, H.: An SMS Spam Filtering System Using Support Vector Machine. Future Generation Information Technology. 577–584 (2010)

[8] Uysal, A.K., Gunal, S., Ergin, S., Sora Gunal, E.: The Impact of Feature Extraction and Selection on SMS Spam Filtering. Electronics and Electrical Engineering. 19, (2013)

[9] Mohd Foozy, C.F., Ahmad, R. and Abdollah, M.F., A framework for SMS spam and phishing detection in the Malay language: A case study. International Review on Computers and Software, 9(7), pp.1248-1254 (2014)

[10] Almeida, T.A., Silva, T.P., Santos, I., Gómez Hidalgo, J.M.: Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering. Knowledge-Based Systems. 108, 25–32 (2016)

[11] Reaves, B., Blue, L., Tian, D., Traynor, P., Butler, K.R.B.: Detecting SMS Spam in the Age of Legitimate Bulk Messaging. Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks. (2016)

[12] Singh, A., Batra, S.: Ensemble based spam detection in social IoT using probabilistic data structures. Future Generation Computer Systems. 81, 359–371 (2018)

[13] Delvia Arifin, D., Shaufiah, Bijaksana, M.A.: Enhancing spam detection on mobile phone Short Message Service (SMS) performance using FP-growth and Naive Bayes Classifier. 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob). (2016)

[14] SMS Spam Collection Dataset. [online] Available at: https://www.kaggle.com/uciml/sms-spam-collection- dataset [Accessed 28 Feb. 2020]

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

[16] 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)

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

[18] Dada, E.G.; Bassi, J.S.; Chiroma, H.; Adetunmbi, A.O.; Ajibuwa, O.E. Machine learning

for email spam filtering: Review, approaches and open research problems. Heliyon, 5, e01802 (2019)

[19] Shah, N.F.; Kumar, P. A comparative analysis of various spam classifications. In Progress in Intelligent Computing Techniques: Theory, Practice, and Applications; Springer: Berlin/Heidelberg, Germany, pp. 265–271 (2018)

[20] Chandrasekar, C.; Priyatharsini, P. Classification techniques using spam filtering email. Int. J. Adv. Res. Comput. Sci., 9, 402 (2018)

[21] Shafi'I, M.A.; Latiff, M.S.A.; Chiroma, H.; Osho, O.; Abdul-Salaam, G.; Abubakar, A.I.; Herawan, T. A review on mobile SMS spam filtering techniques. IEEE Access, 5, 15650–15666 (2017)

[22] J. Hua and Z. Huaxiang, "Analysis on the content features and their correlation of Web pages for spam detection", China Commun., vol. 12, no. 3, pp. 84-94, Mar. (2015)

[23] S.-E. Kim, J.-T. Jo and S.-H. Choi, "SMS spam filterinig using keyword frequency ratio", Int. J. Secur. Appl., vol. 9, no. 1, pp. 329-336, (2015)

[24] E. B. Cleff, "Privacy issues in mobile advertising", Int. Rev. Law Comput. Technol., vol. 21, pp. 225-236, (2007)

[25] O. Osho, O. Y. Ogunleke and A. A. Falaye, "Frameworks for mitigating identity theft and spamming through bulk messaging", Proc. IEEE 6th Int. Conf. Adapt. Sci. Technol. (ICAST), pp. 1-6, Oct. (2014)

[26] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach", J. Netw. Comput. Appl., vol. 36, no. 1, pp. 324-335, Jan. (2013)

[27] O. Osho, V. L. Yisa, O. Y. Ogunleke and S. I. M. Abdulhamid, "Mobile spamming in Nigeria: An empirical survey", Proc. Int. Conf. Cybersp. (CYBER-Abuja), pp. 150-159, Nov. (2015)

[28] Sarker, Iqbal H. “Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective.” (2021).

[29] Nilam Nur Amir Sjarif, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya, Suriani Mohd Sam, SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm. The Fifth Information Systems International Conference, 509–515 (2019)

[30] Luo GuangJun, Shah Nazir, Habib Ullah Khan, and Amin Ul Haq, Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms.Security and Communication Networks, vol. 2020, Article ID 8873639, 6 pages, https://doi.org/10.1155/2020/8873639 (2020)

[31] Kowsari, Kamran, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. "Text classification algorithms: A survey." Information 10, no. 4 (2019): 150.

[32] Gupta, M., Bakliwal, A., Agarwal, S. and Mehndiratta, P. (2018). A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers, 2018 11th International Conference on Contemporary Computing, IC3 2018 pp. 1–7.