A Machine Learning Approach for Efficient Spam Detection in Short Messaging System (SMS)

Robert G. de Luna
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
rgdeluna@pup.edu.ph

Verna C. Magnaye
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
vcmagnaye@pup.edu.ph

Rose Anne L. Reaño
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
ralreano@pup.edu.ph

Karina L. Enriquez
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
klenriquez@pup.edu.ph

Dexter Astorga
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
dpastorga@iskolarngbayan.pup.edu.ph

Trisha Celestial
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
tmcelestial@iskolarngbayan.pup.edu.ph

Aira Mae Española
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
amtespanola@iskolarngbayan.pup.edu.ph

Brian Allen Lanting
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
baqlanting@iskolarngbayan.pup.edu.ph

Danielle Mugar
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
dmmugar@iskolarngbayan.pup.edu.ph

Mateo Ramos
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
mateogramos@iskolarngbayan.pup.edu.ph

Jenjazel Redondo
Polytechnic University of the Philippines
Sto. Tomas City, Batangas, Philippines
jmredondo@iskolarngbayan.pup.edu.ph

***Abstract*—Short Message Service (SMS) is a generally used communication method due to its convenience and affordability. SMS spam message is an unauthorized text message that contains a variety of content types such as advertisements, fraudulent texts, and promotions. These messages can pose a serious threat to mobile phone users as they may contain security threats, malicious activities, and other concerning issues. These can lead to identity theft, financial loss, and other types of fraud. To deal with the problem of spamming, various machine-learning models are applied to develop an optimized model that effectively, reliably, and precisely identifies and filter out spam or junk message from a genuine SMS text.**

The dataset used is a combination of self-acquired data and internet collected dataset with 60-40 ham to spam partitions. With regards to the accuracy of the model, the Bernoulli Naive Bayes achieved the highest performance with 96.63% accuracy upon optimization.

Keywords— spam message, ham message, machine learning, short messaging system

I. INTRODUCTION

Short Message Service (SMS), also known as "text messaging," permits users to send short messages to electronic devices such as smartphones and cellular phones. It is the most basic and widely used method of communication in the world because it does not require an internet connection, and it is a low-cost service provided by most telecommunications service providers. SMS spam message is an unauthorized text message that contains a variety of content types such as advertisements, fraudulent texts, and promotions. One of the purposes of spamming is to entice the user to direct on a link or reveal private details such as banking information, credit card details, addresses, or even social security number. Spam messages are not limited to SMS only, but it is also prevalent in different social media platforms, email, and web pages. The prevalence of SMS as a primary form of communication makes it more vulnerable to different forms of illicit activity through text spamming.

Generally, spam text messages are utilized as a form of promotion of products and services. However, considering they are unsolicited networks, these messages can endanger mobile phone users as they may contain serious security threats, malicious activities, malware problems, and other concerning issues. Using machine learning model, spam text messages may be determined by studying the context and words used. Spam text messages and ham messages may be classified through algorithms. In order to deal with the prevalent spamming, the researchers aim to improve a machine-learning model that effectively, reliably, and precisely assesses whether the provided SMS text message is spam or a genuine text message, which is a ham. The issue of SMS spam messages is widespread and has been the subject of many previous studies, with machine learning being a common approach to solving the problem. However, many of these studies use similar datasets, and there is often a large gap in the quantity of spam and non-spam SMS texts. Researchers also gathered additional data to reduce the gap in the number of spam and non-spam messages.

The proposed SMS spam detection model must be able to determine through natural language processing whether the text message is spam or ham. Applying more classification algorithms will give a higher chance of finding the most effective model; hence, eleven classification methods are used: logistic regression, multinomial naive bayes, decision tree, K-neighbors, support vector, AdaBoost, bagging, extra trees, gradient boosting, random forest, and XGBoost. This will be limited to the classifier

mentioned, and only the top four classifiers with the highest performance accuracy will be optimized. The model could benefit telephone company subscribers who face the danger of spam and scams.

II. Review of Related Works

SMS spamming has become a serious problem for mobile customers (Gomaa, 2020) [1]. Individuals are susceptible on being scammed by text messages with website links, especially with rewards, because of their curiosity (Wali, 2021) [2]. For this reason, systems that aid in determining spam messages are important. These systems can be attained with the help of different techniques like machine learning.

Mukerjee (2020) [3] implemented count-vectorizer (CV) for their feature extraction step to generate distinct features. The result of their experiment shows that count-vectorizer performs well with the simple Naive Bayes (NB), attaining the accuracy of 98%, performing better than Logistic Regression (LR).

In the study of Krishnaveni and Radha (2021) [4], a comparison between NB and SVM algorithms is carried out for spam SMS detection with the use of NLP. Count vectorizer is used to recognize the number of unique words in the provided dataset. Upon comparing the two classifiers, it is found that SVM is better than Naive Bayes in all the standards used. The SVM attained a 94.32% accuracy, precision of 92.84%, recall of 93.07%, and F-measure which is 94%.

Gangare et. al (2022) [5] utilized count vectorizer for feature selection and applied Naïve Bayes multinomial classifier for their model in identifying SMS messages. This model achieved a 94% efficiency. In the study of Abiramasundari (2021) [6], Count vectorizer is used to convert the text into numeric values of the data before integrating the Rule Based Subject Analysis (RBSA) and Semantic Based Feature Selection (SBFS) techniques with Multinomial Naive Bayes, Support Vector Machine, Gaussian Naive Bayes and Bernoulli Naive Bayes. The performance of each classifier is compared using four metrics: precision, f1-score, recall, and support. It is found that SVM gained the best results in the four metrics used.

In the study of Kontsewaya et al (2020) [7], they used CountVectorizer as their embedding technique, along with seven classifications: K-Nearest Neighbors, SVM, Logistic regression, Decision tree, Naive Bayes, and Random Forest. Through the performed hold out validation, Naive Bayes and Logistic Regression performed with 99% accuracy.

Kudupudi and Nair (2021) [8] created a model to detect spam messages with NLP and Term Frequency - Inverse Document Frequency (TF-IDF) vectorizer as word embedding technique. The classifier used in the system, Logistic Regression (LR), achieved 96% accuracy. Similar to this, Nazir et al. (2020) [9] proposed a system adding two other machine learning classifiers in training model, these are Decision Tree (DT) and KNN. The best model is still logistic regression, which has 99% accuracy.

Ora (2020) [10] proposed a model that detects spam messages with low latency. NLP techniques (Bag of Words and TF-IDF) are used as well, while Chi-Square for feature selection to reduce latency. Five machine learning models: Extreme Gradient Boost (XGBoost), Light Gradient Boosting Machine (LightGBM), SVM, BNB, and Random Forest, were then implemented, wherein BNB achieved the maximum accuracy of 96.5% with a latency of 0.157 seconds.

Mohasseb et al. (2020) [11] model for identifying and classifying spam SMS is based on the message's syntactical features and patterns. Among the three implemented learning techniques: Naive Bayes (NB), KNN and Random Forest (RF), the KNN achieved the highest accuracy of 83.9%. The proposed model of Mussa and Jameel (2019) [12] used extreme gradient boosting algorithm (XGBoost) for spam detection and two wrapper feature selection algorithms to select the optimal feature. The algorithm used gave 98.64% of accuracy when used for handling an imbalanced dataset.

Palad et.al (2019) [13] converted their data to Waikato Environment for Knowledge Analysis Weka-suitable format and used three classification algorithms: J48 decision tree, Naive bayes and sequential minimal optimization (SMO). Among the three classifiers, the J48 decision tree achieved the highest accuracy with 79%. Alshahrani (2021) [14] also used WEKA text technique and applied two machine learning classifiers: random forest and decision tree, since those two are used for a large number of datasets with various feature types. The classification system using the random forest method produces the best results, with a 98.2% accuracy rate.

Python based Flask is the platform used by Gupta S. et.al (2021) [15] and TF-IDF vectorization is carried out on generating word cloud vector. With this, they achieved a model with accuracy of 95.90%. Adewale et al. (2021) [16] model is executed on the Python programming platform's Scikit-learn library. Machine learning methods were used for validation, namely as SVM with the RBF Kernel], Logistic Regression, Gaussian Naive Bayes [NB], Random Forests, Multinomial Naive Bayes [NB] and Ada boost. They conclude that enforcing feature selection techniques to normalize and boost the size of SMS messages improved the effectiveness of machine learning algorithms in SMS message classification.

Gadde (2021) [17], used six classification algorithms, which are the Logistic Regression, KNN, Decision tree, Naive Bayes, SVM, and Random Forest. They also used Count Vectorizer, Hashing Vectorizer and TF-IDF Vectorizer as word embedding techniques. For the sampling of data, they used SMOTE, Synthetic Minority Oversampling Technique. Out of all the models they tested, TF-IDF Vectorizer with SVM classifier obtained the best accuracy, with accuracy percentage of 97% in classifying the spam messages.

Liu (2021) [18], used five classification algorithms together with Word2Vectorizer and TF-IDF Vectorizer. They used Syntactic Parsing to analyze the relationship of words in a sentence in the SMS. The best performing model is the Logistic Regression with Word2Vec as the word

embedding technique to classify the spear phishing messages.

Julis (2020) [19] utilized text mining in their data preprocessing, a particular step of this process is homoglyphing, where they detect homoglyphs and convert them to their original meaning. These are similar looking symbols to the English alphabet.

Yerima (2022) [20] proposed a semi-supervised One Class SVM with only non-spam data for training. It achieved an overall accuracy of 98% and true positive rate of 100%, which is better when compared with the seven standard machine learning algorithms that used a bag of words approach.

Sonowal (2020) [21] utilized four feature selection algorithms, the Pearson rank correlation, Spearman's rank correlation, Kendall rank correlation, and point biserial rank correlation. The best ranking algorithm is the Kendall ranking algorithm showed accuracy of 98.40% with AdaBoost Classifier. It also reduced the number of features by 39.47%.

Sjarif (2020) [22] used structural features as the independent variable for their spam classifier model. They also used standard metrics such as accuracy, time, Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and false positive in conjunction with kappa statistics to establish the best performing model based on accuracy. The outcome showed that SVM Classifier 98.9% accuracy as the best model.

Among these studies, they have used at least two classifiers to determine which model is the most suitable. The question of what the results may be if the number of models used is higher needs to be addressed.

Out of all the related works, only a few executed cross validation and optimization. Thus, an optimized and cross validated model may exceed the performance and reliability of existing models.

The performance metric considered in determining the best model on most of the studies is solely based on accuracy. Although some calculate the precision, recall, and F1-score value, the relevance of the following metrics was not indicated.

III. RESEARCH METHODOLOGY

The model can differentiate a ham message, or generally desired legitimate message, to spam message, or any type of undesired message typically sent for scamming.

The SMS spam dataset combines three separate datasets. The researchers performed exploratory data analysis before applying text preprocessing techniques, selection of vectorizer, and eleven machine learning algorithms. These models are evaluated to see the top four performing model. These models are then optimized to improve the model's accuracy. Figure 1 shows the flow of process which researchers utilized to come up with the proposed model.

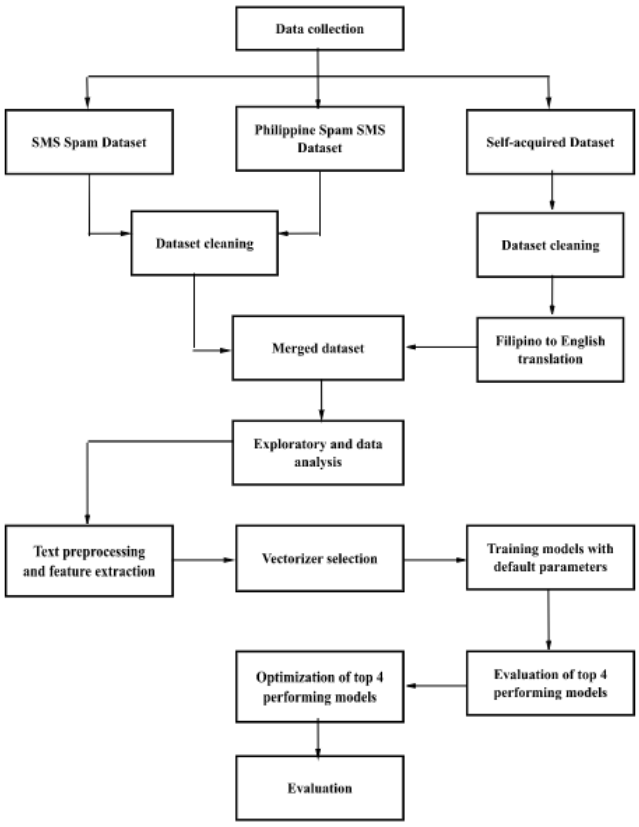

Fig. 1. Process Scheme for SMS Spam Detection.

A. Dataset

The SMS spam dataset used by the researchers is a combination of three different datasets: two from a dataset library site – Kaggle and one self-acquired data by researchers. The dataset SMS spam collection by UCI Machine Learning is composed mostly with ham messages or messages that is considered a genuine text for the recipient and is not considered spam. This dataset has 5169 unique values with two columns for class and SMS. After performing data cleaning, it is seen in Table 1 that the data is clearly imbalanced with an 87-13 partition of ham and spam.

TABLE I. THE PHILIPPINE SPAM SMS DATASET

Number of Ham Messages	Number of Spam Messages	Total
4,516	653	5,169

These leads for the researchers to do add more dataset to somehow balance out the data. The first additional dataset, Philippine Spam SMS, is a collection of personally received spam messages of the author Bwandowando. This includes the cellphone number of the sender of the spam, the date received, and the spam text message. The data cleaning was performed which determine the following results.

The second additional dataset is a self-acquisition of data from the electronics engineering students at Polytechnic University of the Philippines – Sto. Tomas Branch (PUPSTB). The data are random English and Filipino messages.

TABLE II. SPAM SMS DATASET

Number of Ham Messages	Number of Spam Messages	Total
0	79	79

To aid the Filipino messages, the researchers used the Google Translate API to change them into the English language. Once the Filipino messages are translated into English, data cleaning is performed which concluded the data shown in Table 3.

TABLE III. SMS SPAM COLLECTION DATASET

Number of Ham Messages	Number of Spam Messages	Total
0	561	561

The total number of ham messages is 4,516 and spam messages is 1,293. Since the data is still unbalanced, the researchers under sampled ham messages to create a 60-40 data partition, in which 60% of the messages are ham and 40% are spam. The merged three datasets resulted in a new dataset with the quantity of each spam and ham displayed in Table 4. The merged datasets gathered a collection of 15,183 ham words and 18,382 words spam words. The partition of the dataset is 60-40 but after text transformation, removal of non-essential words and reduction of words to their root words, 83 data points were duplicated so they were removed to the training and testing data, yielding to 1,930 ham messages and 1,218 spam messages.

TABLE IV. FINAL DATASET

Number of Ham Messages	Number of Spam Messages	Total
1,930	1,218	3,148

B. Text Preprocessing

SMS Exploratory and data analysis is performed to obtain a preliminary insight into the data. Text preprocessing may undergo through different NLP techniques. Turning letters to lowercase to avoid duplication of the same word with different capitalizations.

Tokenization involves dividing a text into smaller, meaningful units such as words, phrases, or other relevant elements. Removing special characters from text helps simplify the data, as these characters are non-alphanumeric and can impede text processing and analysis. *Stopwords* are words that do not contribute much to the meaning, often redundant, of the text. Removing *stopwords* and punctuation can reduce the size of the text corpus and potentially speed up information retrieval and learning.

Stemming is the reduction of a word to its base. This can help to normalize the text and reduce the quantity of unique words in a corpus. After executing those processes, text data is converted into a numerical format. The dataset has undergone text transformation to identify what will vary in spam and ham identification.

TABLE V. PRE-PROCESSED RESULTS

Count	Before Text Transformation	After Text Transformation
Characters	311,201	180,021
Words	68,412	33,565
Sentences	7,273	3,227

C. Word Embedding Technique

Text data must first be converted into a numerical format for machine learning algorithms that can only work with numerical data. The Count Vectorizer and TF-IDF Vectorizer were used to determine the technique for default parameters.

C.1. Count Vectorizer

This is a text feature extraction technique that converts text data into a numerical matrix. This is done through tallying the frequency of each word in a text, and producing a matrix where rows relate to documents, columns correspond to words. The elements in the matrix represent the frequency of every word in the respective document.

C.2 TF-IDF Vectorizer

The TF-IDF Vectorizer measures the significance of a word to the document in a set. The TF-IDF calculates a numeric score for each word in a document based in the frequency in the document (term frequency), and the rarity across the entire corpus (inverse document frequency). Words that appear more commonly in a document have a higher term frequency score, while words that are less common across the entire corpus have a higher inverse document frequency score. The TF-IDF vectorizer forms a matrix where each row signifies a document, each column represents a word, and the numeric values indicate the relative importance of each word in the corresponding document.

$$TF = (\text{Frequency of word in a document})/(\text{Total number of words in that document})$$

$$DF = (\text{Documents containing word W})/(\text{Total number of documents})$$

$$IDF = (\log(\text{Total quantity of documents}))/(\text{Documents containing word W})$$

$$TF - IDF = TF \times IDF$$

D. Classification Model in Selection of Vectorizer

Naive Bayes is conventionally considered a classification-type model. It has been effectively applied in a range of applications, including spam sorting, text classification, opinion analysis, and recommender systems. Three Naive Bayes variation are used to test the two vectorizers, namely: Gaussian Naive Bayes, Multinomial Naive Bayes, and the Bernoulli Naive Bayes.

E. Machine Learning Techniques

The texts in the datasets have been converted into numerical vectors to apply the machine learning classifiers. These classifiers are: Bernoulli Naive Bayes Classifier, Decision Tree Classifier, K-Nearest Neighbors Classifier, Random Forest Classifier, AdaBoost Classifier, Bagging

Classifier, Extra Trees Classifier, Gradient Boosting Classifier, Extreme Gradient Boosting Classifier, Logistic Regression Classifier, and Support Vector Classifier.

IV. RESULTS AND DISCUSSION

A. Metrics Evaluation

The machine learning algorithms were measured based on its accuracy, precision, recall, and F1 scores, but only the accuracy metric will be used to evaluate the model with the best performance.

Accuracy = ((TP + TN)) / ((TP + TN + FP + FN))

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 Score = (2(Precision × Recall)) / (Precision + Recall)

B. Selection of Vectorizer

In the selection of Vectorizer, CountVectorizer and TF-IDF Vectorizer are tested using three Naive Bayes Models.

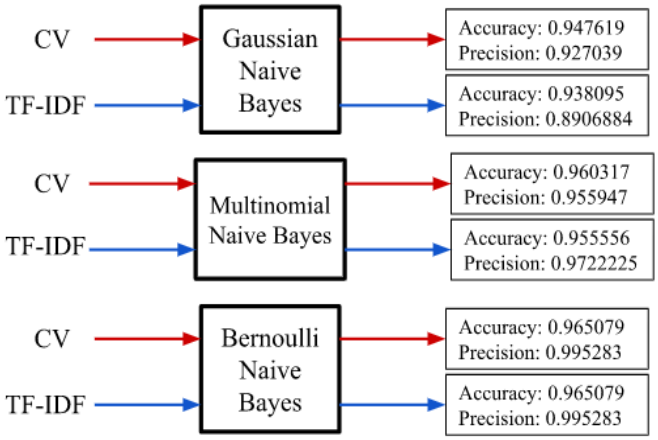

Fig. 2. Comparison of CountVectorizer and TF-IDF Vectorizer in Naive Bayes Model.

The scores indicated that Bernoulli Naive Bayes produces accuracy with highest performance using CountVectorizer and TF-IDF Vectorizer obtaining similar results. Precision, on the other hand, is measured to determine the consistency of the models. The second highest performing model, which is the Multinomial Naive Bayes, is utilized to identify the vectorizer. Since CountVectorizer yielded higher accuracy than TF-IDF Vectorizer, CountVectorizer is chosen as the preferred vectorizer.

C. Cross-Validation Results with Default Parameters

TABLE VI. EXPERIMENT RESULTS WITH MACHINE LEARNING ALGORITHMS

Algorithm	Accuracy, %
LR	95.6164
BNB	95.4270
RFC	95.2354
ETC	94.9811
SVC	94.5684
XGB	93.9011
AdaBoost	92.3768
BgC	92.0271
GDBT	91.2335
DTC	91.1692
KNN	76.6521

Eleven machine learning classification algorithms are trained with default parameters. These are Logistic Regression (LR), Support Vector (SVC), Bernoulli Naive Bayes (BNB), Decision Tree (DTC), K-Nearest Neighbors (KNN), Random Forest (RFC), AdaBoost, Bagging (BgC), Extra Trees (ETC), Gradient Boosting (GDBT), and XGBoost (XGB). Upon checking the results, LR has the highest accuracy with 95.6164% as shown in Table VI.

D. Optimization of Top Four Performing Models

The four top performing models are optimized to observe the effect of optimized parameters in these algorithms. The results of optimization in Table VII shows that all machine learning models improve their accuracy rate. The performance metrics are also determined to evaluate the range and uniformity of the models. BNB achieved the highest accuracy 96.6333%.

TABLE VII. EXPERIMENT RESULTS WITH TOP FOUR MACHINE LEARNING ALGORITHMS

Algorithm	Accuracy	Precision	Recall	F1 Score
BNB	96.6333%	98.4431%	95.5174%	92.7835%
RFC	96.0297%	98.5228%	94.8699%	91.1408%
LR	95.8709%	98.7711%	93.7399%	89.2555%
ETC	95.8070%	98.4885%	94.3495%	90.5663%

E. Comparison of Machine Learning Models

The top four performing models, Logistic Regression, Bernoulli Naive Bayes, Random Forest, and Extra Tree are determined based on their accuracy rate. In the results cross validation with default parameters, the Logistic Regression has the highest accuracy of 95.6164%. Hence, after the optimization, the Bernoulli Naive Bayes achieved the highest accuracy rate of 96.6333% which is determined to be the proposed machine learning model for the SMS Spam Detection.

V. CONCLUSION

In this paper, researchers proposed a machine learning model for spam messages detection. Datasets are derived from three separate datasets. Two datasets are gathered and the other one is self-acquired by researchers. The combined dataset used is composed of 3,233 messages with 60-40 ham and spam partitions. The CountVectorizer is the word embedding technique used to input data into different machine learning classification models.

Eleven machine learning classification models are tested, and the top four performing models proceed to optimization using GridSearchCV. With regards to the accuracy of the model, the Bernoulli Naive Bayes achieved the highest accuracy rate of 96.6333%. which is determined to be the best machine learning model for the SMS Spam detection in this study.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge the support and resources provided by Polytechnic University of the Philippines (PUP). The facilities, funding, and access to relevant literature and research materials have greatly contributed to the success of this project.

REFERENCES

[1] W. H. Gomaa, "The Impact of Deep Learning Techniques on SMS Spam Filtering," International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, 2020, doi: https://doi.org/10.14569/ijacsa.2020.0110167.

[2] H. Wali, "An Empirical Study on Text Phishing", California State University, San Marcos, 2021, http://localhost/files/rv0430686.

[3] A. Mukerjee, "Spam Filtering Using Bag-of-Words" The Start Up, 2020, https://medium.com/swlh/spam-filtering-using-bag-of-words-aac778e1ee0b.

[4] N. Krishnaveni and V. Radha, "Comparison of Naïve Bayes and SVM Classifier for Detection of Spam SMS using Natural Language Processing", International Journal of Semantic Computing, vol. 11, no. 2 pp. 2260-2265, 2021, Doi: 10.21917/ijsc.2021.0323.

[5] A. Gangare, J. Rathore, A. Tadge, A. Shrivastav, R. Yadav, and P. Sisodiya, "Implementation of Spam Classifier using Naïve Bayes Algorithm", International Research Journal of Engineering and Technology (IRJET), Vol. 09, 02,2022, https://www.irjet.net/archives/V9/i2/IRJET-V9I272.pdf.

[6] S. Abiramasundari, "Spam filtering using Semantic and Rule Based model via supervised learning", Rajasthan Cooperative Recruitment Board, Vol. 25, Issue 2, pp. 3975 – 3992, 2021, https://www.annalsofrscb.ro /index.php/journal/article/download/1405/1174.

[7] Y. Kontsewayaa, E. Antonova, and A. Artamonov, "Evaluating the Effectiveness of Machine Learning Methods for Spam Detection" 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: Eleventh Annual Meeting of the BICA Society, 2020, doi: 10.1016/j.procs.2021.06.056.

[8] N. Kudupudi and S. Nair, "Spam message detection using logistic regression", International Journal of Advanced Computer Science and Applications, 9(9), 815-818, 2021, https://www.ijisrt.com/assets/upload/files/IJISRT21SEP728.pdf.

[9] S. Nazir, H. Khan, and A. Haq, "Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms" Security and Communication Networks. 2020 https://doi.org/10.1155/2020/8873639.

[10] A. Ora, "Spam Detection in Short Message Service Using Natural Language Processing and Machine Learning Techniques" master's thesis, Dublin, National College of Ireland, 2020, https://norma.ncirl.ie/id/eprint/4286.

[11] A. Mohasseb, B. Aziz, and A. Kanavos, "SMS Spam Identification and Risk Assessment Evaluations", Proceedings of the 16th International Conference on Web Information Systems and Technologies, 2020, https://doi.org/10.5220/0010022404170424.

[12] D. Mussa, and N. Jameel, "Relevant SMS Spam Feature Selection Using Wrapper Approach and XGBoost Algorithm", Kurdistan Journal of Applied Research, 4(2), pp. 110-120, 2019, https://doi.org/10.24017/science.2019.2.11.

[13] E. B. Palad, M. Tangkeko., L. Magpantay, and G. Sipin, "Classification of Filipino Online Scam Incident Text using Data Mining Techniques", 19th International Symposium on Communications and Information Technologies (ISCIT), 2019, doi: 10.1109/iscit.2019.8905242.

[14] A. Alshahrani, "Intelligent Security Schema for SMS Spam Message Based on Machine Learning Algorithms", Int. J. Interact. Mob. Technol., vol. 15, no. 16, pp. pp. 52–62, Aug. 2021, https://doi.org/10.3991/ijim.v15i16.24197.

[15] S. D. Gupta, S. Saha, and S.K. Das "SMS Spam Detection Using Machine Learning", Journal of Physics: Conference Series (JPCS), 2021, doi: 10.1088/1742-6596/1797/1/012017.

[16] Y. Adewale, "Enhanced Short Message Service Spam Filtering System Based on Normalized and Expanded Text," vol. 63, no. 01, 2021, https://publication.babcock.edu .ng/asset/docs/publications/COSC/9457/6121.pdf.

[17] S. Gadde, A. Lakshmanarao, and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Mar. 2021, doi: https://doi.org /10.1109/icaccs51430.2021.9441783.

[18] M. Liu, Y. Zhang, B. Liu, Z. Li, H. Duan and D. Sun, "Detecting and Characterizing SMS Spearphishing Attacks", 930-943, 2021, doi: 10.1145/3485832.3488012.

[19] M. Julis, and S. AIagesan, "Spam detection in SMS using machine learning through text mining", International Journal of Scıentıfıc & Technology Research, 9 (02), 2020, https://www.academia.edu/download/77424969/Spam-Detection-In-Sms-UsingMachine-Learning-Through-Text-Mining.pdf.

[20] S. Yerima and A. Bashar, "Semi-supervised novelty detection with one class SVM for SMS spam detection", 2022 29th International Conference on Systems, Signals, and Image Processing (IWSSIP), pp. 1 – 4, 2022, doi: 10.1109/IWSSIP55020.2022.9854496.

[21] G. Sonowal, "Detecting Phishing SMS Based on Multiple Correlation Algorithms", SN Comput Sci. 2020; 1(6):361. doi: 10.1007/s42979-020-00377-8. Epub 2020 Nov 2. PMID: 33163974; PMCID: PMC7604914.

[22] N. Sjarif, Y. Yazriwati, C. Suriayati, and A. Nurul, "Support Vector Machine Algorithm for SMS Spam Classification in The Telecommunication Industry", International Journal on Advanced Science, Engineering and Information Technology, 10(2), pp. 635-639, 2020, http://dx.doi.org/10.18517/ijaseit.10.2.10