Analysis of Machine Learning Algorithms Used For Short Message Service (SMS) Spam Classification

Dr. Vidhya P M[1], Joel Eldo[2], Neha Joy[3], Sooryadas Ps[4], Sreerag K[5], Dhakshu Sivan[6]
[1]Associate ProfessorM Dept of CSE
[2, 3, 4, 5, 6]Assistant Professor, Dept of CSE
[1, 2, 3, 4, 5, 6] Sree Narayana Gurukulam College Of Engineering (SNGCE), Kolenchery, Ernakulam

Abstract- *SMS, or short message service, is a vital instrument for communication on a global scale. SMS is a marketing tool used by businesses, but regrettably some people exploit it to send spam. Worldwide, these spam and promotional texts are frequently received by smartphone users. The work[1] reviewed here examines a paradigm for categorizing spam, promotional, and ham messages using standard text messages. 4,125 text messages were used to train the model and 1,260 to test it. The classifiers were evaluated using a 10-fold cross validation approach, and the findings indicate that XGBoost, Multinomial Logistic Regression, Support Vector Machine, and Random Forest are some of the top models for a multi - class classification of useful and spam SMS.*

Keywords- short message, spam, comparative study, machine learning, natural language processing

I. INTRODUCTION

Short Message (or Messaging) Service, a system that enables mobile phone users to send and receive text messages. Our project intends to categorize spam messages apart from otherwise useful messages (ham messages), based on the most suitable natural language processing (NLP) and machine learning (ML) algorithms we can find.

II. PURPOSE

One could consider SMS to be a worldwide form of communication. Unfortunately, this offers people a lucrative opportunity to abuse this service for bad intentions. Textclassification could immensely aid in this endeavor. Models for classifying messages according to their features can be produced via classification algorithms These models can now be used in mobile devices as the processing capability of current smartphones is practically equal to that of computers. This gives smartphone users the chance to utilize an application that can run a classifier to categorize SMS.

III. TRANSACTIONS/JOURNAL PAPERS REVIEWED

The referenced literature helps us to take a deeper look into the world of SMSs, classification of spam messages, methods of analysis, groundbreaking studies and existing methodologies before applying machine learning algorithms on the resultant data.

A. *Short Messaging Service*

Globally, Short Message Service (SMS) is the most widely used form of communication. In 2015, there were 6.1 billion SMS users globally. Comparatively, just 2.6 billion people use email globally [2]. According to the 2013 Ericsson Mobility Report, mobile subscriptions and data traffic have increased globally on average, by 70% during the fourth quarters of 2012 and 2013. The regions with the highest rates of growth include APAC, China, and India [3]. Around 9 billion mobile subscriptions are currently active worldwide, according to a 2019 report by Ericsson [4]. A comparable statistical study on smartphone use in the US was published by the Pew Research Centre. According to the report, text messaging is the most common way that Americans use their smartphones [5]. According to all of these reports, SMS can be viewed as a global form of communication. Unfortunately, this offers people a lucrative opportunity to abuse this service for bad intentions. This project might benefit immensely from text classification. Models for differentiating messages according to their features can be produced via classification algorithms. These models can now be used in mobile devices thanks to the development of modern technology, as the processing capability of current smartphones is practically equal to that of computers. This gives smartphone users the chance to have an application that can run a classifier to categorize SMS. In the next sections we take a look into an independently created dataset of SMS and its investigation in which an unsupervised multilingual sentence boundary detection [5] is used to tokenize the sentences, and a language corpus will be used to train the tokenizer as mentioned in [7].

B. *Emerging Studies*

Numerous researches have attempted to categorize Short Message Service (SMS) messages according to their qualities. A study employs a dataset of 450 ham messages and 425 spam messages made up of British English SMS [8]. It utilizes the Naive Bayes (NB) model to categorize spam SMS. The SMS messages' features were taken. They were used to teach the model how to distinguish between ham and spam messages. The NB technique was adopted because it is easier to compute and retrain, making it possible to easily retrain the model [8]. spam messages are categorized using TF-IDF and Random Forest (RF) [9]. Using a collection of English data, the model was trained. The highest accuracy was achieved when RF and TF-IDF were combined.

A study examined the usage of two types of feature selection approaches with three neighboring algorithms—NB, J48, and Support Vector Machine (SVM)—in an effort to determine the optimal feature selection techniques to utilize for classifying SMS spam [10]. The study only distinguished between ham and spam messages and employed an English dataset. After using a feature selection method, the accuracy of all algorithms—with the exception of SVM—increases. The improvement in accuracy may be explained by the fact that feature selection techniques can improve the efficiency of a conventional machine learning model [11].

Spam is filtered out using Optimum-path Forest-based (OPF) classifiers in another investigation [12]. The paper contrasts this OPF with SVM, artificial neural networks with multilayer perceptrons, K Nearest Neighbor (KNN), and (ANN-MLP). Results indicate that compared to other techniques, training the OPF classifiers involves less time and resources.

Ham messages were correctly detected by OPF, however half of the spam messages were incorrectly identified. Except for SVM, the OPF classifier has a greater accuracy rating than other algorithms. Despite the fact that SVM uses a lot more resources than OPF to categorize messages [12]. Another study [13] suggested using decision trees to filter spam SMS. Other than feature matching, his study analyses communications based on the sender's phone number, time zone, prohibited characters, and whether the message contains terms that are on a blacklist. This filtering technique is simple, but a ham message that possesses one of these characteristics can be labelled as a false positive. Having many false positives is arguably more of a concern than potentially banning ham transmissions.

Combining spam categorization and personality recognition models has been used in an effort to decrease false positives [14], [15]. Results from both researches indicate that a spam classification model that incorporates personality as one of its variables may be able to improve the performance of subpar classifiers and lessen the incidence of false positives. In a different study, ensemble learning was used to try to improve the accuracy of weak classifiers [16]. The outcome is an improvement above prior efforts that just used one classifier. The dendritic cell technique, a novel approach to ensemble learning, has state-of-the-art results combining SVM and NB, reaching an accuracy of 99% on both the training and validation datasets [17].

The classification of spam and junk SMS messages has also been done using convolutional neural networks (CNN) [18]. Semantic-CNN (SCNN), an algorithm, was employed in the study to categorize the communications. The SCNN could only distinguish between spam and ham messages after being trained on an English dataset. Although the accuracy for SMS categorization is 98.65%, the results are comparable to the state-of-the-art.

Another method of spam detection was put out [19] by contrasting the effectiveness of KNN, Decision Tree (DT), and Logistic Regression (LR). The study primarily focuses on using LR to distinguish between spam and ham communications, but it also clearly quantifies the efficacy of each method. The outcome demonstrates that LR outperforms KNN and DT, with the best accuracy of the three at 99%. When compared to KNN and DT, LR's computation time is short.

A gradient boosted tree technique is called XGBoost (XGB). For the best results for the classifier, gradient boosting uses ensemble learning on weak algorithms. Scalable and lightning-fast performance are two features of XGB [20]. This method is thus an excellent substitute for spam classification. Another study that used XGB to categorize spam email got excellent results [21]. Their XGB classifier performed better than earlier research efforts, with an accuracy of 96.88%. A 94.4% accuracy percentage for classifying SMS spam was achieved by another thesis that used XGB [22]. But the findings show that XGB underperformed in comparison to Bernoulli Naive Bayes (BNB).

C. *Existing methodology of analysis*

There are five distinguished stages in the process of designing a machine learning model [1], focused mainly on processing the raw data beforehand the stages are data gathering, data labelling, create bag of words, model training, and result analysis.The process of creating the bag of words is the most crucial and sophisticated. It is again divided into

three steps -text preprocessing, term frequency, and finding text features as shown in Fig 1.1.

i. *Data Gathering and Data Labelling*

[1] claims to have collected the SMS from users' phones and over a span of 4 years have collected above 4000 SMS messages that can be used as training data.Hence a sum of 56% ham communications, 33% advertisements and 10% spam messages make up the training dataset. Later every message in the dataset was manually labelled after it was assembled to which numbers were assigned that represent the various message categories. Ham communications, promotional messages, and spam messages are the three types of messages that were employed in the research.

ii. *Creating a Bag of Words*

SMS is really noisy. Characters like noise might make it difficult for algorithms to effectively classify communications. By iteratively looping through each message in the dataset, we can preprocess the text to eliminate any characters that we find superfluous. Punctuation is an example of a character that is not required in a text classification model because it is a reading aid that helps readers grasp what they are reading. In order to eliminate uppercase letters, the entire dataset was additionally lowercased. This algorithm in particular categorizes ⌈the message based on its word properties. This paper made use of the Tala corpus to find word features that were employed in the local region [9].

iii. *Training the model*

The reviewed work conducted the experiments using eight classifying algorithms. The algorithms used were : Multinomial Naive Bayes (MNB), Multinomial Logistic Regression (MLR), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), Stochastic Gradient Descent (SGd), XGBoost (XGB), and Random Forest (RF) algorithms.

D. Evaluation of Results

The accuracy of almost all algorithms surpassed 94%, with Random Forest (RF) being the most accurate followed by Multinomial Logistic Regression (MLR) and XGBoost (XGB). The first set of average results contains a minor differential margin between all algorithms of only approximately 0.3%, with the exception of KNN, MNB, and DT. The KNN algorithm had the lowest accuracy in both batches, but subsequently had a large decline of roughly 12% in the second batch. Other algorithms, such MNB and DT, also exhibit this trend, with the first batch consistently scoring higher accuracy than the second. This might be the case because KNN, MNB, and DT performance is correlated with the volume of training data. On the second batch of 10-fold results, however, SVM outperformed XGB by 0.39% and obtained the third batch. RF, MLR, and SVM are the most accurate models in that order. The skewed dataset of the first batch may have led to the poor performance, which is why SVM surpassed XGB in the second batch. The difference, however little, demonstrates how a balanced dataset could enhance a model's performance. The fact that the datasets were labelled by various individuals and that the definitions of "ham," "spam," and "promotional messages" varied between the two datasets is one of the numerous factors influencing the test results.

E. *Analysis of algorithms*

i. Multinomial Logistic Regression

Unsurprisingly, the accuracy of the MLR approach in this case was 94% since it is simpler to use, comprehend, and train. It doesn't make any assumptions about how classes are distributed in feature space. It not only offers an assessment of a predictor's suitability (coefficient size), but also the direction of relationship (positive or negative).

The main drawback of logistic regression is that it creates linear boundaries and assumes linearity between the dependent variable and the independent variables. Logistic regression should not be employed if there are less data than features because this could result in overfitting. It only works for forecasting discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.

ii. Multinomial Naive Bayes (MNB)

The Naive Bayes algorithm is efficient and operates quickly and yielded an accuracy of upto 94% . It can be used to resolve multi-class prediction issues. We can argue that for categorical input variables as opposed to numerical variables, Naive Bayes is more appropriate. It can outperform other models and needs a lot less training data if its assumption about the independence of characteristics is correct.

But on the other hand, Naive Bayes makes the uncommon but unfounded assumption that all predictors (or features) are independent. This restricts the algorithm's usability in practical usage cases. This approach encounters the "zero-frequency problem," where it gives a categorical variable with zero probability if its category was not present in

the training dataset but was present in the test data set. To resolve this problem, it would be ideal if you employed a smoothing technique. Its estimations can be wrong in some cases, so we must not take its probability outputs for face value

iii. Support vector machine (SVM)

Good at solving machine learning problems with small samples. It's commended for its ability to handle non-linear feature interactions. It has no need to rely on the entire data as it possesses a strong generalization ability.

The notable con of SVMs is that when the observation sample is large, the efficiency is not very high. SVMs provide no universal solution to nonlinear problems, and sometimes it is difficult to find a suitable kernel function. They are also sensitive to missing data.

iv. K-Nearest Neighbor (KNN)

K-NN has no assumptions; in contrast to linear regression, which requires numerous data assumptions before it can be used, K-NN makes no such assumptions. K-NN just tags new data entry based learning from historical data without explicitly building any model. The majority of classifier algorithms are simple to construct for binary problems but require additional work to implement for many classes, in contrast to K-NN which adapts to several classes automatically. As we add more training data, the classifier adjusts right away. In real-time use, it enables the algorithm to react swiftly to changes in the input. The fact that K-NN may be applied to both classification and regression issues is one of its main advantages. There are several distance criteria available as well.

But there are several areas where K-NN fails. K-NN may be fairly simple to use, but as the size of the dataset increases, the algorithm's effectiveness or speed quickly decreases. When there are few input variables, it functions well, but as the number of variables increases, the K-NN algorithm has trouble predicting the results of new data points. It wants the most neighbors possible. On unbalanced data, k-NN doesn't perform well. The model will finally give A a lot of preference if we consider two classes, A and B, and the majority of the training data is labelled as A. This could lead to incorrectly classifying the less common class B. K-NN is fundamentally incapable of handling the missing value problem.

v. Decision Trees

Decision trees are effective in both classification and regression applications because they may be used to predict both continuous and discrete values. Decision trees need less effort to understand an algorithm because they are basic. It can record relationships that are not linear. Because decision trees do not simultaneously take into consideration numerous weighted combinations, they have the benefit of not requiring any feature modification when working with non-linear data. When compared to KNN and other classification algorithms, they are incredibly quick and effective. Decision trees are simple to comprehend, analyze, and depict. One machine learning approach where we don't worry about feature scaling is the decision tree. Random woods are another. Decision trees are helpful in data exploration because they provide us a solid understanding of the relative relevance of attributes. Because there is no outside influence or impact from missing data in a tree node when using a decision tree, less data is needed.

That noted, the operation's time complexity is extremely high and continues to rise as the number of records increases. In addition, training a decision tree with numerical variables takes a long time. Similar results are obtained with methods like random forests and XGBoost. As the input rises, it takes longer for training-time complexity to rise.

vi. Stochastic gradient descent

Due to the network only processing one training sample, it is simpler to fit in the memory. Because just one sample is processed at a time, it is computationally quick. It can converge more quickly for larger datasets since it updates the parameters more frequently. The steps necessary to reach the loss function's minima have oscillations because of the frequent updates, which can assist you escape the loss function's local minimums (in case the computed position turns out to be the local minimum).

The following are some drawbacks of stochastic gradient descent: The procedures taken to get the minima are highly noisy as a result of the frequent updates. This frequently function minima due to noisy steps. Because one training sample is processed at a time, frequent updates are computationally expensive. As it only handles one sample at a time, it lacks the benefit of vectorized operations.

vii. XGBoost

This algorithm works well. It does effectively with small, big, intricate, and data with subgroups. However, it struggles with sparse data and can also have difficulties with data that is widely spread. On such kinds of data challenges, it

typically performs better than the majority of supervised learning algorithms.

‘

The black box nature is likely the largest restriction. XGBoost won't provide us with effect sizes if we ask for them (though some adaboost-type algorithms can give that result). That part would need to be independently derived and programmed. For those use situations, XGBoost would not be out go-to algorithm given the models that already exist (such as penalized GLMs).

Algorithms	Accuracy	Difficulty	Desirability of result
MLR	94%	Easy	Satisfactory
MNB	upto 94%	Easy	Good
SVM	98%	Hard	Less
KNN	86%	Easy	Less
XGB	92%	Hard	Reliable
SGB	94%	Hard	Reliable
RF	94%	Easy	Satisfactory
DT	93%	Easy	Good

Fig. 1

IV. CONCLUSION

In this paper, we surveyed an example for Short Messaging Service (SMS) classification and the viable algorithms for its implementation. We inferred that SMS message classification works reallywell with conventional machine learning algorithms. Except for K-Nearest Neighbor(KNN)., all the algorithms utilized in this study achieved accuracy rates of at least 90% as shown in Fig.1.

According to results from the testing set,the best result was obtained when the model employed a selection from the dataset with an equal ratio for each class. MLR and SGd, followed by MNB and RF algorithms best meet our purpose and are anticipated to yield the best results when compared to the size of our potential dataset. MLR, XGB, and stochastic gradient descent may all attain the highest degree of accuracy (SGD).In the upcoming paper, we pursue the development of an SMS spam classification system that makes use of the inferences made from the referred work and provides reliable and accurate results.

V. APPENDIX

SMS	:	Short Message Service
ML	:	Machine Learning
MNB	:	Multinomial Naive Bayes
MLR	:	Multinomial Logistic Regression
SVM	:	Support Vector Machine
K-NN	:	K-Nearest Neighbor
DT	:	Decision Tree
SGd	:	Stochastic Gradient Descent
XGB	:	XGBoost
RF	:	Random Forest

REFERENCES

[1] Agustinus Theodorus, Tio Kristian Prasetyo, Reynaldi Hartono, Derwin Suhartono - "Short Message Service (SMS) Spam Filtering using Machine Learning in Bahasa Indonesia" 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT) April 09-11. 2021, ISTTS Surabaya, IndonesiaPortio Research, “SMS: the language of 6 billion people,” Portio Research Limited, p. 49, Jun-2015.

[2] Ericsson AB, “Ericsson Mobility Report: On the Pulse of the Networked Society,” Stockholm, Sweden, 2013.

[3] Ericsson, “Mobility Reports,” Ericsson Mobility Report, 2019.

[4] A. Smith, “U.S. Smartphone Use in 2015,” Pew Research Center: Internet & Technology, 2015. [Online]. Available: https://www.pewresearch.org/internet/2015/04/01/us-smartphone-use in-2015/. [Accessed: 18-Jan-2020].

[5] T. Kiss and J. Strunk, “Unsupervised Multilingual Sentence Boundary Detection,” Comput. Linguist., vol. 32, no. 4, pp. 485-525, Dec. 2006.

[6] F. Tala, “A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,” University of Amsterdam, 2003.

[7] M. Taufiq Nuruzzaman, C. Lee, M. F. A. bin Abdullah, and D. Choi, “ Simple SMS spam filtering on independent mobile phone,” Secur. Commun. Networks, vol. 5, no. 10, pp. 1209-1220, Oct. 2012.

[8] N. N. Amir Sjarif, N. F. Mohd Azmi, S. Chuprat, H. M. Sarkan, Y. Yahya, and S. M. Sam, “ SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm,” Procedia Comput. Sci., vol. 161, pp. 509-515, 2019.

[9] A. Sharaff, “ Spam Detection in SMS Based on Feature Selection Techniques,” in Emerging Technologies in Data Mining and Information Security. Advances in Intelligent Systems and Computing, A. Abraham, P. Dutta, J.

Mandal, A. Bhattacharya, and S. Dutta, Eds. Singapore: Springer Nature Singapore Pte Ltd, 2019, pp. 555-563.

[10] K. Uysal, S. Gunal, S. Ergin, and E. Sora Gunal, "The Impact of Feature Extraction and Selection on SMS Spam Filtering," Electron. Electr. Eng., vol. 19, no. 5, pp. 67-72, May 2013.

[11] D. Fernandes, K. A. P. Da Costa, T. A. Almeida, and J. P. Papa, " SMS Spam Filtering Through Optimum-Path Forest-Based Classifiers," in 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015, pp. 133-137.

[12] I. Androulidakis, V. Vlachos, and A. Papanikolaou, "Fimess: filtering mobile external sms spam," in Proceedings of the 6th Balkan Conference in Informatics, 2013, pp. 221-227.

[13] E. Ezpeleta, I. Garitano, U. Zurutuza, and J. M. G. Hidalgo, "Short Messages Spam Filtering Combining Personality Recognition and Sentiment Analysis," Int. J. Uncertainty, Fuzziness Knowledge-Based Syst., vol. 25, no. Suppl. 2, pp. 175-189, Dec. 2017.

[14] E. Ezpeleta, U. Zurutuza, and J. M. Gómez Hidalgo, "Short Messages Spam Filtering Using Sentiment Analysis," in Text, Speech, and Dialogue, P. Sojka, A. Horák, I. Kopecek, and K. Pala, Eds. Switzerland: Springer International Publishing Switzerland, 2016, pp. 142-153.

[15] V. Gupta, A. Mehta, A. Goel, U. Dixit, and A. C. Pandey, " Spam Detection Using Ensemble Learning," in Harmony Search and Nature Inspired Optimization Algorithms. Advances in Intelligent Systems and Computing, N. Yadav, A. Yadav, J. Bansal, K. Deep, and J. Kim, Eds. Singapore: Springer Nature Singapore Pte Ltd, 2019, pp. 661-668.

[16] A. Al-Hasan and E.-S. M. El-Alfy, "Dendritic Cell Algorithm for Mobile Phone Spam Filtering," Procedia Comput. Sci., vol. 52, pp. 244-251, 2015.

[17] G. Jain, M. Sharma, and B. Agarwal, " Spam Detection on Social Media Using Semantic Convolutional Neural Network," Int. J. Knowl. Discov. Bioinforma., vol. 8, no. 1, pp. 12-26, Jan. 2018.

[18] L. GuangJun, S. Nazir, H. U. Khan, and A. U. Haq, " Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms," Secur. Commun. Networks, vol. 2020, pp. 1-6, Jul. 2020.

[19] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785-794.

[20] B. Mustapha, S. Hasan, S. O. Olatunji, S. M. Shamsuddin, and A. Kazeem, "Effective Email Spam Detection System using Extreme Gradient Boosting," arXivPrepr. arXiv2012.14430, Dec. 2020.

[21] A. Ora, " Spam Detection in Short Message Service Using Natural Language Processing and Machine Learning Techniques," National College of Ireland, 2020.

[22] F. Rahmi, "Aplikasi SMS Spam Filtering pada Android menggunakan Naive Bayes," Universitas Pendidikan Indonesia, 2016