Low Time Complexity Model for Email Spam Detection using Logistic Regression

Zubeda K. Mrisho[1], Anael Elkana Sam[3]
School of Computational and Communication Science and Engineering, The Nelson Mandela Institution of Science and Technology, Arusha, Tanzania

Jema David Ndibwile[2]
College of Engineering
Carnegie Mellon University Africa
Kigali, Rwanda

***Abstract*—Spam emails have recently become a concern on the Internet. Machine learning techniques such as Neural Networks, Naïve Bayes, and Decision Trees have frequently been used to combat these spam emails. Despite their efficiency, time complexity in high-dimensional datasets remains a significant challenge. Due to a large number of features in high-dimensional datasets, the intricacy of this problem grows exponentially. The existing approaches suffer from a computational burden when thousands of features are used (high-time complexity). To reduce time complexity and improve accuracy in high-dimensional datasets, extra steps of feature selection and parameter tuning are necessary. This work recommends the use of a hybrid logistic regression model with a feature selection approach and parameter tuning that could effectively handle a big dimensional dataset. The model employs the Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction method to mitigate the drawbacks of Term Frequency (TF) to obtain an equal feature weight. Using publicly available datasets (Enron and Lingspam), we compared the model's performance to that of other contemporary models. The proposed model achieved a low level of time complexity while maintaining a high level of spam detection rate of 99.1%.**

Keywords*—Machine learning; feature selection; feature extraction; parameter tuning

I. Introduction

Email is an online application that enables the exchange of data using electronic devices [1]. Email communication is quick, inexpensive, easy to duplicate, and widely available. Email can extremely be beneficial to businesses and organizations because it allows for the efficient, productive, and effective transmission of all types of electronic data [2]. Email communication began in the 1960s with the restricted functionality of sending information to users within the same computing environment only [3]. Recently, email has become the most common way of communication [4], serving users across computing platform environments. The average number of emails exchanged per day reached 293 billion in 2019 and is forecasted to reach 347 billion by the end of 2023 [5].

Despite its importance, email has become a vehicle for a variety of malicious programs [6]. It is estimated that 50% of all emails are spam [1]. Email spam, also known as junk mail, refers to any form of undesired, uninvited digital communication sent in large quantities [7]. Spam is usually sent via email [8] but can also be delivered via text messages, phone calls, or other social media platforms. Spam has been a big challenge, disturbing users and consuming their time. Spam also leads to phishing attacks, storage space misuse, decreased internet speed, and theft of critical information [5]. The financial losses caused by email spam are estimated to reach a total of USD 257 billion between 2012 and mid-2020 [9]. As a result, substantial negative impacts on the global economy, such as lower productivity have been identified. These factors hinder the development of the communication sector that can benefit governments, individuals, and business companies [10].

To combat the problems, various scientific research studies have been conducted, including the application of machine learning [11]. Previous scientific studies were categorized into three approaches, single-based machine learning, hybrid, and feature engineering [12]. In the first classification, a specific single machine learning algorithm was used to build a spam detection method [12]. Some popular classifications of machine learning algorithms include Naïve Bayes, Random Forest, Support Vector Machines (SVMs), and K-nearest neighbor (KNN) [5].

Support Vector Machines are supervised learning models, which are mostly used to analyze data for regression analysis and classification [13]. Every data item is plotted as a point in n-dimensional space where n is the number of features present with the value of each feature being that of a certain coordinate in the SVM algorithm. The classification is accomplished by finding the hyper-plane that best differentiates the two classes. Support Vector Machines achieves great accuracy on small, clear datasets but performs poorly on larger, noisy datasets with overlapping classes [14].

Naive Bayes is a machine learning classification algorithm commonly used for binary and multi-class classification problems. This algorithm is based on the Bayes Theorem, which states that given the known independent probability of each event and the reverse conditional probability of the pair of events, one may compute an unknown conditional probability of the pair of events [15]. The disadvantage of this method is that it makes assumptions that all attributes are independent, which is incorrect. In fact, by recognizing that some attributes are related, one can create patterns or common attributes from related attributes to minimize the number of features, hence reducing storage consumption.

Random Forest is a classifier that uses the number of decision trees on separate subsets of a dataset and averages their results to enhance its predicted accuracy [16]. Instead of

relying on one decision tree, Random Forest collects the forecasts from every tree and calculates the final output based on the majority vote of predictions. The technique is well-suited to classification problems with small datasets because a large number of trees may make it slow for real-time prediction.

K-nearest neighbor, also called Lazy Learner is another learning algorithm that works well in simple classifications [17]. When an email is classified, KNN tries to find the K-nearest neighbors by calculating the distance in each prediction. In high dimensional datasets, it becomes challenging for the KNN algorithm to compute the distance in each dimension resulting in poor performance.

A combined machine-learning (hybrid) algorithm generates a new line of spam detection methods. The approach combines a specific machine learning algorithm and other methodologies [12]. Wijaya [18] proposed a hybrid decision tree with logistic regression with a focus on reducing noisy data. Another researcher Dedeturk [5] introduced a model which uses logistic regression combined with an artificial bee algorithm. However, this model faces high computation costs.

The feature engineering classification focuses on offering a new set of features. Farisa [19] proposed an intelligent spam detection method and recognizes the relevant features by categorizing spam features into three categories. These are payload, head features, attachment features. Payload features are those that involve the email body, readability, and lexical features [19], while attachment features are the files that are combined within an email. Despite its benefits, this methodology cannot be used when there is an imbalanced dataset [19].

As reviewed, we identify that machine learning is an efficient method for detecting email spam. However, most of the existing models failed to consider the number of features in high-dimensional datasets, leading to high time complexities. Nevertheless, the finding by Majeed [20] shows that time complexity is an important factor to be considered in model development since it reduces the training speed and decreases the importance of the model to be used in online spam filtering [11]. Time complexity depends on the number of features required in a given model as well as whether the proposed method is linear or nonlinear [21]. Xia [22] proposed an approach based on reducing time complexity in rule-based filtering. Nonetheless, this is not currently a recommended approach due to inefficient results that require every time to change the rule.

High-dimensional datasets are datasets with many features. It is the excess number of features that leads to a high time complexity and sometimes a low detection rate (meaning low accuracy) [23], as illustrated in (1) - (8).

Recall formula for finding accuracy of the model [24].

$$Accuracy = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ prediction} \quad (1)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Key: TN=True negative, TP=True positive, FN=False negative, and FP=False positive.

Example: Let us take 10,000 features for a high dimensional dataset and 2605 features for a low dimensional dataset with a test size of 0.34. For a high dimensional dataset, the number of correct predictions is described by the confusion matrix Table I.

Then from Table I, TP=1521, TN=1700, FP=136, FN=43.

$$Accuracy = \frac{1521+1700}{1521+1700+136+43} \quad (3)$$

From (3); $Accuracy = \frac{3221}{3400}$ (4)

Therefore, the accuracy of a high-dimensional dataset obtained from (4) = 0.947 (5)

For a low-dimensional dataset, the number of correct predictions is described in Table II.

Then from Table II, TP=98, TN=768, FP=0, FN=10

$$Accuracy = \frac{98+768}{768+0+41+98} \quad (6)$$

From (3); $Accuracy = \frac{866}{907}$ (7)

Therefore, the accuracy for a low-dimensional dataset obtained from (7) = 0.955 (8)

So, from (5) and (8), the accuracy of a high-dimensional dataset seems to be low compared to a low-dimensional dataset.

Therefore, this paper proposes an efficient hybrid model [25] of logistic regression, with the consideration of the time complexity in a high dimensional dataset. Our methodology combines feature extraction, feature selection, and parameter tuning methods. This approach will reduce the time complexity on high-dimensional datasets. It will equally reduce equal feature weight, overfitting, increase training speed and boost performance. The model uses the Big O notation to find the time complexity of different existing models with accuracy starting from 90%. The evaluation involves a calculation of time complexity in terms of the steps required to operate an input.

TABLE I. CONFUSION MATRIX FOR A HIGH DIMENSIONAL DATASET

Actual	Predicted	
	Non-spam	*Spam*
Non-spam	1700	136
Spam	43	1521

TABLE II. CONFUSION MATRIX FOR A LOW DIMENSIONAL DATASET

Actual	Predicted	
	Non-spam	*Spam*
Non-spam	768	0
Spam	41	98

The rest of the paper is structured as follows: the materials and methodology are presented in Section II while the results are discussed in Section III. Finally, the conclusion and future research direction are presented in Section IV.

II. Materials and Methods

A. Experimental Setup

The model was developed using Python (v3.7.1) in the Google Colab (GCC 7.5) environment on a 64-bit Windows operating system, equipped with 8GB of computer Random Access Memory (RAM).

B. Dataset

The experiments were carried out using two datasets derived from a public repository. This helped to validate the accuracy of the model for spam detection. The first dataset was obtained from the Kaggle repository, which was the Enron dataset with 10,000 samples, half of which were spam and half legitimate emails. The second dataset was the Lingspam with 2605 samples, out of which 433 were spam and 2172 legitimate emails. We analyzed the dataset in relation to their balance ratio which is computed by dividing the total number of genuine emails by the total number of spam emails. The balance ratios of Enron and Lingspam were 1 and 5 respectively. The dataset was then split into two, 67% for training and 34% for testing as described in Table III.

C. Pre-Processing

This step involved cleaning the data by removing missing values; transforming the data into a direct format that could be used by machine learning and splitting them for training and testing. Data transformation is a data mining approach that involves changing raw data into a usable format. This is because real-world data is usually inconsistent, inadequate, lacking in specific behaviors or patterns, and rife with mistakes [26]. Data preparation is a tried-and-true approach for overcoming such difficulties. Building a high-performing model needs a careful evaluation of the input data quality. Therefore, the dataset was pre-processed for the suggested model to perform intelligent diagnosis by extracting suitable characteristics from the data. The preprocessing involved several steps such as importation of the data and libraries, cleaning the data by removing missing values; converting the data into a direct format that could be used by machine learning, and splitting them for training and testing. The process of removing missing values and stop words is very important because of their non-informative in the email spam detection process. Apart from removing stop words, characters must also be converted to lowercase before tokenization. In our datasets, no missing values were found, and tokenization was done through the Sklearn library. The splitting test size was 0.34, meaning that 3400 samples of emails for the Enron dataset were used for testing and 6600 for training. For the Lingspam dataset 886 were used for testing and 1719 for training as shown in Table III.

D. Feature Extraction

This step involved converting email messages into a format that could be processed by a machine learning algorithm. Email spam features are obtained from three different methods, namely, the Heuristic approach, Term frequency (TF) analysis, and behavior approach [27]. In the first approach, emails are mined to discover and generate patterns and rules, while in the TF analysis; every word in an e-mail is specified as a feature. The behavior approach builds features based on knowledge about spammers' behavior. This is often gathered via header, attachment, and email flows between groups of e-mail users.

In this study, the Term Frequency Inverse Document Frequency (TF-IDF) method was employed as a feature extraction method. It is a combination of TF and IDF [28]. According to Kadhim [29], this helps to capture features that are more important within the body of an email. The importance of this method is that it reduces the limitation of equal feature weight obtained when TF is used. Term frequency is how many times a term appears in an email and IDF is how many times a term appears in all emails. Suppose an email contains 50 terms, where the term "none" occurs 10 times. Term frequency is obtained as shown in (9) and (10):

$$TF(t) = \frac{\text{Total no.of times a term occur in an email}}{\text{total number of terms in an email}} \quad (9)$$

$$TF(t) = \frac{10}{50} = 0.2 \quad (10)$$

Now let's say we have 5000 emails, and the term "none" occurs 50 times in all emails. Then TF-IDF is obtained as shown in (11) and (12):

$$IDF(t) = log\frac{500}{50} = 2 \quad (11)$$

$$TF - IDF(t) = 0.2 \times 2 = 0.4 \quad (12)$$

Therefore from (12) our TF-IDF (t) is 0.4

E. Feature Selection

Due to the presence of many features in a high-dimensional dataset, feature selection is an important step. This step involves picking up items that are more important to be used in model development [5]. Feature selection leads to less time complexity that increases the potential application in online spam filtering. Training an algorithm using all the features requires a large amount of memory and high time complexity [30]. Hence, reducing the number of features is very important, since it permits the machine learning algorithm to train faster due to the reduction of the number of steps taken to train the model. Additionally, reducing the number of features also eliminates overfitting [31]. This happens when the model fits more data than it needs and starts catching noisy and inaccurate data. Hence, the efficiency and accuracy of the model decrease.

To reduce the time complexity problem, our research used the Sklearn library, which implements the SelectKBest feature selection method. This method only selects the highest scoring features. It is a wrapper method that uses the score function of Chi-square to obtain the features. By using Chi-square only 450 features were selected. Chi-square is a mathematical formula used to determine if there is a relationship between the features and select those with the highest score only as indicated in (13).

$$\text{Chi-square formula} = \sum \frac{(O_{I-E_i})^2}{E_I} \quad (13)$$

TABLE III. DATASET SAMPLE, TRAINING AND TESTING SIZE

Dataset type	Sample size	Training	Test
Enron	10,000	6600	3400
Lingspam	2605	1719	886

Whereby O_i is the number of the class observed and E_i is the number of expected classes when there is no relationship between the feature and the target [32].

F. Proposed Model

Logistic regression is among the most commonly used algorithms for classification [33]. It is an efficiency model with low time complexity [25]. It is used to determine discrete data from a set of variables [34]. In logistic regression, instead of applying a line, we apply an "S" shape that determines the two largest values [35]. The "S" shape is called the logistic function [36] as shown in Fig. 1. It is used to convert every real value between 0 and 1 into another value [37]. The function uses the threshold value, which determines the likelihood of either 0 or 1. A value beyond 0.5 is 1 and below 0.5 is 0. A logistic regression formula can be formed from the linear equation as indicated in (14) – (16).

However, in logistic regression, y can be 0 - 1, so we divide (14) by 1-y. When y=0 we get 0, and when y=1 we get infinity. To match the equation, we need to transform (15) into a logarithm.

$$y = c + m_1x_1 + m_2x_2 + \cdots + m_nx_n \quad (14)$$

$$\frac{y}{1-y} = c + m_1x_1 + m_2x_2 + \cdots + m_nx_n \quad (15)$$

$$log[\frac{y}{1-y}] = c + m_1x_1 + m_2x_2 + \cdots + m_nx_n \quad (16)$$

After transformation, the formula obtained in (16) can be used for logistic regression.

In this research, the logistic regression was trained with 450 features obtained from the SelectKBest as identified in Fig. 2. The results were then optimized using random search with several parameters as shown in Table IV.

G. Parameter Tuning

When training the models in this study, the hyper-parameters were searched to find the ones with the best performance. A random search was used with the parameters as described in subsections 1-3 and values are presented in Table IV.

1) Penalty: This parameter has two options; ridge (L2) or lasso(L1). Both parameters are used in a regression method to reduce the time complexity of the model. However, while ridge is better for a high-dimensional dataset, lasso is better for a low-dimensional dataset.

2) Solver: This parameter has five solvers which are lbfgs, liblinear, sag, saga, and newton-cg. Liblinear is a decent choice for small datasets, while sag and saga are quicker for big ones [38]. Only lbfgs, sag, newton-cg, and saga can handle multinomial loss in multiclass issues.

3) The Inverse of Regularization Strength (c): It is a logistic regression trade-off parameter that affects the intensity of regularization. The larger values of c correlate to less regularization (where we can specify the regularization function).

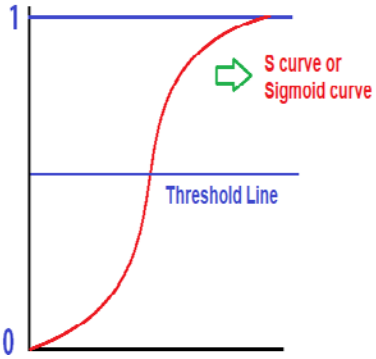

Fig. 1. Logistic Regression Graph [39].

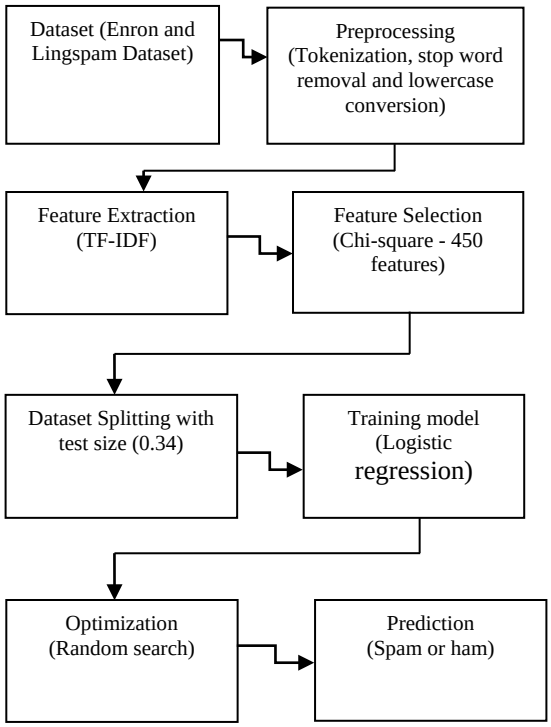

Fig. 2. A Proposed Logistic Regression Model Diagram.

TABLE IV. HYPERPARAMETER USED

Parameter	**Value**
Penalty	L1 and L2
Solver	Saga, sag ,lbfgs, newton-cg, liblinear
C	0.001,0.01,0.1,10,1000

H. Evaluation of the Classifiers

The evaluation of the classifier was evaluated by analyzing the model performance and time complexity during the training and testing procedures. In this study, evaluation was carried out by employing the confusion matrix and Big O notation. The evaluation measures utilized were accuracy, precision, recall, and time complexity each of which is described in subsections 1 – 4 below.

1) Confusion matrix: This is a table that defines the model's overall performance and displays the proper and wrong classifications for each class. Confusion matrix plots are used to show the trained model's ability in guessing the classes of data included in the set of test data. The test set evaluates a model's expected future performance. Table V shows the structure of the confusion matrix using our proposed model. Where TP = True Positive: the number of emails with spam and grouped as having spam, TN = True Negative: the number of emails without spam and grouped as not having spam, FP = False Positive: the number of emails with no spam and grouped as having spam, and FN = False Negative: the number of emails with spam and grouped as not having spam.

a) Accuracy: It computes the frequency with which predictions and labels are equivalent. The accuracy is calculated as shown in (17).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

b) Precision: The number of correctly identified results divided by the total number of positive outcomes results. Equation (18) describes how precision is obtained.

$$Precision = \frac{TP}{TP+FP} \quad (18)$$

c) Recall: The number of accurately recognized positive findings by the total number of samples that should have been positive. A recall is obtained as indicated in (19).

$$Recall = \frac{TP}{TP+FN} \quad (19)$$

2) Big O notation: This is a mathematical study used to describe the complexity of different algorithms [40]. Time complexity is divided into two categories: the number of inputs an algorithm takes to operate and if the algorithm is linear or nonlinear [21]. Each algorithm in machine learning has its formula for finding the time complexity in terms of the steps used to operate an input. This research uses a logistic regression formula to find the time complexity of the proposed model due to its low time complexity compared to others. Table VI presents the formula for finding the time complexity of the different algorithms in machine learning [41]. Given:

O=growth rate of a model.

C=number of the class label for spam is 2spam or ham).

d=number of input/features.

k=number of neighbors, number of support vector.

e=number of epochs.

n=number of neurons.

III. Result and Discussion

This section discusses the results of the proposed model as obtained from the experiments carried out in this research. The results are divided into two parts, the first part shows the time complexity obtained using the Big O notation method as identified in Table VII. The second part shows the performance of the logistic regression when combined with TF-IDF and feature selection method in terms of precision, accuracy, F1-score, and recall as presented in Tables VIII and IX for Enron and Lingspam, respectively.

A. Complexity Result

In machine learning, the time complexity of the model is measured by two things; the type of algorithm used and the number of inputs an algorithm takes to operate. In our model, we used logistic regression which is linear. The advantage of a linear algorithm is its low time complexity relative to non-linear algorithms. Furthermore, when considering the number of inputs, the study used the Big O notation to describe the time complexity of the model as described in Table VI. The result shows that the proposed model attained low time complexity compared to other conventional models as shown in Table VII.

B. Performance Result Analysis

The classifier was evaluated by analyzing the model performance during the training and testing the outcomes. In this study, evaluation was carried out using the confusion matrix as shown in Fig. 3 and 4 whereby 0 represents non-spam and 1 represents spam. The evaluation measures utilized were accuracy, precision, F1 score, and recall. The results showed that saga and L2 parameters are very resourceful parameters in a high-dimensional dataset compared to other solvers because of their performance. Nevertheless, the results show that the feature selection method is an important part to be considered in model development since it reduces the computation time while the optimization process increases model accuracy. The proposed model was compared to other conventional methods and the results showed that the performance of the proposed model was higher than other models as indicated in Table X.

TABLE V. Confusion Matrix Structure

Actual	Predicted	
	Non-spam	*Spam*
Non-spam	TN	FP
Spam	FN	TP

TABLE VI. Formula for Time Complexity

Algorithm	Formula
K-nearest neighbor	O (knd)
Logistic regression	O (nd)
SVMs	O (n^3)
Decision tree	O (n*log (n)*d)
Naïve Bayes	O (n*d)
Deep learning	O(c*d*e*n)

TABLE VII. RESULTS AND COMPARISON ANALYSIS FOR TIME COMPLEXITY

Author	Algorithm and Accuracy obtained	Number of features selected	Time complexity in the training phase(step used)
[5]	Logistic regression- 98.4%	500	O(cd)=2*500 1000 steps
[42]	Deep learning-96.43%	3000	O(c*d*e*n)=2*3000*2*2 24000 steps required
[43]	Naive Bayes-96.87 %	1319	O(c*d)=2*1319 2628 steps required
[44]	Naïve Bayes-96.63%	1000	O(c*d)=2*1000 2000 steps required
[12]	Neural network-96.8%	140	O(c*d*e*n)=2*140*600 168,000 steps required
Proposed approach	Logistic regression- 99.1% & 98.3%	450	O(cd)=2*450

TABLE VIII. RESULTS OF MODEL PERFORMANCE FOR ENRON

Label	Accuracy	Precision	Recall	F1-score
Non-spam(0)	0.98	0.99	0.97	0.98
Spam(1)		0.97	0.99	0.98

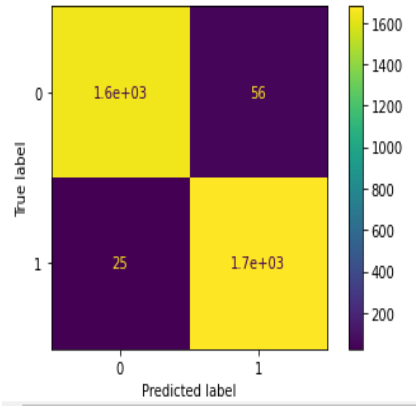

Fig. 3. A Confusion Matrix of Enron Dataset.

TABLE IX. RESULTS OF MODEL PERFORMANCE FOR LINGSPAM

Label	Accuracy	Precision	Recall	F1-score
Non-spam(0)	0.99	0.98	1.00	0.99
Spam(1)		1.00	0.91	0.95

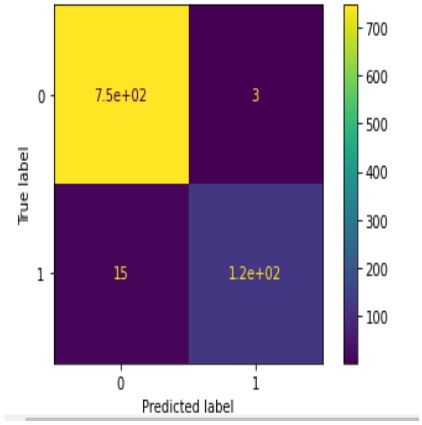

Fig. 4. A Confusion Matrix of Lingspam Dataset.

TABLE X. PERFORMANCE COMPARISON FOR DIFFERENT ALGORITHM

Model	Accuracy
[5]	98.4%
[44]	96.63%
[11]	96.7%
Proposed model	99.1% and 98.3 respectively

IV. CONCLUSION

In this paper, a hybrid logistic regression model was proposed to reduce the time complexity in a high-dimensional dataset that will increase the potential of the model in online spam detection. The model performs three different tasks that are feature extraction, feature selection, and parameter tuning. TF-IDF was used during feature extraction to replace the drawbacks of equal feature weight obtained when TF is used. To increase the training speed in the high-dimensional dataset the model uses Chi-square that helps to select the feature which is related to each other with the highest score only. A random search was used to optimize the model performance. The performed task help to reduce time complexity by decreasing the number of features in a high-dimensional dataset. The model also uses the TF-IDF feature extraction method to reduce the disadvantage of equal feature weight obtained when TF is used. The experiment shows that a better performance of 99.1% is achieved when feature selection is combined with parameter tuning. Overall, it can be concluded that feature selection is an important part of a high-dimensional dataset that helps to reduce an excessive number of features. Nevertheless, for future work, more research is needed in other feature selection and parameter tuning methods.

ACKNOWLEDGMENT

I would like to sincerely thank the Government of the United Republic of Tanzania for funding this research through the Ministry of Education, Science, and Technology (MoEST).

Furthermore, I would like to acknowledge my supervisors, Dr. Jema David Ndibwile and Dr. Anael Sam for their excellent guidance and supervision of this research. Also, I would like to thank my colleague Stephano Amoni for his constant support in this research.

REFERENCES

[1] A. Qashqari, D. Alhbshi, F. Alzahrani, H. Ghwati, and A. Aljahdali, "Electronic Mail Security," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 18, 2020.

[2] P. A. Gloor, A. F. Colladon, and F. Grippa, "The digital footprint of innovators: Using email to detect the most creative people in your organization," *Journal of Business Research*, vol. 114, pp. 254-264, 2020.

[3] M. Kekane, "New technology in business communication " 2020.

[4] C. Dürscheid and C. Frehner, "2. Email communication," in *Pragmatics of computer-mediated communication*, ed: De Gruyter Mouton, 2013, pp. 35-54.

[5] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Applied Soft Computing*, vol. 91, p. 106229, 2020.

[6] T. A. Kemp, M. C. Depaolis, W. R. Gemza, and R. J. Whalen, "Electronic mail security system," ed: Google Patents, 2020.

[7] R. K. Kumar, G. Poonkuzhali, and P. Sudhakar, "Comparative study on email spam classifier using data mining techniques," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2012, pp. 14-16.

[8] G. Fumera, I. Pillai, and F. Roli, "Spam filtering based on the analysis of text information embedded into images," *Journal of Machine Learning Research*, vol. 7, 2006.

[9] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168261-168295, 2019.

[10] R. Broadhurst and M. Alazab, "Spam and crime," *Regulatory Theory*, p. 517, 2017.

[11] A. Zamir, H. U. Khan, W. Mehmood, T. Iqbal, and A. U. Akram, "A feature-centric spam email detection model using diverse supervised machine learning algorithms," *The Electronic Library*, 2020.

[12] H. Faris, A.-Z. Ala'M, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah, *et al.*, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks," *Information Fusion*, vol. 48, pp. 67-83, 2019.

[13] N. Kumar and S. Sonowal, "Email Spam Detection Using Machine Learning Algorithms," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 108-113.

[14] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan, and M. javad Rajabi, "Advantage and drawback of support vector machine functionality," in *2014 international conference on computer, communications, and control technology (I4CT)*, 2014, pp. 63-65.

[15] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve Bayes," *Encyclopedia of machine learning*, vol. 15, pp. 713-714, 2010.

[16] D. DeBarr and H. Wechsler, "Spam detection using random boost," *Pattern Recognition Letters*, vol. 33, pp. 1237-1244, 2012.

[17] L. Firte, C. Lemnaru, and R. Potolea, "Spam detection filter using KNN algorithm and resampling," in *Proceedings of the 2010 IEEE 6th International Conference on Intelligent Computer Communication and Processing*, 2010, pp. 27-33.

[18] A. Wijaya and A. Bisri, "Hybrid decision tree and logistic regression classifier for email spam detection," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2016, pp. 1-4.

[19] Farisa, "An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks," *Information Fusion 48 (2019) 67–83*, vol. 48 2019.

[20] L. Chwif, M. R. P. Barretto, and R. J. Paul, "On simulation model complexity," in *2000 winter simulation conference proceedings (Cat. No. 00CH37165)*, 2000, pp. 449-455.

[21] A. Majeed, "Improving time complexity and accuracy of the machine learning algorithms through selection of highly weighted top k features from complex datasets," *Annals of Data Science*, vol. 6, pp. 599-621, 2019.

[22] T. Xia, "A Constant Time Complexity Spam Detection Algorithm for Boosting Throughput on Rule-Based Filtering Systems," *IEEE Access*, vol. 8, pp. 82653-82661, 2020.

[23] J. Fan, F. Han, and H. Liu, "Challenges of big data analysis," *National science review*, vol. 1, pp. 293-314, 2014.

[24] V. M. Patro and M. R. Patra, "Augmenting weighted average with confusion matrix to enhance classification accuracy," *Transactions on Machine Learning and Artificial Intelligence*, vol. 2, pp. 77-91, 2014.

[25] Y. Han, M. Yang, H. Qi, X. He, and S. Li, "The Improved Logistic Regression Models for Spam Filtering," in *2009 International Conference on Asian Language Processing*, 2009, pp. 314-317.

[26] K. A. Kaufman and R. S. Michalski, "Learning from inconsistent and noisy data: the AQ18 approach," in *International Symposium on Methodologies for Intelligent Systems*, 1999, pp. 411-419.

[27] B. Al-Shboul, H. Hakh, H. Faris, I. Aljarah, and H. Alsawalqah, "Voting-based Classification for E-mail Spam Detection," *Journal of ICT Research & Applications*, vol. 10, 2016.

[28] M. A. Hassan and N. Mtetwa, "Feature extraction and classification of spam emails," in *2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 2018, pp. 93-98.

[29] A. I. Kadhim, "Term weighting for feature extraction on Twitter: A comparison between BM25 and TF-IDF," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 2019, pp. 124-128.

[30] M. Diale, T. Celik, and C. Van Der Walt, "Unsupervised feature learning for spam email filtering," *Computers & Electrical Engineering*, vol. 74, pp. 89-104, 2019.

[31] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, pp. 16-28, 2014.

[32] D. S. Moore, "A chi-square statistic with random cell boundaries," *The Annals of Mathematical Statistics*, pp. 147-156, 1971.

[33] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: a methodology review," *Journal of biomedical informatics*, vol. 35, pp. 352-359, 2002.

[34] R. E. Wright, "Logistic regression," 1995.

[35] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The journal of educational research*, vol. 96, pp. 3-14, 2002.

[36] A. DeMaris, "A tutorial in logistic regression," *Journal of Marriage and the Family*, pp. 956-968, 1995.

[37] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*: Wiley New York, 2000.

[38] Y. Tao, J. Jiang, Y. Liu, Z. Xu, and S. Qin, "Understanding performance concerns in the API documentation of data science libraries," in *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2020, pp. 895-906.

[39] P. Das. (2021). *Logistics Regression in python*. Available: https://www.codespeedy.com/logistics-regression-in-python/.

[40] M. J. Kearns, The computational complexity of machine learning: MIT press, 1990.

[41] A. Abdiansah and R. Wardoyo, "Time complexity analysis of support vector machines (SVM) in LibSVM," *International journal computer and application*, vol. 128, pp. 28-34, 2015.

[42] A. Tyagi, "Content based spam classification-a deep learning approach," Graduate Studies, 2016.

[43] M. Esmaeili, A. Arjomandzadeh, R. Shams, and M. Zahedi, "An anti-spam system using naive Bayes method and feature selection methods," *International Journal of Computer Applications*, vol. 165, pp. 1-5, 2017.

[44] S. Douzi, F. A. AlShahwan, M. Lemoudden, and B. El Ouahidi, "Hybrid Email Spam Detection Model Using Artificial Intelligence," *International Journal of Machine Learning and Computing*, vol. 10, 2020.