Journal of Information Technology and Computing
https://jitc.sabapub.com
ISSN: 2709-5916

2023 Volume 4, Issue 1 : 15 – 28
DOI : 10.48185/jitc.v4i1.653

A Lightweight Machine Learning-Based Email Spam Detection Model Using Word Frequency Pattern

Mohamed Aly Bouke [1,*], Azizol Abdullah[1], Mohd Taufik Abdullah[1], Saleh Ali Zaid[1], Hayate El Atigh [2], Sameer Hamoud ALshatebi[1]

[1]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia.
[2]Departement Computer Engineering, Faculty of Computer Engineering, Bandirma Onyedi Eylul University, Balikesir 10200, Turkey.

Received: 01.02.2023 • Accepted: 06.05.2023 • Published: 27.06.2023 • Final Version: 27.06.2023

Abstract: This paper proposes a lightweight machine learning (ML) based spam detection model using word frequency patterns and the Random Forest (RF) algorithm to address the limitations of existing methods. The proposed model considers the class imbalance issue through a random oversampling strategy to improve the efficiency and effectiveness of spam detection. The performance of the proposed model was evaluated using the spambase dataset, which consists of 4,601 email samples and 58 features sourced from the University of California, Irvine (UCI) ML Repository. Our model achieved an overall accuracy of 97% for precision, recall, and F-score. Additionally, comparisons with existing state-of-the-art methods showed that our model outperforms others, with an improvement of 6% for all evaluation metrics. To address the class imbalance issue in the dataset, we adopted a random oversampling strategy that involved duplicating random instances of the minority class to balance the class distribution. This approach aims to improve the efficiency and effectiveness of the spam detection model by providing a more balanced dataset for training.

Keywords: Spam Detection, Random Forest, Machine Learning

1. Introduction

Email has become an integral part of the lives of millions of people worldwide. Because it is the cheapest, most popular, and fastest mode of communication, it has transformed how people cooperate and work [1]. Email spam is undoubtedly a constant source of frustration for network operators and consumers. Spam imposes various costs, ranging from network bandwidth, processing, and storage expenses to user productivity [2]. Keeping spam away from users without wrongly dropping proper communication is a big challenge; in addition, the storage requirements of maintaining a "trash" folder for users to select if they suspect a lost message is costly [3].

Moreover, spam is an arms race in which spammers continuously create new ways to get around filters by updating their techniques to bypass filtering controls. At the same time, network administrators should continually improve their tools and databases to keep spam out of their users' hands [4]. As a result, while end-users may believe that spam is a largely "fixed" problem, the reality for administrators and operators is quite different [4]. A new line of research has recently arisen that focuses on non-content factors to develop algorithms that distinguish between spam and

[*] Corresponding Author: bouke@ieee.org

a valid email (often referred to as "ham") [5]. In the past, users could manually screen spam emails sent from several addresses. However, spammers today easily avoid all such spam filtering controls. ML algorithms represent a new approach that can detect spam emails. A training dataset is used to train the ML model to achieve this goal. Training datasets are samples of emails that have been pre-classified. ML methodologies can be used efficiently in Email filtering using one of the many algorithms available [6]. These popular methodologies comprise Support Vector Machines (SVM), K-nearest neighbor, Nave Bayes (NB), Decision Tree (DT), RF, and other algorithms [7]. However, developing an effective spam detection model poses a challenge in dealing with imbalanced datasets [8], which can result in biased models performing poorly on the minority class. In this study, we address this issue by adopting a data-balancing approach that duplicates random instances of the minority class to balance the class distribution. This approach aims to improve the efficiency and effectiveness of the spam detection model by providing a more balanced dataset for training. To identify spam emails, we propose a word frequency pattern-based approach. However, achieving an accurate and effective spam detection model requires addressing the class imbalance issue in the spambase dataset. Therefore, we designed a data-balancing approach to overcome this challenge. The data balancing approach contributes to the efficiency and effectiveness of the proposed model by providing a more balanced dataset for training.

This study proposes a lightweight RF algorithm-based spam email detection model to tackle the issue. The proposed model uses a word frequency pattern-based approach to identify spam emails. The following are the primary contributions of this research:

1. A spam detection model based on the RF algorithm is developed to efficiently detect and identify spam emails.

2. A data balancing approach is designed to address the imbalance dataset issue in the spambase1 dataset.

The rest of the paper is structured as follows: Section II reviews similar work on spam detection models. Section III describes and discusses our lightweight spam detection model. The experiments and results discussion of the proposed model is presented in Section IV. Section V compares our work with baseline work, and The final section wraps up the paper and offers the conclusion of this work.

2. Background work

Email spam classification is a crucial issue in today's digital world as it consumes time and irritates recipients. With the widespread use of electronic mail, spam emails have become an increasingly prevalent problem. To handle this issue, ML algorithms have been widely adopted to develop spam detection systems. These algorithms are effective in identifying whether an email is solicited or unsolicited. However, existing spam detection techniques have limitations, such as low detection rates and the inability to handle high-dimensional data. Hence, there is a need for more advanced and efficient spam classification models to ensure a smooth and spam-free email experience for users. In recent years, researchers have proposed several effective ML-based spam detection models, demonstrating promising results in accurately classifying spam emails.

An ML-based methodology for classifying email datasets was proposed by Harisinghaneyin [9], who discovered that the K-nearest neighbors (KNN), Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Naive Bayes algorithms generated superior outcomes.

[1] https://archive.ics.uci.edu/ml/datasets/spambase

According to Debarr and Wechsler [10], logistic regression and support vector machines perform better than naive Bayes; however, the effectiveness of these algorithms depends upon the features in the spam detection model. A spam email filtering system that supports both English and Malay languages and makes use of Term Frequency and Inverse Document Frequency feature selection methods (DFFS) was proposed by Mohamad and Selamat [11]. A feedforward neural network trained with the Krill Herd algorithm was used by Faris et al. [12] to detect spam. A binary differential evolution (BDE) method for support vector machines was given by Hamed et al. [13]. They used the correspondence coefficient as a fitness function, resulting in an overall accuracy of 93.99%. Using the random forest as a classifier and the methods of least redundancy and maximum relevance for feature selection, Sri and Karthika [14] achieved an accuracy of 86%. To reduce feature dimensionality, Saleh [15] suggested a hybrid method employing Chaotic Particle Swarm Optimization and Artificial Bees Colony, which had a 90.81% accuracy. Using the Spambase Dataset, Soleimanian, and Mousavi [16] examined how well different network models performed and discovered that just 10 of the 57 features could obtain a classification accuracy of 91.7%. Last, Khamis et al. [17] suggested an SVM Header-Based Email Spam detection. The tests, which used the Anomaly Detection Challenges and Cyber Security Data Mining 2010 datasets as their test subjects, yielded accuracy rates of 88.80% and 87.20%, respectively, for identifying anomalies in email data.

To this end, the comparison and evaluation of ML-based email spam detection models can be challenging due to various factors such as the size of the dataset, accuracy, and consistency, the number of features, and the experimental parameters. These factors make it difficult to determine the superiority of one ML method over another. To address this challenge, we reviewed previous datasets used for spam classification and compiled a table summarizing their characteristics and limitations (see Table 1). As seen in the table, many previous datasets have limitations, such as the limited number of records, from a single year or only containing a specific type of spam (e.g., image-based or phishing emails).

Table 1. Common Datasets Used for Email Spam Classification.

Dataset	Spam records	Normal records	Year	Limitations
Spamemail	1378	2949	2010	A limited number of records
Hunter	928	810	2008	A limited number of records
Trec 2007	50,199	25,220	2007	The dataset is from a single year
Princeton spam image Benchmark	1071	0	2007	Only contains image-based spam
Dredze image spam Dataset	3297	2021	2007	Only contains image-based spam
Enron-spam	20170	16545	2006	Contains only spam emails from the Enron Corporation
Trec 2006	24,912	12,910	2006	The dataset is from a single year
Gen spam	31,196	9212	2005	A limited number of normal records
Trec 2005	52,790	39,399	2005	The dataset is from a single year
Biggio	8549	0	2005	Only contains image-based spam
Phishing corpus	415	0	2005	Only contains phishing emails
Zh1	1205	428	2004	A limited number of records

PU2	142	579	2003	A limited number of spam records
PU3	1826	2313	2003	A limited number of records
PUA	571	571	2003	The dataset is from a single year
Spamassassin	1897	4150	2002	Only contains a limited number of spam records
Lingspam	481	2412	2000	Only contains English language spam
PU1	481	618	2000	A limited number of records
Spambase	1813	2788	1999	The dataset is from a single year
Spam archive	15090	0	1998	Only contains spam emails from a specific period

This paper presents a novel RF-based email spam detection model that considers the dataset's imbalance before building the prediction model. The proposed model is tested on a real-world dataset containing 4601 records and follows the standard ML process, particularly in the early stages of model development.

3. Methodology

This section presents our lightweight email spam ML-based detection model methodology. We started by exploring the email spam dataset, preprocessing raw data, and developing the classification model. Figure 1 presents the overall architecture of the proposed model. The proposed model is based on the popular ML-supervised RF algorithm mentioned above. The research framework involves the following main steps :

1) Dataset selection

2) Dataset cleaning, languages, tools selection, and experimental environment setup.

3) Dataset balancing status investigation and implementation of the random oversampling strategy.

4) The fourth step includes data scaling using the standard scaler method.

5) Dataset splitting, we split the oversampled dataset into training and testing sets; 20% of the dataset is used as testing samples, while 80% is used for training.

6) Training the model with the resampled training set (80% of the resampled dataset)

7) Test the model using the test set (20% of the resampled dataset).

8) Evaluating the model performance using evaluation metrics such as Fscroce, Recall, Precision, and Accuracy. Additionally, We used visualization tools to evaluate the model intensely, such as the confusion matrix and ROC.

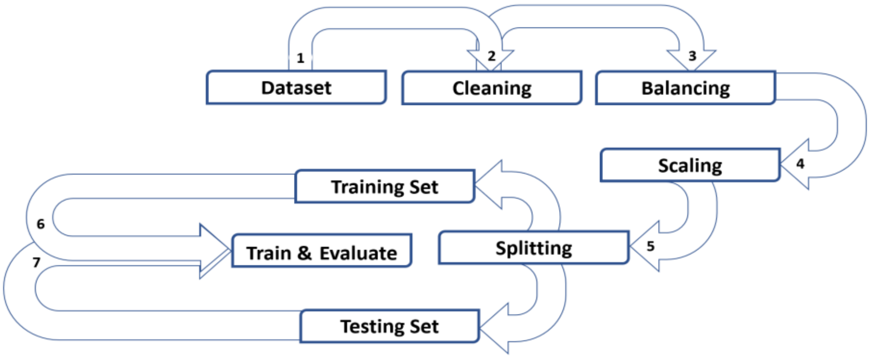

Figure 1. The structure of the suggested model.

3.1. Spam Email Dataset

Hopkins and colleagues collected and developed the spambase dataset used in this study [18]. The dataset contains 4601 email samples with 58 features (see Table 2) , including 1813 spam and 2788 ham samples. The first 48 features in the dataset were generated from commonly occurring words, while the following six features represent the frequency of specific characters. The final three features capture the repetition of letters (upper and lower case) in the email text. The class label attribute in the dataset indicates whether an email is a spam (1) or ham (0). While the spambase dataset is widely used and well-established in the field, it is essential to note that it has some limitations and biases. For instance, the dataset was collected in 1999 and may not represent the current spam email landscape. Additionally, the dataset may be biased towards English-language emails, limiting the results' generalizability to other languages. Therefore, caution should be exercised in interpreting the results, and further research may be needed to validate the findings on other datasets.

Table 2. Spambase dataset features.

	Feature	Type		Feature	Type
1	word_freq_make	Float	30	word_freq_labs	Float
2	word_freq_address	Float	31	word_freq_telnet	Float
3	word_freq_all	Float	32	word_freq_857	Float
4	word_freq_3d	Float	33	word_freq_data	Float
5	word_freq_our	Float	34	word_freq_415	Float
6	word_freq_over	Float	35	word_freq_85	Float
7	word_freq_remove	Float	36	word_freq_technology	Float
8	word_freq_internet	Float	37	word_freq_1999	Float
9	word_freq_order	Float	38	word_freq_parts	Float
10	word_freq_mail	Float	39	word_freq_pm	Float
11	word_freq_receive	Float	40	word_freq_direct	Float
12	word_freq_will	Float	41	word_freq_cs	Float

13	word_freq_people	Float	42	word_freq_meeting	Float
14	word_freq_report	Float	43	word_freq_original	Float
15	word_freq_addresses	Float	44	word_freq_project	Float
16	word_freq_free	Float	45	word_freq_re	Float
17	word_freq_business	Float	46	word_freq_edu	Float
18	word_freq_email	Float	47	word_freq_table	Float
19	word_freq_you	Float	48	word_freq_conference	Float
20	word_freq_credit	Float	49	char_freq_;	Float
21	word_freq_your	Float	50	char_freq_(	Float
22	word_freq_font	Float	51	char_freq_[	Float
23	word_freq_000	Float	52	char_freq_!	Float
24	word_freq_money	Float	53	char_freq_$	Float
25	word_freq_hp	Float	54	char_freq_#	Float
26	word_freq_hpl	Float	55	capital_run_length_average	Float
27	word_freq_george	Float	56	capital_run_length_longest	Integer
28	word_freq_650	Float	57	capital_run_length_total	Integer
29	word_freq_lab	Float	58	is_spam	Binary

3.2. Language and Tools

In this study, we utilized Python programming language, specifically version 3.9.13, due to its versatility, simplicity, and suitability for creating practical ML and artificial intelligence applications [19]. Relevant libraries used in this study include library sci-kit-learn version 1.2.1, matplotlib version 3.7.1, numpy version 1.24.2, and Pandas version 1.4.4. Furthermore, Jupyter Notebook was used to manage the implementation environment. Jupyter Notebook is an open-source, browser-based application that has become a powerful tool for academic purposes by allowing for the sharing of documentation and source codes. Jupyter Notebook provides an interactive computational environment that combines code, text, and visualizations, making it easy to develop, test, and document code. One significant advantage of Jupyter Notebook is its ability to allow users to execute code in blocks or cells, allowing it to test small portions of code before running the entire program. In this study, we conducted the empirical experiments on a computer with an Intel Core i5-5300U CPU, running at 2.30 GHz, with 8 GB of RAM, and a 64-bit Windows 10 operating system.

3.3. Preparing Raw Email Spam Data

Data preparation includes feature encoding, scaling, and balancing depending on the spam dataset's properties. However, since all dataset features are numeric, as shown in Table 1, We will skip the features encoding step.

3.4. Feature scaling:

Feature scaling is essential in preparing the data for machine learning models. Scaling helps to prevent algorithms from being affected by significant or small-scale differences in the input data. For example, consider a dataset with vastly different ranges of values, where one feature varies between 0 and 1 while the other ranges between 1 and 1000. Without feature scaling, the model may give more weight to the feature with a more extensive range, leading to biased results.

Normalization and standardization are two commonly used methods for feature scaling. Normalization scales the values of each feature to a range of 0 to 1, while standardization scales the values to have a mean of 0 and a standard deviation of 1. Both methods have advantages and disadvantages, and the choice depends on the nature of the data and the model being used. For example, normalization is preferred when the data is uniformly distributed, while standardization is useful for normally distributed data. To illustrate the concept of feature scaling, consider a dataset containing two features: age and income. Age ranges from 18 to 80, while income ranges from $10,000 to $100,000. Without scaling, the income feature would have a much more extensive range of values and could dominate the prediction model. By applying feature scaling, age and income can be scaled to the same range, preventing this bias and improving the model's accuracy. To better understand the effects of feature scaling, Figure 2 illustrates the distribution of two features, height, and weight, before and after scaling. The left plot shows the original data, where height ranges from 140 to 200 cm and weight ranges from 40 to 120 kg. The right plot shows the same data after normalization, where height and weight are scaled from 0 to 1. The figure highlights the importance of feature scaling in preparing the data for machine learning models.

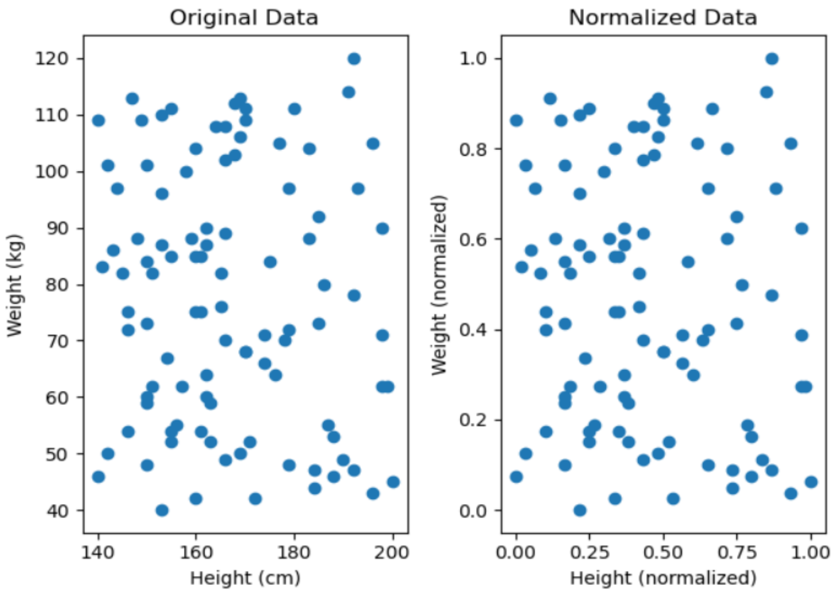

Figure 2. The distribution of two features, height and weight, before (left) and after (right) normalization.

The standardization process is shown in equation (1).:

(1) $$z = \frac{x - u}{\sigma}$$

The new scaled value, Z, is calculated based on the distribution mean, μ, and the standard deviation σ of the distribution.

3.5. Data Balancing

An imbalanced classification problem in ML projects arises when a significantly unequal distribution of samples across different classes leads to classification challenges. This imbalance can range from a slight discrepancy to a severe imbalance, with one instance in the minority class and hundreds, thousands, or even millions in the majority class or classes [20]. In our case, the spambase dataset is unbalanced, as illustrated in Figure 2 (before). Moreover, 60.6% of the instances belong to the class ham, whereas only 39.4% belong to the class spam, indicating that our dataset is imbalanced. Because most ML algorithms for classification were created to assume equal class distribution, classifying imbalanced learning is a complex task.

As a result, models will have low prediction accuracy, particularly for the minority class, because the majority class is often more abundant than the minority [21]. To address this imbalance, two standard methods are used: the data-level approach, which adjusts the class imbalance ratio to balance the class distribution, and the algorithm-level approach, which improves the learning process for the minority class [21]. However, the algorithm-level method is ineffective when the imbalance ratio is high, so the data-level approach is preferred and involves modifying the class composition of the data [22], [23]. One commonly used method is resampling, which increases the minority class by removing samples from the majority class and adding samples from the minority class. Two types of resampling, under-sampling, and oversampling, are considered promising.

Random oversampling (ROS) and Random Undersampling (RUS) are the two main ways to resample an imbalanced dataset randomly. ROS lengthens the learning process, mainly if the original dataset is enormous. However, When the dataset size is small, this strategy is ideal. On the other hand, RUS is a form of data sampling that haphazardly chooses some majority class instances and withdraws them from the dataset until the aimed class distribution is attained [24], [25]. As the spambase dataset is small, the loss of some samples due to RUS will significantly impact the dataset quality. As a result, we selected a ROS-based sampling approach. Therefore, we focused on the data-level approach and used the ROS.

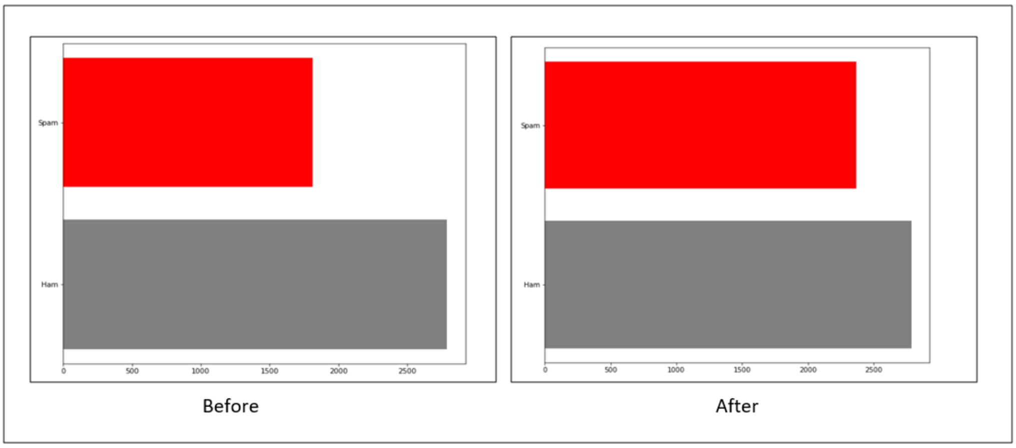

Figure 3. Data Balancing Status.

Figure 3 illustrates the distribution of the two classes before and after resampling using the ROS method. The left plot shows the original dataset, where the majority class (class 1) is represented by the blue bars, and the minority class (class 0) is represented by the orange bars. The right plot shows the dataset after resampling, where the number of instances in the minority class has been increased through duplication. The figure shows that the dataset has been balanced after resampling, with a significantly reduced distribution gap. Moreover, Table 3 shows the dataset balancing statistics before and after resampling using the ROS method. The original dataset contains 1813 instances of class 0 and 2788 instances of class 1. After applying the ROS method, the number of instances in the minority class increased to 2369 to reduce the gap with the majority class, resulting in a balanced dataset with a total of 5157 instances.

Table 3. Dataset balancing statistics before and after resampling using the ROS method.

Class	Original	After ROS
0 (Spam)	1813	2369
1 (Ham)	2788	2788
Total	4601	5157

4. Experiments

This section summarises the experiments utilizing the spambase dataset explored earlier in the study.

4.1. Performance Metrics

The performance of the proposed model was evaluated using three metrics: Accuracy, Recall, F-score, and Precision. These metrics were calculated using the following quantities:

- True Positives (TP) is the number of instances correctly classified as spam.
- True Negatives (TN) is the number of instances correctly classified as ham.
- False Positives (FP) is the number of ham instances incorrectly classified as spam.
- False Negatives (FN) is the number of spam instances incorrectly classified as ham.

$$Precision\ = TP/(TP+FP) \qquad (2)$$

$$Recall\ (Sensitivity)\ = TP/((TP+TN)\ ) \qquad (3)$$

$$Fscore = 2 \times (Precision \times Recall)/(Precision + Recall) \qquad (4)$$

$$Accuracy = \ ((TP+TN))/((TP+TN+FP+FN)\ ) \qquad (5)$$

4.2. Random Forest Classifier

The suggested model employs the RF, a variant of the decision tree and one of the most popular and powerful ML classification algorithms. Our findings indicate that the proposed model achieved an overall score of 97% in Precision, Recall, and Fscore for the majority class (Ham), While this value slowed down by 1% for the minority class. The fact of imbalanced class distribution can explain this. However, the output is considered good, and the model's accuracy is 97%. Table 4 shows the full classification report for the proposed work.

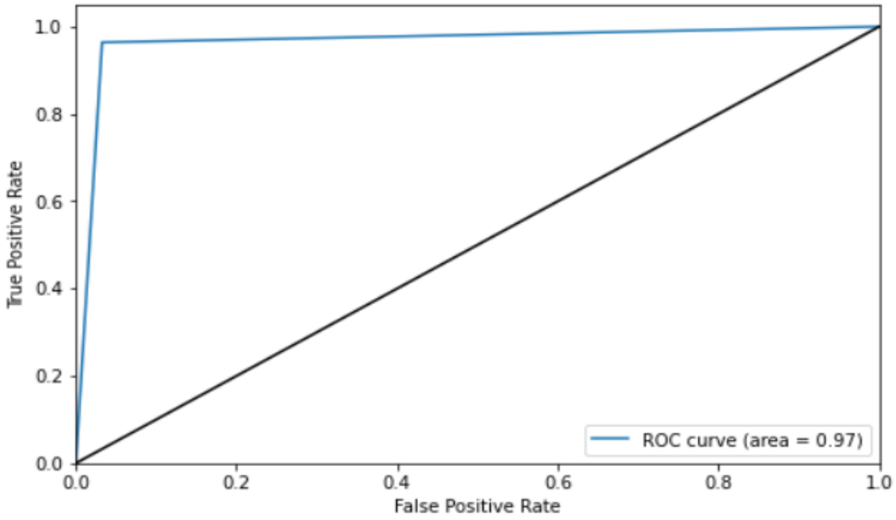

Figure 4. Our proposed model's ROC curve.

Moreover, to confirm the accuracy of the model's output, we utilized receiver operating characteristic (ROC) curves to contrast the True Positive Rate with False Positive Rate over a range of values to predict binary outcomes. ROC is considered a standard way to evaluate ML-based models' results in imbalanced learning [26]. The predictive model is better if the Area under the curve (AUC) is higher. Our proposed model's ROC curve, with AUC = 0.97, is shown in Figure 4.

Table 4. Full classification report of random forest.

	Precision	**Recall**	**F1-score**	**Support**
Ham	0.97	0.97	0.97	831
Spam	0.96	0.96	0.96	717
Accuracy	0.97			
Macro avg	0.97	0.97	0.97	1548
Weighted avg	0.97	0.97	0.97	1548

A confusion matrix is an efficient visualization tool for model output interpretation, allowing us to see all possible classification scenarios. Figure 5 shows the confusion matrix for the proposed model. The matrix indicates that for the class spam, out of 717 instances existing in the testing set, 688 cases are recognized by the model correctly, while 29 are classified wrongly as ham. For the class ham from the total cases in the testing set, which is 831, the model correctly recognized 806 and misclassified 25.

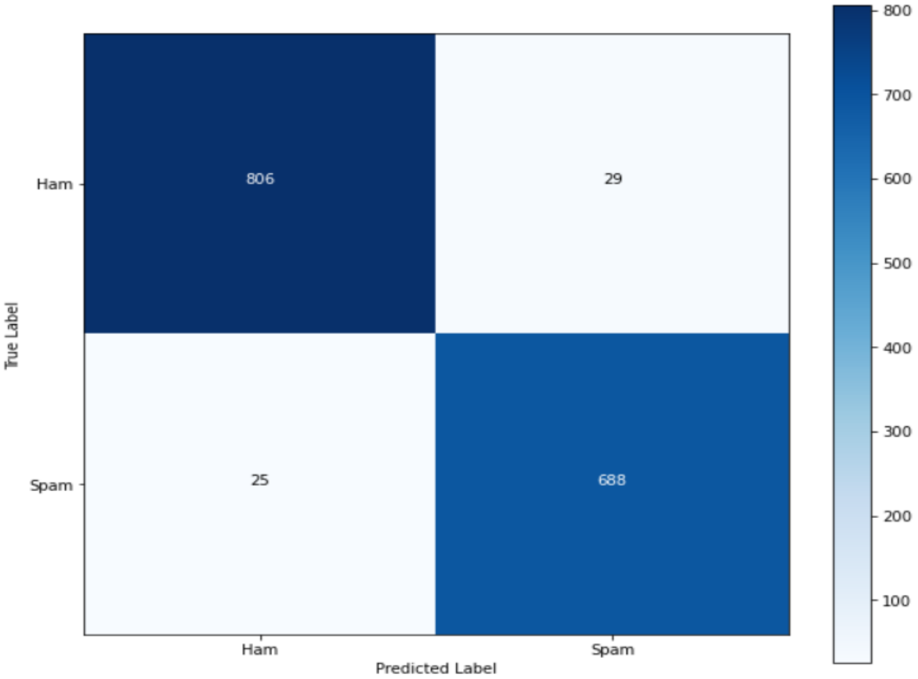

Figure 5. Confusion matrix of the random forest.

5. Work Comparison

Existing research in [16] used the spambase dataset for the proposed work and implemented an RF-based model for classification. The overall score of this work was 92.21%. Nayak et al. [27] proposed an XGBoost-based spam classification model using Spam Email Dataset (SMD), and the outcome achieved an overall accuracy of 88.12%. However, the dataset contains only 1000 records. We used a supervised learning model based on an RF and the spambase dataset in our work. In addition, our approach takes into count the data imbalance status. We used a random oversampling strategy to address this issue, and the model achieved an accuracy of 97% (see Figure 7). Table 5 shows the proposed model's performance compared to benchmark work in Precision, Recall, and Fscore. According to the results, our models performed better for all metrics than other work.

Table 5. Performance of the Proposed Model in Comparison to Other Approaches.

					Dataset	
	Algorithm	Precision	Recall	Fscore	Name	Size
Proposed Work	RF	97%	97%	97%	Spambase	4601
Soleimanian et al. (2020)	RF	89.16%	90.37%	89.76 %	Spambase	4601
Nayak et al. (2021)	XGBoost	92%	88%	88%	SMD	1000

The proposed work's performance was compared to other state-of-the-art methods and traditional machine learning models, namely logistic regression, K-nearest neighbors, decision trees, and support vector machines. The results in Table 6 show that the proposed approach outperformed all traditional models regarding accuracy, precision, recall, and F-score. This suggests that using the random oversampling strategy for addressing imbalanced datasets and the RF-based model can improve classification accuracy.

Table 6. Comparing our model with traditional ML models.

Models	Accuracy	Precision (%)	Recall (%)	Fscore (%)	Dataset
Proposed Work	97	97	97	97	Spambase
Log	92	92.8	88	90	Spambase
KNN	92.3	95.3	84.8	89.9	Spambase
DT	93.8	99.7	88.2	90	Spambase
SVM	93.4	95	89.2	91.7	Spambase

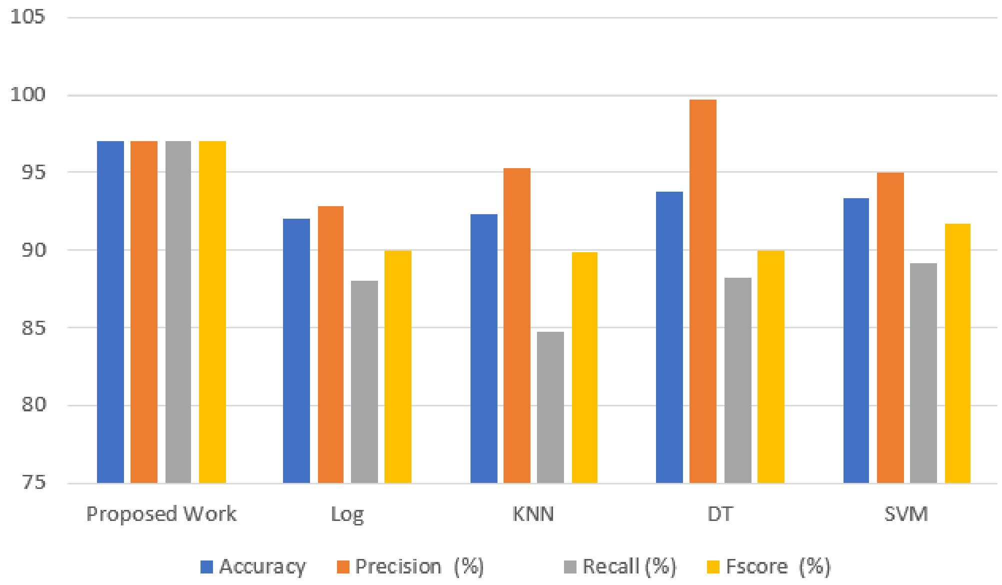

Figure 6. Comparison of proposed models with traditional ML models.

Figure 6 compares the performance of the proposed approach with traditional machine learning models in terms of accuracy, precision, recall, and Fscore. As seen in the figure, our proposed approach outperforms all other models regarding all evaluation metrics. Specifically, the proposed model achieves an accuracy of 97%, significantly higher than all traditional models' accuracy. Furthermore, regarding precision, recall, and Fscore, the proposed approach achieves higher values than all other models. This indicates that our approach is more effective in identifying spam emails and achieving a balance between precision and recall.

Moreover, the figure highlights the importance of addressing imbalanced data in machine learning models. The traditional models perform poorly in imbalanced data, as seen in the lower precision and recall values. However, the proposed approach, which utilizes a random oversampling strategy to address the imbalanced data, achieves significantly higher precision and recall values. This demonstrates the effectiveness of our approach in handling imbalanced data and achieving higher accuracy and performance in spam email classification. Furthermore, our approach achieves higher accuracy, precision, recall, and Fscore values, indicating its potential to be an effective solution for spam email classification.

To this end, the proposed approach shows promise for improving the classification accuracy of imbalanced datasets, and the results highlight the importance of addressing data imbalance in machine learning tasks.

6. Conclusion

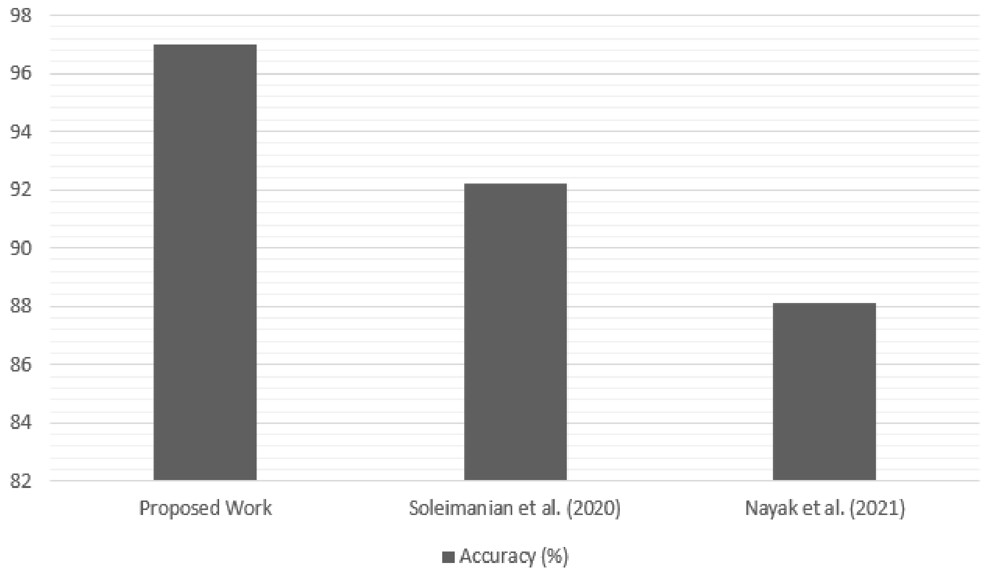

Figure 7. Accuracy comparison of the proposed work with baseline papers.

This research introduces a lightweight machine-learning-based email spam detection model using a word frequency pattern. We first considered the data balancing status in our approach and designed a random oversampling strategy to handle the spambase dataset class imbalance. This was done to make the spam detection model effective in prediction accuracy for unknown test samples and efficient by reducing the model bias toward the majority class. Finally, we tested the performance of our model using the testing dataset. To assess the performance of the resulting spam detection model, we compared the model results with recent similar works (Figure 7). The performance evaluation used Accuracy, Precision, Fscore, and Recall metrics. Furthermore, we used different output interpretation and validation tools like ROC and confusion matrix. It is important to note that although our proposed approach performed well, some limitations still need to be addressed. One limitation is that the proposed approach requires the dataset to be balanced. While our random oversampling strategy addressed this issue, it may not work well with highly imbalanced datasets. Furthermore, the proposed approach may not be suitable for datasets with many features, as the RF-based model can become computationally expensive. Future research could explore the use of other models, such as deep learning

References

[1] S. Whittaker, V. Bellotti, and P. Moody, "Introduction to this special issue on revisiting and reinventing e-mail," *Human--Computer Interact.*, vol. 20, no. 1–2, pp. 1–9, 2005.

[2] H. Faris *et al.*, "An intelligent system for spam detection and identification of the most relevant features

based on evolutionary Random Weight Networks," *Inf. Fusion*, vol. 48, no. June 2018, pp. 67–83, 2019, doi: 10.1016/j.inffus.2018.08.002.

[3] E. S. M. El-Alfy and R. E. Abdel-Aal, "Using GMDH-based networks for improved spam detection and email feature analysis," *Appl. Soft Comput. J.*, vol. 11, no. 1, pp. 477–488, 2011, doi: 10.1016/j.asoc.2009.12.007.

[4] E. P. Sanz, J. M. Gómez Hidalgo, and J. C. Cortizo Pérez, "Chapter 3 Email Spam Filtering," *Adv. Comput.*, vol. 74, no. 08, pp. 45–114, 2008, doi: 10.1016/S0065-2458(08)00603-7.

[5] Y. Hu, C. Guo, E. W. T. Ngai, M. Liu, and S. Chen, "A scalable intelligent non-content-based spam-filtering framework," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8557–8565, 2010, doi: 10.1016/j.eswa.2010.05.020.

[6] Y. Cohen, D. Gordon, and D. Hendler, "Early detection of spamming accounts in large-Scale service provider networks," *Knowledge-Based Syst.*, vol. 142, pp. 241–255, 2018, doi: 10.1016/j.knosys.2017.11.040.

[7] J. D. Rosita P and W. S. Jacob, "Multi-Objective Genetic Algorithm and CNN-Based Deep Learning Architectural Scheme for effective spam detection," *Int. J. Intell. Networks*, vol. 3, no. December 2021, pp. 9–15, 2022, doi: 10.1016/j.ijin.2022.01.001.

[8] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, "Addressing the class imbalance problem in Twitter spam detection using ensemble learning," *Comput. Secur.*, vol. 69, pp. 35–49, 2017, doi: 10.1016/j.cose.2016.12.004.

[9] A. Harisinghaney, A. Dixit, S. Gupta, and A. Arora, "Text and image based spam email classification using KNN, Na{\"\i}ve Bayes and Reverse DBSCAN algorithm," in *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, 2014, pp. 153–155.

[10] D. Debarr and H. Wechsler, "Spam detection using Random Boost," *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1237–1244, 2012, doi: 10.1016/j.patrec.2012.03.012.

[11] M. Mohamad and A. Selamat, "An evaluation on the efficiency of hybrid feature selection in spam email classification," in *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, 2015, pp. 227–231.

[12] H. Faris, I. Aljarah, and J. Alqatawna, "Optimizing feedforward neural networks using krill herd algorithm for e-mail spam detection," in *2015 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)*, 2015, pp. 1–5.

[13] N. O. Hamed, A. H. Samak, and M. A. Ahmad, "Cloud e-mail security: An accurate e-mail spam classification based on enhanced binary differential evolution (BDE) algorithm," *J. Intell. \& Fuzzy Syst.*, no. Preprint, pp. 1–13, 2021.

[14] V. Sri Vinitha and D. Karthika Renuka, "MapReduce mRMR: Random Forests-Based Email Spam Classification in Distributed Environment," in *Data Management, Analytics and Innovation*, Springer, 2020, pp. 241–253.

[15] H. M. Saleh, "An Efficient feature selection algorithm for the spam email classification," *Period. Eng. Nat. Sci.*, vol. 9, no. 3, pp. 520–531, 2021.

[16] F. Soleimanian Gharehchopogh and S. K. Mousavi, "A new feature selection in email spam detection by particle swarm optimization and fruit fly optimization algorithms," *Comput. Knowl. Eng.*, vol. 2, no. 2, pp. 49–62, 2020.

[17] S. A. Khamis, C. F. M. Foozy, M. F. A. Aziz, and N. Rahim, "Header based email spam detection framework using Support Vector Machine (SVM) Technique," in *International conference on soft computing and data mining*, 2020, pp. 57–65.

[18] "UCI Machine Learning Repository: Spambase Data Set." https://archive.ics.uci.edu/ml/datasets/spambase (accessed May 07, 2022).

[19] A. Boschetti and L. Massaron, *Python data science essentials: become an efficient data science practitioner by thoroughly understanding the key concepts of Python*. 2015.

[20] J. Brownlee, "Imbalanced Classification with Python," *Mach. Learn. Mastery*, p. 463, 2020.

[21] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *Int. J. Adv. Soft Comput. its Appl.*, vol. 7, no. 3, pp. 176–204, 2015.

[22] D. Zhang, W. Liu, X. Gong, and H. Jin, "A novel improved SMOTE resampling algorithm based on fractal," *J. Comput. Inf. Syst.*, vol. 7, no. 6, pp. 2204–2211, 2011.

[23] Y. Pristyanto, I. Pratama, and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," *2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018*, vol. 2018-Janua, pp. 310–314, 2018, doi: 10.1109/ICOIACT.2018.8350792.

[24] J. Prusa, T. M. Khoshgoftaar, D. J. DIttman, and A. Napolitano, "Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data," *Proc. - 2015 IEEE 16th Int. Conf. Inf. Reuse Integr. IRI 2015*, pp. 197–202, 2015, doi: 10.1109/IRI.2015.39.

[25] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results," *2020 11th Int. Conf. Inf.*

Commun. Syst. ICICS 2020, no. April, pp. 243–248, 2020, doi: 10.1109/ICICS49469.2020.239556.

[26] J. Brownlee, *Data Preparation for Machine Learning (Data Cleaning, Feature Selection, and Data Transforms in Python)*. Machine Learning Mastery, 2020.

[27] R. Nayak, S. Amirali Jiwani, and B. Rajitha, "Spam email detection using machine learning algorithm," *Mater. Today Proc.*, no. xxxx, Apr. 2021, doi: 10.1016/j.matpr.2021.03.147.