# Preventing Spam with Deep Learning based email filtering using Explainable AI

SK.M.M. Samir Shakir , Ananya Rahaman , and Muhtadi Zubair

Department of Computer Science and Engineering
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{*sk.m.m.samir.shakir,ananya.rahaman, muhtadi.zubair*}*@g.bracu.ac.bd*
*rezatanzim@gmail.com*

***Abstract*—Email is crucially important in both official and unofficial use. However, as the necessity of email increases so does the problem of spam email. Spam emails can be very harmful and might carry viruses or other malware. This can further compromise personal or official cyber security. For these reasons, the need to build and improve spam detection systems is increasing. Machine learning has been used for a while to build such models. However, only increase in accuracy is not good enough of an evaluation metric for this problem as one cannot afford to let an important email sent to spam. So an AI system that can classify and differentiate spam from regular email with optimal evaluation matric such as accuracy, precision, recall, etc. is in need. The purpose of our study is to evaluate a few ML models like Naïve Bayes, logistic regression, Random forest classifier, Ada Boost classifier, and finally deep learning LSTM(Long Term Short Memory) for this particular task and also use an Explainable AI model such as LIME to analyze the inner work of the models and try to compare them. Our study shows Random Forest classifier has the highest precision of 94.77 percent closely followed by Naïve Bayes. Also, the Ada Boost classifier scored the highest recall of 98.32 percent**

***Index Terms*—Naïve Bayes Classifier, Logistic regression, Random Forest Classifier, AdaBoost Classifier, LSTM, Explainable AI, LIME**

## I. Introduction

The Internet brought forward the fourth industrial revolution. Sharing of information has become very easy and fast. Many platforms are available for users to share and acquire information anywhere across the world. Among all, information-sharing mediums, email is one of the oldest among digital systems. However, due to their simplicity, emails are vulnerable to different kinds of attacks, and the most common and perhaps one of the most dangerous ones are spam [1]. No one wants to receive emails unrelated to their work or interest because they waste the time and resources of the recipient. Furthermore, these emails may contain malicious content hidden in the form of attachments or URLs that may compromise the host system's security system. Also, another concern is important to email ends up in spam where they might be overlooked [2]. Both the above-mentioned scenarios are seriously harmful to the users. A user might have different needs regarding security. Some users might need security above anything else, others might not want to miss out on important emails even at the cost of extra security. That's why we will use different evaluation matrices like precision, recall, and f1 alongside accuracy to fully assess the strength and weaknesses of each model. We will be using a few supervised machine learning systems to classify emails into spam or ham [3]. Our dataset is labeled that's why supervised methods are used. We will be using logistic regression, Naive Bayes Classifier, Random Forest Classifier, and AdaBoost Classifier. Our main objective is to evaluate each of these models using different parameters and matrices. For our experiment, we will use emails from different datasets used in previous experiments and merge them to create a new dataset so that our findings are not biased toward a particular domain. We will ultimately try to determine which fundamental machine-learning algorithm is best for spam detection. Furthermore, a recent issue is that AI models are getting complex and it's often hard to judge why a certain model is predicting a certain result. That's why we need Explainable AI models to fully evaluate and explain an AI model. That's why we will see the inner workings of the models with Explainable AI like LIME to further validate our work. Furthermore, our work will be useful as a review of these models for further research.

## II. Literature Review

Spam detection is a rapidly developing topic of interest in the field of Machine Learning particularly Natural Language Processing in recent times. In this section, we review some high-quality research that is relevant to our work and aid in the overall experiment. The review will mostly focus on the similarity to our work and articulate the literature gap elucidated by our research. Leug's [4] research paper is one of the earlier literature focusing on the area of email spam filtering systems in a more theoretical manner. The authors presented brief research aimed at formulating relations between information retrieval, Information Filtering, and spam detection systems in a more theoretically grounded manner. However, the research did not venture into the field of Machine Learning. Furthermore, An experiment with publicly available datasets of email with automated tools and a proper architecture for classification is also not presented in their work. Additionally, It also falls short of presenting and evaluating the parameters used by previous researchers in evaluating other proposed techniques. Wang [5] reviewed the different methods used to filter out unprompted spam emails. Additionally, The paper categorized email spam into different folders in a hierarchical manner and automated a few of the tasks required to respond to an email message. However, the research was not done with a domain-agnostic dataset and a comparative machine-learning algorithm review was not present. Further various evaluation matrices were not used to validate the model. Zhong [6] researched email spam filtering systems in detail and published a paper titled "Spam filtering and email-mediated applications". In this paper, the authors used ensemble learning algorithms to propose a framework for classifying emails. Furthermore, the article also illustrated the concept of an operable email(OE) system in an email-mediated real-world application and demonstrated an email assistant with various features in real-world applications. However, the paper did not go through different Machine learning algorithms as a comparative analysis evaluating the performance of its systems. Also, the sample size of the training data was not adequate to create a non-bias system. A review paper by Cormack [7] titled "Email spam filtering: a systematic review" compared previously proposed spam filtering algorithms with an emphasis on efficiency. The research also compared email spam filtering systems with other variations of spam identification systems. However, a few important evaluation matrices other than efficiencies like f1, precision, and recall were not compared. Furthermore, the comparison of efficiency was heavily dependent on research works done before rather than objective implementation-based evaluation. Additionally, each of the results compared was mostly biased towards the datasets used in each experiment rather than a unified unbiased evaluation.

## III. Background

In this section, we will discuss the necessary background leading up to our work. We will briefly discuss Supervised Machine Learning, Logistic Regression, Naïve Bayes Classifier, Random Forest Classifier, AdaBoost classifier, LSTM, and Explainable AI.

### A. Supervised Machine Learning

Supervised learning is the type of machine learning in which prediction models are trained using well "labeled" training data, and on the basis of that data, models predict the output. The labeled data means all of the input data is already classified which is assumed correct. All of the input data has one or more features and more than one class which can also be viewed as output. The entire dataset is split into training and testing sections. Then the training portion is used to train the model to find the correlation between the features and class. This correlation function can be viewed as a function between input and output. This trained model is then evaluated by testing on the test set by letting it predict the classes of the test inputs and then comparing the predictions with the correct classification. Furthermore, the correctness of this model can be quantified by various evaluation matrices such as accuracy, precision, recall, f1, etc. These kinds of algorithms are called Supervised Machine Learning because here the training set acts as a supervisor to mold the model for testing.

*1) Logistic Regression:* Logistic regression is one of the most popular and effective Machine Learning algorithms used for classification purposes. This algorithm falls under the class of Supervised Machine Learning algorithms. When the classification variable of the input data is categorical in nature then Logistic Regression is often a good choice for the prediction model. Categorical variable refers to the type of variables with domain consisting of a fixed set of categories rather than continuous values. Example: 0 or 1, True or false, or even more than 2 categories like grades(A, B, C, D, F) etc. However, rather than providing the exact category as output the model calculates the probability of each input belonging to a certain category. Then the category with the highest probability can be taken as the output. Logistic Regression and Linear Regression are different in the sense that Logistic regression is used to predict categorical values and Linear Regression is used to predict continuous values. Mathematical function for Logistic Regression is as follows:

$$y = \frac{e^(b + b1x)}{1 + e^(b + b1x)} \tag{1}$$

*2) Naïve Bayes Classifier:* The Naive Bayes classifier is a probabilistic Machine Learning model that's used for classification purposes similar to Logistic Regression, which means it deals with categorical variables like 0 or 1 rather than continuous variables. The classifier is fundamentally based on the Bayes theorem. The classifier is particularly useful for tasks related to Natural Language Processing. It treats each text as a bag of words and uses the frequency of occurrence of each word to classify the texts. For our research, we used the Multinomial Naive Bayes classifier to predict if the email is spam or not spam. However, The biggest disadvantage of the Multinomial Naive Bayes classifier is that it does not take the intra-contextual relation between words into consideration while making the classification. Regardless, it's still very useful for NLP (Natural Language Processing) and performs decently well in most cases. The mathematical formulation of Bayes theorem is as follows:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(A)} \tag{2}$$

*3) Random Forest Classifier:* Random Forest is one of the most popular and effective Machine Learning algorithms used for classification and regression purposes. This algorithm also belongs to the group of Supervised Machine Learning algorithms. A key difference is this algorithm can be used for both classification and regression-based problems in Machine Learning. During the process of Random Forest classification, the dataset is repeatedly subsampled. Then each sample is classified by a simpler classifier called Decision Tree. Each of the decision trees can be viewed as an individual classifier. Additionally, for the prediction, each of the Decision Trees will predict the output individually and the final output will be decided by the majority vote of the Decision Trees. This process is also known as ensemble learning. The higher number of Decision Trees lead to better accuracy of the overall Random Forest classifier.

*4) AdaBoost Classifier:* The mathematical called Adaptive Boosting is a technique in Machine Learning similar to Random Forest which is based on the Ensemble Method. The main idea is to combine a high quantity of weak classifiers to create a strong classifier. Furthermore, the weak classifiers are almost always decision trees with one level which is also commonly referred to as a stump. At first, all the data points are assigned equal weights. Then the weights are updated and more weights are given. The mathematical data points were wrongly classified by the stumps.

### *B. Deep Learning*

*1) LSTM:* networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail. Talking about RNN, is a network that works on the present input by taking into consideration the previous output (feedback) and storing it in its memory for a short period of time (short-term memory). Also, it resolves the Vanishing Gradient problem of RNN. It has 3 gates Forget Gate, Input Gate, and Output Gate. forget gate decides which information must be passed through the cell state and which information must be removed from the cell state. The input gate decides whether to allow new information in the cell state and store it in the memory cell. Output gate which information flows into the rest of the LSTM networks.

### *C. Explainable AI*

As AI becomes more advanced, analysts are challenged to comprehend and retrace how the algorithm came to a result. The entire process of computation and prediction turned into a black box system that we can not look into or analyze. Even the data scientist or the engineers building the system might have a hard time understanding and explaining why and exactly how the model predicts a certain output and what is happening inside the black box. As a result, it is hard to have confidence in such a model especially if it is deployed in a high-risk situation. Explainable AI is a group of methods that allow users to look into the inner workings of a Machine Learning system.

## IV. Dataset

### *A. Data Collection*

Collecting data is difficult due to numerous constraints, such as the volume of data and the throughput required for proper and timely ingestion. We collected four datasets from various sources and previous research. A csv file called SMS spam was collected from the UCI machine learning repository, a public repository of datasets previously used in various Machine Learning tasks. It was originally created by Almeida [8] for his research regarding spam detection. This dataset consists of 4825 non-spam and 747 spam entries. So, the dataset is clearly not balanced and will lead to a non-spam-biased model if trained on this dataset alone without balancing. Next, we collected another dataset from Apache spam assassin's public dataset. However, this is also imbalanced with 500 spam and 2500 non-spam entries. Our third dataset was the Enron dataset collected from Kaggle which has 1499 spam and 3672 non-spam once again heavily biased. Lastly, our final dataset was taken from G. Sakkis [9] research where 433 entries were spam and 2172 entries were non-spam.

### *B. Data Preprocessing*

After getting the required data set, we have done some prepossessing where we initially cleaned the data set using OpenRefine but we did not remove any of the null values from the data set as it might hold some clue regarding the classification we are aiming for. Then the class level has been encoded and the features were vectorized to convert textual data to numeric data so that it can be used as input in the prediction model. The punctuations has been removed so that it becomes more easier to work with using such a large dataset.

### *C. Dataset Description*

Next, we merge all the datasets and shuffle it. This is done to create a model that is not inclined toward any particular email environment. For example, the Enron spam dataset consists of emails of Enron employees which is not a general environment and might cause our model to work only in similar situations. After the merge we down sample the dataset by randomly removing the minimum portion of non-spam emails to make the dataset balanced. After the down sampling, we have 4575 spam and 4575 non-spam entries. We use this dataset as our training set.

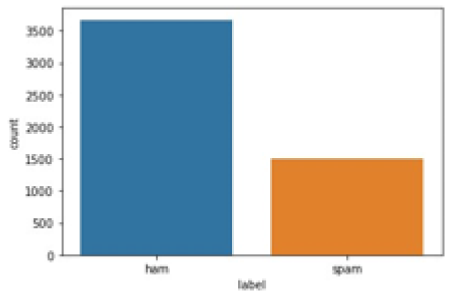

Fig. 1. Dataset visualization.

In Figure 1, We can see the visualization of the Dataset. We can easily see that there is an imbalance in the dataset but after doing the preprocessing, it is balanced.

## V. METHODOLOGY

The experimentation is done using Python language in the Jupyter notebook platform. We use various libraries to aid our experiment. Pandas were used for their high-performance data frame structure. Numpy is used for working with arrays. Most importantly, we used sklearn to build our machine learning models, fitting the data and evaluating the result. We build Naïve Bayes Classifier, Logistic Regression, Random Forest Classifier, and AdaBoost Classifier from sklearn. We also use matplotlib and seaborn for data visualization. For deep learning we use Tensorflow. Lastly, we use LIME as our XAI model to evaluate the inner workings of our machine learning models. The following flowchart.2 summarizes our workflow on a high level.

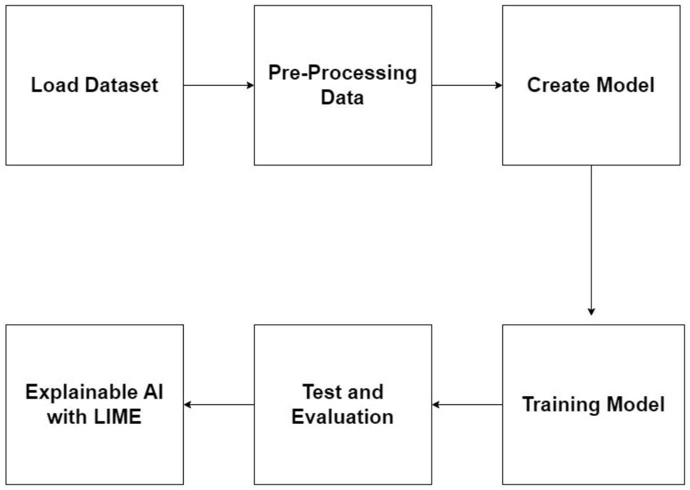

Fig. 2. workflow.

Our project can be segmented into five parts. On a high level first, we load the datasets. After that we clean and process the data for our models. After that, we merge the datasets and shuffle them. Lastly, we downsample the dataset to make it balanced. This is basically preprocessing. We did not delete the entries with empty email bodies or null values, as that might be a key indicator for detecting spam. After that, we split the data into 85 percent split and 15 percent for the test. The train section will be later used for fitting the machine learning models and deep learning models and the test section will be used for evaluation of the predictive capability of the fitted models. Then we convert each of the texts into sparse matrices for further classification using the CountVectorizer method. In this matrix, the rows correspond to the input and we have columns for each unique word. The values in that matrix correspond to the number of specific words separated in that input. This numerical transformation enables us to train the data with our machine-learning models. Afterward, we build four machine learning models namely Naïve Bayes Classifier, Logistic Regression, Random Forest Classifier, and AdaBoost classifier, and LSTM, and train them using our cleaned and pre-processed data. 14040000 (240*150*390) inputs to the first layer of our RNN model. First, we feed the data to two LSTM layers with 128 units each. After that, we do the same with another LSTM layer with 64 units. Finally, we put everything through a time-distributing dense layer with an activation function of softmax, which results in 6 element vector. We choose the softmax activation function in the last layer since we want our model to output probability, not a strict prediction. This also helps us to figure out where our model is making mistakes and its confidence in its prediction. The model has 50 percent dropouts after the four LSTM layers. Then we use Batch normalization after the dropout. We look for the sign-ending image so the model can stop taking input images after the sign is done. The model uses categorical cross-entropy as the loss function. Our RNN architecture has a 6-dimensional softmax output and our ground truth 6-dimensional vector is one hot, thus this loss function of our model works great. After fitting the models, we evaluate them by comparing their predictions with the test values. We evaluate them by comparing their evaluation matrix and confusion matrix. After getting the results we use LIME as our XAI model and use it to further evaluate each of our classifiers and their inner functionality. LIME is almost agnostic to different types of ML models and treats them as Black Box for evaluation.

## VI. Results

Result evaluation is one of the most important parts of our project. Here we comprehensively compare various evaluation matrices of different classifiers. We are presenting 2 confusion matrix (Adaboost and LSTM) which works best for the performance. Here, in figure 3 we can see that the True positive

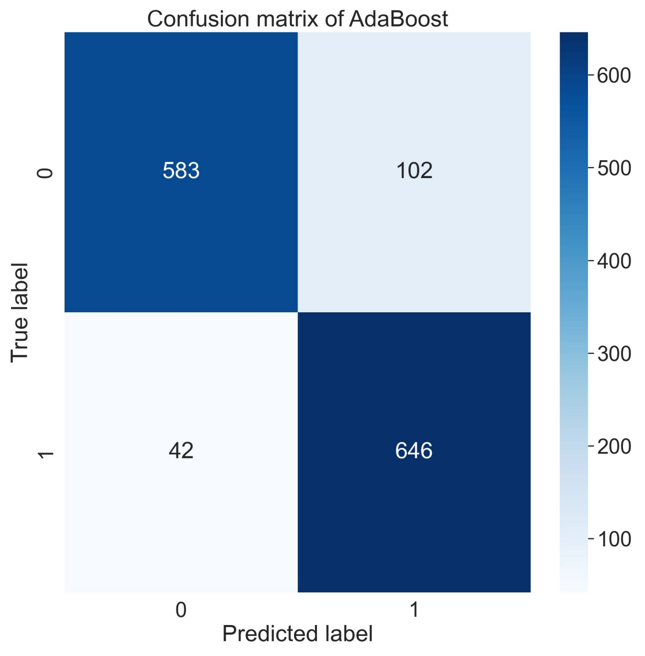

Fig. 3. Confusion matrix of Adaboost classifier.

value is 583, and the False positive value is 102. On the other hand, the True negative value is 42, and the False-negative value is 646. From figure 4, it can be seen that the True positive value is 621, and the False positive value is 64. On the other hand, the True negative value is 53, and the False-negative value is 635.

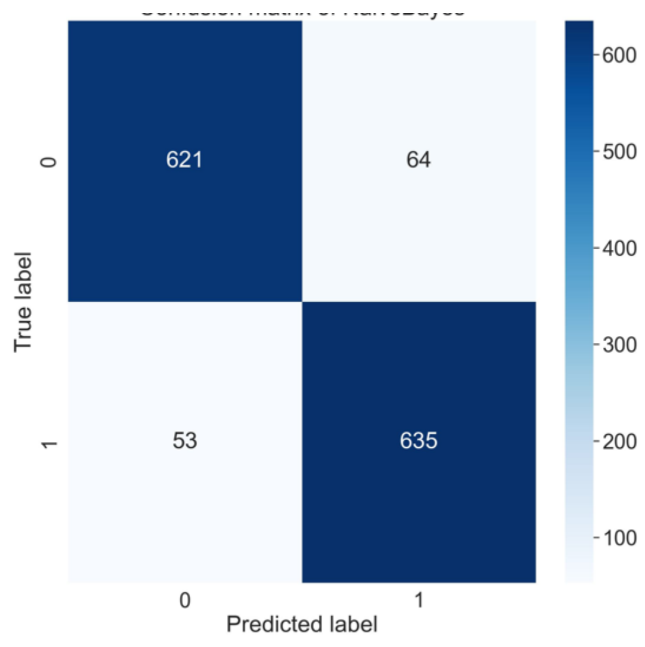

Fig. 4. Confusion matrix of LSTM classifier.

From the confusion matrix, we get a quick overview of the performance of each classifier. From the highest True Negative goes to the Multinomial LSTM classifier. Lowest False Positive value goes to the Logistic regression and lowest False Negative value goes to the Multinomial Naive Bayes classifier. So, it seems although the confusion matrix is great, on the other hand, we need more evaluation to make a better judgment. The following table compares the Precision, Recall, F1, and accuracy of each classification.

TABLE I
EVALUATION OF THE MATRICES COMPARISON BETWEEN CLASSIFIERS

| Comparison | Accuracy | precision | recall | F1 |
|---|---|---|---|---|
| NB | 90.68 | 0.8982 | 0.9190 | 0.9084 |
| LR | 94.61 | 0.9267 | 0.9696 | 0.9477 |
| RF | 94.10 | 0.9224 | 0.9224 | 0.9427 |
| AdaBoost | 89.00 | 0.8497 | 0.9493 | 0.8968 |
| LSTM | 95.10 | 0.9377 | 0.9563 | 0.9068 |

We can see that both Logistic Comparison and Random Forest classifiers have the highest F1 score while Adaboost has the lowest F1 score. The accuracy is also too close to differentiate between Logistic Regression and Random Forest. So, Logistic regression is the clear winner in terms of recall and F1 closely followed by the Random Forest classifier. But in terms of accuracy LSTM will be clearly the winner. Here, LSTM also has the highest Recall and precision value. Finally, we can see for a particular input the inner workings of a particular model. The following figure 5 shows the rationale behind the LSTM in terms of accuracy and precision. That is the reason for using LSTM.

From the above picture 5, Explainable AI(XAI) is Shown for LSTM. We can see LSTM can classify almost all emails very perfectly which are spam and which are not. We are going

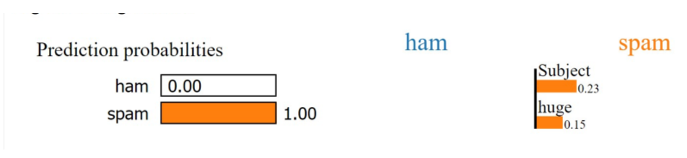

Fig. 5. LIME for LSTM

to show the Lime implementation of Adaboost classifier. From

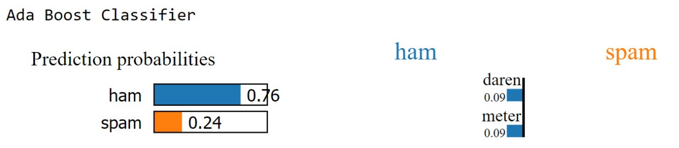

Fig. 6. LIME for Adaboost

the picture 6, Explainable AI(XAI) is Shown for Adaboost. We can see Adaboost can classify most of the emails but the accuracy is not good as LSTM.

## VII. CONCLUSION

In this paper, we classify spam from regular messages with four different machine-learning classifiers and one deep-learning model. Additionally, we mesh four different datasets previously used in similar research and balanced them to make an unbiased judgment regarding the performance of the algorithms. We have found that every model has its own strength and weakness. One model is not better than another in all aspects. However, Logistic regression and Random Forest seem to perform the best overall. As we use more and more data online spam will be more of an issue in the future. So much more research is needed in this particular area. Trying unsupervised methods on the data might be a work for the future as most of the data we have in this regard is unlabeled so LSTM works best in this regard.

## REFERENCES

[1] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, “A support vector machine based naive bayes algorithm for spam filtering,” 12 2016, pp. 1–8.

[2] C. Palanisamy and T. Kumaresan, “E-mail spam classification using s-cuckoo search and support vector machine,” *International Journal of Bio-Inspired Computation*, vol. 9, p. 142, 01 2017.

[3] N. Parmar, H. Jain, A. Sharma, and A. K. Kadam, “Email spam detection using naive bayes and particle swarm optimization,” 2020.

[4] C. Lueg, “From spam filtering to information retrieval and back: Seeking conceptual foundations for spam filtering,” *Proceedings of the American Society for Information Science and Technology*, vol. 42, 10 2006.

[5] X.-L. Wang and I. Cloete, “Learning to classify email: a survey,” 09 2005, pp. 5716 – 5719 Vol. 9.

[6] W. Li, N. Zhong, Y. Yao, J. Liu, and C. Liu, “Spam filtering and email-mediated applications,” 12 2006, pp. 382–405.

[7] G. Cormack, “Email spam filtering: A systematic review,” *Foundations and Trends in Information Retrieval*, vol. 1, pp. 335–455, 01 2006.

[8] T. Almeida, J. Gomez Hidalgo, and A. Yamakami, “Contributions to the study of sms spam filtering: New collection and results (preprint),” 10 2013.

[9] G. Sakkis, “A memory-based approach to anti-spam filtering for mailing lists,” *Information Retrieval*, vol. 6, pp. 49–73, 01 2003.