International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Email Monitoring System Using Various Machine Learning Approaches

Ndivhuwo Netshamutshedzi[1], Netshikweta Rendani[1] and Ibidun Christiana Obagbuwa*[2]

Submitted: 12/05/2024 **Revised:** 26/06/2024 **Accepted:** 03/07/2024

Abstract*:* A large portion of email traffic is made up of spam, which has caused issues throughout the world. Spammers always employ new techniques, making managing or preventing spam messages difficult. In today's world, both businesses and educational institutions heavily rely on email communication. This study aims to compare the predictive performance of Machine Learning (ML), Deep Learning (DL), and Ensemble Learning (EL) in the context of email monitoring systems. In our research, we build upon previous studies addressing the spam problem to enhance accuracy. We employ a variety of methods, including Naive Bayes (NB), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Adaptive Boosting and Random Forest. The paper findings reveal that LSTM achieved the highest level of accuracy, reaching 99.88%. Consequently, LSTM stands out as a potent machine-learning system with potential benefits for future studies in this field.

Keywords: *Machine learning, Deep learning, Ensemble learning, Naive Bayes, Support Vector Machine, Convolutional neural network, AdaBoost Classifier and Long short-term memory.*

1. Introduction

The internet has become an integral part of human life, with over four and a half billion users finding it convenient for various purposes. Email, in particular, has emerged as a trusted means of communication among internet users Kumar et al. (2020). As email services have advanced over time, they have become powerful tools for exchanging different types of information. However, the widespread use of email has also led to an increase in spam attacks. These unsolicited and unwanted emails, sent by individuals with deceptive intentions from anywhere in the world, often contain fake content and links designed for phishing attacks and other threats Jain et al. (2019). The ultimate goal behind these spam emails is to obtain users' personal information for malicious purposes, such as identity theft or financial gain Masood et al. (2019). Such emails may contain malicious content or URLs that direct recipients to harmful websites, earning them the label of phishing emails.

Despite the progress made in spam filtering technology, distinguishing between legitimate and malicious emails remains challenging due to the constantly evolving nature of spam content. Spam emails have been a problem for several decades, and despite the availability of various anti-spam measures, even non-expert users can still fall victim to them Akhtar et al. (2017). In email management systems, spam filters work to identify spam and redirect it to a designated space, like a spam folder, giving users the option to decide whether to review them. Various spam filtering tools, including corporate email systems, email filtering gateways, contracted anti-spam services, and user education, can effectively handle spam emails written in English or other languages Akhtar et al. (2017). However, they often struggle to filter spam in languages that have recently been digitized.

The suggested study leverages established artificial intelligence models to identify spam emails. This paper outlines the process of training machine learning (ML), deep learning (DL) and ensemble learning (EL) models like SVM, Naive Bayes, CNN, LSTM and Adaptive Boosting which is a type of recurrent neural network, for the purpose of detecting spam emails. Additionally, this paper provides insights into the development and training of diverse machine learning models. It utilizes precision, recall, f-measure and ROC-AUC as crucial evaluation metrics for comparing Naive Bayes and SVM. In contrast, it employs evaluation parameters, namely Model Loss, to assess deep learning models like CNN, and LSTM and ensemble learning models like AdaBoost. Ultimately, the study concludes by comparing all the models to determine the highest accuracy and the most favourable evaluation parameter values achieved by deep learning, machine learning and ensemble models (Drucker et al., 1999).

Ndivhuwo Netshamutshedzi, Department of Mathematical and Computational Science, University of Venda, South Africa, Email: emailndivhuwon.nn@gmail.com
Netshikweta Rendani, Department of Mathematical and Computational Science, University of Venda, South Africa. Email: rendani.netshikweta@univen.ac.za
Ibidun Christiana Obagbuwa, Department of Computer Science and Information Technology, Sol Plaatje University, South Africa, ibidun.obagbuwa@spu.ac.za
Correspondence: ibidun.obagbuwa@spu.ac.za*

The integration of machine learning (ML) and deep learning (DL) models into an email monitoring system is motivated by the need to streamline and optimize email management processes. By harnessing the power of ML and DL algorithms, this system aims to significantly reduce the manual effort required for sorting through large volumes of emails while improving overall efficiency. Moreover, the design rationality behind incorporating these models lies in their ability to enhance email security and ensure compliance with data protection regulations Jain et al. (2019). ML and DL models can be trained to detect and flag suspicious emails, such as phishing attempts or malware, thereby bolstering the systems security measures. Additionally, these models enable personalized responses and suggestions by analyzing the content and context of emails, leading to more meaningful interactions and efficient handling of inquiries. Furthermore, the system is designed to continuously learn and improve over time, adapting to evolving patterns and trends in email communication. This continuous learning process not only enhances the accuracy and effectiveness of the system but also ensures its scalability and adaptability to meet changing requirements and usage patterns. Furthermore, the motivation and design rationality behind leveraging ML and DL models in an email monitoring system revolve around optimizing efficiency, enhancing security, providing personalized experiences, enabling continuous improvement, and ensuring scalability and adaptability (Suryawanshi et al., 2019).

The proposed research primarily centres on identifying spam within email datasets. The unique contribution to this study lies in utilisation of the spam dataset to assess a range of machine learning, deep learning, and ensemble learning techniques. Specifically, the paper introduced CNN and LSTM deep learning models as well as employed AdaBoosting within the ensemble learning framework. These models have seldom been applied in previous studies addressing similar issues. Impressively, they have yielded outstanding results in detecting spam emails within the dataset. Notably, LSTM achieved the highest accuracy at 99.88%, closely followed by AdaBoost at 99.85%. These results are noteworthy because the study approach differs significantly from previous methodologies. The primary contributions of this research include:

The paper introduces an advanced model aimed at enhancing the detection of spam messages, employing a range of machine learning algorithms such as SVM, Naive Bayes, LSTM, CNN, Adaptive Boosting, and Random Forest. Evaluation of these models includes metrics like accuracy and receiver operating characteristic scores for comparison. Furthermore, the study presents demonstrations and comparisons of the proposed model against other competitive baseline models. It acknowledges the potential concerns regarding privacy infringement with the implementation of an email monitoring system and suggests addressing these through clear organizational policies communicated to employees. The presented method undergoes training and testing on various algorithms, resulting in a notable increase in spam detection rates on imbalanced datasets, assessed through metrics including accuracy, precision, recall, specificity, F1-score, and ROC (AUC).

1.1 Organization of the Paper

The rest of the paper is structured as follows: Section 2 covers the related works. Section 3 explains the architecture of our model. The results and analysis of the experiments are discussed in Section 4. Finally,Section 5 is dedicated to the conclusion.

2. Literature Review

The global academic community is becoming increasingly interested in email monitoring systems. In this chapter, an overview articles of similar studies undertaken for email monitoring system using different machine learning algorithms. Reviews that are comparable to those that have been published in this field's literature have been offered. This approach is taken to articulate the problems that still need to be solved and to draw attention to the distinctions between current evaluation and the previous one.

(Vyas et al., 2015) evaluated the effectiveness of supervised ML techniques in spam filtering and discovered that the Naive Bayes approach outperformed all other techniques (excluding SVM and ID3) in terms of both precision and speed. Although SVM and ID3 provide more precision than Naive Bayes, they require additional time to construct a system. Thus, the method depends on the particular circumstances, required precision, and available time. The study suggests that for a better future, the paper need to use spam filtering architecture, and all aspects of email should be considered. In the present paper we compare the predictive performance of the Naive Bayes, SVM, CNN, LSTM and AdaBoost models for email monitoring system. The paper shows that when it comes to accuracy, recall and f-measures, we found that Naive Bayes is more successful and gives superior outcomes, but when compared to Naive Bayes, SVM has the best precision. The comparison analysis of the findings prove that Naive Bayes produces better outcomes in terms of accuracy, SVM performs better when it comes to precision, recall, and f-measure.

(Hossain et al., 2021) reviewed a model that divides emails into spam and junk mail. To produce a comparison analysis, the proposed approach is used for both ML and DL. In the ensemble technique for machine learning implementation, Multinomial Naive Bayes (MNB), Random Forest (RF), K-Nearest Neighbor (KNN), and Gradient Boosting (GB) are employed. Implementing deep learning using recurrent neural networks (RNN), gradient descent (GD), and artificial neural networks (ANN). The output of numerous classifiers is combined using an ensemble approach. Compared to a single classifier, the ensemble approaches enable the creation of predictions with higher prediction accuracy. More pertinent attributes need to be extracted in this effort in order to build a model for an email monitoring system. This paper's focus aligns with the aim of my study, which is to compare the predictive performance of the ML, DL, and EL models for an email monitoring system. The paper found out that LSTM which is DL model perform better than ML and EL model.

(Ferrag et al., 2020) offered an analysis of spam diagnosis datasets and intrusion detection system deep learning algorithms. They examined various DL-based detection techniques and assessed the performance of those models. The cyber datasets that are widely used were examined and categorized into seven distinct groups by the researchers. The categories included datasets that pertained to various types of traffic such as network traffic, Intranet traffic, electrical network, virtual private network, Android applications, Internet of Things traffic, and traffic from internet-connected devices. These categories were used as the basis for the analysis conducted by the researchers. They arrive at the inference that when it comes to detecting intrusions and spam, deep learning models have the potential to surpass traditional machine learning and lexical models in performance. Their paper focuses on DL approaches, where this paper examines a variety of model such as ML, DL and EL. The present paper also found out that DL has the potential to classify spam as spam and ham as ham.

(Jain et al., 2019) constructed a semantic LSTM that is optimized for spam detection. They have used a developing approach known as the deep learning technique in their work. To classify spam, a specific structure called LSTM is utilized, which belongs to a larger class of networks known as Recursive Neural Network (RNN). In contrast to conventional classifiers, where the features are manually created, it has the potential to learn abstract features. Through the use of word2vec, WordNet, and ConceptNet, word vectors with semantic meaning are created from the text before being fed into the LSTM for task classification. The results of classification are contrasted with classifiers that compare SVM, Naive Bayes, ANN, k-NN, and Random Forest. Additionally, using solely on LSTM classifiers alone may not always result in improved predictive accuracy that is where the paper include EL to improve accuracy.

(Siddique et al., 2021) created three different models NB and SVM as machine learning models, and CNN as a DL model using a long short-term memory (LSTM) and compared their accuracy. The LSTM model outperformed the others with an accuracy rate of 98.40%, while NB achieved 98.00%, SVM achieved 97.50%, and CNN achieved 96.20%. The SVM model had a precision rate of 97%, a recall rate of 92.50%, and an F1 score of 95%. In comparison, the NB model had a precision rate of 96.50%, a recall rate of 95%, and an F1 score of 96%. Their paper focuses on ML and DL model, this paper include EL models for better comparison in email monitoring system. The present paper shows that LSTM outperform other models with 99.88%.

(Breiman, 1996) introduced the bagging algorithm, which is a widely used ensemble method in machine learning that improves the performance of predictive models. The paper details the algorithm's principles and its various applications. The study presents empirical results that demonstrate the efficacy of bagging in reducing prediction error and increasing the stability of model performance. The paper concludes that bagging is a potent and versatile technique that can greatly enhance the precision and robustness of predictive models. In this paper we conclude that DL outperforms other model since it has high accuracy than other models.

(Biggio et al., 2011) provides a fresh strategy for enhancing the robustness of classifiers against poisoning attacks in adversarial classification tasks. The writer introduces a bagging-based ensemble technique that employs multiple base classifiers to mitigate the results of adversarial attacks on how well the model performs. The paper presents empirical outcomes demonstrating how well the suggested approach works to increase categorization accuracy and model robustness against poisoning attacks. The study suggests that bagging-based ensemble methods are a promising technique for enhancing the security and reliability of classification models in adversarial settings. In this paper variety of model to make well informed decision as to which model perform better are compared.

(Gangavarapu et al., 2020) presented an in-depth study of the literature on the application of ML techniques for stopping phishing and spam emails. The article offers a comprehensive summary of the various approaches and algorithms used in the field, including both traditional and deep learning-based methods, and discuss their limitations and challenges. The paper also proposes novel approaches for improving the accuracy and robustness of email filters using machine learning. Overall, the paper offers a useful tool for researchers and practitioners captivated by leveraging machine learning in email filtering. Spam is a major problem in modern world, since people nowadays heavily relied on network, that is where spammer tries to scam use their trick. In this paper a way to fight this is employed using different method.

(Bazzaz Abkenar et al., 2021) tackled Twitter's spam issue, given its widespread usage and appeal to spammers. Its aim is to spot and filter out spam tweets and their creators for a spam free Twitter environment. To enhance Twitter's spam detection, a novel hybrid method combining Synthetic Minority Over sampling Technique (SMOTE) and Differential Evolution (DE) strategies is introduced. SMOTE addresses imbalanced Twitter dataset classes, while DE fine tunes Random Forest (RF) classifier hyper parameters. In comparison to existing methods, this approach significantly improves classification performance, particularly for imbalanced datasets. The optimized RF classifier boasts a remarkable 98.97% detection rate, alongside exceptional F1-score and Area Under the Receiver Operating Characteristic Curve (AUROC) values of 0.999. This underscores the method's impressive efficiency and effectiveness in combatting Twitter spam. This paper employs a spam dataset to assess and contrast the predictive capabilities of machine learning (ML), deep learning (DL), and ensemble learning (EL) models within an email monitoring system. Evaluating this method on using different method like DL and EL may yield different results.

(Mishra and Thakur, 2013) addressed the pervasive issue of spam mail in the context of increasing internet users. It emphasizes the challenges researchers face in reducing spam, which is typically defined as unsolicited email messages. The primary objective of the paper is to categorize spam mail and address various problems

associated with online communication. To achieve this, the study explores the application of machine learning algorithms for classifying spam and legitimate emails. It utilizes a benchmark dataset containing 9324 records with 500 attributes for both training and testing. The research aims to identify the most effective classification approach for distinguishing between spam and legitimate messages. Three supervised machine learning algorithms Naive Bayes, Random Tree, and Random Forest are employed on the spam mail dataset. Additionally, two feature selection algorithms are used in the analysis. The paper's findings and methodology have the potential to contribute significantly to combatting unsolicited commercial emails, viruses, Trojans, worms, frauds, and other undesirable electronic communications. This study builds upon Mishra et al.'s work by extending the analysis to include deep learning and ensemble learning models, with the aim of further enhancing the accuracy and efficiency of email monitoring systems.

(Smith and Tabak, 2009) explored the complex issue of monitoring employee emails in the workplace. The paper investigates the legal, ethical, and practical implications of such monitoring, as well as the potential impact on employee privacy and trust. Through a comprehensive analysis of various case studies, surveys, and legal frameworks, the study conclude that while monitoring employee emails can be a necessary tool for ensuring organizational security and preventing misconduct, it should be done with careful consideration of employee privacy and legal requirements. The paper offers valuable insights for managers and policymakers dealing with the challenges of balancing privacy and security concerns in the workplace. This paper address these concerns, were organizations can establish clear policies around email monitoring and communicate these policies to their employees beforehand to avoid any confusion between manger and employee.

(Friedman and Reed, 2007) discussed the implications of employee monitoring for the law and employee relations on email users in the workplace. The study acknowledge that employers have a legitimate interest in protecting their business interests through email monitoring, but emphasize the need to balance this with employees right to privacy. Legal considerations, including the Fourth Amendment and federal wiretapping laws, are discussed. The paper also addresses the psychological and social effects of email monitoring on employees and offers suggestions for striking an appropriate balance between privacy and employer interests. This paper underscore the importance of creating equitable policies for email monitoring in the work environment.

(Chory et al., 2016) conducted a survey to investigate the impact of computer-mediated workplace communication under organizational surveillance on the privacy of employee's concerns and responses. The study involved 304 working adults, and the findings suggested that employees who perceived extensive monitoring were more likely to feel negative emotions and have privacy invasion concerns. Furthermore, the paper found that employees who strongly valued privacy as a fundamental right were more likely to resist surveillance measures, whereas those who prioritized organizational outcomes over privacy concerns were more accepting of surveillance. This paper emphasizes the need to balance organizational goals with employee privacy concerns when implementing workplace surveillance measures.

This paper explores the utilization of ML, DL and EL in email monitoring systems, with a particular emphasis on spam filtration. To achieve this goal, two main ML, two DL and two EL techniques are implemented. This study analysed a range of papers on this subject, examining the techniques proposed and the obstacles encountered when detecting spam and monitoring emails. Moreover, the study assesses the strengths and shortcomings of the suggested spam prevention methods and detection, that have not been thoroughly examined before. In recent years, numerous researches have been done employing ML techniques like RNN, ANN, LSTM, SVR, and many more. This study evaluates the predicted accuracy of email monitoring systems using Naive Bayes, SVM, CNN, LSTM, Random Forest and AdaBoost Classifier.

3 Methodology

The supervised machine learning models are the emphasis of this study, namely, Naive Bayes, SVM, LSTM network and DL, CNN, and ensemble classifier, AdaBoost, for the email monitoring system. Include ensemble learning and deep learning to be able to do a good comparative study. We use four metric evaluation methods Accuracy, Precision, Recall, and F1 measure to choose the best model with the highest performance prediction accuracy. Since Accuracy and Precision compare a limited number of models, the model with the highest Accuracy, precision, Recall, and F1 measure values is preferred.

3.1 Data Collection and Preparation

The data utilised in this study was gathered from the website Kaggle, and machine learning models were trained using this data. The information was acquired In CSV format, it was initially accessible in the English language (Iyengar et al., 2017).

3.1.1 Dataset

The dataset was translated with the help of the Python package and we found out there were 5 columns and 5572. The authors then manually corrected the translated data. Our dataset contains 5572 spam and ham emails. We created our own dataset because there were some 403 duplicate rows in the dataset, then we removed them and left with 5169 emails, see Figure 1. The dataset has many fields, and some of these columns of the dataset are not required. So we remove (drop) some columns which are not necessary. We turn spam or ham into numerical data and create a new column called spam.

	Catergory	Text	spam
0	ham	Go until jurong point, crazy.. Available only ...	0
1	ham	Ok lar... Joking wif u oni...	0
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	1
3	ham	U dun say so early hor... U c already then say...	0
4	ham	Nah I don't think he goes to usf, he lives aro...	0
...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	1
5568	ham	Will Ì_ b going to esplanade fr home?	0
5569	ham	Pity, * was in mood for that. So...any other s...	0
5570	ham	The guy did some bitching but I acted like i'd...	0
5571	ham	Rofl. Its true to its name	0

5169 rows × 3 columns

Figure 1: Before preprocessing, the dataset is checked for spam and ham emails.

Table 1. Extracted dataset definition report.

Dataset feature	Values
Number of variables	3
Number of observations	5572
Missing cells	0
Missing cells %	0.0%
Duplicate rows	403
Duplicate rows %	4.8%
Total size in memory %	87.2+ KB
Average record size in memory %	16.0 B

Table 1 provide an overview of the dataset's structure, completeness, and memory usage, which are important considerations for data analysis and processing.

3.1.2 Data Pre-processing

In order to train and evaluate various learning models, raw data must first be organised and managed. This process is referred to as preprocessing. Preprocessing, to put it simply, is an ML strategy that transforms raw data into a practical and useful structure (Afzal and Mehmood, 2016). Preprocessing serves as the starting stage in constructing a machine-learning model. It involves the conversion of real-world data, often riddled with imperfections, inaccuracies, and deficiencies due to faults, into precise, reliable, and usable input variables and patterns (Akhtar et al., 2017).

3.1.3 Import Data

The dataset must be imported in its original form after being downloaded from Kaggle and converted to CSV (Venkatesh, 2021) released the spam or not spam dataset. The dataset contains 5573 emails, including 653 spam emails and 4516 ham emails. On the Kaggle website, you can access the dataset called spam or not spam at https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

3.1.4 Tokenization

This crucial pre-processing stage collects and counts every word contained in the email frequency of each word and records where each word appears (Chen et al., 2009). We were able to identify terms that appeared more than once in our sample using the Count Vectorizer. The words are referred to as tokens since each one has a specific number that indicates how frequently they appear in the text. Unique feature values are included in 1e tokens, which will later aid in the development of feature vectors. Each word receives its own token during a tokenization step.

3.1.5 Stop Word Removal

The next stage is to remove every pointless word and punctuation mark, the dataset has been separated by commas, full stops, colons, and semicolons to convert into distinctive tokens (Karim et al., 2019). The technique of removing pointless words is known as stop word deletion. Natural Language Toolkit (NLTK), a built-in library for Python, is frequently used in language processing.

3.1.6 Stemming

Stemming the tokens is the next action after they have been created. Stemming is a technique for reverting the dataset's derived terms to their original versions (Chen et al., 2009). Prefixes and suffixes are first removed from base words. The process of stemming is then utilised to turn altered and misspelt words into their root or stem words. In order to successfully complete the stemming process for this stage as well, we employed the NLTK Python Library. Spam phrases can be quickly discovered after the content of emails has been stemmed (Drucker et al., 1999).

3.1.7 Selection and Feature Extraction

The process of turning a sizable raw dataset into a more manageable format is called feature extraction. This stage can involve extracting, regardless of the underlying dataset, any variable, attribute, or class (Androutsopoulos et al., 2000). In order to train the model and produce more accurate and dependable results, attribute extraction is a critical step. The process of picking a few crucial variables that accurately characterise data from among the many potential qualities throughout the feature extraction process is known as feature selection (Sharma and Bhardwaj, 2018). Following that, the model is built using the chosen attributes or variables. In turn, the model-building process will be faster if feature selection is done correctly.

3.2 Machine Learning, Deep Learning and Ensemble Classifier Models

The approaches used in ML are supervised machine learning, DL method and EL model to email monitoring system in this study are listed below.

3.2.1 Machine Learning (ML)

Spam filtering is primarily a classification problem from the standpoint of machine learning, where we attempt to identify emails as spam or junk on its feature. As an example, depending on whether x is a dimensional vector containing the features or a value of 1 or 0, the data point

(x, y) can signify either spam or ham. Machine learning algorithms can be trained or taught how to classify emails. ML model's primary goal is to naturally understand new information without human involvement. Three main types of machine learning, employed for many different applications, are available. Over the past ten years, to enhance email communication, study participants have kept working on many projects (Alpaydin, 2020). One of the foremost imperative ways to defend email networks is by spam filtering of emails. This study aims to build a reduced alternative to several ML models and methodologies now utilized for email monitoring systems. Additionally, the most popular machine learning techniques are evaluated in this study that is SVM, LSTM, and Naive Bayes.

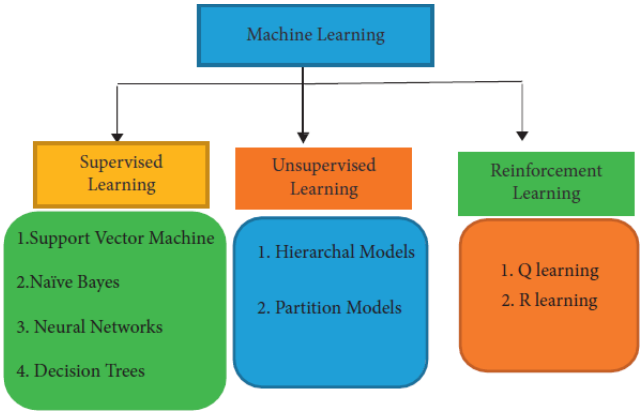


Figure 2: Types of machine learning, adapted from (Ahmed et al., 2022).

Figure 2 demonstrates various kinds of machine learning. Machine learning facilitates the processing of vast quantities of data. (Though it typically provides faster and more accurate results to detect unwanted content, it can also require extra time and resources to train its models for a high level of performance. Integrating machine learning with AI and cognitive computing Garcez et al. (2019) can make handling massive amounts of data even more powerful.

3.2.1.1 Naive Bayes (NB)

An algorithm for supervised learning is the Naive Bayes (NB) classifier that employs Bayes' Theorem. It primarily hinges on the ability to distinguish between various entities using predetermined attributes. Naive Bayes detects a word or event that occurred in a prior context and computes the probability of that word or event recurring in the future (Kumaresan and Palanisamy, 2017). Naive Bayes classifier technique can be used for classifying spam emails as word probability plays main role here. If there is any word which occurs often in spam but not in ham, then that email is spam. This paper uses Naive Bayes classifier algorithm because has become a best technique for email filtering and Naive Bayes always provide an accurate result. For this the model is trained using the Naive Bayes filter very well to work effectively. The Naive Bayes always calculates the probability of each class and the class having the maximum probability is then chosen as an output. It is used in many fields like spam filtering as discussed in equation (1)

$$P(C|D) = \frac{P(D|C) * P(C)}{P(D)} \qquad \text{(Equation 1)}$$

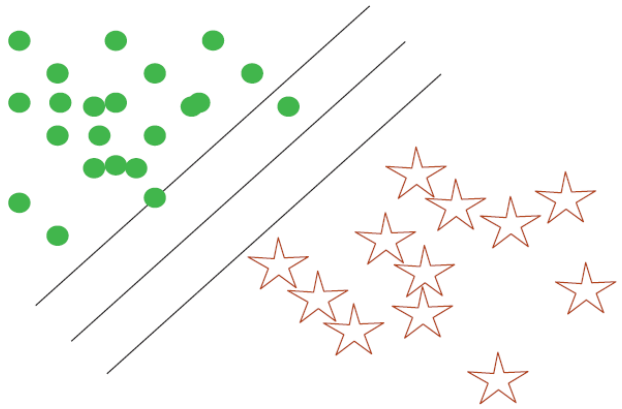


Figure 3: Support vector machine classification, (Ahmed et al., 2022).

Where C and D are events and *P (D)* 0.

The prior probabilities of witnessing D and C without taking into account one another are P(D) and P(C), respectively.

P (*D*|*C*) is the likelihood of seeing occurrence D if C is accurate.

3.2.1.2 Support Vector Machine (SVM)

The SVM represents another supervised machine learning technique, specifically designed for datasets that have been categorized. In SVM training, both positive and negative datasets are typically employed. Notably, negative datasets are exclusive to SVM training and are not utilized in the preparation of other machine-learning models. SVM stands out as one of the most frequently used models for both classification and regression tasks (Olatunji, 2019). This paper uses SVM to classify email if it is spam or ham and SVM it offers a higher level of reliability compared to alternative models when it comes to data classification. In situations where the available labelled data is limited, SVM emerges as the fastest and most dependable classification model. The Support Vector Machines totally founded on the idea of Decision points. The main resolution of Support Vector Machine algorithm is to create the line or decision boundary. The SVM model utilizes a hyperplane to segregate positive and negative values (spam and ham) within the dataset and subsequently determines whether the values closely align with the decision surface, Figure 3 illustrates the SVM.

SVM algorithms are very potent for the identification of patterns and classifying them into a specific class or group. They can be easily trained and according to some researchers, they outperform many of the popular email spam classification methods Scho¨lkopf and Smola (2002).

This is because during training, SVM use data from email corpus. However, for high dimension data, the strength and efficacy of SVM diminish over time due to computational complexities of the processed data Yu and Xu (2008). According to Chhabra et al. (2010), SVM is a good classifier due to its sparse data format and satisfactory recall and precision value. SVM has high classification accuracy. Moreover, SVM is considered a notable example of "kernel methods", which is one of the central areas of machine learning.

3.2.2 Deep Learning (DL)

Neural networks, a key component of deep learning, are a powerful technology that has become increasingly important when it comes to artificial intelligence and ML. The models consist of multiple hidden layers, each with adjustable weights that allow them to learn increasingly complex representations of the input data (Jain et al., 2019). When given an input, the DL model processes it through the hidden layers, which enable it to make accurate predictions based on the learned patterns.

3.2.2.1 Long Short-Term Memory (LSTM)

Within the realm of deep learning models, Long Short-Term Memory (LSTM) stands as a prominent contender. LSTM, classified under the category of recurrent neural networks (RNN) (Hochreiter and Schmidhuber, 1997), possesses a network structure replete with feedback connections. This architecture exhibits remarkable versatility, accommodating both entire data sequences and individual data points with equal proficiency (Rana et al., 2011). The LSTM approach incorporates a number of gates that significantly improve memory in the context of time series monitoring systems (Hochreiter and Schmidhuber, 1997). The paper uses LSTM because it can handle large size of data and LSTM excels in handling unsegmented data as well as data organized in time series, making it a robust choice for tasks involving classification and prediction. More size of data will provide more info to the model and therefore more generalized will be the model. If the input values are (x_1, x_2, ..., x_t) and the output values are (y_1, y_2, ..., y_t) of the current historical data to be monitored system, then the main phases of the LSTM network in a unit are as follows:

Inputgate: The input gate calculates the amount of input that is allowed to pass through it and is calculated using Equation (2).

$$i_t = \sigma(x_t U^i + s_{t-1} W^i), \quad \text{(Equation 2)}$$

The sigmoid function maps the value of the input between [0, 1] and this value is multiplied by the weight vector (U^i). This helps the gate manage the amount of input which is passed through the input gate.

Forgetgate: The forget gate helps the network choose what and how much information from the previous level to pass to the next level. The sigmoid function maps the value of this function between 0 and 1. It is given by Equation (3)

$$f_t = \sigma(x_t U^f + s_{t-1} W^f), \quad \text{(Equation 3)}$$

If no input needs to be passed to the next level, the previous memory is multiplied with the zero vector, which makes the input value zero. Similarly, if the memory at st−1 needs to pass to next level it is multiplied by 1 vector. If only some portion of the input is to be passed, then the corresponding vector is multiplied with the input vector.

Outputgate: The output gate, defines the output passed at each step of the network. It is given by Equation (4)

$$o_t = \sigma(x_t U^o + s_{t-1} W^o), \quad \text{(Equation 4)}$$

In the case of spam classification, the final output is the classification label. The output at each time step is required in the problems like language modelling Sundermeyer et al. (2012), and language translation Sutskever et al. (2014). The equations of the three gates described above are the similar equations with different parameter matrices *U* and *W*. Each LSTM unit also perform various calculations given below.

Candidate hidden state *g* of the network is calculated by Equation (5)

$$g_t = tanh(x_t U^g + s_{t-1} W^g), \quad \text{(Equation 5)}$$

c_t is the internal memory of the LSTM unit. It is calculated by Equation (6)

$$h_t = c^0_{t-1} f + g^0 i, \quad \text{(Equation 6)}$$

It can be seen from the equation that the cell's memory is the combination of the portion of the previous cell state c_{t-1}. The information from the previous memory and the new calculated hidden state multiplied (element wise) by current input state gives the current cell memory.

The output hidden state st is calculated as the product of the internal memory and the output gate. It is calculated by Equation (7)

$$s_t = tanh(c_t)^o o. \quad \text{(Equation 7)}$$

In this equations, x_t denotes the input vector, st the output vector, σ the sigmoid function, *tanh()* the hyperbolic tangent function, *U* the weight vector, W the weights, ct the cell state vector, *g()* the candidate hidden state, and i_t, *ft* and o_t the block gates.

In an email monitoring system utilizing ML, the significance of the LSTM architecture lies in its proficiency in processing sequential data. LSTM models are particularly adept at understanding the intricate dynamics within email conversations due to their ability to retain memory over extended sequences. This capability allows them to capture the temporal dependencies and subtle nuances present in email threads Zamir et al. (2020). In tasks such as email classification and priority labeling, where the context and order of information are crucial, LSTM models prove invaluable.

3.2.2.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks, also known as CNNs, are a popular class of deep learning models characterized by their space-invariant artificial neural networks (SIANN) with weights that are mutual. CNNs are also known as fully connected networks or multilayer where every node in subsequent layers is interconnected to each neuron in the previous layer (Punisˇkis et al., 2006). These models typically include hidden layers composed of convolutional layers, where tensors are fed into the network and features are extracted through convolutions. The output of these convolutional layers is then passed to subsequent layers for prediction. Overall, CNNs are a powerful tool for image and pattern recognition, with broad applications in disciplines like speech recognition, computer vision, and processing using natural language. The paper use CNN because it is good with pattern recognition, it is good for pridicting spam problem.

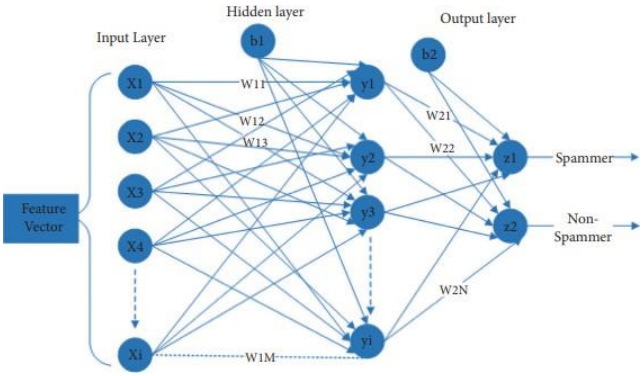


Figure 4: Convolutional neural network (CNN), (Siddique et al., 2021).

The CNN model diagram is presented in Figure 4. Convolutional layers, which conduct convolutions, are among the hidden layers in CNN. Tensors are fed into these convolutional layers, which extract features. The tensors were then moved on to the next layer, which resulted in a prediction.

The ability of CNNs to save the time and effort of feature extraction and selection motivated us to employ them in our work. A CNN receives raw data (images to be classified in our application) and, as they are passed through its layers, where fine representative features are extracted. However, CNNs training is a time-consuming process. The use of pre-trained models is the golden key to reduce the computational time and resources by applying CNN concept. However, a rescaling step is needed to fit the data to the input layer size of the pretrained model Punisˇkis et al. (2006). Data augmentation step is performed prior to the training procedure in order to increase the number of training data samples. This technique improves system performance and overcomes the lack of sufficient data samples.

3.2.3 Ensemble Classifiers

The ensemble classifier is an innovative classification technique that involves grouping various classifiers for training and then assembling them to enhance the method's accuracy on similar tasks such as spam filtering. The two most often used ensemble classifiers are the bagging classifier and the boosting classifier (Biggio et al., 2011). The bagging ensemble classifier has been employed to prevent poisoning attacks on spam filters (Netsanet et al., 2018). Empirical results indicate that the boosting-based classifier, that is, AdaBoost performs better with the specified feature set, as it does not involve random feature extraction. This review discusses the commonly used ensemble techniques in the domain of spam filtering. Figure 5 demonstrates various kinds of ensemble classifier.

3.2.3.1 AdaBoost Classifier

Boosting is a highly structured method that incorporates various weak learners to create a stronger learner that is more powerful compared to individual counterparts. One example of a boosting ensemble classifier is the AdaBoost system, which can produce better results even if the performance of weak learners is not good. These algorithms are highly effective in solving spam problems, with surveys indicating that they can generate superior classification results in contrast to techniques like Naive Bayes and Decision Tree (Gangavarapu et al., 2020). The AdaBoost algorithm is straightforward, fast, and simple to use, and requires minimal parameter tuning (except T). Furthermore, it is a versatile classifier that can be used with any kind of data, be it textual, numeric, or discrete. Moreover, it has been extended to address other classification problems beyond binary classification. The paper uses Adaptive Boosting because it is use to create a strong classifier using a number of weak classifier. Boosting is complete by creation a model from a training data sets, then create another model that will precise the faults of the first model and is highly effective in solving spam problems.

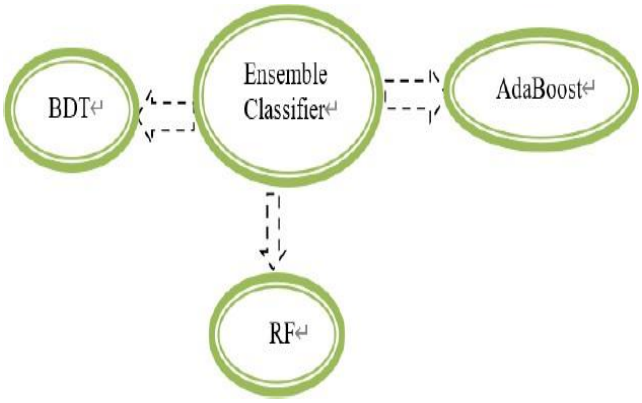


Figure 5: Ensemble Classifier, (Nisar et al., 2021).

3.2.3.2 Random Forest Classifier

A Random Forest classifier is a sophisticated ensemble tree classifier comprising diverse decision trees with

varying shapes and sizes. It employs random sampling of training data during tree construction and selects random subsets of input features when making node splits. This intentional introduction of randomness helps reduce the correlation among decision trees, ultimately enhancing the generalization capability of the ensemble by ensuring that the features of the individual trees do not resemble one another closely. The paper use Random Forest model to enhance the capability and reduce the correlation among decision trees on email monitoring system.

3.3 Perfomance Analysis

For the purpose of comparing the classification abilities of various classifier algorithms, four assessment metrics accuracy, precision, recall, and F1 score are established.

Accuracy: This metric evaluates the average number of correctly classified emails over the whole email dataset. Its equation is (8)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \qquad \text{(Equation 8)}$$

Precision: This metric indicates the reliability of the filter by measuring the proportion of true positive results to the total of positive results in the dataset as a whole. Its equation is (9)

$$Precision = \frac{TP}{TP + FP}, \qquad \text{(Equation 9)}$$

Recall: Recall evaluates the classifier sensitivity. The formula of recall is given in (10)

$$Recall = \frac{TP}{TP + FN}, \qquad \text{(Equation 10)}$$

F1-score: This metric reflects the balance between precision and recall, as shown in (11)

$$F1\_score = 2 * \frac{(Precision * Recall).}{(Precision + Recall)} \qquad \text{(Equation 11)}$$

In the context of classification, there are four potential results: true positive (TP) occurs when spam samples are accurately categorized as spam, true negative (TN) occurs when ham samples are correctly classified as ham, false positive (FP) occurs when spam samples are mistakenly classified as ham and false negative (FN) occurs when ham samples are mistakenly classified as spam.

In order to assess the effectiveness of a classifier, its diagnostic ability is measured by a receiver operating characteristic curve (ROC). This curve illustrates how the true-positive rate and the false-positive rate change at various threshold values. To compare classifiers, the area under the ROC curve (AUC) is used, with a higher AUC indicating better classification ability for the classifier. Six well-known classifier algorithms Spam emails are classified using Naive Bayes, SVM, CNN, LSTM, AdaBoost and Random Forest and their performance is statistically assessed.

Model loss refers to the classifier's inability to accurately forecast negative outcomes in each example. The ideal model presupposes that there will not be any losses. Otherwise, the loss would be larger if the model cannot make accurate predictions.

4. Results And Analysis

In this section, we carried out multiple experiments using various forms of text representation to assess the effectiveness of our method and compare it to other models and ML, DL and EL algorithms. In the course of this experimental analysis, testing was done on classification algorithms on two ML: Naive bayes and SVM, two DL models: LSTM and CNN and two ensemble model: AdaBoosting and Random Forest. The goal was to contrast the predictive performance to identify the most effective of the classifier methods. The same data distribution, 80% for training and 20% for testing was used across all experiments.

The choice of model in the analysis process depends on the specific objectives of the study. In this case, the goal is to identify spam in email. The effectiveness of the models can be evaluated based on two key factors: speed and accuracy. If the primary objective is to identify spam in email at a fast pace, models with faster computation times would be more suitable. These models may sacrifice some accuracy for speed. On the other hand, if accuracy is of utmost importance, models with higher accuracy rates should be prioritized, even if they have longer computation times. Considering the focus of this research on identifying depression signs in real-time as email come in, it is crucial to have a model that can classify them quickly and accurately. Therefore, it is necessary to analyze both the computational time and accuracy of the models to make an informed decision.

Based on the results shown in Table 6, the LSTM model stands out as the most effective option. It achieved the highest accuracy rate of 99.88% while maintaining a relatively low computational time. As we can see, the DL model LSTM is the most accurate model, but it requires a lot of training time. This combination of high accuracy and lot of computation makes it a strong contender for solving the spam identification problem in real-time email analysis. Looking at this, this paper can clearly state the LSTM model emerges as the most suitable choice. It balances both accuracy and but not computational time, making it an effective tool for identifying spam in email. SVM and Naive Bayes are ML models with around the same accuracy percentage as EL models AdaBoosting.

The Adaptive Boosting Classifier achieved an accuracy of 99.85% in this study, indicating its strong performance in detecting spam sentiment in email. Compared to other models in the study, the Adaptive Boosting Classifier had

the second highest accuracy score. Additionally, one previous study by Krause et al. (2019) reported a high accuracy of 98.88% for the Adaptive Boosting Classifier, further highlighting its effectiveness. The Adaptive Boosting Classifier's ability to capture complex patterns and interactions in the data likely contributed to its successful performance.

Overall, the Adaptive Boosting Classifier shows promise as a reliable model for spam detection in email. On the other hand, the random forest model presents mixed results. While it achieved the highest accuracy score in the study by Kumar et al. (2012) at 99.0%, it obtained a relatively lower score in the current study, with 98.60%. These variations could be attributed to different datasets or other factors. Overall, considering the consistently high accuracy scores and the specific requirements of the paper, LSTM emerged as the best model choice. However, the inclusion of other models such as SVM, Naive Bayes, CNN and Adaptive Boosting Classifier allows for a comprehensive comparison and exploration of their performance in sentiment analysis.

ML models that is SVM and Naive Bayes are employed in the computation of evaluation metrics like accuracy, precision, recall, and f-measures, which are described in Table 2. When it comes to accuracy, recall and f-measures, we found that Naive Bayes is more successful and gives superior outcomes, but when compared to Naive Bayes, SVM has the best precision. The comparison analysis findings presented in Table 2 prove that Naive Bayes produces better outcomes in terms of accuracy, and SVM performs better when it comes to precision, recall, and f-measure. The visual representation of the contrast between the outcomes produced by Naive Bayes and SVM is found in Figures 14 presented an insightful rendering of the Naive Bayes and SVM models performance through the confusion matrix and ROC curve. The AUC-ROC curve emerges as a quintessential tool for validation, graphically depicting model discrimination. FPR and TPR values grace the axes, graphics portraying efficacy. A high AUC signals superior class separation 0 as 0 and 1 as 1. Figures 14c, 14d, 17a, 17b, 22a and 22b show LSTM pronounced AUC advantage over CNN, Naive Bayes, SVM, AdaBoost and RF, solidifying its prowess in distinguishing ham from spam. In this, the Naive Bayes model reigns supreme, illuminating with unmatched brilliance.

Evaluation metrics determined for DL models are LSTM and CNN and we also find accuracy and loss of the model. As shown in Table 3 and Figures 16a - 16d, for both LSTM and CNN, model accuracy for training decreases as the epochs increase. Increased epoch count results in validation rate decreases. Model loss for both LSTM and CNN for training decreases as the number of epochs increases while validation slightly increases as the number of epochs increases. Finally, the training accuracy and test accuracy results were contrasted. We discovered that the LSTM is more accurate. The comparison study results demonstrate that LSTM gives superior results versus CNN training accuracy and test accuracy results.

EL models which are AdaBoosting and random forest a method that are employed to determine assessment metrics including accuracy, precision, recall, and f-measures, which are explained in Table 4. We discovered that AdaBoosting is more effective and yields superior outcomes in terms of accuracy, recall, and f-measures in terms of recall percentage when compared to accuracy, precision and f1-measure, Figures 18, 19, 20 and 21 show metrics for AdaBoost and Random Forest. The comparison analysis findings, which are shown in Table 5 show train accuracy and test accuracy, AdaBoost has more successful outcomes in terms of training accuracy. Additionally, each model's accuracy was determined, and assessment metrics including precision, recall, and f-measure for SVM and Naive Bayes and AdaBoosting for CNN and LSTM we find the training accuracy and test accuracy to compare and assess. The results showed that the LSTM model outperformed the other models with 99.88% of accuracy.

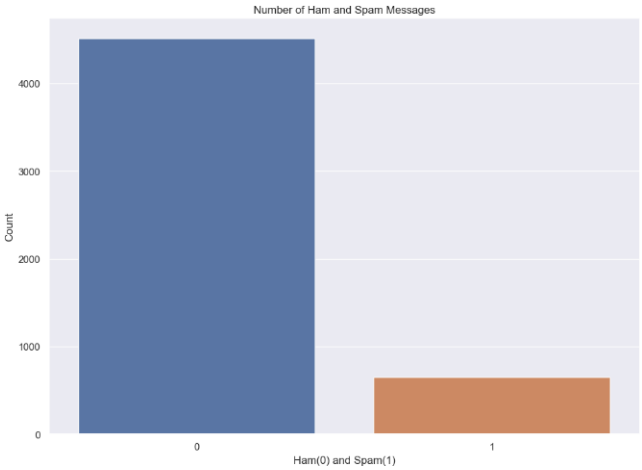


Figure 6: Number of ham and spam messages.

Figure 6 shows the percentage of spam and ham. It observed that spam emails contain a spam percentage of 12.633004449603405 % and ham emails contain a ham percentage of 87.3669955503966 %. But we can see from the graph, that ham emails contain a high percentage as compared to spam emails. As we can see, the classes are imbalanced, because there is a very high difference between the spam email and the ham email. Then we can say our dataset is an imbalanced dataset.

In this study, after preprocessing the datasets, various supervised machine learning algorithms were applied to train and test spam datasets, including Naive Bayes (NB), SVM, CNN, LSTM, Adaptive Boosting and Random Forest. StratifiedKFold was performed on spam datasets by specifying "shuffle = False". StratifiedKFold is a variation of cross-validation that leads to stratified folds MINASTIREANU and MESNITA (2020). In such case, in binary classification, each fold has preserved the proportion of classes of the original dataset. So, it assures that each fold is a representative of the whole dataset. For example, in spam datasets, StratifiedKFold ensures that it

maintains the 1:19 spam to ham ratio in each fold. In general, StratifiedKFold is a better method, in terms of both bias and variance, than KFold cross-validation. The reason for applying this technique in our experiment is mainly related to the bias of most classification algorithms which tend to weigh each sample equally, meaning that the dominant class gain as much weight, so the model created by machine learning algorithms is oriented and unidirectional; the classifier performance will be very low and usually the bias will increase. We consider 10 for K value in StratifiedKFold. It is evident that accuracy is not an appropriate measure to evaluate a model in an imbalanced dataset. The results of conducting six machine learning algorithms on spam datasets by applying Stratified10Fold and default parameters of classifiers in Scikit-learn are illustrated in figures 23a and 23b. The evaluation results in revealed that in the balanced datasets (the ratio of spam to ham in spam datasets was 1:1), the highest F1-score was related to NB classifier with 100% and 97% for SVM, respectively and AdaBoost has Max Accuracy of 98.84% while RF has 99.61%. While in imbalanced spam datasets (in which the spam to ham rate was 1:19), the highest F1-score belonged to NB, with 99% and 98% for SVM, respectively and AdaBoost has 94.2% while RF has 98.6%. The evaluation parameter results indicate a significant challenge in imbalanced datasets. In real-world datasets like spam datasets, in which the number of spam email was much lower than ham samples (1:19 spam to ham ratio), the performance of classifiers degraded dramatically. In an imbalanced dataset, standard classification algorithms tend to the majority classes; as a result, the model created by machine learning algorithms is seems biased, and the accuracy will be very low. Furthermore, we apply AUROC to measure how much each model is capable of distinguishing between spam and ham classes. Figures 25a, 25b, 27a and 27b shows that in balanced datasets, NB has the highest accuracy of 1.00, and 0.978 for SVM, respectively while AdaBoost has 98.84 and 99.61 for RF. In Figures 25a and 25b, the AUCs are reduced in imbalanced datasets to 0.989 and 0.980 in the NB classifier and SVM. On the other hand, the results of Figure 25 indicate that the rate of spam detection in the datasets gathered continuously is better than the randomly collected email. The evaluation results in balanced datasets are satisfactory, while a significant degradation is noticed in imbalanced datasets. The low detection rate arises the need for presenting a new method to assist in enhancing email spam detection in imbalanced datasets. Since NB has the best result in the analyses illustrated in Figure 25, and it is observed on Figure 27 RF has the best results, it is employed as an optimization model. NB is consisted of a large number of SVM that joint together to constitute an ensemble. So, the most frequent prediction will be the overall output of the model. Hence, a combination of over sampling technique and evolutionary algorithm is offered to optimize NB and RF, which improves spam detection rate considerably. After pre-processing the dataset, one can use the SMOTE algorithm to increase the minority-class samples (samples of spam class) to solve the class imbalance problem in an imbalanced datasets Bazzaz Abkenar et al. (2021). In our case, this technique will increase the number of spam instances in spam datasets.

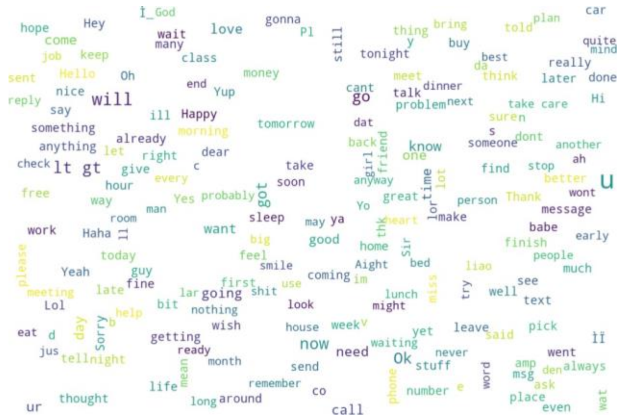


Figure 7a: Ham word cloud.

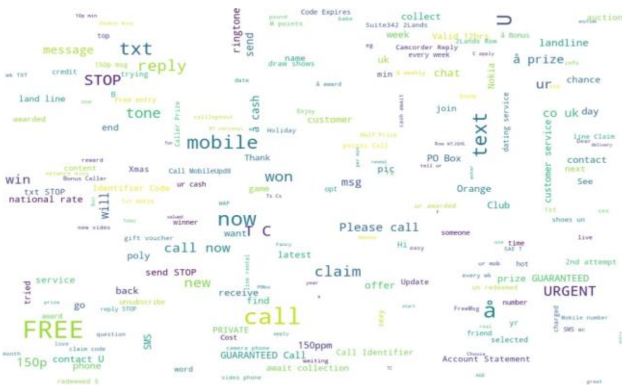
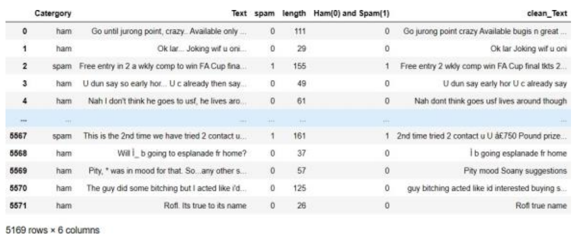


Figure 7b: Spam word cloud.

Figure 7: Ham and Spam word cloud

In Figures 7a - 7b depicts the most common words in the spam and ham messages. So we use the Word Cloud library, Figure 7a shows the ham word cloud (legitimate words) and Figure 7b shows the spam word cloud of those tempting words.

Figure 8 show that this dataset contained 5169 rows × 6 columns. From the six features, we have a category where we categorise spam and ham as a text message that is being sent through email, we created a column called spam in the form of binary where 1 represent spam and 0 for ham, we also have the length of the message in each email sent and only one of the six features were concentrated on which is the Text. The clean Text column was then added that contained the cleaned tweets.



	Catergory	Text	spam	length	Ham(0) and Spam(1)	clean_Text
0	ham	Go until jurong point, crazy.. Available only ...	0	111	0	Go jurong point crazy Available bugis n great ...
1	ham	Ok lar... Joking wif u oni...	0	29	0	Ok lar Joking wif u oni
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	1	155	1	Free entry 2 wkly comp win FA Cup final tkts 2...
3	ham	U dun say so early hor... U c already then say...	0	49	0	U dun say early hor U c already say
4	ham	Nah I don't think he goes to usf, he lives aro...	0	61	0	Nah dont think goes usf lives around though
...	...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	1	161	1	2nd time tried 2 contact u U â£750 Pound prize...
5568	ham	Will Ì_ b going to esplanade fr home?	0	37	0	Ì b going esplanade fr home
5569	ham	Pity, * was in mood for that. So...any other s...	0	57	0	Pity mood Soany suggestions
5570	ham	The guy did some bitching but I acted like i'd...	0	125	0	guy bitching acted like id interested buying s...
5571	ham	Rofl. Its true to its name	0	26	0	Rofl true name

5169 rows × 6 columns

Figure 8: Data cleaning and clean text

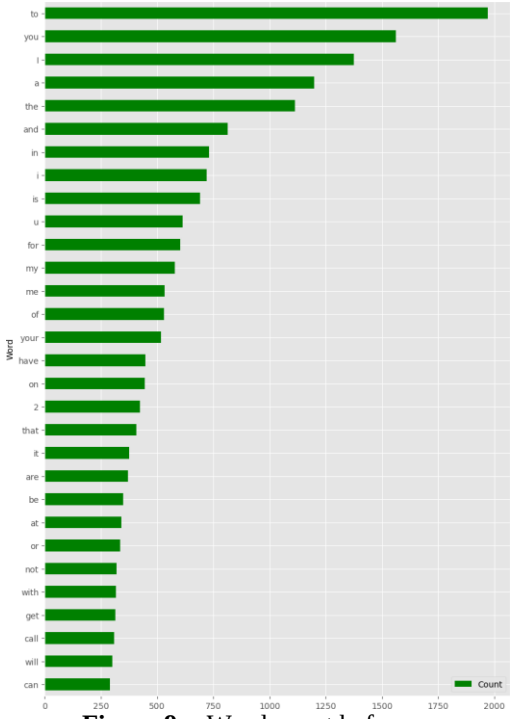


Figure 9a: Word count before pre-processing

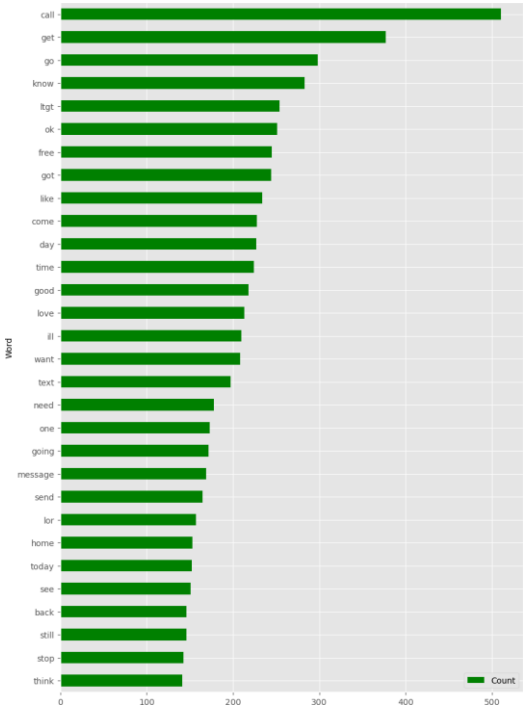


Figure 9b: Word count after pre-processing.

Figure 9: Word count before and after preprocessing

Figure 9a shows the word count before doing the preprocessing techniques; it shows the most appeared words in the emails. The plot shows the count of different words present in our dataset. For this, we are creating a function named word count plot and Figure 9b shows the word count after we completed the data preprocessing techniques, and plot the word count once again to see the most frequent words.

	num_characters	num_words	num_sentence
count	5572.000000	5572.000000	5572.000000
mean	80.118808	18.695621	1.970747
std	59.690841	13.742587	1.417778
min	2.000000	1.000000	1.000000
25%	36.000000	9.000000	1.000000
50%	61.000000	15.000000	1.000000
75%	121.000000	27.000000	2.000000
max	910.000000	220.000000	28.000000

Figure 10: Stats of characters, words and sentence in spam dataset.

4.1 Correlation Analysis

To analyse the relationship between public sentiment and spam figures, we determined the Pearson Correlation Coefficients. The significance of the correlation being non-zero was denoted by the reported p-value. The proportion of spam Against ham was computed, and the correlation was assessed using data smoothed by a 7-day moving average. Figure 11 displays these findings. Scatterplots were employed to illustrate the correlation coefficient visually.

Figure 10 the dataset which comprises 5572 observations of text data. On average, each observation contains 80.12 characters, 18.70 words, and nearly 2 sentences. Text lengths vary, with a standard deviation of 59.69 for characters, 13.74 for words, and 1.42 for sentences. The shortest observation has 2 characters, 1 word, and 1 sentence, while the longest spans 910 characters, 220 words, and 28 sentences. These statistics offer valuable insights into the distribution and structure of the text data.

The figures 11 and 12 shows that the proportion of number of characters, words and sentence with the same variables has a strong positive association, while the proportion of number of characters, words and sentence has a moderate negative association with the different variables. Because the total number of spam and ham are all cumulative numbers, these associations can also be explained by the correlation on Figure 12 with time. Notably, the corresponding value of statistical significance for the correlation coefficient is zero, representing statistically significant results.

4.2 Machine Learning Results

Here we compare the accuracy and performance of SVM with Naive Bayes Classifier for the same set of data. The following images consist of the factors that are being compared.

Figures 13a - 13b demonstrates the components, including the confusion matrix, classification report, and f1 measure for Naive Bayes and SVM Classifier.

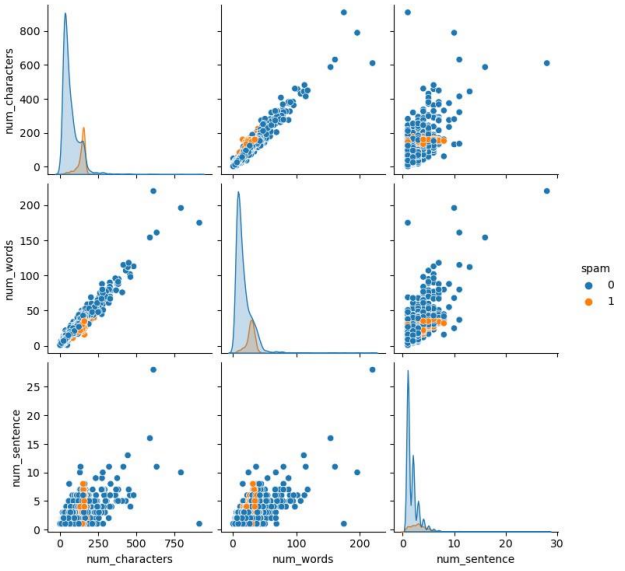


Figure 11: Correlation coefficient plots

4.3 Deep Learning Results

In this section, we present the construction of our deep learning model, featuring LSTM and CNN. By employing these components, the model autonomously extracts essential features, eliminating the need for an independent and resource-intensive feature extraction process.

In Figures 15a - 15b, we present using the Keras API the parameters architectures of CNN and LSTM. This Figure shows that LSTM has high Parameters as compared with CNN. In these Figures 15a - 15b, hyperparameters are utilized to define and configure LSTM and CNN models for a natural language processing task, likely sentiment analysis or text classification. The choice of hyperparameters significantly influences the model's ability to generalize and make accurate predictions.

Figures 16a - 16d demonstrate the model accuracy and model loss for CNN and LSTM for each epoch. The graph line is decreasing as the epochs increase, as can be seen. When the number of epochs is increased, the model loss rate decreases.

Table 3 displays training and test accuracy for two deep learning models: LSTM and CNN. LSTM achieved higher training accuracy 99.88% than CNN 99.71%, implying LSTM learned the training data better. Similarly, LSTM also showed better test accuracy of 98.36% compared to CNN 97.78%, indicating its superiority in generalising to new data. Both models performed well, with LSTM consistently outperforming CNN in accuracy metrics.

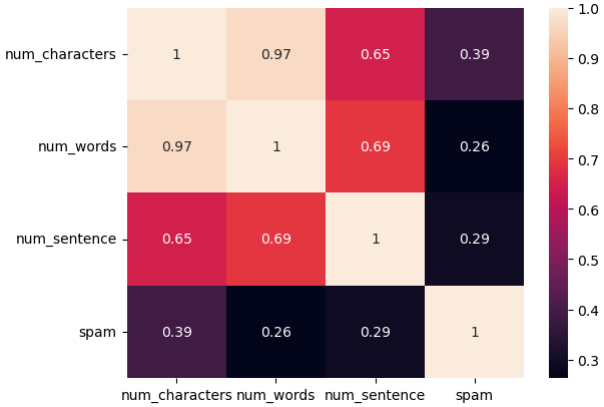


Figure 12: Correlation coefficient plots.

	precision	recall	f1-score	support
0	0.98	0.99	0.98	905
1	0.93	0.84	0.88	129
accuracy			0.97	1034
macro avg	0.95	0.91	0.93	1034
weighted avg	0.97	0.97	0.97	1034

```
test set
\Accuracy Score: 0.971953578336557
f1 Score: 0.8816326530612245
Recall: 0.8372093023255814
Precision: 0.9310344827586207
```

Figure 13a: Metrics for NB.

	precision	recall	f1-score	support
0	0.99	0.98	0.99	909
1	0.88	0.91	0.90	125
accuracy			0.97	1034
macro avg	0.94	0.95	0.94	1034
weighted avg	0.98	0.97	0.98	1034

```
test set
\Accuracy Score: 0.9748549323017408
f1 Score: 0.8976377952755905
Recall: 0.912
Precision: 0.8837209302325582
```

Figure 13b: Metrics for SVM

Figure 13: Metrics for NB and SVM.

4.4 Ensemble Learning Results

An ensemble model of these is made to get the best possible results among ML and DL models for the spam reviews dataset. A model is created from training data sets to finish off the process of boosting, after which a second model is created to correct the previous flaws.

Above we see that measuring performance change every time running code. So cross validation is applied to get the best possible result, also want to get classification report for best measured performance and its confusion matrix shown in Figures 23a and 23b, 24 and 26. Roc curve for NB, SVM, AdaBoost and RF shows perfect splitting data see Figure 25a, 25b and 27.

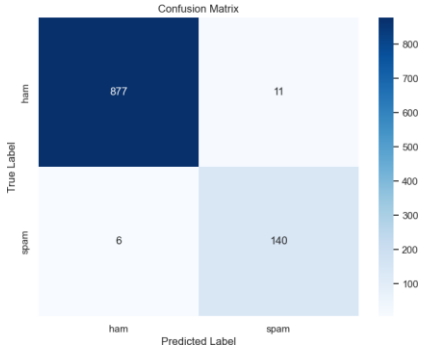


Figure 14a: Confusion Matrix for NB.

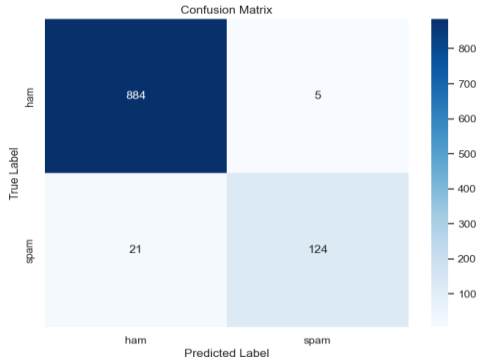


Figure 14b: Confusion Matrix for SVM.

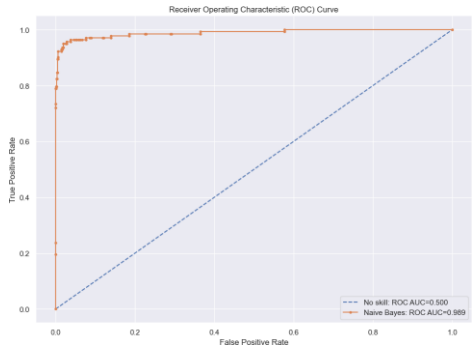


Figure 14c: ROC of NB.

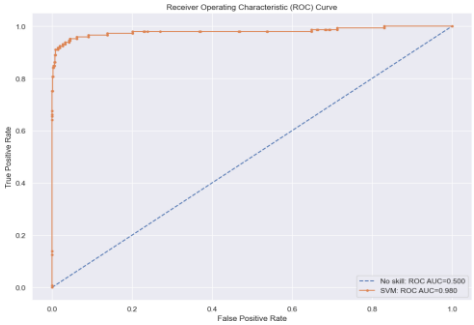


Figure 14d. ROC of SVM.

Figure 14. Confusion Matrix and ROC curve of NB and SVM model.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 79, 32)	248256
lstm (LSTM)	(None, 100)	53200
dropout (Dropout)	(None, 100)	0
dense (Dense)	(None, 20)	2020
dropout_1 (Dropout)	(None, 20)	0
dense_1 (Dense)	(None, 1)	21

Total params: 303497 (1.16 MB)
Trainable params: 303497 (1.16 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 15a: Parameters of LSTM architecture

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 79, 32)	248256
conv1d (Conv1D)	(None, 77, 128)	12416
max_pooling1d (MaxPooling1D)	(None, 25, 128)	0
dropout_2 (Dropout)	(None, 25, 128)	0
conv1d_1 (Conv1D)	(None, 23, 128)	49280
global_max_pooling1d (GlobalMaxPooling1D)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_4 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 32)	2080
dropout_5 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 1)	33

Total params: 320321 (1.22 MB)
Trainable params: 320321 (1.22 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 15b: Parameters of CNN architecture

Figure 15: Parameters of LSTM and CNN architecture.

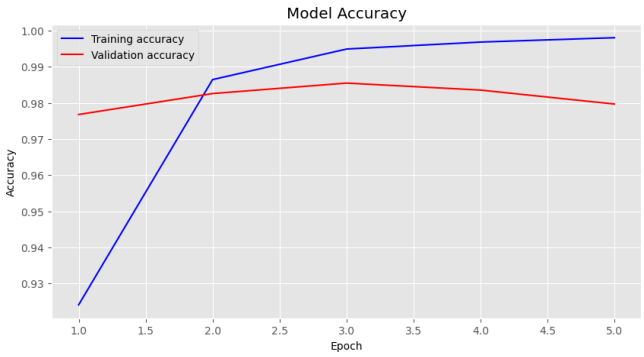


Figure 16a: Model accuracy for LSTM.

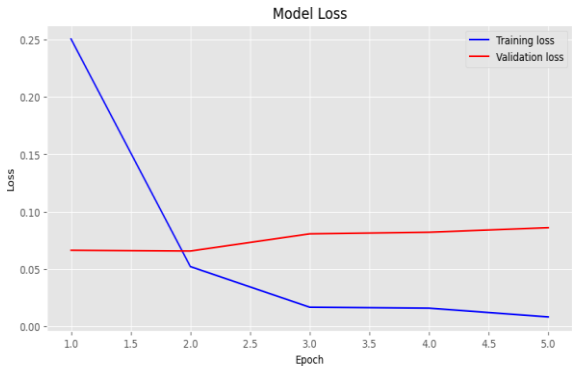


Figure 16b: Model loss for LSTM.

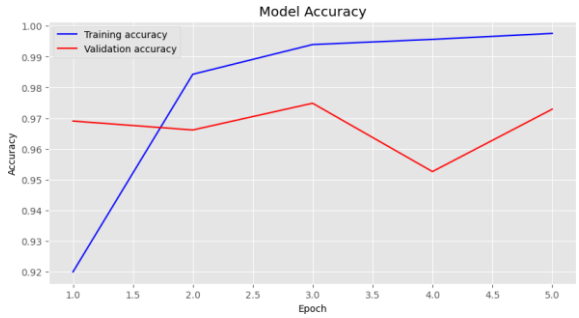


Figure 16c: Model accuracy for CNN.

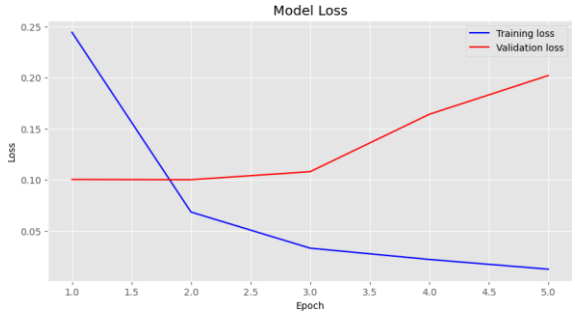


Figure 16d: Model loss for CNN.

Figure 16: Model accuracy and loss for LSTM and CNN model using our dataset.

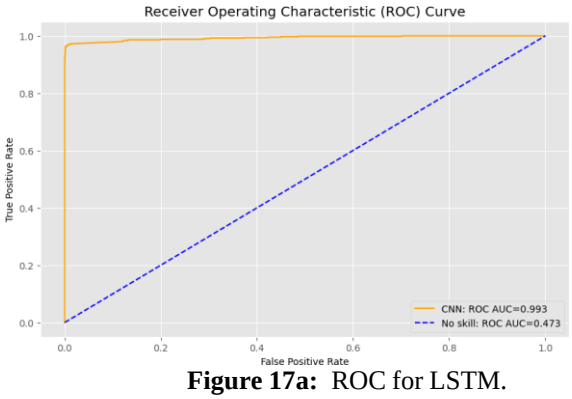


Figure 17a: ROC for LSTM.

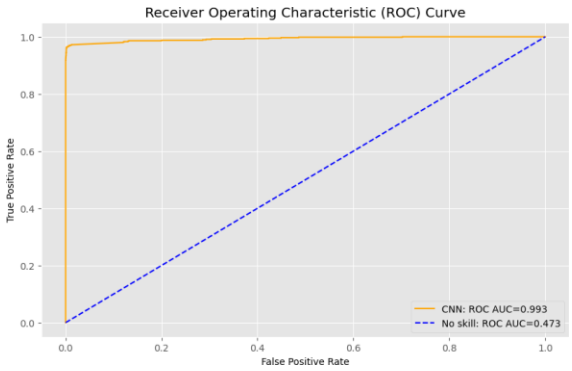


Figure 17b. ROC for CNN.

Figure 17: ROC for LSTM and CNN

	precision	recall	f1-score	support
0	0.98	0.99	0.98	1132
1	0.93	0.84	0.89	161
accuracy			0.97	1293
macro avg	0.95	0.92	0.94	1293
weighted avg	0.97	0.97	0.97	1293

test set
\Accuracy Score: 0.9729311678267595
f1 Score: 0.8859934853420195
Recall: 0.84472049689441
Precision: 0.9315068493150684

Figure 18. Matrix for Adaptive Boosting.

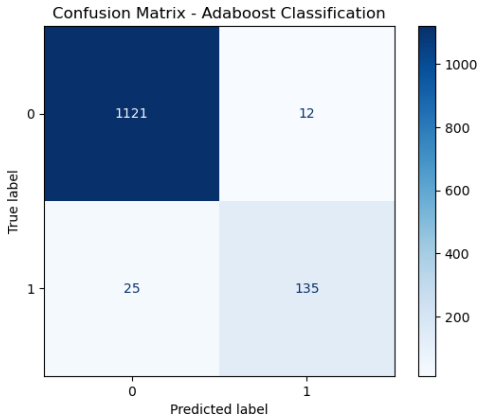


Figure 19: Confusion matrix for Adaptive Boosting

Train Results

Confusion Matrix for Train :
[[3377 0]
[494 5]]
Accuracy Score for Train : 0.8725490196078431
ROC AUC for Train : 0.9828208385975385

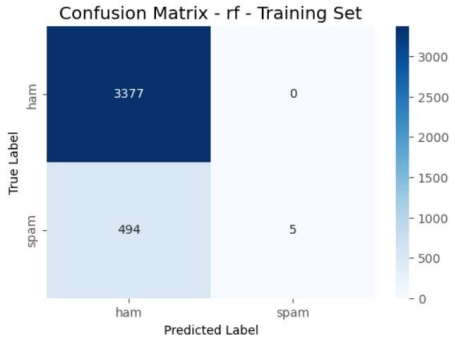


Figure 20: Confusion matrix for train random forest

```
Train Results

Confusion Matrix for Train :
 [[3377    0]
 [ 494    5]]
Accuracy Score for Train :  0.8725490196078431
ROC AUC for Train :  0.9828208385975385
```

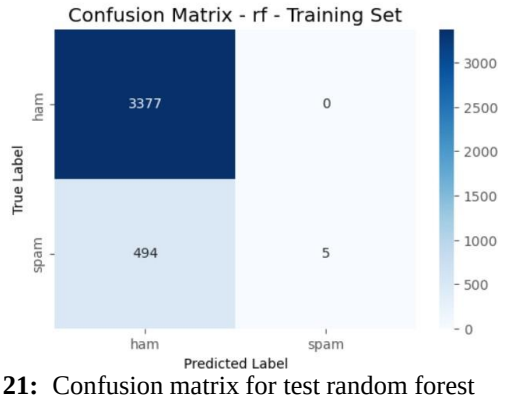


Figure 21: Confusion matrix for test random forest

.

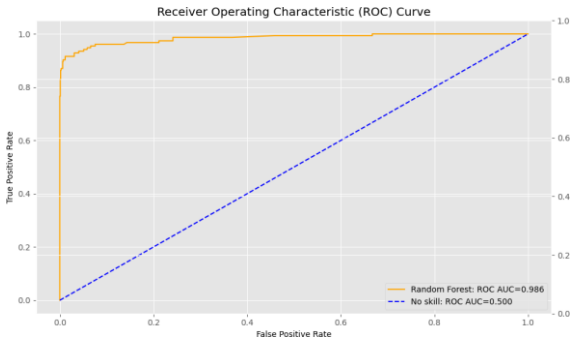


Figure 22a: ROC for Random Forest.

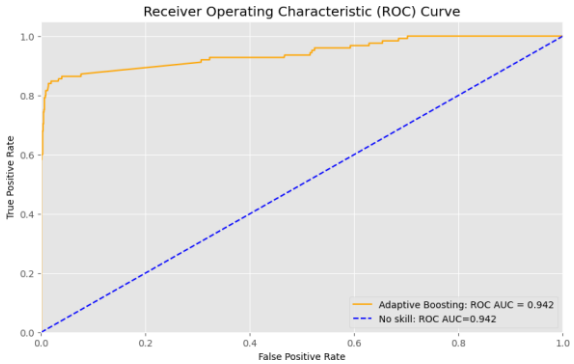


Figure 22b: ROC for Adaptive Boosting.

Figure 22: ROC for Random Forest and Adaptive Boosting

	precision	recall	f1-score	support
Ham	1.00	1.00	1.00	226
Spam	1.00	1.00	1.00	33
accuracy			1.00	259
macro avg	1.00	1.00	1.00	259
weighted avg	1.00	1.00	1.00	259

Figure 23a: NB matrix for cross validation.

	precision	recall	f1-score	support
Ham	0.99	1.00	1.00	226
Spam	1.00	0.94	0.97	33
accuracy			0.99	259
macro avg	1.00	0.97	0.98	259
weighted avg	0.99	0.99	0.99	259

Figure 23b: SVM matrix for cross validation.

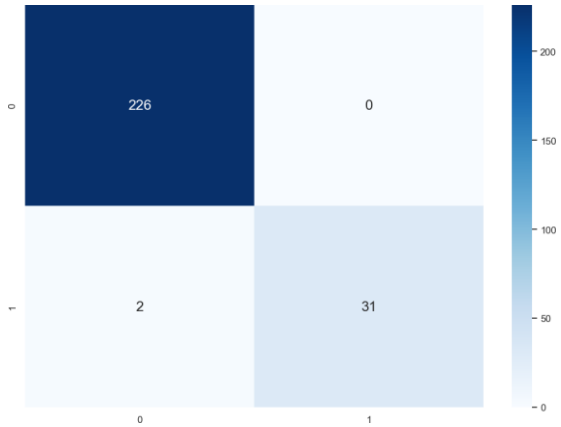


Figure 23: Matrix for NB and SVM

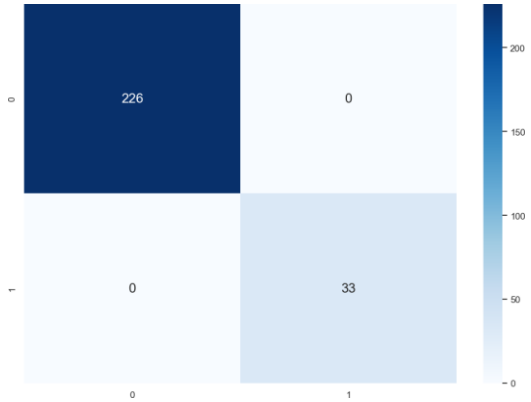


Figure 24a: NB confussion matrix for crossvalidation.

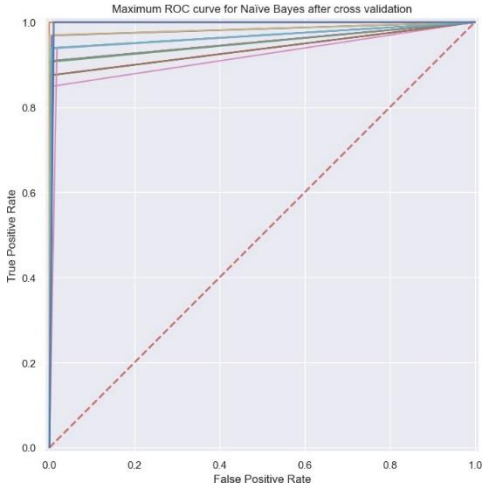


Figure 24b: SVM confussion matrix for crossvalidation.

Figure 25a: NB ROC for cross validation.

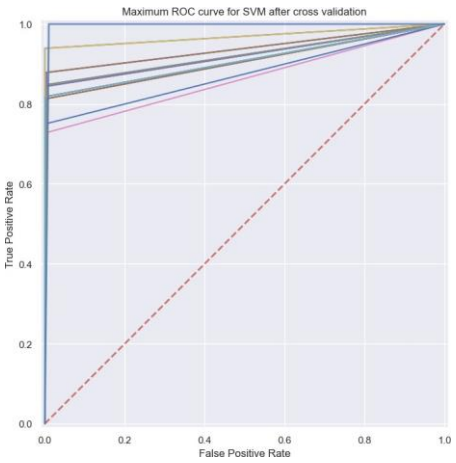


Figure 25b: SVM ROC for cross validation.

Figure 25. ROC for NB and SVM with cross validation.

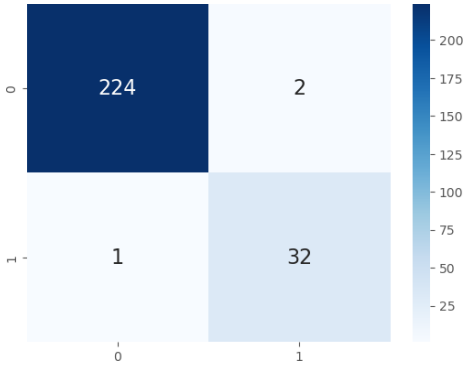


Figure 26a: AdaBoost confussion matrix for cross validation

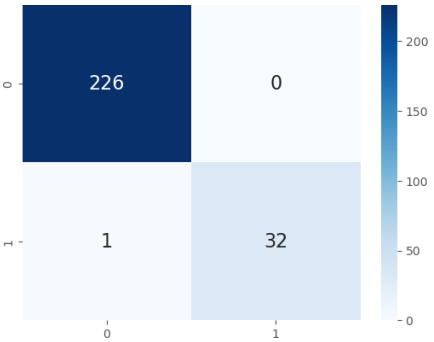


Figure 26b: RF confussion matrix for cross validation.

Figure 26. Matrix for AdaBoost and RF.

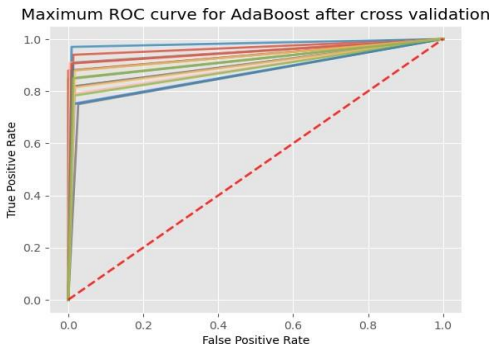


Figure 27a: AdaBoost ROC for cross validation.

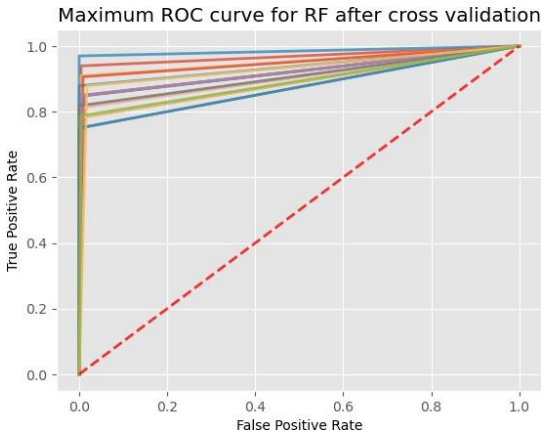


Figure 27b: RF ROC for cross validation

Figure 27. ROC for AdaBoost and RF with cross validation.

Table 2: Evaluation parameter values of ML models.

ML models	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Naive Bayes	97.58	89.33	93.71	91.47
SVM	97.49	96.12	85.52	90.51

Table 3: Training and test accuracy of DL models.

DL models	Training Accuracy (%)	Test Accuracy (%)
LSTM	99.88	98.36
CNN	99.71	97.78

Table 4: Evaluation parameter values of EL models.

EL models	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
AdaBoost	91.65	84.40	98.70	90.99
Random forest	98.85	97.58	96.41	98.78

Table 5: Training and test accuracy of EL models.

EL models	Training Accuracy (%)	Test Accuracy (%)
AdaBoost	99.85	97.14
Random forest	87.25	88.24

Table 6: The accuracy of various models.

Models	Accuracy (%)	ROC AUC (%)
Naive Bayes	97.58	98.9
SVM	97.49	98.0
LSTM	99.88	99.8
CNN	99.30	99.3
AdaBoosting	99.85	98.6
Random forest	98.6	94.2

From Table 6 the LSTM and CNN demonstrate superior performance in both accuracy and ROC AUC. The LSTM model achieves the highest accuracy of 99.88% and the highest ROC AUC of 99.8%. Similarly, the CNN model also performs impressively well, with an accuracy of 99.30% and a ROC AUC of 99.3%. These results indicate that both LSTM and CNN models are highly accurate in predicting class labels, and they also excel in distinguishing between classes, as evidenced by their high ROC AUC scores. Therefore, in terms of both accuracy and ROC AUC, the LSTM and CNN models outperform the other models listed in the table.

4.5 Comparison with Existing Studies

To demonstrate the competitiveness of our proposed email monitoring system, a comparison is provided in Tables 7 showing related recent studies results, whose experiments were conducted over the same or different datasets.

Table 7: Existing work related to email spam detection.

Reference	Algorithm used	Dataset	Accuracy (%)
Kumar et al. (2012)	ML	TANAGRA data mining	99.0
Salama et al. (2023)	DL	Dredze dataset, ISH dataset and improved dataset	99.87
Agarwal and Kumar (2018)	ML and DL	Ling Spam dataset	95.50
Hossain et al. (2021)	ML and DL	spam-base dataset	100
Jain et al. (2019)	ML and DL	SMS spam collection dataset and Twitter dataset	97.30
Magdy et al. (2022)	ML	SpamBase dataset	99.83
Siddique et al. (2021)	ML and DL	Urdu Emails Dataset	98.40
Krause et al. (2019)	ML and EL	CSDMC2010 dataset	98.88
Our work	ML, DL nd EL	SMS Spam Collection Dataset	99.88

The conducted comparative study shows that our suggested models are competitive with those found in the literature.

The findings of this study hold considerable significance for both companies and employees, particularly in the realm of email monitoring systems. Not only will the insights gleaned from this research contribute to the development of more precise email monitoring systems, but they will also enhance employee awareness, thereby reducing susceptibility to spam and minimizing its detrimental impact on the company. Moreover, this investigation represents a significant advancement in understanding the intricacies of email monitoring systems, paving the way for further knowledge enhancement in this domain. By assessing the results and their effects, valuable data will be generated, aiding companies in implementing effective strategies to combat malicious activities such as phishing and spam, whether perpetrated by employees or external threats. Ultimately, this study has the potential to serve as a cornerstone for future research endeavours in related fields within the realm of data science.

5. Conclusion

Email has become the predominant mode of communication in the modern era, enabling the global dissemination of messages through internet connectivity. Each day witnesses an exchange of over 270 billion emails, with roughly 57% of this volume comprising unsolicited spam messages Fallows (2002). These Spam emails, often known as non-authentic communications, encompass emails that are harmful or commercially oriented. They have the potential to compromise personal information, including financial data such as bank details, and can cause distraction on individuals, corporations, or even entire communities. Beyond their promotional aspect, these emails might embed links leading to websites engaged in phishing or hosting malware, aiming to illicitly obtain confidential data. The issue of spam is not merely an inconvenience for end-users; it carries financial repercussions and poses a significant security threat. Consequently, an intricately devised system has been formulated to identify and intercept unwarranted and intrusive emails, effectively curbing the influx of spam messages. The successful implementation of this system would prove immensely advantageous to both individuals and companies. Organisations can create clear policies for email monitoring, convey these policies to their staff, and inform their staff about the dangers that may be posed by spam as well as the effects of becoming a victim of it Magdy et al. (2022).

The rapid growth of using email, along with the ever increasing tendency of people on this platform, attracts many spammers from all over the world. Most papers discussed in related work were conducted without taking class imbalance into account so their findings are related to balanced datasets, whereas, the spam detection rate was still low in those papers which considered imbalanced datasets. We conducted an empirical study of six machine learning algorithms on spam datasets which revealed that LSTM performed better in detecting spammers, although the detection rate was still low, especially in imbalanced datasets. Therefore, to mitigate the class imbalance problem in spam detection, a combination of StratifiedKFold algorithm was employed on the SVM, AdaBoost, RF and NB classifier in this paper. Since

standard classification algorithms have a tendency towards the majority classes, in an imbalanced dataset, the model created by machine learning algorithms is biased, and the accuracy will be very low. StratifiedKFold was employed to tackle the imbalanced class distribution of spam datasets. The proposed method was evaluated on real imbalanced datasets. Cross-validation in DL might be a little tricky because most of the CV techniques require training the model at least a couple of times Powers and Atyabi (2012).

In this study, a ML, DL, and EL algorithms for identifying spam emails has been developed. To conduct the study, a dataset containing both spam and legitimate (ham) emails is gathered from Kaggle. This dataset undergoes preprocessing for various analytical methods. The paper use a confusion matrix to investigate the accuracies of the models. The performance evaluation is carried out using metrics like accuracy, precision, recall, F-measure, ROC-AUC for SVM, Naive Bayes and AdaBoosting, for LSTM and CNN model accuracy and model loss are determined for comparison. The analysis of this paper find out that NB perform better compared to SVM on cross validation. The paper compared the performance of ML, DL and EL models in Table 6. The study findings suggest that deep learning models excel at distinguishing spam emails. Specifically, the LSTM algorithm stands out with an impressive estimated accuracy rate of 99.88% and a low test accuracy of 98.36%. Although training LSTM might take a bit longer compared to CNN, SVM, Naive Bayes, AdaBoosting, and Random forest its effectiveness and accuracy surpass the other approaches. The future extension of this work includes the use of cross validation techniques in deep learning models. Furthermore, in the extension of this work, we are interested in testing the model on new datasets to detect spam.

Data Availability Statement

The datasets used and analyzed during the current study are available on the Kaggle website, and the link is https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset.

Funding

This project is sponsored by DSI-NICIS National e-Science Postgraduate Teaching and Training Platform (NEPTTP).

Acknowledgements

The author would like to acknowledge University of the Witwatersrand and DSI-NICIS National e-Science Postgraduate Teaching and Training Platform (NEPTTP) Electrical and School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa, for providing the technical facilities to carry out this work.

Conflict Of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Ethics Approval And Consent To Participate

Not applicable.

References

[1] Afzal, H. and Mehmood, K. (2016). Spam filtering of bi-lingual tweets using machine learning. In 2016 18th International conference on advanced communication technology (ICACT) (IEEE), 710–714

[2] Agarwal, K. and Kumar, T. (2018). Email spam detection using integrated approach of naive bayes and particle swarm optimization. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (IEEE), 685–690

[3] Ahmed, N., Amin, R., Aldabbas, H., Koundal, D., Alouffi, B., and Shah, T. (2022). Machine learning techniques for spam detection in email and iot platforms: analysis and research challenges. Security and Communication Networks 2022, 1–19

[4] Akhtar, A., Tahir, G. R., and Shakeel, K. (2017). A mechanism to detect urdu spam emails. In 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) (IEEE), 168–172

[5] Alpaydin, E. (2020). Introduction to machine learning (MIT press)

[6] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., and Spyropoulos, C. D. (2000). An evaluation of naive bayesian anti-spam filtering. arXiv preprint cs/0006013

[7] Bazzaz Abkenar, S., Mahdipour, E., Jameii, S. M., and Haghi Kashani, M. (2021). A hybrid classification method for twitter spam detection based on differential evolution and random forest. Concurrency and Computation: Practice and Experience 33, e6381

[8] Biggio, B., Corona, I., Fumera, G., Giacinto, G., and Roli, F. (2011). Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In Multiple Classifier Systems: 10th International Workshop, MCS 2011, Naples, Italy, June 15-17, 2011. Proceedings 10 (Springer), 350–359

[9] Breiman, L. (1996). Bagging predictors. Machine learning 24, 123–140

[10] Chen, X.-l., Liu, P.-y., Zhu, Z.-f., and Qiu, Y. (2009). A method of spam filtering based on weighted support vector machines. In 2009 IEEE International Symposium on IT in Medicine & Education (IEEE), vol. 1, 947–950

[11] Chhabra, P., Wadhvani, R., and Shukla, S. (2010). Spam filtering using support vector machine. Special Issue IJCCT 1, 3

[12] Chory, R. M., Vela, L. E., and Avtgis, T. A. (2016). Organizational surveillance of computer-mediated workplace communication: Employee privacy

concerns and responses. Employee Responsibilities and Rights Journal 28, 23–43

[13] Drucker, H., Wu, D., and Vapnik, V. N. (1999). Support vector machines for spam categorization. IEEE Transactions on Neural networks 10, 1048–1054

[14] Fallows, D. (2002). Email at work (Pew Internet & American Life Project)

[15] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., and Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications 50, 102419

[16] Friedman, B. A. and Reed, L. J. (2007). Workplace privacy: Employee relations and legal implications of monitoring employee e-mail use. Employee Responsibilities and Rights Journal 19, 75–83

[17] Gangavarapu, T., Jaidhar, C., and Chanduka, B. (2020). Applicability of machine learning in spam and phishing email filtering: review and approaches. Artificial Intelligence Review 53, 5019–5081

[18] Garcez, A. d., Gori, M., Lamb, L. C., Serafini, L., Spranger, M., and Tran, S. N. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. arXiv preprint arXiv:1905.06088

[19] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation 9, 1735–1780

[20] Hossain, F., Uddin, M. N., and Halder, R. K. (2021). Analysis of optimized machine learning and deep learning techniques for spam detection. In 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (IEEE), 1–7

[21] Iyengar, A., Kalpana, G., Kalyankumar, S., and GunaNandhini, S. (2017). Integrated spam detection for multilingual emails. In 2017 International Conference on Information Communication and Embedded Systems (ICICES) (IEEE), 1–4

[22] Jain, G., Sharma, M., and Agarwal, B. (2019). Optimizing semantic lstm for spam detection. International Journal of Information Technology 11, 239–250

[23] Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., and Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. IEEE Access 7, 168261–168295

[24] Krause, T., Uetz, R., and Kretschmann, T. (2019). Recognizing email spam from meta data only. In 2019 IEEE Conference on Communications and Network Security (CNS) (IEEE), 178–186

[25] Kumar, N., Sonowal, S., et al. (2020). Email spam detection using machine learning algorithms. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (IEEE), 108–113

[26] Kumar, R. K., Poonkuzhali, G., and Sudhakar, P. (2012). Comparative study on email spam classifier using data mining techniques. In Proceedings of the international multiconference of engineers and computer scientists (Newswood Limited, Hong Kong), vol. 1, 14–16

[27] Kumaresan, T. and Palanisamy, C. (2017). E-mail spam classification using s-cuckoo search and support vector machine. International Journal of Bio-Inspired Computation 9, 142–156

[28] Magdy, S., Abouelseoud, Y., and Mikhail, M. (2022). Efficient spam and phishing emails filtering based on deep learning. Computer Networks 206, 108826

[29] Masood, F., Almogren, A., Abbas, A., Khattak, H. A., Din, I. U., Guizani, M., et al. (2019). Spammer detection and fake user identification on social networks. IEEE Access 7, 68140–68152

[30] MINASTIREANU, E.-A. and MESNITA, G. (2020). Reducing type ii errors in credit card fraud detection using xgboost classifier. In Proc. 19th Int. Conf. INFORMATICS Econ. Educ. Res. Bus. Technol. 174–182

[31] Mishra, R. and Thakur, R. (2013). Analysis of random forest and naive bayes for spam mail using feature selection categorization. International Journal of Computer Applications 80, 42–47

[32] Netsanet, S., Zhang, J., and Zheng, D. (2018). Bagged decision trees based scheme of microgrid protection using windowed fast fourier and wavelet transforms. Electronics 7, 61

[33] Nisar, N., Rakesh, N., and Chhabra, M. (2021). Review on email spam filtering techniques. International Journal of Performability Engineering 17

[34] Olatunji, S. O. (2019). Improved email spam detection model based on support vector machines. Neural Computing and Applications 31, 691–699

[35] Powers, D. M. and Atyabi, A. (2012). The problem of cross-validation: averaging and bias, repetition and significance. In 2012 Spring Congress on Engineering and Technology (IEEE), 1–5

[36] Punisˇkis, D., Laurutis, R., and Dirmeikis, R. (2006). An artificial neural nets for spam e-mail recognition. Elektronika ir Elektrotechnika 69, 73–76

[37] Rana, S., Jasola, S., and Kumar, R. (2011). A review on particle swarm optimization algorithms and their applications to data clustering. Artificial Intelligence Review 35, 211–222

[38] Salama, W. M., Aly, M. H., and Abouelseoud, Y. (2023). Deep learning-based spam image filtering. Alexandria Engineering Journal 68, 461–468

[39] Scholkopf, B. and Smola, A. J. (2002). Learning with kernels: support vector machines, regularization, optimization, and beyond (MIT press)

[40] Sharma, P. and Bhardwaj, U. (2018). Machine learning based spam e-mail detection. International Journal of Intelligent Engineering & Systems 11

[41] Siddique, Z. B., Khan, M. A., Din, I. U., Almogren, A., Mohiuddin, I., and Nazir, S. (2021). Machine learning-based detection of spam emails. Scientific Programming 2021, 1–11

[42] Smith, W. P. and Tabak, F. (2009). Monitoring employee e-mails: Is there any room for privacy? Academy of Management Perspectives 23, 33–48

[43] Sundermeyer, M., Schlu¨ ter, R., and Ney, H. (2012). Lstm neural networks for language modeling. In Interspeech. vol. 2012, 194–197
[44] Suryawanshi, S., Goswami, A., and Patil, P. (2019). Email spam detection: an empirical comparative study of different ml and ensemble classifiers. In 2019 IEEE 9th International Conference on Advanced Computing (IACC) (IEEE), 69–74
[45] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems 27 [Dataset] Venkatesh, R. (2021). Spam mails dataset
[46] Vyas, T., Prajapati, P., and Gadhwal, S. (2015). A survey and evaluation of supervised machine learning techniques for spam e-mail filtering. In 2015 IEEE international conference on electrical, computer and communication technologies (ICECCT) (IEEE), 1–7
[47] Yu, B. and Xu, Z.-b. (2008). A comparative study for content-based dynamic spam classification using four machine learning algorithms. Knowledge-Based Systems 21, 355–362
[48] Zamir, A., Khan, H. U., Mehmood, W., Iqbal, T., and Akram, A. U. (2020). A feature-centric spam email detection model using diverse supervised machine learning algorithms. The Electronic Library 38, 633–657