International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-12, Special Issue-1, March-2024
https://doi.org/10.55524/CSISTW.2024.12.1.47
Computing for Sustainable Innovation: Shaping Tomorrow's World (CSISTW-2024) on 16th-17th February, 2024
Organized by B.N. College of Engineering and Technology, Lucknow, India

Spam Email Detection Using Deep Learning Model

Rajnikant[1], and Mr.Vinay Kumar[2]

[1,] P.G. Scholar, Department of CSE, BN College of Engineering and Technology, Lucknow, Uttar Pradesh, India
[2]Assistant Professor, Department of CSE, BN College of Engineering and Technology, Lucknow, Uttar Pradesh, India

Correspondence should be addressed to Rajnikant; rajnikant16071997@gmail.com

ABSTRACT- The persistent menace of spam emails poses a formidable cybersecurity threat, demanding the implementation of robust classification models for prompt detection and prevention. This research addresses the intricate challenge of classifying spam emails by evaluating diverse machine learning and deep learning models. The focal point of the identified problem lies in the inherent complexity of distinguishing spam emails from legitimate ones, leading to potential security vulnerabilities. Existing models' limitations, particularly in managing false positives and false negatives, underscore the imperative for enhanced techniques. The chosen methodology incorporates conventional machine learning models, such as Random Forest, Multinomial Naive Bayes (MNB), and Support Vector Machine (SVM), in conjunction with the Netcraft dataset. Additionally, deep learning models, including Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Bidirectional LSTM (BiLSTM), are deployed with meticulous hyperparameter tuning. An attention mechanism is introduced to dynamically allocate distinct weights to email headers and bodies based on their significance. The outcomes unveil the consistently superior performance of the Random Forest model, attaining a flawless F1-score of 1.0 for spam email classification. The SVM model also exhibits excellence with impeccable precision and recall across all folds. Nevertheless, the MNB model encounters difficulties in classifying legitimate emails, influencing overall metrics. The deep learning models, especially LSTM, CNN, BiL STM, and RNN, demonstrate exceptional efficacy with remarkably minimal test losses and elevated accuracies, underscoring their prowess in spam email classification tasks.

KEYWORDS: Spam Email, SVM, Random Forest, Deep Leaning, Machine learning

I. INTRODUCTION

The expansive growth of the digital realm has led to substantial shifts in how individuals interact online. Regrettably, it has also spawned a troubling trend - a swift surge in security threats, marked by the constant emergence of new and varied types of attacks. These attacks not only jeopardize the integrity of victims' systems but also aim at depleting their financial resources [1][2]. In this ever-changing landscape, a particular form of attack prominently stands out - the spam email attack. This illicit activity exploits social networking technologies and platforms to amass a target's identity data and account information. In the effective counteraction of spam emails, diverse techniques and principles have been devised, where email filtering assumes a pivotal position [3][4]. Email filtering involves an array of approaches, incorporating mechanisms like blacklisting, visual similarity assessments, heuristics, and machine learning algorithms. These approaches serve as valuable means for identifying spam emails to a certain degree, furnishing a degree of defense against such threats. Nevertheless, they do possess inherent constraints. The primary goal of email filtering mechanisms is to identify and isolate potentially harmful or fraudulent emails before they reach the recipient's inbox. Blacklisting, for instance, involves maintaining a list of known malicious email addresses, domains, or IP addresses. When an incoming email matches an entry on this blacklist, it is flagged as suspicious. Visual similarity checks analyze the content of emails, looking for visual cues and patterns that resemble known Spam emails. Heuristic approaches rely on predefined rules and patterns to identify suspicious characteristics in emails. Machine learning algorithms use historical data to train models capable of recognizing Spam patterns. While these techniques prove beneficial, there might be instances where they are less effective, especially when confronted with spam emails featuring dubious IP-linked URLs and incongruent URLs. Cybercriminals have honed their skills in camouflaging their intentions by utilizing URLs that do not immediately raise suspicion. Moreover, they may utilize IP-linked URLs that, while less conventional, pose an equally menacing threat. Traditional email filtering approaches may not sufficiently tackle these sophisticated tactics, potentially exposing users to vulnerabilities in spam attacks. Emails containing URLs with IP addresses pose a more significant threat because, when users click on them, they are often redirected to deceptive web pages that cunningly request confidential information. This not only jeopardizes the user's financial security but can also lead to identity theft [5]. Numerous approaches have been suggested, integrating machine learning models with diverse classifiers to categorize emails based on a set of extracted attributes. These attributes encompass variations in href tags, the occurrence of commonly exploited phrases such as 'click,' 'here,' and 'click here,' the presence of JavaScript code, the quantity of dots in domain names, and the existence of specific HTML codes. Other pivotal features include the tally of links, associations with domains, and URLs incorporating IP addresses, along with assessments for body-to-domain alignment and the presence of keywords like 'update,' 'confirm,' 'verify account,' 'restrict,' and 'suspend.' An

alternate approach, outlined in the paper [6], utilizes the 'bag of words' concept. In this method, all words present in the email content are isolated, and the frequency of each word acts as a feature for classifying spam emails. Employing machine learning, these techniques aim to enhance email security and assist users in recognizing messages that may pose potential harm. Insights from the Anti-Spam Working Group (APWG) reveal the concerning escalation of Spam emails. The reported count of Spam emails skyrocketed from 87,390 in 2019 to an astonishing 266,342 in 2020 [7][8]. The extensive variety of Spam email categories, reaching approximately 100,000 by December 2020, emphasizes the flexibility and continually changing nature of this cyber menace. The heightened frequency of Spam email attacks has led to significant monetary losses, especially in advanced nations such as Australia, the United Kingdom, and the United States. These financial setbacks are estimated at around 600 million dollars per year. The repercussions of such attacks extend widely, impacting both individuals and organizations, emphasizing the pressing requirement for enhanced email security measures. Emails containing IP address-based URLs are particularly pernicious due to their potential to redirect users to deceptive websites soliciting sensitive information. Machine learning models have been instrumental in developing techniques to classify such emails based on various extracted features. These features include HTML disparities, the presence of commonly abused phrases, JavaScript code, domain characteristics, and specific keywords. Another approach involves using a 'bag of words' model to analyze the frequency of words within email content for classification.The economic consequences of these Spam attacks are significant, with annual losses estimated at around 600 million dollars in developed nations such as Australia, the United Kingdom, and the United States. This underscores the immediate requirement for effective email security measures to safeguard individuals and organizations from the escalating threat posed by Spam emails.The rising threat of Spam emails underscores the vital importance of user awareness. Individuals should avoid opening links in suspicious emails and refrain from sharing confidential information. Simultaneously, alongside this heightened awareness, it is crucial to implement an effective approach for detecting and preventing Spam emails to minimize potential losses [9] [10].

II. LITERATURE SURVEY

Spam emails, a notorious category within spam attacks, pose a significant challenge in the domain of email security. The efficacy of detection and mitigation of these threats depends on the classifiers employed within the detection models. Additionally, the features extracted and utilized during the training and testing phases play a crucial role in the success of these models. In a recent investigation detailed in the paper [12], a pioneering approach was introduced, relying on the incorporation of ten distinct features to represent Spam emails. This feature-rich representation served as the basis for training and testing a robust email security model. Notably, the study employed the Random Forest algorithm, a potent ensemble learning method that constructs multiple decision trees to collectively make well-informed classification decisions. The decision trees generated within the Random Forest model played a crucial role in the email security process. They worked by examining the presence of the same set of features in incoming emails. If any of these features were detected in an email, the model flagged it as a potential Spam attempt. The results of this approach were highly promising, with an impressive accuracy rate of nearly 96%. This level of accuracy significantly contributes to the overall security of email communication. Notably, the model's performance in terms of false negatives and false positives was equally remarkable. The false negative rate, which reflects the model's ability to correctly identify genuine Spam emails, stood at a mere 4%. Additionally, the false positive rate, indicating the likelihood of legitimate emails being misclassified as Spam threats, was an impressively low 0.1%. These outcomes showcase the efficiency and reliability of the model, underscoring its potential to significantly enhance email security by effectively reducing Spam threats.In another innovative approach, the concept of a multi-layer classifier was introduced in a different study [11]. This method incorporates a three-tier classifier system, enhancing the model's ability to detect Spam emails effectively. The multi-layer classifier approach capitalizes on a hierarchical classification process, allowing for a more thorough and accurate assessment of email content. The significance of these advancements in email security is clear. With the ever-evolving landscape of Spam attacks, having reliable and robust models for detection is essential. The feature-rich representation and the use of advanced classification techniques, such as Random Forest and multi-layer classifiers, empower email security systems to stay ahead of emerging threats.These models not only improve the accuracy of identifying Spam emails but also reduce false negatives and false positives, ultimately enhancing the overall security and trustworthiness of email communications. A cutting-edge approach involves a multi-layer classifier system, consisting of three tiers. If the initial two tiers fail to effectively classify an email, the third tier classifier comes into play to make the final determination. This comprehensive approach leads to an average accuracy rate exceeding 96%, which is a commendable achievement. However, there are trade-offs involved, as the utilization of a multi-layer classifier demands more time and memory resources [12][13][29]. Another approach to Spam email detection relies on a clustering method, a concept rooted in combining supervised classification algorithms with unsupervised clustering techniques. This approach introduces a level of adaptability and synergy between these two methodologies, aiming to enhance the accuracy of email classifications. Following this, a consensus clustering technique is used in the training of the model. This method streamlines the classification process, offering improved accuracy compared to the traditional k- means classification algorithm. In a separate framework for Spam email detection, the concept of fuzzy logic is employed. Fuzzy logic introduces a degree of uncertainty and imprecision, allowing for more nuanced decision-making. This approach capitalizes on the subtle, non-binary nuances present in email content, enhancing the model's ability to detect Spam emails effectively [15][16]. These diverse approaches highlight the constant evolution and innovation within email security. While the multi-layer classifier system boasts high

accuracy rates, it requires substantial time and memory resources. The clustering method leverages a combination of supervised and unsupervised techniques, leading to more accurate classifications. Lastly, the fuzzy logic framework introduces a level of adaptability and nuance in email detection.In an era where Spam attacks are increasingly sophisticated, these advanced approaches are essential for enhancing email security. They not only improve the accuracy of identifying Spam emails but also offer flexibility and adaptability to stay ahead of emerging threats. Ultimately, these innovative methods contribute to more reliable and effective email security systems, safeguarding individuals and organizations from the ever-present risk of Spam attacks. In this process, 21 features are initially extracted during preprocessing. In the subsequent stage, feature reduction is employed. The retained features serve as the basis for categorizing emails as either legitimate or Spam. Remarkably, this model achieves an approximateaccuracy of 97%, underscoring its effectiveness in distinguishing between the two email categories [17][8].
A model is introduced, utilizing logistic regression as a classifier for email classification between legitimate and non-legitimate emails. Training on a noisy dataset, it incorporates 10 features, including URLs and href disparities. Notably, the model achieves an impressive accuracy of 95%, demonstrating its proficiency in email classification. Furthermore, the model boasts a remarkably low false positive rate of just 0.03%, signifying its ability to effectively identify legitimate emails while minimizing the misclassification of non-legitimate ones, thus enhancing email security[18][19]. A Spam email detection model is introduced, employing support vector machines for classification. The model is trained and tested using the KDDCup99 dataset, achieving an accuracy rate of up to 87%. This demonstrates its effectiveness in identifying and mitigating Spam threats [7][20]. Another model utilizes features extracted from email

Table 1: State of the art of Spam email detection

References	Classifiers Used	Accuracy
[21]	Random Forest	0.956
[22]	SVM + J48	0.937
[5]	Support Vector Machine	0.795
[4]	Random forest andDecision Tree	0.899
[15]	C5.0	0.957
[2]	Bayes Net	0.9711
[23]	Naïve Bayes ,SVM andLogistic Regression,	0.981

URLs. URLs undergo scrutiny and are compared to legitimate URL forms. If any discrepancies or suspicious elements are detected, the email is promptly routed to the spam folder, enhancing email security[7][8].

III. PROPOSED METHODOLOGY

The proposed method for email detection involves several stages, as depicted in Figure 1. Initially, the input dataset undergoes HTML parsing, extracting features crucial for subsequent analysis and comparison. The model is trained on both balanced and unbalanced datasets, enhancing feature recognition accuracy. Following feature extraction, a Convolutional Neural Network computes the features, facilitating the classification of emails into spam or non-spam categories. Analysis of the email header and body occurs at both character and word levels. If the email matches features indicative of spam, such as suspicious links or extensions, it is promptly identified as spam and directed to the spam folder. Conversely, if the email lacks these spam features, it is categorized as non-spam anddirected to the legitimate email folder. This dual-trained model ensures a higher probability of accurate feature recognition, enhancing overall result precision.

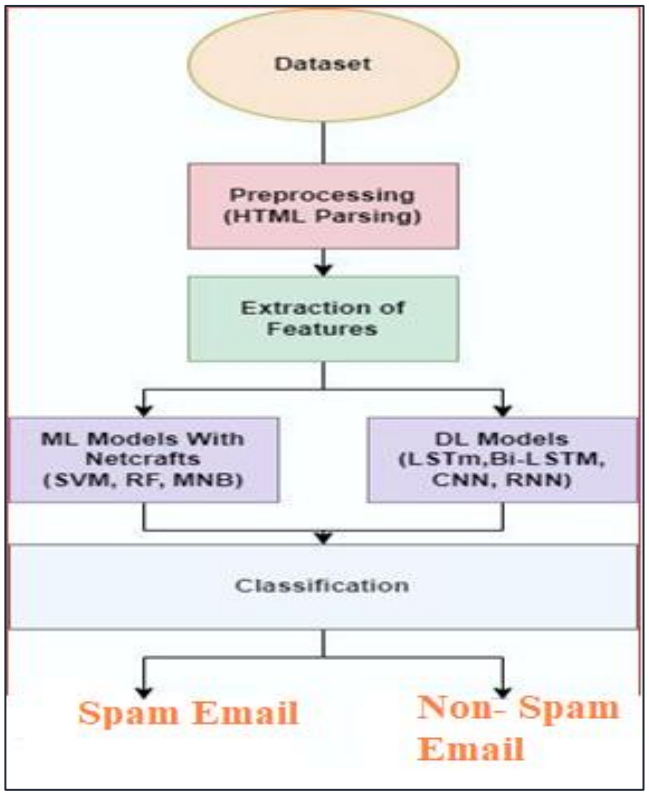

Figure 1: Model of Spam email detection

- ***Dataset***

The We have collected the dataset in which 5k is Spam emails and 5k non-Spam emails. We have downloaded the emails using following code snippet as shown in Fig 2 and Fig 3. We also have used the email extension for specifying the Spam emails as shown Figure. 4.

The description of both Spam and non- Spam emails are shown in Fig 5(a) and 5(b) respectively

```
result_status, items= Server.search(None,"UNSEEN")
items= items[0].split()
global id
if len(items)<1:
    print "[-] Whoo! No new Eamils"
    #close thw database connection
    con.close()
    #exit
elif len(items)>0:
    print"[+] we have new %d Emails "%len(items)

    print"[+] Start Phising analysis...."
```

Figure 2: New emails Downloading List

```
def flag(emasil_id):
    result_status,items = server.search(None,"SEEN")
    items = items[0].split()
    EmailToDelete = emasil_id
    for EmailToDelete:
        server.store("1:{0}".format(EmailToDelete),'+X-GM-LABELS','\\ spam')
        server.expunge()
        print server.expunge()
        return EmailToDelete
```

Figure 3: Non-legitimate emails moving into spam

```
extension_blacklist = [
    'ade', 'adp', 'bat', 'chn', 'cmd', 'com', 'cpl', 'exe', 'hta', 'ins', 'isp',
    'jse', 'lib', 'mde', 'msc', 'msp', 'mst', 'pif', 'scr', 'sct', 'shb', 'sys',
    'vb', 'vbe', 'vbs', 'vxd', 'wsc', 'wsf', 'wsh', 'htm', 'htm']
```

Figure 4: Malicious File Extensions

	label	title	content
0	phish	240_False.txt	Subject: Confirm Your NCUA Identity NCUA Home ...
1	phish	303_False.txt	Subject: Reactivate Your Account\n Amazon\ncom...
2	phish	130_False.txt	Subject: CONSUMER ALERT : Protect Yourself Aga...
3	phish	86_False.txt	Subject: PostBank OnlineBanking Postbank Onlin...
4	phish	404_False.txt	Subject: System maintenance: update your Feder...

Figure 5(a): Samples of Spam emails

label	title	content
ham	Summary.txt	Legitimate\n----------\n- Owner: kitchen-l\n- ...
ham	Summary.txt	Legitimate\n----------\n- Owner: williams-w3\n...
ham	Summary.txt	Legitimate\n----------\n- Owner: beck-s\n- Tot...
ham	Summary.txt	Legitimate\n----------\n- Owner: lokay-m\n- To...

Figure 5(b) Samples of non-Spam email (ham)

IV. RESULTS AND DISCUSSION

The testing set, derived from the dataset, integrates with the proposed model, incorporating precise pointers for evaluation. True Negative signifies authentic emails correctly classified, while False Positive represents genuine emails misclassified as illicit. False Negative indicates Spam emails misclassified as legal, and True Positive denotes non-legitimate emails misclassified as Spam. Comparing outcomes with the entire email, utilizing only the email body and header reveals discrepancies. The model relies on the email body due to its tendency to contain suspicious content, challenging baseline machine learning. Addressing this, an attention mechanism assigns superior weight to the email header, enhancing Spam email detection accuracy through dynamic weightage adjustments across various experiments. We have applied the different combination of model such as Random forest with Net craft whose result is shown in Table 2.

Table 2: Results of the Random Forest + Net craft

Fold	Precision	Recall	F1-Score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
Avg	1.00	1.00	1.00

These results indicate that the model achieved high precision, recall, and F1-scores for the "spam" class, demonstrating its strong performance in classifying Spam emails. Here's a combined Table 3 summarizing the results of the MNB (Multinomial Naive Bayes)
+ Net craft evaluation.

Table 3: Results of the MNB + Net craft

Fold	Precision	Recall	F1-Score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00

This Table 4 provides a comprehensive summary of the evaluation results for the SVM (Support Vector Machine) + Net craft model, including precision, recall, and F1-scores for both "spam" and "ham" classes across different folds, as well as overall metrics. The evaluation results of three different models combined with Netcraft for Spam email classification have been summarized in the provided tables. For the Random Forest model, the results demonstrated consistent high performance with aperfect F1-score of 1.0 for the "phish" class across all folds. The average accuracy and other metrics also indicated strong classification capabilities.

Table 4: results of the SVM (Support Vector Machine) + Netcraft

Fold	Precision (spam)	Recall (spam)	F1-Score (spam)	Precision (Ham)	Recall (Ham)	F1-Score (Ham)
0	1.00	1.00	1.00	-	-	-
1	1.00	1.00	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	-	-	-
3	1.00	1.00	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00	1.00	1.00
Avg	1.00	1.00	1.00	1.00	1.00	1.00

On the other hand, the Multinomial Naive Bayes (MNB) + Netcraft model exhibited perfect precision and recall for "spam" in all folds but performed poorly for the "ham" class, resulting in an average ham precision and recall of 0.0. This impacted overall metrics, such as F1 scores, which were lower due to the ham class's underperformance. The Support Vector Machine (SVM) + Netcraft model achieved perfect precision and recall for both "spam" and "ham" classes in all folds, leading to excellent overall metrics, with an average accuracy of 1.0.

The evaluation results of three different models combined with Netcraft for Spam email classification have been summarized in the provided tables. For the Random Forest model, the results demonstrated consistent high performance with a perfect F1-score of 1.0 for the "spam" class across all folds. The average accuracy and other metrics also indicated strong classification capabilities On the other hand, the Multinomial Naive Bayes (MNB) + Netcraft model exhibited perfect precision and recall for "spam" in all folds but performed poorly for the "ham" class, resulting in an average ham precision and recall of 0.0. This impacted overall metrics, such as F1 scores, which were lowerdue to the ham class's underperformance.

The Support Vector Machine (SVM) + Netcraft model achieved perfect precision and recall for both "spam" and "ham" classes in all folds, leading to excellent overall metrics, with an average accuracy of 1.0. These results highlight that the Random Forest and SVM models offer consistent high performance in Spam email classification, whereas the MNB model struggles to classify the "ham" class effectively. Users can reference the individual tables for more detailed information on each model's performance. We have also used the LSTM, Bi-LSTM, CNN and RNN model for the classification of the Spam emails in which we have used the hyper parameter tuning using GridSearch which gives the best parameters (vocab size = 10000, embedding_dim = 100, max_sequence_length = 200, epochs 10, batch Size =32).

```
Epoch 1/10
110/110 [==============================] - 35s 282ms/step - loss: 0.0721 - accu
Epoch 2/10
110/110 [==============================] - 31s 283ms/step - loss: 0.0070 - accu
Epoch 3/10
110/110 [==============================] - 43s 394ms/step - loss: 0.0071 - accu
Epoch 4/10
110/110 [==============================] - 38s 339ms/step - loss: 0.0071 - accu
Epoch 5/10
110/110 [==============================] - 31s 281ms/step - loss: 0.0070 - accu
Epoch 6/10
110/110 [==============================] - 36s 331ms/step - loss: 0.0070 - accu
Epoch 7/10
110/110 [==============================] - 52s 476ms/step - loss: 0.0070 - accu
Epoch 8/10
110/110 [==============================] - 32s 286ms/step - loss: 0.0070 - accu
Epoch 9/10
110/110 [==============================] - 33s 296ms/step - loss: 0.0071 - accu
Epoch 10/10
110/110 [==============================] - 31s 283ms/step - loss: 0.0071 - accu
47/47 [==============================] - 4s 62ms/step - loss: 0.0055 - accuracy
Test loss: 0.005537998862564564
Test accuracy: 0.9993342161178589
```

Figure 6: LSTM model result

The deep learning model, comprising two LSTM layers over an embedding layer, achieved exceptional performance on the test dataset. It exhibited a remarkably low test loss of 0.0055 and an impressive test accuracy of 99.93% as shown in Figure 6. This indicates that the model effectively learned to classify text data into binary categories, showcasing its strong predictive capabilities in text classification tasks. The CNN model, with an embedding layer followed by convolutional and pooling layers, achieved remarkable performance on the test data. It demonstrated an exceptionally low test loss of 0.00013 and a perfect test accuracy of 100% as shown in Figure 7. This underscores its effectiveness in classifying text data into two categories, making it a powerful model for text classification tasks.

```
Epoch 1/10
110/110 [==============================] - 9s 71ms/step - loss: 0.1109 - accuracy: 0.9874
Epoch 2/10
110/110 [==============================] - 6s 51ms/step - loss: 0.0051 - accuracy: 0.9991
Epoch 3/10
110/110 [==============================] - 7s 64ms/step - loss: 0.0042 - accuracy: 0.9991
Epoch 4/10
110/110 [==============================] - 6s 57ms/step - loss: 0.0041 - accuracy: 0.9991
Epoch 5/10
110/110 [==============================] - 8s 69ms/step - loss: 0.0033 - accuracy: 0.9991
Epoch 6/10
110/110 [==============================] - 7s 61ms/step - loss: 0.0036 - accuracy: 0.9991
Epoch 7/10
110/110 [==============================] - 6s 51ms/step - loss: 0.0022 - accuracy: 0.9991
Epoch 8/10
110/110 [==============================] - 8s 70ms/step - loss: 9.8121e-04 - accuracy: 0.9997
Epoch 9/10
110/110 [==============================] - 6s 52ms/step - loss: 3.1625e-04 - accuracy: 1.0000
Epoch 10/10
110/110 [==============================] - 8s 70ms/step - loss: 1.3425e-04 - accuracy: 1.0000
47/47 [==============================] - 1s 14ms/step - loss: 1.3329e-04 - accuracy: 1.0000
CNN Test loss: 0.00013328532804735005
CNN Test accuracy: 1.0
```

Figure 7: CNN model Results

```
Epoch 1/10
110/110 [==============================] - 73s 569ms/step - loss: 0.0694 - accuracy: 0.9957
Epoch 2/10
110/110 [==============================] - 63s 577ms/step - loss: 0.0073 - accuracy: 0.9991
Epoch 3/10
110/110 [==============================] - 61s 556ms/step - loss: 0.0074 - accuracy: 0.9991
Epoch 4/10
110/110 [==============================] - 61s 558ms/step - loss: 0.0046 - accuracy: 0.9994
Epoch 5/10
110/110 [==============================] - 61s 559ms/step - loss: 1.3949e-04 - accuracy: 1.0000
Epoch 6/10
110/110 [==============================] - 63s 576ms/step - loss: 1.1845e-05 - accuracy: 1.0000
Epoch 7/10
110/110 [==============================] - 62s 560ms/step - loss: 4.9559e-06 - accuracy: 1.0000
Epoch 8/10
110/110 [==============================] - 61s 554ms/step - loss: 3.2512e-06 - accuracy: 1.0000
Epoch 9/10
110/110 [==============================] - 61s 554ms/step - loss: 2.3513e-06 - accuracy: 1.0000
Epoch 10/10
110/110 [==============================] - 62s 559ms/step - loss: 1.7717e-06 - accuracy: 1.0000
47/47 [==============================] - 9s 155ms/step - loss: 1.2990e-06 - accuracy: 1.0000
BiLSTM Test loss: 1.2990226423426066e-06
BiLSTM Test accuracy: 1.0
```

Figure 8: Bi-LSTM Results

The Bidirectional LSTM (BiLSTM) model, incorporating an embedding layer and two BiLSTM layers, delivered outstanding performance on the test dataset. It achieved anexceptionally low test loss of approximately 1.3e-06 and a perfect test accuracy of 100% as shown in Figure 8. This signifies the model's proficiency in classifying text data into binary categories, establishing it as a robust choice for text classification tasks. The RNN model, featuring anembedding layer followed by two RNN layers, exhibited remarkable performance on the test dataset. It achieved a low test loss of 0.0056 and an impressive test accuracy of 99.93%. as shown in Figure 9. This highlights its capability to effectively classify text data into binary categories, showcasing its strong predictive abilities in text classification tasks.

```
Epoch 1/10
110/110 [==============================] - 17s 132ms/step - loss: 0.0424 - accur
Epoch 2/10
110/110 [==============================] - 15s 133ms/step - loss: 0.0074 - accur
Epoch 3/10
110/110 [==============================] - 15s 134ms/step - loss: 0.0074 - accur
Epoch 4/10
110/110 [==============================] - 15s 135ms/step - loss: 0.0073 - accur
Epoch 5/10
110/110 [==============================] - 15s 132ms/step - loss: 0.0072 - accur
Epoch 6/10
110/110 [==============================] - 15s 132ms/step - loss: 0.0071 - accur
Epoch 7/10
110/110 [==============================] - 15s 133ms/step - loss: 0.0073 - accur
Epoch 8/10
110/110 [==============================] - 15s 134ms/step - loss: 0.0073 - accur
Epoch 9/10
110/110 [==============================] - 15s 132ms/step - loss: 0.0072 - accur
Epoch 10/10
110/110 [==============================] - 14s 132ms/step - loss: 0.0075 - accur
47/47 [==============================] - 2s 27ms/step - loss: 0.0056 - accuracy:
Simple RNN Test loss: 0.005641018971800804
Simple RNN Test accuracy: 0.9993342161178589
```

Figure 9: RNN model result

V. CONCLUSION

The Spam email detection is a process in which the suspicious links, malicious files or image are detected in a particular email which is threat the user's confidential data as well as financial threat also. In the proposed method we have used baseline ML model and Deep learning models with the help of which model does the comparison on word level and character level of the particular email. The evaluation of Spam email classification models, including Random Forest, Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and deep learning models such as RNN, LSTM, CNN, and Bi-LSTM, has provided valuable insights into their performance. The Random Forest model consistently demonstrated high precision, recall, and F1-scores for the "spam" class across all folds. Its perfect F1-score of 1.0 indicates robust capabilities in accurately classifying Spam emails. The SVM + Netcraft model also exhibited outstanding performance, achieving perfect precision and recall for both "spam" and "ham" classes in all folds, resulting in exceptional overall metrics with an average accuracy of 1.0. In contrast, the MNB + Netcraft model, while achieving perfect precision and recall for "spam," struggled with the "ham" class, leading to lower overall metrics. The average precision and recall for "ham" were both 0.0, impacting the F1 scores negatively. The deep learning models, including RNN, LSTM, CNN, and Bi-LSTM, were highly effective in classifying Spam emails. The hyperparameter-tuned RNN model, with a test loss of 0.0056 and a test accuracy of 99.93%, demonstrated strong predictive abilities. The LSTM model exhibited an even lower test loss of 0.0055 and an impressive test accuracy of 99.93%. The CNN model showcased remarkable performance with a test loss of 0.00013 and a perfect test accuracy of 100%. The Bidirectional LSTM (BiLSTM) model achieved outstanding results with an incredibly low test loss of approximately 1.3e-06 and a perfect test accuracy of 100%. These deep learning models excelled in text classification tasks, effectively learning and classifying text data into binary categories. The results underscore their potential as powerful tools for Spam email detection.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest

REFERENCES

1. Herzberg and A. Gbara, "Trustbar: Protecting (even naive) web users from spoofing and Spam attacks," Comput. Sci. Dep. Bar Ilan …, no. January, pp. 1–28, 2004.
2. T. Meyer and B. Whateley, "SpamBayes: Effective open-source, Bayesian based, email classification system," Proc. First Conf. Email Anti-Spam, vol. 98, pp.1–8,2004,[Online].Available: ftp://ftp.research.microsoft.com/users/joshuago/conference /papers-2004/136.pdf
3. S. S. M. Motiur Rahman, T. Islam, and M. I. Jabiullah, "PhishStack: Evaluation of Stacked Generalization in Spam URLs Detection," Procedia Comput. Sci., vol. 167, no. 2019, pp. 2410–2418, 2020, doi: 10.1016/j.procs.2020.03.294.
4. A. A. Akinyelu and A. O. Adewumi, "Classification of Spam email using random forest machine learning technique," J. Appl. Math., vol. 2014, 2014, doi: 10.1155/2014/425731.
5. N. B. Harikrishnan, R. Vinayakumar, and K. P. Soman, "A machine learning approach towards Spam email detection CEN-Security@IWSPA 2018," CEUR Workshop Proc., vol. 2124, no. March 2020, pp. 21–28, 2018.
6. Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Spam Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism," IEEE Access, vol. 7, pp. 56329–56340, 2019, doi: 10.1109/ACCESS.2019.2913705.
7. P. Lawson, C. J. Pearson, A. Crowson, and C. B. Mayhorn, "Email Spam and signal detection: How persuasion principles and personality influence response patterns and accuracy," Appl. Ergon., vol. 86, no. March, p. 103084, 2020, doi: 10.1016/j.apergo.2020.103084.
8. E. Zhu, Y. Ju, Z. Chen, F. Liu, and X. Fang, "DTOF-ANN: An Artificial Neural Network Spam detection model based on Decision Tree and Optimal Features," Appl. Soft Comput. J., vol. 95, p. 106505, 2020, doi: 10.1016/j.asoc.2020.106505.
9. A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for Spam detection," J. King Saud Univ. - Comput. Inf. Sci., no. xxxx, 2020, doi: 10.1016/j.jksuci.2019.12.005.
10. N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, "Adopting automated whitelist approach for detecting Spam attacks," Comput. Secur., vol. 108, p. 102328, 2021, doi: 10.1016/j.cose.2021.102328.
11. K. Selvan and M. Vanitha, "Detection of Spam Web Pages Based on Features Vector and Prevention using Multi Layered Authentication," vol. 119, no. 15, pp. 565– 573, 2018.
12. A. Bergholz, G. Paaß, F. Reichartz, S. Strobel, and J. H. Chang, "Improved Spam detection using model-based features," 5th Conf. Email Anti-Spam, CEAS 2008, no. January 2008, 2008.
13. K. A. Molinaro and M. L. Bolton, "Evaluating the applicability of the double system lens model to the analysis of Spam email judgments," Comput. Secur., vol. 77, pp. 128–137, 2018, doi: 10.1016/j.cose.2018.03.012.
14. D. Ranganayakulu and C. C., "Detecting Malicious URLs in E-mail – An Implementation," AASRI Procedia, vol. 4, pp. 125–131, 2013, doi: 10.1016/j.aasri.2013.10.020.
15. O. Doctor, "Dynamic Evolving Neural Fuzzy Framework for Spam E-Mail Detection," no. March, pp. 3960– 3964, 2013.
16. F. A. Aloul, "The Need for Effective Information Security Awareness," J. Adv. Inf. Technol., vol. 3, no. 3, pp. 176– 183, 2012, doi: 10.4304/jait.3.3.176-183.
17. A. Vazhayil, N. B. Harikrishnan, R. Vinayakumar, and K. P. Soman, "PED-ML: Spam email detection using classical machine learning techniques CENSec@Amrita," CEUR Workshop Proc., vol. 2124, pp. 69–76, 2018.
18. S. Smadi, N. Aslam, and L. Zhang, "Detection of online Spam email using dynamic evolving neural network based on reinforcement learning," Decis. Support Syst., vol. 107, pp. 88–102, 2018, doi: 10.1016/j.dss.2018.01.001.
19. M. Volkamer, K. Renaud, B. Reinheimer, and A. Kunz, "User experiences of TORPEDO: TOoltip-poweRed Spam Email DetectiOn," Comput. Secur., vol. 71, pp. 100–113, 2017, doi: 10.1016/j.cose.2017.02.004.
20. L. Gallo, A. Maiello, A. Botta, and G. Ventre, "2 Years in the anti-Spam group of a large company," Comput. Secur., vol. 105, p. 102259, 2021, doi: 10.1016/j.cose.2021.102259.
21. E. G. Dada and S. B. Joseph, "Random Forests Machine Learning Technique for Email Spam Filtering," vol. 9, no. 1, pp. 29–36, 2018.
22. H. Mi, Z. Wang, and A. Ittycheriah, "Supervised attentions for neural machine translation," EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., no. 4, pp. 2283–2288, 2016, doi: 10.18653/v1/d16-1249.
23. J. Drew and T. Moore, "Automatic identification of replicated criminal websites using combined clustering," Proc. - IEEE Symp. Secur. Priv., vol. 2014-Janua, pp. 116– 123, 2014, doi: 10.1109/SPW.2014.26.
24. Bashir, T. Agbata B.C, E. Ogala, and W. Obeng-Denteh,

"The Fuzzy Experiment Approach for Detection and Prevention of Spam attacks in online Domain," East African Sch. J. Eng. Comput. Sci., vol. 3, no. 10, pp. 205– 215, 2020, doi: 10.36349/easjecs.2020.v03i10.001.

25. T. H. Nguyen and R. Grishman, "Modeling skip-grams for event detection with convolutional neural networks," EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., pp. 886–891, 2016, doi: 10.18653/v1/d16-1085.
26. M. Nguyen, T. Nguyen, and T. H. Nguyen, "A deep learning model with hierarchical LSTMs and supervised attention for anti-Spam," CEUR Workshop Proc., vol. 2124, no. Iwspa, pp. 29–38, 2018.
27. X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," Proc. 28th Int. Conf. Mach. Learn. ICML 2011, no. 1, pp. 513–520, 2011.
28. P. Sabeeha, MD; Karimullah, SK; Babu, "Detection of Malicious URLs by Correlating the Chains of Redirection in an Online Social Network (Twitter)," Int. J. Res. Stud. Comput. Sci. Eng., vol. 1, no. 3, pp. 33–38, 2014.

Lakhani, S., Yadav, A., & Singh, V. (2022, December). Detecting SQL Injection Attack using Natural Language Processing. In 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (pp. 1-5). IEEE.