Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 3s (2025)

Reliable Email Spoofing Detection using Enhanced Cybersecurity Approaches

[1*]Deepak Mane, [2]Geetanjali Sharma, [3]Sandip Shinde, [4]Abhijit Banubakode, [5]Sunil Sangve, [6]Maqsood Ahmed Ansari

[1,3,5]Vishwakarma Institute of Technology, Bibwewadi, Pune-411037, Maharashtra, India
[2]Pimpri Chinchwad College of Engineering, Pune- 411044, Maharashtra, India
[4]MET Institute of Computer Science, Mumbai- 400050, Maharashtra, India
[6]JSPM Narhe Technical Campus, Pune-411041, Maharashtra, India
[1*]dtmane@gmail.com, [2]geetu2k3@gmail.com, [3]sandeep.shinde@vit.edu, [4]abhijitsiu@gmail.com, [5]sunilsangve@gmail.com, [6]maqans@gmail.com
Correspondence author: dtmane@gmail.com

Article History:

Received: *30-09-2024*

Revised: *21-11-2024*

Accepted: *30-11-2024*

Abstract:

In today's modern era Email spoofing is a serious danger to email communication because it allows hackers to impersonate legitimate senders and change email headers like the email ID and IP address. And email headers are vital parts of any emails that carry important details about the sender, recipients, email delivery method, and email content. These headers are essential for email management and communication since they fulfill a variety of functions, such as email filtering, security, and traceability. So, if the non-legitimate user sends an email then mainly any of the important fields in headers is mismatched. So the proposed mechanism in this paper provides a reliable email spoofing detection at the server level, utilizing cutting-edge cyber security practices and artificial intelligence, in response to this expanding concern.

Keywords: Cybersecurity, Artificial Intelligence, Email Security, Malware, Flask, Backend, Frontend, Node Modules.

1. Introduction

In recent years, the rapid and effective transmission of information made possible by email communication has become a crucial component of modern life. The security and dependability of this communication method are, however, seriously hampered by the email spoofing. Through the use of email spoofing, non-legitimate users can create false sender identities, alter email headers, and eventually trick receivers. This research aims to create a reliable system for the server-level identification and filtration of faked emails in order to counteract this growing threat. These email headers include crucial details about the sender, recipients, route taken by the email, and content. This paper examines the significance of these headers and shows how crucial they are to email management and security. The proper mechanism has been proposed that improves email authentication, anomaly detection, and machine learning for the proactive detection of faked emails. This method makes use of cutting-edge cybersecurity tools and artificial intelligence. This strengthens confidence and reliability

in electronic communication by protecting the integrity of email communication and the larger field of email security.

This paper focussed on finding the spoofed email with the help of an proposed algorithm and helping users to be safe from the harmful emails. Proper Dataset and Machine Learning model has been created to detect the emails because mainly most of the spoofing has been occurring due to the change in email headers only. The proper spoof guardian application has been created so that it will be a guardian for the user and help them detect the spoofed email.

2. Literature survey

[1] In this paper the authors have focused on evaluating SPF and DMARC adoption in high-profile and defensively registered domains, exposing widespread configuration issues that facilitate email spoofing. The study found that 65.9% of top 500 domains use SPF, and 34.3% use DMARC. It introduces subdomain spoofing and offers a methodology for better management. Notably, a successful remediation effort through CSIRT notifications improved SPF records for vulnerable domains. However, the study's limited focus on specific domain types and the assumption that CSIRT notifications are universally effective present notable research limitations. Future studies should encompass a broader domain spectrum and consider diverse remediation strategies.

[2] This paper mainly focused on investigating the detection of email spoofing through email header extraction and classification using a random forest algorithm, emphasizing the importance of email communication and the growing threats of spoofing. The study introduces a Python script to perform these tasks, achieving a notable accuracy of 0.99 and minimizing the skipping of emails, while also showing minimal impact on system resources during execution. Previous research on email security, particularly the work by R. Padmavathi Iyer and Manoj Mistra in 2017, laid the groundwork for this study's approach. Despite its successes, the research is limited by its focus on Gmail as the email provider, potentially limiting its applicability across different platforms. Additionally, the study acknowledges class imbalance issues in email classification, suggesting potential improvements for future research to address these limitations.

[3] In this paper researchers have given a detailed overview of Email Spoofing and its impact. Email spoofing is a prevalent issue in email systems, involving the forging of email source addresses. While prior research has addressed this problem, this paper introduces an innovative approach using memory forensics to detect whether a client received or replied to spoofed emails. Memory forensics leverages the fact that all actions on a computer, including email communications, are recorded in physical memory, ensuring non-repudiation. The study's results show that the approach takes around 12 minutes for accurate detection, with efficient resource management. However, there is a need for optimization to reduce execution time for real-time applications. Additionally, the study does not address the detection of spoofed emails sent by malicious clients from spoofing websites, a limitation that could be explored in future research to enhance email security.

[4] In this research authors have given the deep idea about the escalating challenges of spoofed emails which is notably affecting business and e-commerce. They have proposed an innovative approach to enhance sender domain authentication. Despite the effectiveness of authentication methods like SPF, DKIM, and DMARC, their tendency to occasionally misclassify legitimate emails, such as forwarded

messages, remains a concern. Focusing on DMARC's reporting capabilities, the study introduces a method that employs X-means clustering and DMARC report data to detect legitimate IP addresses, aiming to mitigate false positives in sender domain authentication. However, this method is reliant on organizations or email providers generating DMARC reports, which could pose practical challenges if not universally implemented. The research provides insights into improving the accuracy of authentication but does not explore potential false negatives or resource requirements in detail, aspects that warrant consideration for comprehensive practical implementation.

[5] In this the researchers address the critical issue of email spoofing by employing memory forensics for email trace analysis, enabling the identification of spoofing-based email attacks. Notably, it streamlines the memory capture process by focusing solely on relevant processes, reducing memory dump size, and expediting detection. An innovative URL extractor mechanism, driven by machine learning and novel URL features, identifies live email processes, facilitating header field extraction. Authentication header fields like SPF, DKIM, DMARC, and ARC are closely scrutinized for received emails, while novel header fields and MX record verification are applied for replied emails. The incorporation of an email attack alert system notifies IT administrators of potential threats. The approach delivers rapid email detection, approximately 35 seconds, with minimal false positives and resource efficiency. Nonetheless, the method primarily caters to web-based email clients, and future work could explore compatibility with other client-side applications. Additionally, extending the scheme to mobile devices and evaluating its effectiveness on Android and Apple operating systems presents a promising avenue for further research.

[6] This paper confronts the issue of email sender spoofing, a common problem that undermines the reliability and security of email communication. The study explores the behavior of different email client applications when receiving sender-spoofed emails and introduces an investigative algorithm aimed at identifying email sender spoofing by extensively analyzing email header fields, particularly Received SPF, DKIM, DKIM-Signature, and DMARC. The algorithm scrutinizes these fields for valid values, distinguishing between legitimate and unauthorized emails. The creation of a spoofed and legitimate email dataset aids in algorithm development and evaluation, demonstrating its effectiveness in detecting address-spoofed emails. However, the study acknowledges the need for ongoing algorithm refinement to address evolving spoofing techniques, and it emphasizes the importance of real-world applicability, scalability, and compatibility with diverse email clients for future research and practical implementation.

[7] [8] This paper mainly focuses on addressing the critical issue of email date and time spoofing within email security, recognizing the absence of adequate proactive mechanisms to safeguard email systems. The research emphasizes the significance of digitally signed messages for authenticating sender identity as a primary countermeasure. Using a dataset of spoofed and legitimate emails, the study proposes an algorithm for email forensics, focusing on header information and date-time-related fields. By scrutinizing sent-date and received-date fields and detecting predefined differences, the algorithm identifies instances of email spoofing. While the study outlines future plans for automating header analysis and spoofing detection, it does not explore the potential complexities of real-world implementation. Additionally, the research is limited in scope to date and time spoofing, highlighting

the need for broader research into email security encompassing various aspects, including sender identity and content manipulation, for a comprehensive solution.

3. Methodology

This research has been conducted in phases. Here, initial phases involved accessing emails, extraction of email headers, analysis, building a machine learning model to detect spoofing and then building an application that would help users in interacting with the proposed system.

Here in the initial phase

1. Sending Spoofed Emails

Sending the spoofed emails in the initial phase helped us in creating the dataset. This was done by using Kali Linux, which provided us the necessary tools and resources to stimulate and analyze the email spoofing attempts effectively.

We created a Simple Mail Transfer Protocol (SMTP) server on Brevo. By employing the Kali Linux and the SMTP server we were able to conduct controlled email spoofing stimulations.

This helped us in getting the valuable insights and also in creation of our dataset as it was not readily available.

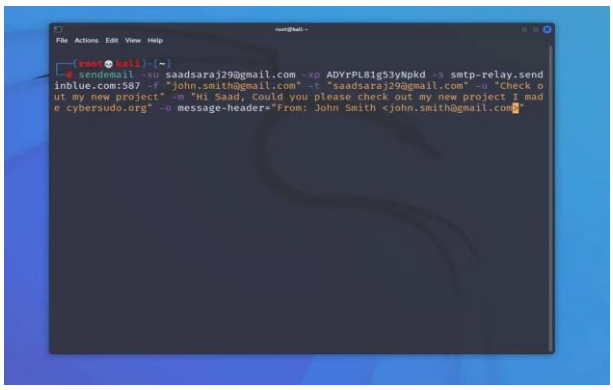

Fig 1. Kali Linux Terminal

2. Accessing the Email

To gain access to Gmail emails for the purpose of implementing server-level email spoofing detection, a secure and privacy-conscious approach is crucial. Gmail provides us with robust security features, including two-factor authentication (2FA), which enhances the protection of email accounts. In this methodology, we outline the process of accessing Gmail emails with 2FA.

But, similarly due to security purposes accessing emails without using Gmail application required additional permissions. To access emails, in this paper, we have to first enable 2FA. It is essential to set up 2FA for the Gmail account being used for this research. 2FA adds an additional layer of security. 2FA allows us to create app-passwords. App-passwords are created for specific apps, it provides user privacy by only giving access to our system for the email task.

Gmail accounts with 2FA enable third-party applications to access data. An App Password is a unique, application-specific password generated within the Gmail account settings. It is used instead of the account's regular password for secure access.

We have proposed using app-password instead of Gmail password as no user prefers to give Google account access to any third party application.

In this paper we have used IMAP server to access mails using the generated app-passwords. IMAP (Internet Message Access Protocol) is a widely used email protocol that allows email clients to access and manage emails stored on a remote email server. It allows us to access mail and then process it.

In this paper, by utilizing 2FA and generating App Passwords, we ensure that the access to Gmail emails remains highly secure and compliant with Gmail's security policies. The next phase in this paper focuses on intercepting and inspecting incoming emails in real-time to detect potential email spoofing attempts.

3. Extracting Email Headers

The second phase in this paper focuses on extracting the email headers. There are many important headers which are highly affected by any malicious activity.

We extracted headers - DKIM, DMARC, SPF, From, To, Return Path, Message-ID, DKIM Domain, DKIM Signature, Authentication Results, Subject and Date

DKIM - Email message authenticity is checked using the DKIM (DomainKeys Identified Mail) email authentication mechanism. Email headers are added a digital signature, which is then used to verify that the message was not tampered and came from the sender's domain that was indicated.

DMARC - DMARC stands for Domain-based Message Authentication, Reporting, and Conformance. It works with SPF (Sender Policy Framework) and DKIM (Domain Keys Identified Mail) as well as other email authentication techniques.

SPF - SPF stands for Sender Policy Framework. In email headers, SPF aids in the prevention of email spoofing and phishing by enabling the domain owner to select which mail servers are permitted to send email on their domain behalf.

SPF can assumed values None, Neutral, Pass, Fail, SoftFail, Temperror, Permerror

From - From contains sender email address, this fail is spoofed in email spoofing attacks.

To - To contain the receiver email address.

Return–Path: Return-Path contains the email address, where the email is sent back if there is any failure in email delivery.

Message-ID: Message-ID header serves as a distinctive identifier. The email client or email server generates it when the message is created and adds it to the email header as a means to specifically identify and refer to the message.

DKIM Signature -

DKIM (DomainKeys Identified Mail) signature in email headers is a cryptographic signature added to an outgoing email message by the sending email server. It is signed using the sender's private key and verified using the sender's public key, published in the DNS(Domain Name system) records.

DKIM Domain: DKIM domain refers to the domain specified in the DKIM signature

Subject - Subject contains the email Subject.

Date- Date contains the date and time when the email is sent.

Emails raw data was parsed and regular expressions were used in order to extract these headers, and email processing libraries and tools made specifically for retrieving and interpreting header data were also used. After, being extracted these email headers were analyzed to know discrepancies being introduced in spoofed emails as compared to legitimate mails

4. Email Header Analysis

The analysis and assessment of these headers was done in this phase. Its analysis allowed us to use the information and differences found in them to judge the validity of incoming emails and create a robust email spoofing detection system.

Here, we observed all the mails and found how header value changes as somebody tries to spoof it.

If the mails do not contain DKIM signature, or DKIM signature, it truly indicates that the message has been tampered midway.

After further analysis, we proposed an algorithm to detect email spoofing attacks.

```
if(dmarc==pass){
        return true
}
else if(dmarc==fail){
        return false
}
else if(return_path==null){
        return false
}
else if(DKIM!=null && DKIM.domainName==from.domainName){
        return true
}
else if(dkim_signature==null){
        return false
}
else if(spf!=pass && spf!=neutral && spf!=null){
        return false
}
else if(spf==pass && dkim==pass){
        return true
}
else if(dmarc=null && dkim=(null || pass) && (spf==neutral || null){
        if(DKIM.domainName==from.domainName){
                return true
        }
        else{
                return false
        }
}
else if(dmarc=null && dkim=null && spf=null){
        return false
}
```

Fig 2. Email Spoofing

As per fig 2, we observed that the following conditions hold when we try to detect email spoofing attacks.

We observed a spoof email,

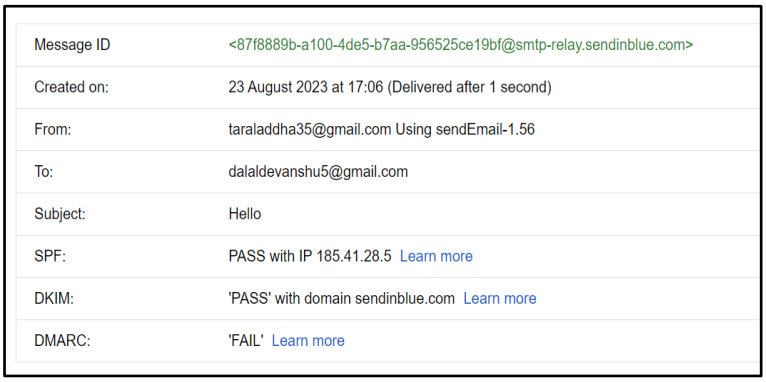

Fig 3. Spoofed Email

As in fig 3, we can observe that if any of the conditions fails, as we can see here in this image the DMARC fields have been failed, which indicates that the email is spoofed.

We have created a Kali Linux server that helps us to send spoofed emails.

5. Dataset Creation

After extracting the headers, we have created our own dataset to build an ML model. As today, there is no such dataset openly available.

We have used fields like DKIM, DMARC, SPF, Message-ID Domain, DKIM Domain, Spoofed status to create dataset.

After creation of the dataset, we have applied appropriate data cleaning steps to convert into processed data that could be used to train a ML model.

6. ML model Development

7. Developing UI:

Based on the above proposed architecture, we have built a system, which would ask the user to register himself.

Proposed system requires a user email address and email app password. Our proposed system asks the user for the above details while registering himself, with the help of the above details, our system fetches the emails and displays them on the screen as spoofed or legitimate. This system allows the user to protect himself from email spoofing.

4. Experiment Result and Discussion

We have done email header analysis. We observed that among all the headers in email, SPF, DKIM and DMARC are widely used in Email Spoofing Detection. It was observed that the presence of various online tools to send Spoofed emails, has made its detection highly important.

To understand it in more detail, we built our server on Kali Linux to send mails and experimented with it, to see how the header values change as we send Spoofed emails. It was observed that along with the above 3 mentioned headers(SPF, DKIM, DMARC), DKIM signature is also widely used as a tool to

detect any man in the middle attacks and to make sure that the email has not been tampered in its journey.

From	To	Return_Path	Message-ID	Message-ID-Domain	SPF-Result	DKIM	DMARC	DKIM-Domain	Authentication-Results	DKIM-Signature	Spoofed
Gmail Tea	Tara Laddi	Field not found.	<CAAxB4v9pCBhyC	mail.gmail.com	none	none	none	none	Field not found.	Field not found.	No
Gmail Tea	Tara Laddi	Field not found.	<CAAxB4v_REyf0y	mail.gmail.com	none	none	none	none	Field not found.	Field not found.	No
Gmail Tea	Tara Laddi	Field not found.	<CAAxB4v_fNwFF-	mail.gmail.com	none	none	none	none	Field not found.	Field not found.	No
"Google+ t	taraladdhc	<noreply-daa26fef@p	<Ciihm9_bx8oCFU	plus.google.com	none	none	none	plus.google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google Pla	taraladdhc	<3K9DtWA8KDFASwEi	<46439cfe47a5e4f	google.com	none	none	none	google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google Pla	taraladdhc	<3BIcNWQ8KDBcA1JF	<46439cfe47a5e4f	google.com	pass	pass	pass	google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google Pla	taraladdhc	<3OLZYWQ8KDNcG79	<46439cfe47a5e4f	google.com	pass	pass	pass	google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google <n	taraladdhc	<3GZ1nWQgTDMIvw-j	<tntrfL2CN8oLiiAr	notifications.google.com	pass	pass	pass	accounts.google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google Pla	taraladdhc	<3xWQgWQ8KDNwLC	<46439cfe47a5e4f	google.com	pass	pass	pass	google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google Pla	taraladdhc	<3r7_pWQ8KDKoXQg	<46439cfe47a5e4f	google.com	pass	pass	pass	google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google Pla	taraladdhc	<34V37WQ8KDGALCU	<46439cfe47a5e4f	google.com	pass	pass	pass	google.com	mx.google.com;	v=1; a=rsa-sha256;	No
Google Pla	taraladdhc	<3UWQFWg8KDAYtk2	<46439cfe47a5e4f	google.com	pass	pass	pass	google.com	mx.google.com;	v=1; a=rsa-sha256;	No

Fig 4. Dataset

The above Fig 4 is representing the dataset on which the training and testing has been done. This dataset includes both spoofed and non-spoofed emails with the key email headers columns.

After this analysis, we prepared our own dataset as seen in fig 4. We use our mails to get the header values and generate the dataset.

From the above analysis, we have proposed our algorithm as seen in fig 2. to detect email Spoofing attacks.

We then trained our model on the above dataset.

We tried various models: Neural Network(NN), SVM model, KNN classifier, LDA, Gaussian Naive Bayes and NN+SVM and Decision tree classifier.

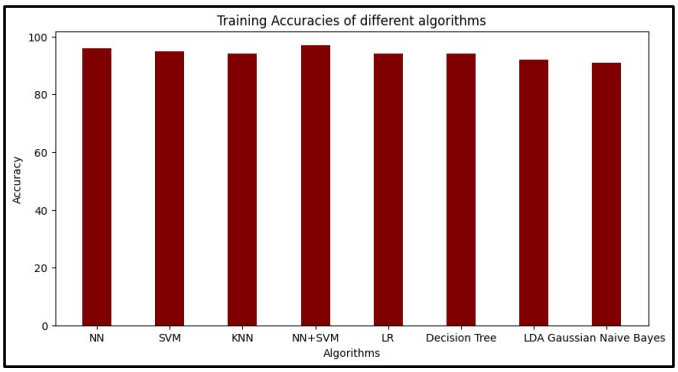

Fig 5. Training accuracies of different algorithms

The overall analysis of the training accuracies of different algorithms are mentioned in the form of a bar graph. Neural Network + SVM has the maximum training accuracy of 98%.

The accuracy of all the models can be seen in fig 5.

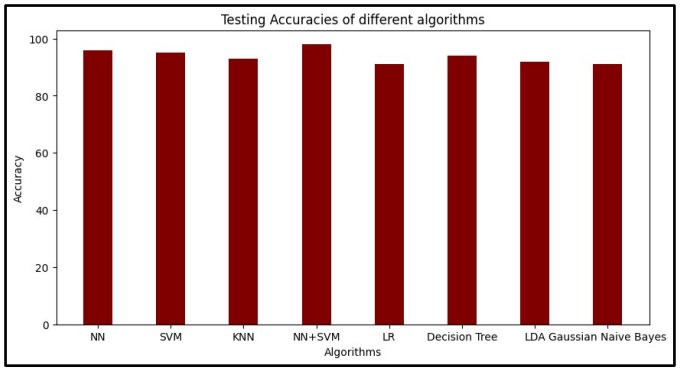

Fig 6. Testing accuracies of different algorithms

The overall analysis of the testing accuracies of different algorithms are mentioned in the form of a bar graph. Neural Network + SVM has the maximum testing accuracy.

As seen in fig 6, Neural Network+SVM has the highest testing accuracy 97%.

We have below plotted the training loss and validation loss of the Neural Network + SVM model.

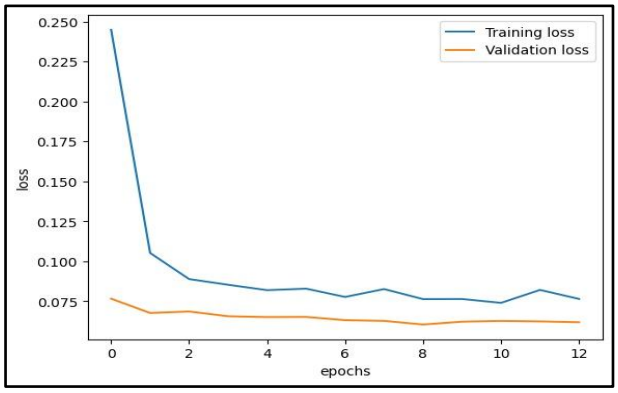

Fig 7. Training Loss and Validation Loss Chart

As seen in fig 7, we can see that loss decreases with increasing epochs, indicating the model is turning wheel with increasing number of epochs.

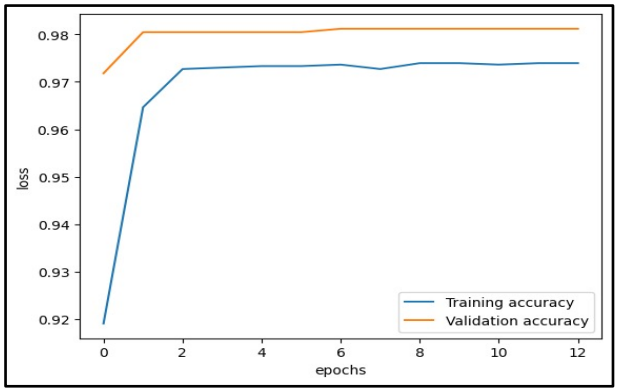

Fig 8. Training Accuracy and Validation Accuracy Chart

Here, in fig 8, we have plotted the line chart for training accuracy vs validation accuracy for Neural Network + SVM model.

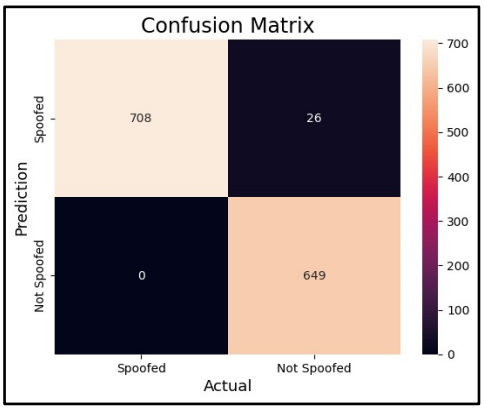

Fig 9. Confusion Matrix for Spoofed and not Spoofed

The above confusion matrix represents the analysis of the number of spoofed and non-spoofed emails and it gives the best way in analyzing how many emails are legitimate and how many are spoofed.

Here, in fig 9, we have plotted a confusion matrix for our Email Spoofed Detection.

As per the fig 8, we can see our recall comes out to be 1, which is really good as there are no cases of false negatives and our model is able to detect all spoofed emails.

We have prepared a web application using the MERN stack as seen in fig 10.

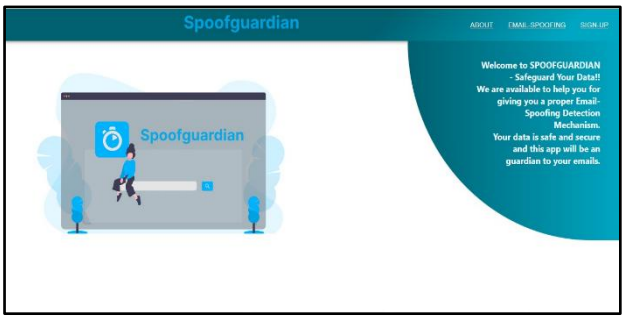

Fig 10. Home Page

The above image in Fig 10, shows the homepage of the Spoofguardian application which includes the details of the Email spoofing detection and its features.

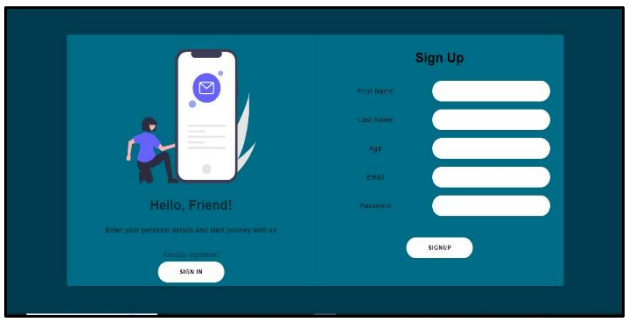

Fig 11. Login and Signup

For the authentication and allowing only legitimate users to access the platform the application has a login and signup feature as seen in fig 11.

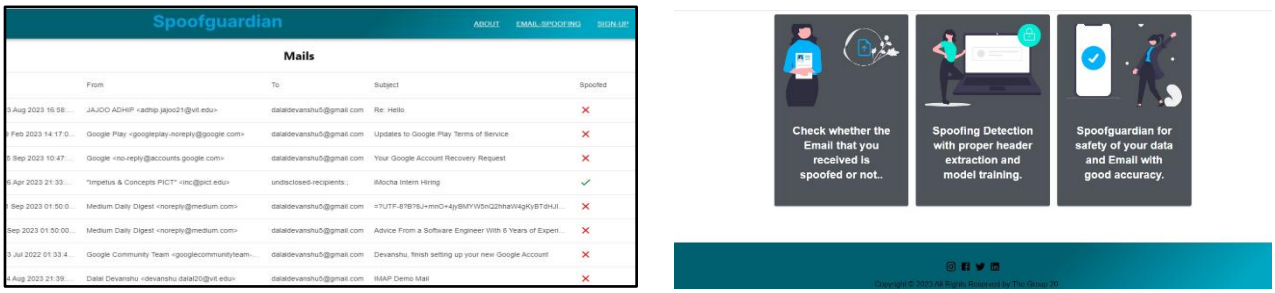

Fig 12. Spoofing Detection Section

The main section of the Spoofguardian application is represented here in fig 12, which shows the spoofed and non spoofed emails and helpful for the user to detect the harmful mails immediately.

We are storing the results of the ML model in the mongo DB database. This allows us to access the results of previously detected mails in a quick manner, as the IMAP server takes a lot of time.

```
_id: ObjectId('651d208b5822daca5315d6c0')
User: "dalaldevanshu5@gmail.com"
From: "Google <no-reply@accounts.google.com>"
To: "dalaldevanshu5@gmail.com"
Date: "Mon, 14 Aug 2023 15:48:12 GMT"
Subject: "Security alert"
Spoofed: 0
```

Fig 13. User DB Schema

As seen in fig 13, this is the User DB schema which includes the id, from, to, date, subject and spoofed and non-spoofed email boolean value. These are the detected mails data, where spoofed

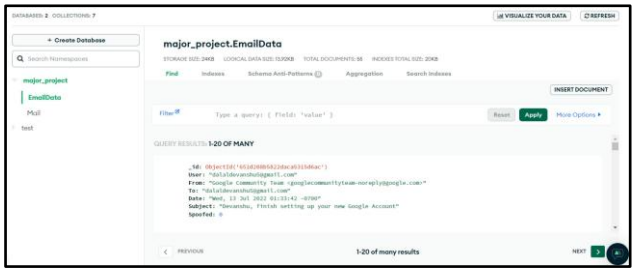

Fig 14. MongoDB Cluster

We have created a cluster as seen in fig 14, to connect to the database in which we are storing the user and mail details.

This developed application allows users to detect spoofed emails. User needs to create an account here with his email and app password(Generated as per the steps in methodology). We have then integrated the above ML model with the above web application, this helps us to fetch email in real time and show it to use with its spoofed status.

5. Conclusion

This paper's research represents an important development in the ongoing fight against email spoofing. The proposed results presents several potential to improve email communication security, safeguard users from cyberthreats, and hone the systems to adjust to the always changing environment of email-based attacks. And even future research in these fields will help email security in the digital age and

help users to protect the important emails. This study proposed a modern solution with the help of Cybersecurity and Artificial Intelligence as a technology.

6. Future Scope

In future the proper threat analysis system can be created and continuous monitoring and rapid response mechanisms can be developed to stay ahead of evolving spoofing techniques.

And will try to work on increasing the scalability of the technique to effectively manage high email volumes. Particularly in large organizations or email service providers, research can concentrate on optimizing the system to process a huge quantity of emails in real-time. So for that GPU's can be used to deal with the training and testing of large amounts of data.

Even the Blockchain technology can be used in investigating the use of decentralized identity management in conjunction with email authentication. This might result in a system for email validation that is more reliable and safe on a global scale

7. Limitations

1) The research was constrained as the existing dataset was not available, reasoning for the creation of a new dataset. A larger dataset could have resulted in more robust results.

2) The research needs access to the emails and for secure access to the emails it was mandatory to enable two-factor authentication (2FA) and generating app password. This method greatly improves email security and privacy, but it was difficult for participants who were hesitant to give access to their emails to the third party applications.

8. References

[1] S. Maroofi, M. Korczyński, A. Hölzel and A. Duda, "Adoption of Email Anti-Spoofing Schemes: A Large Scale Analysis," in *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3184-3196, Sept. 2021, doi: 10.1109/TNSM.2021.3065422

[2] Oluwaseun Odunibosi, "Classification of email headers using Random Forest Algorithm to detect Email Spoofing", 2019

[3] Iyer, R., Atrey, P.K., Varshney, G., & Misra, M.K. (2017). Email spoofing detection using volatile memory forensics. 2017 IEEE Conference on Communications and Network Security (CNS), 619-625.

[4] Kanako Konno, Naoya Kitagawa, and Nariyoshi Yamai. 2020. False Positive Detection in Sender Domain Authentication by DMARC Report Analysis. In Proceedings of the 3rd International Conference on Information Science and Systems (ICISS '20). Association for Computing Machinery, New York, NY, USA, 38–42. https://doi.org/10.1145/3388176.3388217

[5] Sanjeev Shukla, Manoj Misra, and Gaurav Varshney, "Forensic Analysis and Detection of Spoofing Based Email Attack Using Memory Forensics and Machine Learning" Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering Security and Privacy in Communication Networks, 2023, p. 491-509

[6] S. Gupta, E. S. Pilli, P. Mishra, S. Pundir and R. C. Joshi, "Forensic analysis of E-mail address spoofing," 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), Noida, India, 2014, pp. 898-904, doi: 10.1109/CONFLUENCE.2014.6949302.

[7] P. Mishra, E. S. Pilli and R. C. Joshi, "Forensic Analysis of E-mail Date and Time Spoofing," *2012 Third International Conference on Computer and Communication Technology*, Allahabad, India, 2012, pp. 309-314, doi: 10.1109/ICCCT.2012.69.

[8] M. Tariq Banday, Farooq A. Mir, Jameel A. Qadri, Nisar A. Shah, (2011) Analyzing Internet e-mail date-spoofing,Digital Investigation, Volume 7, Issues 3–4,2011, Pages 145-153, https://doi.org/10.1016/j.diin.2010.11.001.