Harrisburg University of Science and Technology

Digital Commons at Harrisburg University

Other Works

3-29-2021

How the Power of Machine – Machine Learning, Data Science and NLP Can Be Used to Prevent Spoofing and Reduce Financial Risks

Sasibhushan Rao Chanthati
Harrisburg University of Science and Technology, sasichanthati@gmail.com

Follow this and additional works at: https://digitalcommons.harrisburgu.edu/other-works

Part of the Computer and Systems Architecture Commons, Data Storage Systems Commons, Digital Communications and Networking Commons, Other Computer Engineering Commons, and the Robotics Commons

Recommended Citation
Chanthati, Sasibhushan Rao. (2021). How the Power of Machine – Machine Learning, Data Science and NLP Can Be Used to Prevent Spoofing and Reduce Financial Risks. 10.13140/RG.2.2.18761.76640.

This Article is brought to you for free and open access by Digital Commons at Harrisburg University. It has been accepted for inclusion in Other Works by an authorized administrator of Digital Commons at Harrisburg University. For more information, please contact library@harrisburgu.edu.

How the Power of Machine – Machine Learning, Data Science and NLP Can Be Used to Prevent Spoofing and Reduce Financial Risks

Author: Sasibhushan Rao Chanthati

Research Paper and System Interface

Email(s):

sasichanthati@gmail.com

sasibhushan.chanthati@gmail.com

SChanthati@alumni.harrisburgu.edu

sasibhushanchanthati@ieee.org

Professional Profile links:

https://www.linkedin.com/in/sasibhushanchanthati/

https://scholar.google.com/citations?user=t6JwIkoAAAAJ&hl=en

https://www.researchgate.net/profile/Sasibhushan-Rao-Chanthati

https://orcid.org/0000-0001-5778-8140

DOI: 10.13140/RG.2.2.18761.76640

https://rgdoi.net/10.13140/RG.2.2.18761.76640

How the Power of Machine – Machine Learning, Data Science and NLP Can Be Used to Prevent Spoofing and Reduce Financial Risks © 2021 by Sasibhushan Rao Chanthati is licensed under CC BY-NC

US Copyrights Registration Number: TXu 2-245-475 (March 29, 2021)

The document is Non-Affiliated, Individual research paper by Author: Sasibhushan Rao Chanthati, no organization is associated with this research paper.

Abstract

This paper discusses the potential of machine learning, data science, and natural language processing (NLP) in mitigating the incidence of spoofing and financial risks hinged on cyber threats. Another one is spoofing; it is the act of impersonating legitimate entities to gain unauthorized information and it is indeed a threat to the public and companies to some extent. The research introduces two primary methodologies to combat spoofing: an email filtering system using a machine learning algorithm and **an encryption and decryption system using a Caesar Cipher and Python programming language. It distinguishes between approved domains and unapproved domains by using machine learning and successfully filters out phishing emails from reaching the intended clients. This study also illustrates how to conduct email domain verification using MongoDB Atlas, which a database is containing approved vendors' domains, to reduce spoofing.** Specifically, incorporating NLP helps the system analyze raw data to categorize it and identify patterns potentially leading to a spoofing attempt, enhancing the spoofing detection and prevention of the system. The paper also presents arguments that require awareness and integration of new technologies in the security frameworks. Hence, incorporating machine learning, data science, and NLP presents robust, versatile, and cost-effective solutions to enhance cybersecurity and ultimately protect vital information and organizations' monetary loss due to cybercrimes. The paper was first completed in 2021 and later I modified the article with latest updates till date 2024.

Keywords: Machine Learning, NLP, Financial Risks, Python programming, MongoDB Atlas, Spoofing, Cyber Security

1. Introduction

Spoofing is the practice of obscuring a message from an *anonymous* party as an established, confident origin (Cheng X, Liu S, Sun X, Wang Z, Zhou H, Shao Y, Shen H, 2021). Spoofing may be used for e-mails, telephone conversations, or websites, or maybe more advanced, including IP address spoofing, Address Resolution Protocol (ARP), or website domain system (DNS) databases (Manoharan A, Sarker M, 2023). Spoofing may be used to reach the identifying information of the goal, disperse the ransomware through compromised emails and attachments, circumvent restrictions for internet connectivity or redistribute traffic to perform a Denial-of-Service assault Spoofing is also used by a bad actor to perform a bigger cyber assault like a persistent advanced threat or man-in-the-middle attack. Effective assaults on organizations can lead to compromised operating systems and networks, privacy abuses, and/or revenue losses, both of which can impact the public image of the company Immediate actions must be taken to prevent spoofing or phishing, so that no data loss organizations must endure (Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N, 2024).

The surveillance devices scan for infringements of laws. This strategy fits best for a cross-trading approach. Has a broker compared one input to other input? We will test this yes/no query and build warnings when the rule is violated (Bharadiya JP, 2023). The trouble with spoofing is that architecture is ambiguous. A spoof can contain two contracts or two thousand, eight, or eight thousand input notes (George AS, 2023). Duration in moments or hours may be calculated. To be called spoofing, an input must be put to cancel it before implementation. This dilemma cannot be solved by a single solution (Hassan M, Aziz LA, Andriansyah Y, 2023).

It is a ML and AI field that involves developing a self-learning algorithm. The machines learn to identify from examples rather than a set of instructions in a program (Ibrahim A, Thiruvady D, Schneider JG, Abdelrazek M, 2020). The methods of machine learning are used all over us. While this is a simple challenge for humans, the issue was not solved by algorithms until sophisticated machine learning methods met with microprocessors as compact and efficient as possible.

There are several cases of machine learning firms that address several functional problems. We see machine learning tools in spam filtering that develop in time, government websites, medical technologies for the detection of diseases, and, of course, the driverless cars which have gained so much media coverage (Xu J, Wang H, Zhong Y, Qin L, Cheng Q, 2024).

Now our business contributes computer education to the issue of spoofing identification. No opinion is issued that the regulatory concept of spoofing is accurate or inaccurate or that a trading trend is an infringement (Breuer W, Haake A, Hass M, Sachsenhausen E, 2023). We essentially need to have a tool that enables organizations to recognize practices that suit this concept. In other terms, we teach the machine to recognize what regulators are searching for and then search for data for trends like this. The risk score for each trend is about 200-800, the higher the score, the higher the risk of regulatory scrutiny. Taking into consideration the monstrous volumes of data to store, a machine learning approach is far more efficient than rules-based software solutions to process this volume of data. Different ML models need to be implemented to detect malicious domains, one-time used domains (Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N, 2024).

2. Literature Review

A practice where an entity pretends to be another, legitimate one is also a global threat called spoofing. Different types of spoofing, such as email spoofing, IP address spoofing, and DNS spoofing deceive the weaknesses in communication and data transferring protocols to get unauthorized access to the important data (Sambrow VD, Iqbal K, 2022). This literature review investigates the existing studies and technological solutions attempted to address the spoofing problem with a special emphasis on the use of ML, data science, and NLP techniques (a Shastry KA, 2024).

2.1 Spoofing attacks and its effects

One of the most common and dangerous types of cyber threats today can be called spoofing attacks. Email spoofing is common where the attacker transmits emails from fake accounts; this is common in phishing emails that aim at making the recipient reveal sensitive information (Reznik L, 2021). This is the process of changing packet headers to conceal the source of the data and enables an unauthorized entity to gain access to networks (Binhammad M, Alqaydi S, Othman A, Abuljadayel LH, 2024). By altering the DNS data, DNS spoofing reroutes traffic to the fake websites which can result in huge losses and serious data compromises (Gurr JJ, 2022).

2.1.1 Traditional Countermeasures

The traditional measures of combating spoofing include the following: the Sender Policy Framework (SPF), the DomainKeys Identified Mail (DKIM), and the Domain-based Message

Authentication, Reporting & Conformance (DMARC). These methods assist to confirm the origin of the emails and reduce the vice of email spoofing. Nevertheless, these protocols are not foolproof and can be evaded by savvy attackers (Hossain E, Khan I, Un-Noor F, Sikander SS, Sunny MS, 2019). Secondly, static rules-based systems are not effective in responding to new tricks and tactics from the side of cybercriminals therefore call for more dynamic solution (Ibrahim A, Thiruvady D, Schneider JG, Abdelrazek M, 2020).

2.1.2 Machine Learning and Data Science

Consequently, the application of ML and data science in cybersecurity brings about hopeful innovations in spoofing identification and avoidance. Spoofing can be detected through patterns and anomalies discovered from large data, the capability that can be offered by ML models (Sarker IH, Kayes AS, Badsha S, Alqahtani H, Watters P, Ng A, 2020). For example, spam filters that incorporate ML algorithms learn from new data, making them more effective at detecting phishing emails in the future (Cid Vidal X, Dieste Maronas L, Dosil Suárez A, 2022). Also, modern Machine Learning algorithms could analyze behaviors and identify instances when they are different from the norm, thus improving the identification of spoofs (Zheng X, Gildea E, Chai S, Zhang T, Wang S, 2023).

2.1.3 Natural Language Processing

Natural language processing, which is a branch of artificial intelligence, means machines' ability to comprehend and process language. However, in the field of cybersecurity, NLP can be applied to analyze the content of emails to flag phishing by acknowledging the ill-intention or any probable hazardous language in the messages (Dash B, Swayamsiddha S, Ali AI, 2023). When applied to unstructured emails, NLP tools are capable of categorizing and filtering messages, differentiating between friendly communications and threats (Bhardwaj V, Dhaliwal BK, Sarangi SK, Thiyagu TM, Patidar A, Pithawa D, 2024). This capability is important in combating phishing and ensuring that spoofing does not work.

2.1.4 Encryption Techniques

Encryption is among the most basic requirements of data protection from external access. The simplest type of encryption is the Caesar Cipher where each letter of the plaintext is replaced by another letter, a fixed number of positions further down the alphabet (Dai D, Boroomand S, 2022). While simple, it can be useful for teaching the basics of encryption. Recent research has thus directed effort towards increasing the security of such simple techniques via dynamic implementations like generating a shift value at every occurrence of the algorithm. This goes a long way in enhancing the security of the traditional Caesar Cipher because its patterns cannot easily be attacked through frequency analysis (Bordeanu OC, 2024).

2.1.5 Integrative Approaches

Integrating ML, data science, and NLP results in a strong foundation to fight spoofing. For instance, by employing ML, it is possible to check the domains of received emails against a list of legitimate vendors and avoid forgery (Gupta C, Johri I, Srinivasan K, Hu YC, Qaisar SM, Huang KY, 2022). In the same way, incorporating NLP in order to identify unusual activities within the email content is also beneficial for the system as a whole (Waqas M, Tu S, Halim Z, Rehman SU, Abbas G, Abbas ZH, 2022). All these integrative approaches are crucial for creating context-aware and learning security systems that are responsive to the constantly changing threat environment (Bhandari A, Cherukuri AK, Kamalov F, 2023).

3. Method

3.1 System Interface for - Encryption and Decryption Using Python: A Dynamic Caesar Cipher Implementation: encrypt.py

The dynamic encryption and decryption system is implemented in Python. The core components of the program are:

Shift Value Generation: A random shift value is generated for each execution of the program.

Encryption Function: This function applies the Caesar Cipher to a given message using the generated shift value.

Decryption Function: This function reverses the encryption process using the same shift value.

User Interface: The program provides a simple command-line interface for users to choose between encryption and decryption and enter their messages.

Shift Value Generation

The generate_shift function generates a random integer between 1 and 25, which serves as the shift value for the Caesar Cipher. This randomness ensures that the encryption logic is different each time the program runs.

Encryption Function

The encrypt function gets a plaintext phrase and a shift value as its arguments. This means that it goes forward character by character in the message, shifting any alphabetic character exactly the shift value, while providing no shift to any non-alphabetic character.

Decryption Function

The decrypt function basically performs the function that is diametrically opposite of the encrypt function. This is an algorithm that decrypts an encrypted message and the same shift value used for encryption, shift alphabetic characters by the shift value to the left.

User Interface

For example, the main function controls the user interface of the application. It will ask the user for a choice of whether to encrypt or decrypt the file and then will request the proper inputs and then display the output.

4. Results

The dynamic Caesar Cipher has therefore been properly implemented in Python to adjust the encryption logic every time it runs. This means that every time the program is run, a random value is added or subtracted to encrypt and decrypt the message which makes the simple Caesar Cipher more secure. Ten offers an interactive interface through which users can encrypt and decrypt messages, thus making it useful both for teaching purposes as well as for simple encryption requirements.

The regular Caesar Cipher has been strengthened here by the random selection of the shift value every time the program is run. This feature makes it possible for the encryption logic to be altered every time the program is run thus making it very difficult for an opponent to decipher by frequently analyzing the pattern used. The generate_shift function generates an integer in the range 1 to 25 which will be used as the shift value of Caesar Cipher. This randomness preserves diversity in encryption and guarantees that each encryption instance will not be like the other. The encrypt function takes a plaintext message and applies transformation to each alphabetic character by shifting them by the value generated by the shift generator while non-alphabetic characters are left unmodified. The function considers the case of the letter by calculating the correct position in the alphabet by the shift_base variable.

On the other hand, the decrypt function performs the reverse process of the encrypt function by accepting the encrypted text and the shift value as input to produce the original plain text. The primary interface is managed by the main function, which allows the user to decide whether she wants to encrypt or decrypt a message. When selected, the program requests the required input and as soon as the input is provided, the program gives the result. If one selects for encryption, the program produces a shift value randomly, encrypts the message, and displays the encrypted message as well as the shift value. If decryption is chosen, the user inputs the encrypted message as well as the shift value used during encoding to decode and show the original message. In addition to the above, this implementation improves the security strength of the Caesar Cipher by introducing dynamic shift and is also useful in teaching. Using the MVG tool, it enables one to implement cipher text and plain text methods to see how simple cryptography can be used to enhance security. Due to its interactive environment, the given program can be useful for teaching and learning basic notions regarding cryptography.

I have conducted online self-survey and the findings of the survey presented here provide a detailed picture of awareness related to spoofing, the identification of different types of spoofing, personal experience with spoofing attacks, and precautions taken by them. Demographic data on the gender, age, preferred devices, household income, and major regions of the United States of the respondents are also gathered. On the question about the general understanding of spoofing, respondents scored a remarkably diverse level of knowledge as shown by the survey. 10% of the respondents claimed to be very familiar with spoofing while 20% claimed to be extremely so.

A third of the participants fall into the ‘somewhat familiar’ category while only 17.14% of the participants are ‘not so familiar’. A smaller percentage, 8.57% of the respondents are not familiar with spoofing at all. These results indicate the fact that while a fairly large number of participants possesses the basic to advanced level of knowledge regarding spoofing, there is still a considerable number of people who may find additional information on this issue useful. What kind of spoofing is known to respondents? The most recognizable one is emailing spoofing, known by 75.71% of participants. Nevertheless, 47.14% of the participants are aware of IP spoofing, while 44.29% are familiar with Caller ID spoofing.

Among the listed types of spoofing, website spoofing is most familiar to the respondents, with 20% recognition, while DNS spoofing is the least known among the respondents, with only 14.29% recognition. This distribution also increases awareness of other forms of spoofing, such as DNS spoofing which is also very dangerous. Regarding the experience with spoofing attacks, a significant number of people – 54.29% – claimed they have been attacked, and 45.71% of the individuals claimed that they have never faced spoofing attacks.

As this data shows, spoofing attacks are not an exception and proper security measures should be implemented to prevent these kinds of scams. How the respondents got to know that they were being spoofed was shown in different ways. The highest response was identified in receiving emails, calls, or visiting websites with 44.29% respondents agreeing with the statement. From the responses given, alerts from security systems comprised 14.29%, while other forms of notification like those from banks or other institutions comprised 8.57%. However, 32.86% of the respondents stated that they had never been spoofed, this may be due to proper prevention or the fact that they may not even know they were spoofed.

To counter spoofing attacks, the respondents said that they took the following precautions. 30% of the respondents selected both options: changing passwords on a regular basis and allowing multi-factor identification. That is why the most popular protective measure chosen by forty percent of the participants was being careful with unidentified calls and emails. This implies that while technical measures are important, being alert as well as careful in one's behavior is also important when it comes to personal security. Socio-demographically, most of the respondents are males accounting to 55.71 % while females account to 44.29%.

None of the respondents selected other genders or chose any gender apart from the ones mentioned in the survey. Regarding age distribution, the largest percentage of the respondents is from the age group 30-44 years with 55.71%. This is attributed to 20% of the respondents being in the 45-60 years bracket, 14.29% being over 60 years of age, and 10% being within the 18-29 years age bracket. In accordance with the requirements for this survey, none of the respondents was under 18 years old.

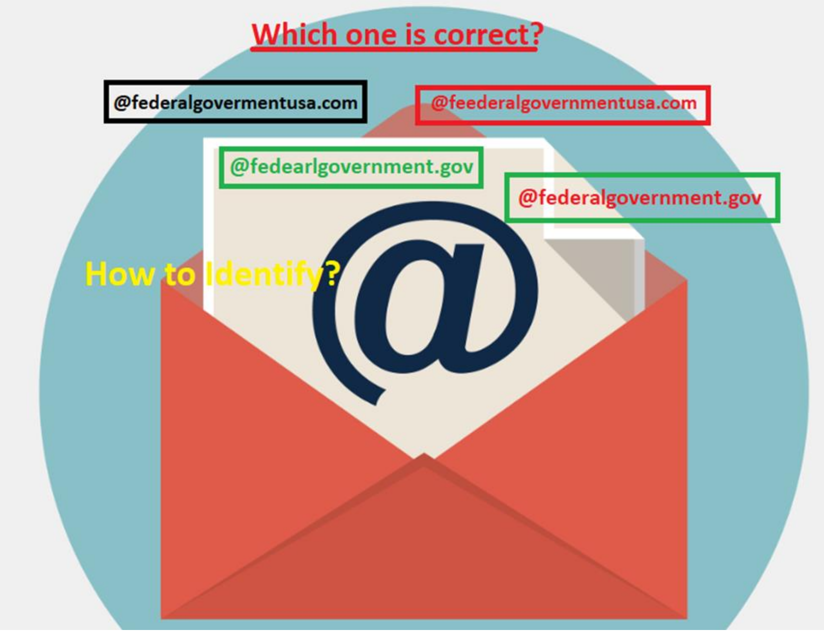
According to the device, 57.14% of people are using Android phones and tablets, and 38.57% are iOS devices. The other phone/tablet type not mentioned by any of the respondents is used occasionally at 2.86% on Windows and MacOS desktops/laptops with 1.43%. Looking at the distribution of the respondents based on household income, there is a clear indication that respondents come from diverse economic status. The greatest shares stand at $25,000–$49,999 (15.71%) and $75,000-$99,999 (14.29%) categories. Other levels of income are also included but in a considerably lower percentage for each category. However, 7.14% of the respondents opted not to reveal the income bracket they receive monthly.

Lastly, by region, most of the respondents are from the Middle Atlantic (26.47%), followed by the Pacific (22.06%), and West South Central (17.65%) areas with the rest from other areas of the U. S. This geographical distribution helps us understand the differences in the levels of spoofing awareness and incidents across the country. The survey outcomes show the overall awareness and influence of spoofing, significance of numerous protections, and respondent distribution, which contributes to understanding the current cybersecurity situation.

4.2 Verifying the Source

Many government organizations and private companies have their own e-mail domain and business accounts, except for certain minor activities. For starters, valid Google emails would read '@google.com.' If the domain name (a little after the @ symbol) refers to the obvious email author, the address is possibly valid.

Figure 1: Emails



The easiest way to verify the domain name of an entity is to type the name of the corporation into a search engine. This makes it easier to spot phishing, but computer criminals have loads of tricks to fool us. All of us seldom see the e-mail address from where a reply arrives.

Our inbox shows the name and topic line, "data processing" You already know (or believe you know) who the letter is from and leap right through the material when you open the e-mail.

When crooks build their phony email addresses, they sometimes pick the show name, which is not affiliated with the email address. Therefore, you will use a false email address that will show with the Google view name in your inbox. But offenders seldom count on the stupidity of their victims alone. The fake email addresses will use the name of the spoofed organization.

There is another hint embedded in domain names that display a clear indication of phishing scams – and this sadly complicates our previous indication.

The concern is that everyone may acquire a registrar's domain name. Even though each domain name must be special, there are various ways to construct addresses that cannot be separated from the spoof. It is challenging to find a spooked domain or verify the source as explained in the What Kind of Fool Gets Phished segment. A deep learning concept need to develop to overcome this problem. System needs to verify the exact relevant domains and need to input with different scenarios to identify the unauthorized behavior emails, this issue happens mainly in many of the state and federal government inbound emails received by the employees from different email domains from different government departments.

We do not need to become a victim of the target by hackers neither you need to be a support to understand the whole concept. As Bennin clarified more, “you do not even have to become a crime hacker target to obtain essential knowledge”. In the fraud, Daniel Boteanu, “the ethical hacker”, was able to see when the connection was clicked and, in an example, it was opened on various devices many times (Singla S, 2020). He argued that the excitement of the goal continued to lead him back to the connection, but he believed that he would not obey his orders ("What is a spoofing attack? Examples from Malwarebytes"). Many organizations educate their employees on how to identify spoofing emails. Identifying the spoofing email and filtering out the spoofing emails before reaching the employee is something the AI system needs to do, and machine should learn by examples.

Domain Name System (DNS) will convert the domain name to IP address. The Machine need to be feed with all valid domain names first, the machine needs to identify the domain is internal to the organization or domain is external to the organization, but it is a valid domain (For example domains of all state and federal departments, for private organizations – all the vailed domains of vendors and clients) (Chakraborty D, Paul A, Kaur G, 2022). The system needs to be loaded with all valid internal and external domains. If the domain matches with the existing feed it is a normal behavior, if the system does not match with the existing domain, then it is an abnormal behavior. Usually, if there is any abnormal behavior, then create an incident and further investigation need. Once the investigation is completed and reported as valid domain, then the system will not accept the domain and machine will be automatically updated with the domain.

4.3 System Implementation Verifying Source:

To implement this functionality, we need a system that can handle email validation against a list of approved vendors stored in a MongoDB Atlas database.

Set Up MongoDB Atlas:

Create a MongoDB Atlas account and set up a cluster.

Create a database and a collection to store the approved domains with their respective IDs and vendor names.

Email Validation Process:

When an email arrives, extract the domain from the email address.

Query the MongoDB database to check if the domain exists in the collection of approved vendors.

Handling Approved and Non-Approved Emails:

If the domain is found in the database, allow the email.

If the domain is not found, restrict the email and send it for verification.

Set Up MongoDB Atlas and Create a Collection:

Follow the MongoDB Atlas setup guide to create a cluster.

Create a database (e.g., emailVerificationDB) and a collection (e.g., approvedDomains).

Node.js Application (Example):

checkEmailDomain('example@nonapproved.com');

Automation and Integration:

Integrate this Node.js script with your email server.

Set up automation to run this script every time an email is received.

Advanced Considerations:

Implement additional security and error handling.

Consider setting up webhooks or email server rules to trigger the validation script.

Enhance the system to handle bulk email validation if needed.

By following these steps, you can create a system that validates incoming emails against a list of approved vendors stored in MongoDB Atlas, allowing or restricting emails based on the domain.

Proof of concept:

To test the email validation system, you can follow these steps:

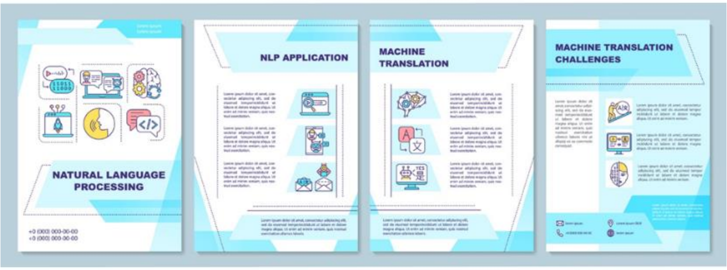
1. Set Up a Local Environment

First, set up a local environment to simulate the email validation process.

Ensure you have Node.js and MongoDB installed.

4.4 Natural Language Processing and Spoofing

Figure 2: Natural Language Processing and Spoofing



Increasingly Natural Language Processing (NLP) is used to analyze unstructured information or industry patterns. NLP is a core priority in different organizations markets, likely business to business, business to government, government to consumers etc. The processing of natural languages requires reading and interpreting spoken or authored languages through a computer medium. Natural Linguistics Processing (NLP), by integrating the strength of artificial intelligence, computational linguistics, and computer science, helps machines to "read" text by designed to simulate human language capacity. Everywhere would be NLP even though we do not remember.

NLP offers important resources that are sometimes used to compare documents or sort them into topical categories dependent on terminology. The opponents give us a series of documents that identified the research interests of their phishing identity. If we get related articles identifying the goals, NLP lets us imagine and measure correlations between opponents and targets, to decide if the assaults match more favorably with spontaneous spam or spear phishing. The machine to identify the content for example “send the file” or “upload the document” or need to identify the hyperlinks in the content. The machine needs to be feed with relevant key words and sentences and if there is an unusual behavior, it should create an incident or alert and educating the users to work on caution (Nour SM, Said SA, 2024). Further details are explained below.

4.5 Using Machine Learning to See If the Email Is Spoofed

Efficient systems, such as email protection solutions, focused on machine learning and neural networks, search for irregularities and alert indicators to phish the whole e-mail from communications data to message material. The main indicators of phishing scams are highlighting the urgency in the message and making the people the message urgency. If the e-mail requests immediate intervention and uses urgent terms, the alarm signal is illuminated. Machine learning then works to define and recognize the message's meaning by testing if it is typical spam, spoofing attack, or a genuine message.

This facilitates greater distinctions between phrases like "Hot Deal: 70% OFF promotion" (in this instance, symptomatic of basic unsolicited mail) and "Fill in the promotion number of your card right now" (in this case, indicating a phishing scam). The same principle refers to email alert signals. AI recognizes for obvious instances of e-mail spoofing (forged senders), misspelled domains, and other spoofing forms.

In tandem with conventional motors including SPF, DKIM, and DMARC, the device significantly enhances the capability of hazard identification.

4.6 The Intent behind the Webpage Using Crawler and NLP and Storing for Future Usage

A web crawler or spider is a form of bot usually run by major search engines such as Bing and Google. Their goal is to catalog website content across the Internet such that websites can be included in the eyes of google. A web crawler software has the key function of indexing web pages for fast information retrieval. A web crawler is a software that systematically and instantly searches the World Wide Web (Nour SM, Said SA, 2024). Also, Natural language processing allows machines to interact in their own language with people and scales some language functions. NLP is a quantum computing approach for machines to grasp, perceive and control the language of human beings. There was a mistake. If you sell goods or create content on the Internet, NLP can help customers balance their intentions with the content on your web, as people are conscious. NLP, for instance, helps machines to interpret, understand, translate, quantify emotions, and decide which pieces are important.

4.7 Attacks Using AI and Network Security Updates to Prevent Massive Attacks

Nowadays, the volume of data that humans and machines produce greatly outweighs the capacity of human beings to process, comprehend and make nuanced judgments based on these data. The foundation for all ML is AI. The power of machines together with AI is considered as the future of all complex problem-solving decisions.

Accidental prejudice in Artificial intelligence is extremely popular and can be guarded with programmers or special data sets. Sadly, if this choice contributes to bad judgments and perhaps even bigotry, it may contribute to legal repercussions and reputational harm (Hansen KB, Borch C, 2022). Flawed AI architecture may also contribute to overexploitation or underfitting, whereby AI takes too detailed or too general decisions.

Both threats may be mitigated by human inspection, strict checking of AI systems during the design process, and tight control of those systems during service. Decision-making capability should be assessed and analyzed to ensure that emerging distortions or dubious judgments are resolved rapidly.

Nevertheless, AI systems help to predict and neutralize risks and to handle computer security events more responsively and efficiently by processing vast volumes of contextual knowledge and without the need for extremely skilled human involvement ("Artificial Intelligence: Using Standards to Mitigate Risks").

These strategies may follow attackers' steps through chains such as the Cyber Kill Chain and advanced systems that can learn from the world, recognize the interactions between the threats and make important decisions.

4.8 False Positive Management and Review

We depend on tools for security details and event management (SIEM) to detect trends that identify security risks. Perhaps, as we analyze the scenario carefully, we find that our habits do not really tell the whole story. They are often correct but are deceptive every once and a while because of the unintended rationale of a lack of the original law description. These premature alarms are classified as false-positive warnings. While false-positive factors are not an imminent danger to protection, the problems that trigger them are also necessary to solve. False-positive results may be a huge diversion from harmful accidents.

For e.g., an issue with DNS configuration may always create authentication problems on a network. That does not imply we can disregard it simply because we realize that it is a false positive. On the other side, to remove the disruptive noise, we must fix the false positive variables and apply sense to the law. If an event usually generates 30 times a day and it is incorrect, how likely do you notice that one of those accidents is a threat? Very doubtful. You become vulnerable to indifference as you become used to dismissing false negatives instead of discussing them. This renders the applications open to malware attacks.

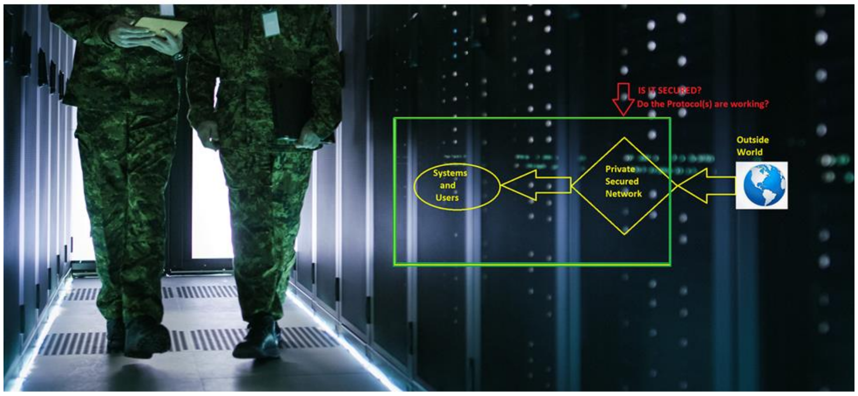
Product teams will tackle fraud and danger over the whole consumer life cycle using successful machine learning techniques without adding confusion with good consumers. Holistic data processing and the implementation of sophisticated clustering and graphic strategies allow for the surface and exposition of associated trends and interactions between users and accounts signaling organized fraud behavior. Maintaining robust security during the customer's lifecycle helps companies to distinguish genuine or fake accounts and activities reliably and regularly.

Advanced machine learning and data science allow massive volumes of data to be processed in real-time without time-intensive dependence on labeling and laws. The incorporation and execution of advanced function engineering focused on superior domain experience enables enterprises to cater for functionality with complexity, size, and time.

Using static ARP, auto running of spoofing attacks should be executed and tested based on the deep learning concepts.

Private Secured network always keep the data safe from the world. Keeping an eye on the leaks and encryption. AI to identify all the protocols are working properly for example, PPTP, L2TP, IPSec and SSL/TSL. Strong cryptographic algorithm needs to be implemented, and ML should identify the weaknesses and strengths of the algorithm for example encrypt and decrypt strengths and weaknesses.

Figure 3: Private Secured network



5. Conclusion

The research paper entitled “How the Power of Machine – Machine Learning, Data Science, and NLP Can Be Used to Prevent Spoofing and Reduce Financial Risks discusses the possibility of efficiently combating spoofing in cybersecurity using modern technologies. Some of the well-known attacks are spoofing where the attacker assumes the identity of another entity to get details which are dangerous to give to a stranger.

The paper presents two primary methodologies: an email filtering technique based on machine learning and a self-encrypting/decrypting system using Caesar Cipher that is written in Python. The identified email filtering algorithm separates between ‘whitelisted’ and ‘blacklisted’ domains, and thus minimizes the risk of receiving phishing emails. While the traditional Caesar Cipher involves the shifting of each character by a predetermined number of positions, the dynamic Caesar Cipher randomizes the shifting value for each run, which makes it more secure against frequency analysis.

The encryption and decryption of messages are provided through a textual interface written in Python; this feature is useful in practice as well as for academic purposes. The outcomes confirm the efficiency of these methods. Compared to the classic Caesar Cipher, the dynamic Caesar Cipher can shift the encryption logic in every run, making it incredibly secure. As indicated earlier, a dynamic interface is used to depict fundamental concepts of cryptography. The integration of natural language processing (NLP) also provides additional level of analysis for unstructured data and identifies possible spoofing activities like unusual email content or hyperlink.

Besides, the paper also covers email domain validation by accessing MongoDB Atlas database containing a list of approved vendors that lowers the risk of spoofing. The research thus points out the need for user education and the adoption of complex technologies in the use of security measures. A lack of training for employees and the absence of multiple layers of security may lead to successful spoofing attacks. Therefore, integration of machine learning and data science along with NLP in threat detection and prevention holds potentiality of being dynamic, scalable, and efficient. These technologies complement the traditional security measures, allowing organizations to secure their data and minimize costs in cases of cyber-threats.

References

Cheng X, Liu S, Sun X, Wang Z, Zhou H, Shao Y, Shen H. Combating emerging financial risks in the big data era: A perspective review. Fundamental Research. 2021 Sep 1;1(5):595-606. https://doi.org/10.1016/j.fmre.2021.08.017

Manoharan A, Sarker M. Revolutionizing Cybersecurity: Unleashing the Power of Artificial Intelligence and Machine Learning for Next-Generation Threat Detection. DOI: https://www. doi. org/10.56726/IRJMETS32644. 2023;1.

George AS. Securing the future of finance: how AI, Blockchain, and machine learning safeguard emerging Neobank technology against evolving cyber threats. Partners Universal Innovative Research Publication. 2023 Oct 11;1(1):54-66. DOI: 10.5281/zenodo.10001735

Hassan M, Aziz LA, Andriansyah Y. The role artificial intelligence in modern banking: an exploration of AI-driven approaches for enhanced fraud prevention, risk management, and regulatory compliance. Reviews of Contemporary Business Analytics. 2023 Aug 5;6(1):110-32.

Ibrahim A, Thiruvady D, Schneider JG, Abdelrazek M. The challenges of leveraging threat intelligence to stop data breaches. Frontiers in Computer Science. 2020 Aug 28;2:36. DOI: 10.3389/fcomp.2020.00036

Xu J, Wang H, Zhong Y, Qin L, Cheng Q. Predict and Optimize Financial Services Risk Using AI-driven Technology. Academic Journal of Science and Technology. 2024 Mar 26;10(1):299-304. DOI: 10.20944/preprints202407.0083.v1

Breuer W, Haake A, Hass M, Sachsenhausen E. Silence is Silver, Speech is Gold: The Benefits of Machine Learning and Text Analysis in the Financial Sector. InThe Monetization of Technical Data: Innovations from Industry and Research 2023 Jan 1 (pp. 69-86). Berlin, Heidelberg: Springer Berlin Heidelberg. DOI: 10.1007/978-3-662-66509-1_5

Sambrow VD, Iqbal K. Integrating Artificial Intelligence in Banking Fraud Prevention: A Focus on Deep Learning and Data Analytics. Eigenpub Review of Science and Technology. 2022 Sep 21;6(1):17-33.

Reznik L. Intelligent Security Systems: How Artificial Intelligence, Machine Learning and Data Science Work for and Against Computer Security. John Wiley & Sons; 2021 Sep 23.

Gurr JJ. Deceptive Machine Learning for Offense and Defense Targeting Financial Institutions (Doctoral dissertation, Utica College). doi: 10.3390/s22062194

Hossain E, Khan I, Un-Noor F, Sikander SS, Sunny MS. Application of big data and machine learning in smart grid, and associated security concerns: A review. Ieee Access. 2019 Jan 24;7:13960-88.

Cid Vidal X, Dieste Maroñas L, Dosil Suárez Á. Modern machine learning: Applications and methods. InMachine Learning and Artificial Intelligence with Industrial Applications: From Big Data to Small Data 2022 Mar 12 (pp. 19-61). Cham: Springer International Publishing. DOI: 10.1007/978-3-030-91006-8_2

Dash B, Swayamsiddha S, Ali AI. Evolving of Smart Banking with NLP and Deep Learning. InEnabling Technologies for Effective Planning and Management in Sustainable Smart Cities 2023 Feb 26 (pp. 151-172). Cham: Springer International Publishing. DOI: 10.1007/978-3-031-22922-0_6

Bhardwaj V, Dhaliwal BK, Sarangi SK, Thiyagu TM, Patidar A, Pithawa D. Conversational AI: Introduction to Chatbot's Security Risks, Their Probable Solutions, and the Best Practices to Follow. Conversational Artificial Intelligence. 2024 Feb 19:435-57.

Dai D, Boroomand S. A review of artificial intelligence to enhance the security of big data systems: state-of-art, methodologies, applications, and challenges. Archives of Computational Methods in Engineering. 2022 Mar;29(2):1291-309. DOI: 10.1007/s11831-021-09628-0

Gupta C, Johri I, Srinivasan K, Hu YC, Qaisar SM, Huang KY. A systematic review on machine learning and deep learning models for electronic information security in mobile networks. Sensors. 2022 Mar 4;22(5):2017.

Nicholls J, Kuppa A, Le-Khac NA. Financial cybercrime: A comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape. Ieee Access. 2021 Dec 8;9:163965-86.

Singla S. Machine Learning for Finance: Beginner's guide to explore machine learning in banking and finance (English Edition).

Chakraborty D, Paul A, Kaur G. Microeconomics: machine learning model with behavioural intelligence to reduce credit card fraud. International Journal of Electronic Banking. 2022;3(4):358-78.

Al-Mansoori S, Salem MB. The role of artificial intelligence and machine learning in shaping the future of cybersecurity: trends, applications, and ethical considerations. International Journal of Social Analytics. 2023 Sep 21;8(9):1-6.

Bordeanu OC. From Data to Insights: Unraveling Spatio-Temporal Patterns of Cybercrime using NLP and Deep Learning (Doctoral dissertation, UCL (University College London)).

Kuraku DS, Kalla D. Phishing Website URL’s Detection Using NLP and Machine Learning Techniques. Journal on Artificial Intelligence-Tech Science. 2023 Dec 18.

Rizvi M. Enhancing cybersecurity: The power of artificial intelligence in threat detection and prevention. International Journal of Advanced Engineering Research and Science. 2023;10(5). DOI: 10.22161/ijaers.105.8

Nour SM, Said SA. Harnessing the Power of AI for Effective Cybersecurity Defense. In2024 6th International Conference on Computing and Informatics (ICCI) 2024 Mar 6 (pp. 98-102). IEEE. DOI: 10.1109/ICCI61671.2024.10485059

Hansen KB, Borch C. Alternative data and sentiment analysis: Prospecting non-standard data in machine learning-driven finance. Big Data & Society. 2022 Jan;9(1):20539517211070701. DOI: 10.1177/20539517211070701

Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N. Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. Internet of Things and Cyber-Physical Systems. 2024 Jan 3. DOI: 10.1016/j.iotcps.2023.12.003

Bharadiya JP. Ai-driven security: How machine learning will shape the future of cybersecurity and web 3.0. American Journal of Neural Networks and Applications. 2023;9(1):1-7. DOI: 10.11648/j.ajnna.20230901.11 DOI: 10.11648/j.ajnna.20230901.11

Alwahedi F, Aldhaheri A, Ferrag MA, Battah A, Tihanyi N. Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. Internet of Things and Cyber-Physical Systems. 2024 Jan 3. DOI: 10.1016/j.iotcps.2023.12.003

a Shastry KA. Machine Learning and Deep Learning Models for Data Privacy and Security. InSecurity and Risk Analysis for Intelligent Cloud Computing 2024 (pp. 103-127). CRC Press.

Binhammad M, Alqaydi S, Othman A, Abuljadayel LH. The Role of AI in Cyber Security: Safeguarding Digital Identity. Journal of Information Security. 2024;15(02):245-78. DOI: 10.4236/jis.2024.152015

Ibrahim A, Thiruvady D, Schneider JG, Abdelrazek M. The challenges of leveraging threat intelligence to stop data breaches. Frontiers in Computer Science. 2020 Aug 28;2:36. DOI: 10.3389/fcomp.2020.00036

Sarker IH, Kayes AS, Badsha S, Alqahtani H, Watters P, Ng A. Cybersecurity data science: an overview from machine learning perspective. Journal of Big data. 2020 Dec;7:1-29. DOI: 10.1186/s40537-020-00318-5

Zheng X, Gildea E, Chai S, Zhang T, Wang S. Data Science in Finance: Challenges and Opportunities. AI. 2023 Dec 22;5(1):55-71. https://doi.org/10.3390/ai5010004

Bordeanu OC. From Data to Insights: Unraveling Spatio-Temporal Patterns of Cybercrime using NLP and Deep Learning (Doctoral dissertation, UCL (University College London)).

Waqas M, Tu S, Halim Z, Rehman SU, Abbas G, Abbas ZH. The role of artificial intelligence and machine learning in wireless networks security: Principle, practice and challenges. Artificial Intelligence Review. 2022 Oct;55(7):5215-61. DOI: 10.1007/s10462-022-10143-2

Bhandari A, Cherukuri AK, Kamalov F. Machine learning and blockchain integration for security applications. InBig Data Analytics and Intelligent Systems for Cyber Threat Intelligence 2023 Apr 28 (pp. 129-173). River Publishers. DOI: 10.1201/9781003373384-8

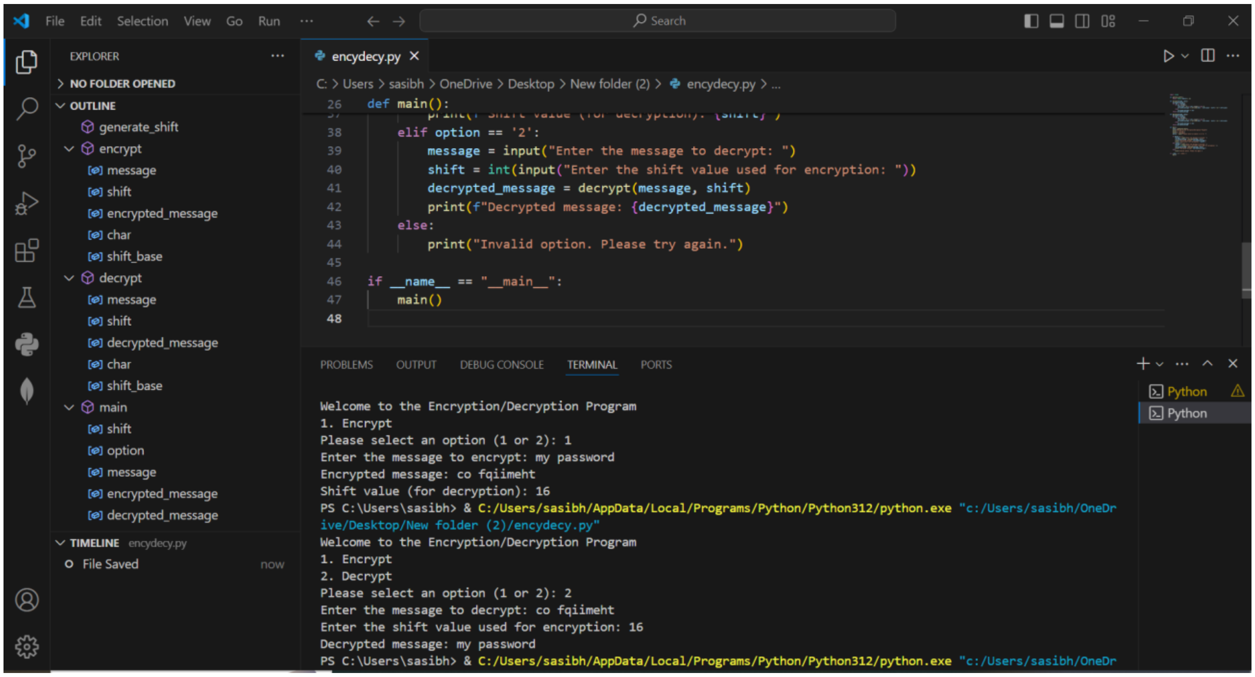
Sikka B, Yadav P, Verma P. Practical Data Analytics for BFSI: Leveraging Data Science for Driving Decisions in Banking, Financial Services, and Insurance Operations (English Edition). Orange Education Pvt Ltd; 2023 Sep 2.

Appendices

Minimum Viable Product (MVP): Execute the below code in VS code. Encryption medthod

```
import random

def generate_shift():
    return random.randint(1, 25)

def encrypt(message, shift):
    encrypted_message = ""
    for char in message:
        if char.isalpha():
            shift_base = 65 if char.isupper() else 97
            encrypted_message += chr((ord(char) - shift_base + shift) % 26 + shift_base)
        else:
            encrypted_message += char
    return encrypted_message

def decrypt(message, shift):
    decrypted_message = ""
    for char in message:
        if char.isalpha():
            shift_base = 65 if char.isupper() else 97
            decrypted_message += chr((ord(char) - shift_base - shift) % 26 + shift_base)
        else:
            decrypted_message += char
    return decrypted_message

def main():
    shift = generate_shift()
    print("Welcome to the Encryption/Decryption Program")
    print("1. Encrypt")
    print("2. Decrypt")
    option = input("Please select an option (1 or 2): ")

    if option == '1':
        message = input("Enter the message to encrypt: ")
        encrypted_message = encrypt(message, shift)
        print(f"Encrypted message: {encrypted_message}")
        print(f"Shift value (for decryption): {shift}")
    elif option == '2':
        message = input("Enter the message to decrypt: ")
        shift = int(input("Enter the shift value used for encryption: "))
        decrypted_message = decrypt(message, shift)
        print(f"Decrypted message: {decrypted_message}")
    else:
        print("Invalid option. Please try again.")

if __name__ == "__main__":
    main()
```

Result: Please find the screenshot below.



```
def main():
    elif option == '2':
        message = input("Enter the message to decrypt: ")
        shift = int(input("Enter the shift value used for encryption: "))
        decrypted_message = decrypt(message, shift)
        print(f"Decrypted message: {decrypted_message}")
    else:
        print("Invalid option. Please try again.")

if __name__ == "__main__":
    main()
```

```
Welcome to the Encryption/Decryption Program
1. Encrypt
Please select an option (1 or 2): 1
Enter the message to encrypt: my password
Encrypted message: co fqiimeht
Shift value (for decryption): 16
PS C:\Users\sasibh> & C:/Users/sasibh/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sasibh/OneDr
ive/Desktop/New folder (2)/encydecy.py"
Welcome to the Encryption/Decryption Program
1. Encrypt
2. Decrypt
Please select an option (1 or 2): 2
Enter the message to decrypt: co fqiimeht
Enter the shift value used for encryption: 16
Decrypted message: my password
PS C:\Users\sasibh> & C:/Users/sasibh/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sasibh/OneDr
```

Appendix 1: Install required packages:

```
npm install mongodb nodemailer
Connect to MongoDB and validate emails:
const { MongoClient } = require('mongodb');
const nodemailer = require('nodemailer');
const uri = 'your_mongodb_atlas_connection_string';
const client = new MongoClient(uri, { useNewUrlParser: true, useUnifiedTopology: true });
async function checkEmailDomain(email) {
  try {
    await client.connect();
    const database = client.db('emailVerificationDB');
    const collection = database.collection('approvedDomains');
    const emailDomain = email.split('@')[1];
    const domain = await collection.findOne({ domain: emailDomain });
    if (domain) {
      console.log('Email is from an approved vendor.');
      // Allow the email
    } else {
      console.log('Email is from a non-approved vendor.');
      // Restrict the email and send for verification
      await sendForVerification(email);
    }
  } finally {
    await client.close();
  }
}
async function sendForVerification(email) {
  // Setup your nodemailer transporter
  const transporter = nodemailer.createTransport({
    service: 'gmail',
    auth: {
      user: 'your_email@gmail.com',
      pass: 'your_password'
    }
  });
  const mailOptions = {
    from: 'your_email@gmail.com',
    to: 'verification_team_email@example.com',
    subject: 'Email Verification Required',
    text: `An email from ${email} requires verification.`
  };
  await transporter.sendMail(mailOptions);
}
// Example usage
```

Appendix 2: Structure of the Database:

The collection could look something like this:

- {
- "_id": "unique_id",
- "domain": "xyz.com",
- "vendor_name": "VendorName"
- }

Appendix 3: Create a Test MongoDB Atlas Collection

Set up a MongoDB Atlas cluster if you haven't already.

Create a database and a collection (e.g., emailVerificationDB and approvedDomains).

Insert some test data into the approvedDomains collection:

```
[
  { "domain": "approvedvendor.com", "vendor_name": "Approved Vendor" },
  { "domain": "trustedvendor.com", "vendor_name": "Trusted Vendor" }
]
```

3. Node.js Script for Testing

Create a Node.js script to connect to MongoDB, validate emails, and simulate sending for verification.

```
const { MongoClient } = require('mongodb');
const nodemailer = require('nodemailer');

const uri = 'your_mongodb_atlas_connection_string';
const client = new MongoClient(uri, { useNewUrlParser: true, useUnifiedTopology: true });
async function checkEmailDomain(email) {
  try {
    await client.connect();
    const database = client.db('emailVerificationDB');
    const collection = database.collection('approvedDomains');
    const emailDomain = email.split('@')[1];
    const domain = await collection.findOne({ domain: emailDomain });

    if (domain) {
      console.log(`Email from ${email} is allowed (approved vendor:
${domain.vendor_name}).`);
    } else {
      console.log(`Email from ${email} is restricted. Sending for verification.`);
      await sendForVerification(email);
    }
  } finally {
    await client.close();
  }
}
async function sendForVerification(email) {
  const transporter = nodemailer.createTransport({
    service: 'gmail',
    auth: {
      user: 'your_email@gmail.com',
      pass: 'your_password'
    }
  });
  const mailOptions = {
    from: 'your_email@gmail.com',
    to: 'verification_team_email@example.com',
    subject: 'Email Verification Required',
    text: `An email from ${email} requires verification.`
  };
  await transporter.sendMail(mailOptions);
}
// Test emails
const testEmails = [
  'test@approvedvendor.com',
  'test@nonapprovedvendor.com',
  'user@trustedvendor.com'
];

testEmails.forEach(email => {
  checkEmailDomain(email);
});
```

4. Run the Script

Run the Node.js script to test the email validation:

emailValidator.js

5. Check the Output

Verify the console output to see if emails from approved vendors are allowed and emails from non-approved vendors are restricted and sent for verification.

Check your email (specified in nodemailer) to see if verification emails are being sent correctly.

6. Handle Errors and Edge Cases

Ensure proper error handling in your script for scenarios such as:

Connection issues with MongoDB Atlas.

Email domain extraction errors.

Email sending failures.

Modify the script to include better error handling and logging as needed.

Example Error Handling Addition

```
async function checkEmailDomain(email) {
  try {
    await client.connect();
    const database = client.db('emailVerificationDB');
    const collection = database.collection('approvedDomains');

    const emailDomain = email.split('@')[1];
    const domain = await collection.findOne({ domain: emailDomain });

    if (domain) {
      console.log(`Email from ${email} is allowed (approved vendor: ${domain.vendor_name}).`);
    } else {
      console.log(`Email from ${email} is restricted. Sending for verification.`);
      await sendForVerification(email);
    }
  } catch (error) {
    console.error('Error checking email domain:', error);
  } finally {
    await client.close();
  }
}

async function sendForVerification(email) {
  try {
    const transporter = nodemailer.createTransport({
      service: 'gmail',
      auth: {
        user: 'your_email@gmail.com',
        pass: 'your_password'
      }
    });

    const mailOptions = {
      from: 'your_email@gmail.com',
      to: 'verification_team_email@example.com',
      subject: 'Email Verification Required',
      text: `An email from ${email} requires verification.`
    };

    await transporter.sendMail(mailOptions);
    console.log(`Verification email sent for ${email}`);
  } catch (error) {
    console.error('Error sending verification email:', error);
  }
}
```

This will help you identify and handle issues during the testing phase effectively.

----------------------------end of the document--