Phishing Link Detection Using Machine Learning

Shashank Naik[1], Chaitra Kedaliya[2], Naman S[3], Kartik[4], Sujay[5]
[1,3,4,5]*Student, Dept of Information Science and Engineering, Srinivas Institute of Technology, Mangalore, Karnataka, India.*
[2]*Assistant Professor, Dept of Computer Science & Design, Srinivas Institute of Technology, Mangalore, Karnataka, India.*
doi.org/10.64643/IJIRTV12I12-206838-459

***Abstract*—Phishing threats continue to pose a challenge and are among the greatest threats to internet users due to their exploitation by use of phishing URLs. The paper suggests an algorithmic way of detecting phishing links that takes into account the scalability of today's research approaches. The algorithm is based on heuristics and scalable learning technology which ensures real-time phishing detection at a minimum computational cost. The system employs lexical, structural, and domain-based features to effectively classify URLs. Phishing threats continue to pose a challenge and are among the greatest threats to internet users due to their exploitation by use of phishing URLs. The paper suggests an algorithmic way of detecting phishing links that takes into account the scalability of today's research approaches. The algorithm is based on heuristics and scalable learning technology which ensures real-time phishing detection at a minimum computational cost. The system employs lexical, structural, and domain-based features to effectively classify URLs**

I. INTRODUCTION

Phishing Phishing attacks are the cyber-crimes that people come across the most. These attacks usually happen on websites where people give out their personal account passwords and usernames without realizing it. For a time, there were two ways to detect phishing attacks. The first way is the blacklisting method and the other way is the signature detection method. However, these two methods are not good enough to detect phishing websites that may appear in the future. We need a way to detect these websites in real time by looking at their features and using special techniques to figure out what they are.

The internet is used more and more for things like talking to people doing transactions and sharing data. This has made it easier for people to do things. They can do them from anywhere. It has also given people who want to do bad things on the internet a chance to do so. One of the common and scary types of attacks, on the internet is still phishing attacks. Phishing attacks are a problem because people can lose their personal information and money if they are not careful. Phishing attacks are the problem that people have been dealing with for a long time.of attack.

II. LITERATURE REVIEW

In the past people used to find phishing attacks by looking at the features of a website and using things like Decision Trees and Random Forest to help them. They did this by checking the websites address like how long it was, what characters it used and what the domain was like. Now people are using ways to find phishing attacks like Deep Learning models such as CNN and RNN. These models are good because they can teach themselves from the websites address. They need more computer power to work.

The way people research phishing link detection has changed a lot. It used to be about rules but now it’s about using machine learning and deep learning.

- On people used models that looked at things like how long the websites address was, how old the domain was and if it had bad words in it.
- These models were not too hard on computers. They were not very good at finding phishing attacks.
- Now people are making learning models that can look at the websites address and find patterns without needing someone to tell them what to look for.
- Models like CNNs and RNNs are very good at finding phishing attacks by looking at patterns in

the websites address.

- Some people use a mix of new ways to make sure their models are accurate and do not use too much computer power.
- There are still problems like not having enough data phishing attacks changing all the time and needing a lot of computer power.

The goal of this research is to make something that's easy to use and accurate. Most of the time people do not think about how easy it's to use something when they are making it. This research is trying to make a model that's easy to use and still very good, at finding phishing attacks. The phishing attack detection model should be user friendly.

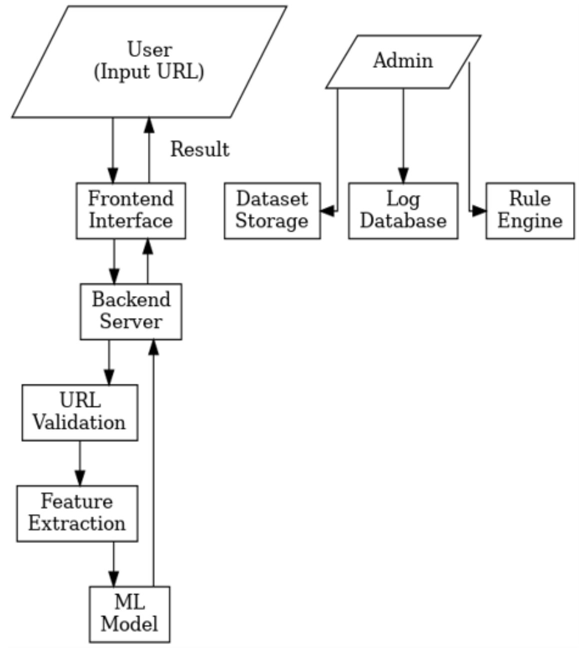
III. SYSTEM DESIGN

The phishing detection system is expected to give a smooth interface to interact between the user and the backend processing unit. In other words, it involves three parts: the frontend part, backend, and machine learning algorithms.

The first interaction comes from the user who enters a website address (URL). After this action, the backend starts its work by validating the data. After successful validation, the system proceeds to extract features, including domain, URL characteristics, and other security-related data. Next, these features are provided to the machine learning algorithm to classify the URL into the two categories, malicious or harmless. Moreover, there are additional modules involved, such as updating the dataset, model training, monitoring.

The system follows a client-server architecture. Users enter URLs via a frontend part of the software. Then, the backend processes requests according to the detection pipeline. The pipeline involves validation, feature extraction, and classification of the URL.

The feature extraction process involves lexical features, domain characteristics, and structural characteristics of the URL. The classification process implements heuristic rules to improve efficiency and reduce latency. Also, the pipeline architecture enables us to integrate different machine learning algorithms in the future.



IV. IMPLEMENTATION

The system uses a framework to handle things behind the scenes. This framework does its job. The system looks at data and sorts it out on the server. This makes it simple to add users and keep the system running. The system is getting things. These include connecting to things. The system uses things like HTML and CSS and JavaScript to make the interface work properly. The system uses Python to handle things behind the scenes. It also uses Flask. This helps with the website address. The system talks to machine learning models. The system stores data and logs in a MySQL database. When a user asks for something, the system gets the request. It looks at the data. Then it tells the user if the website address is bad or good with the help of a model. The system sends the answer back to the user. This way the system can change easily. It can also connect to things like applications and browser extensions.

V. MODEL TRAINING

The Software testing is the process used to help identify the correctness, completeness, security, and quality of developed computer software. This includes the process of executing the program or application with the intent of finding errors. Quality is not an absolut[1]e; it is value to some pe[2]rson. Testing forms the first step in determining the errors in a program. Clearly, the success of testing in revealing errors in programs depends critically on the test cases their accuracy. The model that performs better is chosen for deployment in the system.

VI. RESULT AND DISCUSSION

Different types of URLs were used for testing purposes. This allowed us to assess how effective the tool works in various situations. Results prove that the system successfully detects malicious URL patterns and makes accurate classifications. The frontend reacts promptly while the backend processes the information quickly enough. Limitations of the current algorithm include false negatives and positives because of the heuristic rules used. Testing is a critical part of the project that helps to verify the performance. There are several kinds of tests applied to make sure everything functions correctly. Unit tests check every single component separately while integration tests verify the communication with the frontend. The system testing allows assessing all processes. Several test cases were created to assess how the system responds to different URLs. Test data includes legitimate, malicious, and invalid URL addresses. The tests proved that the system works effectively in classifying malicious links and delivering proper outputs.

The first interaction comes from the user who enters a website address (URL). After this action, the backend starts its work by validating the data. After successful validation, the system proceeds to extract features, including domain, URL characteristics, and other security-related data. Next, these features are provided to the machine learning algorithm to classify the URL into the two categories, malicious or harmless

VII. USER STUDY

In order to see how efficient the system works in practice, Microsoft Google Docs, Google Sheets, and Google Slides are a word processor, a spreadsheet and a presentation program respectively, all part of a free, web-based software office suite offered by Google within its Google Drive service. The three apps are available as web applications, mobile apps for Android and iOS, and desktop applications on Google's Chrome OS. The apps are compatible with Microsoft Office file formats. The suite also includes Google Forms (survey software), Google Drawings (diagramming software), Google Sites (web building software), Google My Maps (map overlay editor), Google Apps Script (code editor for the G-Apps Script coding language) and Google Fusion Tables (database manager; experimental).

The suite allows users to create and edit files online while collaborating with other users in real-time. Edits are tracked by user with a revision history presenting changes. An editor's position is highlighted with an editor-specific color and cursor. A permissions system regulates what users can do. Updates have introduced features using machine learning, including "Explore", offering search results based on the contents of a document, answers based on natural language questions in a spreadsheet, and dynamic design suggestions based on contents of a slideshow, and "Action items", allowing users to assign tasks to other users.

VIII. CONCLUSION

This paper is about a system that can find phishing links. The system is good at finding these links. Can do it while people are actually using the internet. The people who made this system want to add things to it like better machine learning models and ways to make it work faster. This paper talks about a system that uses machine learning to find phishing links. This system can find links, on websites. The people who made this system want to add things to it like better deep learning models and update the lists of bad links every day. They also want to make it into something that people can add to their internet browser.

ACKNOWLEDGMENT

We are thankful to Prof. Athmaranjan K for the guidance he provided us throughout our research work. We would like to express our gratitude to Prof. Sudarshan K and Dr. Shrinivasa Mayya D for their continuous help and guidance throughout this project.

REFERENCES

[1] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," *Telematics and*

Informatics, vol. 38, pp. 161–173, May 2019.

[2] R. Verma and K. Dyer, "On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers," Proc. 5th ACM Conf. on Data and Application Security and Privacy (CODASPY), pp. 111–122, 2015.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[4] A. K. Jain and B. B. Gupta, "A Machine Learning Based Approach for Phishing Detection Using Hyperlinks Information," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 5, pp. 2015–2028, 2019.

[5] PhishTank, "PhishTank: Join the fight against phishing," OpenDNS/Cisco, Available: https://phishtank.org.

[6] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent Phishing Website Detection System Using Fuzzy Techniques," Proc. 3rd Int. Conf. on Information and Communication Technologies: From Theory to Applications (ICTTA), pp. 1–6, 2008.

[7] Machine Learning in Cybersecurity, Springer, Available: https://link.springer.com/journal/42400.

[8] IEEE Security & Privacy, IEEE Computer Society, Available: https://www.computer.org/csdl/magazine/sp.