International Journal of Applied Engineering Research ISSN 0973-4562 Volume 19, Number 1 (2024) pp. 76-81
© Research India Publications. https://dx.doi.org/10.37622/IJAER/19.1.2024.76-81

Detection of Potentially Hazardous Websites Utilizing Hidden Markov Models

Dr.S.R.Menaka [1], K.Divya[2], P.Priyanka[3], S.Yuvasree[4]

[1] Assistant professor, Department of Information Technology, K.S.R. College of Engineering, Tiruchengode, Tamilnadu, India

[2][3][4] Student, B.Tech-Information Technology, K.S.R. College of Engineering, Tiruchengode, Tamilnadu, India

ABSTRACT

Malicious URLs pose a significant cybersecurity threat due to spread malicious software and obtain personal data, and initiate phishing attempts. Traditional techniques for identifying fraudulent URLs include blacklists and heuristics, are losing their effectiveness as new evasion strategies are created by attackers. A study titled "Malicious URL Detection based on Machine Learning" introduces a novel approach using Hidden Markov Models (HMM) to quantify and predict the behavior of Malicious Web Services. This approach not only measures but also predicts the response time of these services, allowing for a quantitative ranking rather than a qualitative assessment. The proposed methodology aims to automatically select the most reliable Malicious Web Service by considering metrics like system predictability and response time variability. The experimental data shows that the new model performs much better than the baseline. Comparing the recommended model to the state-of-the-art, the results show that it is more stable and has accuracy and recall rates of up to 99.2% and 99.16%, respectively.

Keyword: Machine Learning, Malicious URL Detection, Adversarial Attacks, Malicious Web Services

INTRODUCTION

MACHINE LEARNING

Artificial intelligence (AI) has a branch called machine learning that makes computers capable of learning without actual programming. Through training on data, computers can identify patterns and make predictions. Machine learning algorithms find applications in various domains, including spam filtering, fraud detection, product recommendation, and image recognition. The three categories of these algorithms are reinforcement learning, unsupervised learning, and supervised learning. Machine learning proves to be a powerful tool for solving diverse problems. Nonetheless, it is critical to recognize that the completeness and quality of the training data has a significant impact on how well machine learning algorithms perform. Biased or incomplete training data can lead to biased or inaccurate predictions.

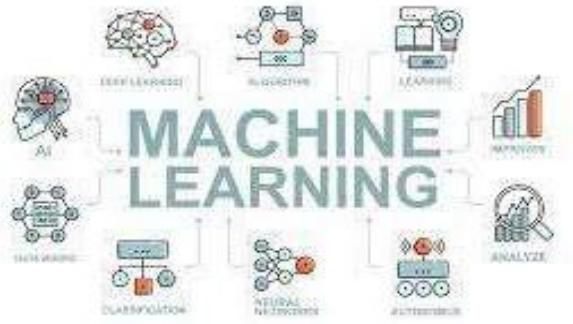

Figure 1. Machine learning

MALICIOUS URL DETECTION

Malicious URL detection involves the identification of URLs that direct users to malicious websites. These websites can distribute malware, steal personal information, or launch phishing attacks. Traditional methods of malicious URL detection, such as blacklists and heuristics, are progressively losing their effectiveness as attackers develop new evasion techniques. Machine learning presents a promising approach to address this issue. By training machine learning algorithms, patterns in malicious URLs that are difficult for humans to detect can be identified. These patterns may include the utilization of specific keywords or domains, the presence of suspicious characters in the URL, and the reputation of the hosting website.

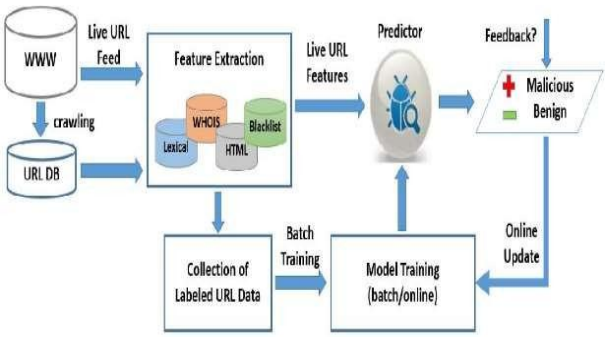

Figure 2. Malicious URL Detection

ADVERSARIAL ATTACKS

Adversarial attacks involve carefully crafted inputs that aim to deceive machine learning algorithms, causing them to make errors. These attacks can target various types of machine learning algorithms, including those used for classifying images and detecting objects. Adversarial attacks are typically created by making subtle changes to the input data, which may be imperceptible to humans. For example, adding a small amount of noise to an image could be an adversarial attack on an algorithm for classifying images. Although this noise may go unnoticed by humans, it can be sufficient to trick the algorithm into misclassifying the image. Adversarial attacks pose a significant threat to the security of machine learning systems. If successfully executed, an attacker could potentially gain control of the system or manipulate it to make harmful decisions.

MALICIOUS WEB SERVICES

Malicious web services refer to web services that are intentionally designed to carry out malicious activities, such as distributing malware, stealing personal information, or launching phishing attacks. These services often masquerade as legitimate ones, making them challenging to identify. Malicious web services can be utilized in various ways, including, Malware distribution: Malicious web services serve as a means to distribute malware, such as viruses, trojans, and worms. This can be achieved by embedding malware within web pages, scripts, or other downloadable files. Personal information theft: Additionally, credit card numbers, Social Security numbers, names, addresses, and other personal information are stolen through malicious web services.

LITERATURE REVIEW

M. Kathiravan [1] et.al has proposed in this paper Malicious websites often impede the development of web service infrastructure and assist in the rise of online criminal activities. Thus, the necessity for a comprehensive plan to deter visitors from visiting these websites online is urgent. We support an approach that classifies websites as safe, spammy, or harmful using machine learning. Rather of looking at websites' contents, the suggested system can just look at the URL. Consequently, it eliminates run-time delays and vulnerabilities depending on browsers. Because the suggested solution uses learning methods, it

Performs better in terms of generality and coverage than blacklisting services. Website addresses fall into three different kinds. Websites that are neutral provide mediocre, risk-free features. Any effort to inundate a user with adverts or websites (such phony surveys and online dating services) is referred to as "spam" on a website. A website created by hackers with the intention of damaging computers and stealing personal information is known as malware. The experimental data shows that the new model performs much better than the baseline.

Daojing He [2] et.al has proposed in this paper in our increasingly interconnected world, network and cyberspace security is still difficult, and threats may arise from malevolent cyber activity like phishing. This article suggests a tiny-Bert stacking- based phishing website detection methodology as a countermeasure to phishing. The main idea is to learn the semantic and long-range dependent properties of URLs by using tiny-Bert to extract features from website URL strings. Next, we construct a classifier based on the stacking method, which consists of four fundamental learners: the first-level learners are Cat Boost, XG Boost, and Light GBM, while the second-level learner is GBDT. With the help of this detection model, phishing websites may be identified

Without the need for human feature extraction. Additionally, by compensating for each other's mistakes during the classification process, basic learners of stacking can increase accuracy and generalization. A data set derived from actual phishing websites is used to assess the suggested model. The findings indicate that the suggested model is more stable and has accuracy and recall rates of up to 99.14% and 99.13%, respectively, compared with the state-of-the-art.

Lizhen Tang [3] et.al has proposed in this paper Phishing attackers use social media, text messaging, and email to disseminate phishing links. They deceive people into accessing phishing websites and submitting vital personal information by using social engineering techniques. Ultimately, the pilfered personal data is used to exploit the credibility of reputable websites or financial establishments in order to reap illicit gains. The advancement and use of machine learning technology have led to the proposal of several machine learning- based methods for phishing detection. Certain solutions depend on characteristics that are retrieved via rules, while other aspects need third-party services, leading to instability and time-consuming problems in the prediction service. In this study, we offer a system for phishing website detection based on deep learning. We have

Integrated the framework as a browser plug-in that can instantly identify whether a person is about to be tricked by phishing and alert them with a warning. The real- time prediction service uses a combination of techniques, such as whitelist filtering, blacklist interception, and machine learning (ML) prediction, to increase accuracy, decrease false alarm rates, and shorten computation times. We examined numerous machine learning models utilizing various datasets in the ML prediction module. The RNN-GRU model demonstrated the viability of the suggested solution with the greatest accuracy of 99.18% based on the experimental data.

Shahd Albelali [4] et.al has proposed in this paper the digital world has come a long way in the last few years, especially with regard to the Internet, which is crucial since most of our daily activities are now done online. The likelihood of a cyberattack is increasing quickly due to the creative tactics used by attackers. One of the most dangerous types of assaults is the malicious URL, which aims to get unwanted information by deceiving unsuspecting end users. This may compromise a user's system and result in annual losses of billions of dollars. Consequently, the importance of website security is growing. In this study, we provide a comprehensive analysis of the literature emphasizing the

Primary methods for machine learning- based malicious URL identification, while accounting for detection technologies, feature kinds, datasets, and constraints in the literature.

Furthermore, we emphasize the directions of research in this area since there aren't many studies on the identification of harmful Arabic websites. Finally, we discuss issues that might deteriorate the quality of malicious URL detectors, along with potential remedies, as a consequence of the analysis we performed on the chosen research.

Boyang Yu [5] et.al has proposed in this paper Users' privacy and security on the internet are seriously threatened by malicious websites. Conventional methods for locating these malicious websites have had difficulty keeping up with the rapid evolution of attack techniques. Language models have been a viable option for successfully identifying and classifying dangerous websites in recent years. Based on the Transformer encoder architecture, this paper presents a unique Bidirectional Encoder Representations from Transformers (BERT) model intended to capture relevant features of malicious web addresses. Large-scale language models are also used for interpretability analysis, dataset evaluation, and training. With an astounding accuracy rate of 94.42%, the assessment results show how well the big language model works in identifying harmful websites. This performs better than current language models. In addition, the interpretability analysis clarifies the model's decision-making process, improving our comprehension of the categorization results. In summary, the Transformer encoder architecture-based suggested BERT model demonstrates strong performance and interpretability in identifying dangerous websites. It has potential as a way to improve network users' security and lessen the dangers brought on by malevolent online activity.

EXISTING SYSTEM

This article presents a methodology to identify web application vulnerabilities using machine learning (ML). Because of their diversity and the extensive usage of custom programming techniques, web applications present unique challenges for analysis. As a result, by using manually labeled data, machine learning (ML) enhances web application security by allowing automated analysis tools to incorporate human understanding of web application semantics. In order to create Mitch, the first machine learning solution for the black-box detection of Cross-Site Request Forgery (CSRF) vulnerabilities, the suggested methodology was used. Mitch's use resulted in the discovery of 35

New CSRFs on 20 prominent websites and 3 new CSRFs on production software.

PROPOSED SYSTEM

The proposed system utilizes a Hidden Markov Models (HMM) to quantify and predict the behavior of Malicious Web Services (MWSs). Initially, a set of features is extracted from the behavior of MWSs, which can include factors such as response time, size, and content of the responses sent by MWSs. These extracted features are then inputted into the HMM model, which is trained to predict MWS behavior specifically in terms of response time. Once the HMM model is trained, it can be utilized to rank MWSs in a quantitative manner. This ranking system aids in identifying the most reliable MWSs, which are those that are more likely to exhibit consistent and predictable behavior. The proposed system offers several benefits, including enhancing the accuracy of malicious URL detection systems, improving the performance of security applications reliant on MWSs, and reducing the cost and complexity associated with managing MWSs.

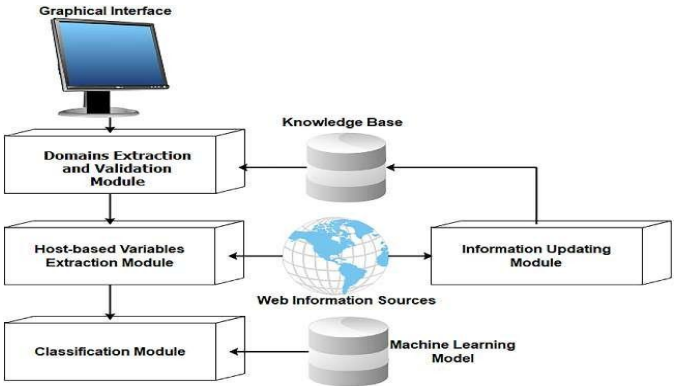

Figure 3. Detect potentially risky websites

USER INTERFACE AND WEBSITE VISITING

To establish a connection with the server, users are required to provide their username and password. Only after doing so, they will be able to successfully connect to the server. If the user already has an existing account, they can directly log in to the server. However, if the user is new, they must register their details, including their username, password, email address, city, and country, in order to create an account on the server. The database will then store the account information for all users, enabling the maintenance of upload and download rates. The user's name will be set as their user ID. Logging in is typically utilized to access a specific page, where the user's query will be searched and displayed. While the internet is intended to be a global network connecting the entire world, it is

Worth noting that certain websites are limited to specific countries. Consequently, it is not surprising that piracy rates are higher in countries where content is not legally available. Some services employ DNS wizardry to overcome this limitation.

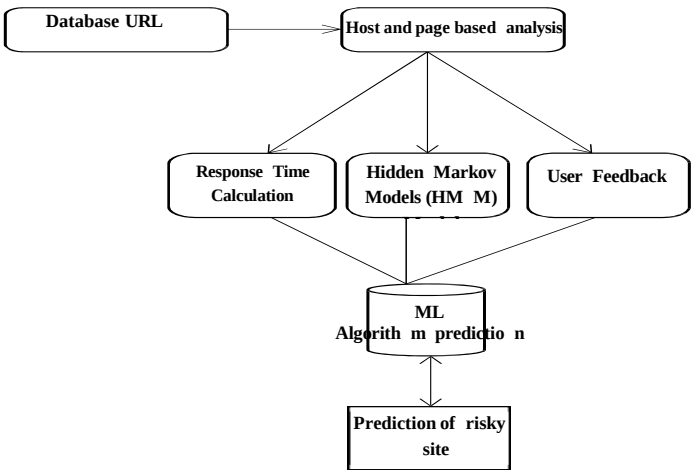

Figure 4. Block diagram

Response Time Calculation

Response time refers to the total duration required to respond to a service request. This service can encompass various tasks, ranging from fetching data from memory, performing disk input/output operations, executing complex database queries, to loading complete web pages. Response time is the total of service time plus wait time, omitting transmission time. In the context of technology, response time can also denote the time delay between the input and output signals, which is contingent upon the values of passive components employed. The total of the service and wait times is the response time, excluding transmission time. The time it takes for a functional unit or general system to respond to an input can also be used to define response time. Conversely, responsiveness refers to the speed at which an interactive system reacts to user input.

User Feedback

This module is utilized to incorporate user feedback regarding Risky Web Services. Feedback is crucial for the functioning and longevity of regulatory mechanisms in both living and nonliving systems, as well as in man-made systems such as education and economy. Information regarding reactions to a product or an individual's task performance serves as a foundation for enhancement. It is also important to modify or control a process or system based on its

Results or effects, like in a biochemical pathway or behavioral response.

Service Improvement

Healthcare organizations can improve the quality, productivity, and efficiency of patient care by implementing quality and service improvement tools in their setting. When used appropriately, these resources and methods allow medical professionals to quickly and economically identify and address issues, as well as guarantee the long-term viability of any patient outcomes improvements.

Algorithm Details

Hidden Markov Models (HMMs) are probabilistic models widely used in various fields, particularly in speech recognition, natural language processing, and bioinformatics. The algorithm involves a sequence of observable events generated by an underlying sequence of hidden states, where transitions between states are modeled by probabilities. HMMs consist of three main components: the initial probability distribution over hidden states, the transition probabilities between states, and the emission probabilities determining the likelihood of observable events given the hidden state. The Forward algorithm is commonly employed to compute the probability of observing a particular sequence, while the Viterbi algorithm finds

The most likely sequence of hidden states given observed data. The Baum-Welch algorithm is utilized for training HMMs by iteratively adjusting model parameters using the Expectation-Maximization (EM) algorithm. These algorithms collectively enable HMMs to capture temporal dependencies and uncertainties in sequential data, making them valuable tools in various applications.

A. Hidden Markov Models Algorithm

The Hidden Markov Model (HMM) algorithm can be implemented using the following steps:

Step 1: Define the state space and observation space

The state space is the set of all possible hidden states, and the observation space is the set of all possible observations.

Step 2: Define the initial state distribution

This is the probability distribution over the initial state.

Step 3: Define the state transition probabilities

These are the probabilities of transitioning from one state to another. This forms the transition matrix, which describes the probability of moving from one state to another.

Step 4: Define the observation likelihoods:

These are the probabilities of generating each observation from each state. This forms the emission matrix, which describes the probability of generating each observation from each state.

Step 5: Train the model

The parameters of the state transition probabilities and the observation likelihoods are estimated using the Baum- Welch algorithm, or the forward-backward algorithm. This is done by iteratively updating the parameters until convergence.

Step 6: Decode the most likely sequence of hidden states

Given the observed data, the Viterbi algorithm is used to compute the most likely sequence of hidden states. This can be used to predict future observations, classify sequences, or detect patterns in sequential data.

Step 7: Evaluate the model

The performance of the HMM can be evaluated using various metrics, such as accuracy, precision, recall, or F1 score.

To summarize, the HMM algorithm involves defining the state space, observation space, and the parameters of the state transition probabilities and observation likelihoods, training the model

Using the Baum-Welch algorithm or the forward-backward algorithm, decoding the most likely sequence of hidden states using the Viterbi algorithm, and evaluating the performance of the model.

B. Pseudo Code

Initialize:

- Define states S = {s1, s2, ..., sN}
- Define observations O = {o1, o2, ..., oT}
- Define transition probabilities A = {a_ij} where a_ij = P(s_j | s_i)
- Define emission probabilities B = {b_jk} where b_jk = P(o_k | s_j)
- Define initial state distribution π = {π_i} where π_i =

P(s_i at t = 1)

Forward Algorithm:

for t = 1 to T: for j = 1 to N:

if t == 1:

forward [j][t] = π_j * b_j(o1) else:

forward [j][t] = sum(forward[i][t-1]

* a_ij) * b_j(ot)

Backward Algorithm:

for t = T to 1: for i = 1 to N:

if t == T: backward[i][t] = 1

else:

backward [i][t] = sum(a_ij * b_j(ot+1) * backward[j][t+1])

Decode (Viterbi Algorithm):

for t = 1 to T: for j = 1 to N:

if t == 1:

Viterbi [j] [t] = π_j * b_j (o1) else:

Viterbi [j] [t] = max (viterbi[i] [t-1] * a_ij) * b_j(ot)

backpointer[t] = argmax(viterbi[:, t])

Estimation (Baum-Welch Algorithm): E-step:

Calculate forward and backward probabilities using the forward-backward algorithm.

M-step:

Update transition probabilities A and emission probabilities B based on the forward and backward probabilities obtained in the E-step.

Repeat E-step and M-step until convergence.

RESULT ANALYSIS

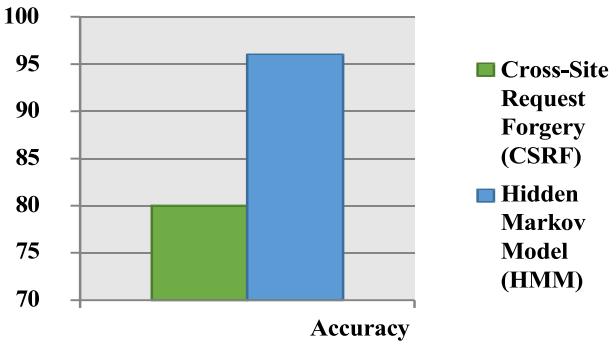

Figure 5. Comparison graph

The paper compares the performance of an existing algorithm with a new one, introducing a fresh approach to dangerous URL detection. The suggested algorithm achieves a higher accuracy rate of 96%, compared to the existing algorithm's 80% accuracy rate. By utilizing Hidden Markov Models and taking into account dynamic parameters like response time variability and system predictability, the suggested methodology is able to quantify and anticipate harmful online service behavior with a 5% increase in accuracy. This shows that the novel method offers a more efficient way to recognize and classify dangerous URLs, possibly providing a significant improvement over the state-of- the-art cybersecurity tactics. To evaluate the suggested algorithm's practical applicability and flexibility in the face of changing risks, however, is still necessary.

Table 1. Comparison table

Cross-Site Request Forgery (CSRF)	80
Hidden Markov Model (HMM)	96

CONCLUSION

In conclusion, the proposed system has the potential to enhance the accuracy, performance, and cost-effectiveness of security applications that rely on Malicious Web Services (MWSs). The system operates by extracting features from the behavior of MWSs and utilizing a Hidden Markov Models (HMM) model to predict their response time. Once trained, the HMM model can quantitatively rank MWSs, enabling the identification of reliable ones with predictable and consistent behavior. Compared to traditional methods like blacklists and heuristics, the proposed system offers several advantages. Firstly, it is more effective in detecting new and emerging threats. Secondly, it can predict MWS behavior in terms of response time. Lastly, it helps reduce the complexity and cost associated with managing MWSs.

FUTURE WORK

To further enhance the accuracy of the system, it is recommended to gather more data on MWSs and utilize it for training the Hidden Markov Models (HMM) model. Additionally, employing more sophisticated HMM models can also contribute to improved accuracy. In terms of performance, the system can benefit from utilizing more efficient HMM models.

Furthermore, implementing the system on a distributed platform can enhance its overall performance.

REFERENCES

[1] M. Kathiravan, V. Rajasekar, S. J. Parvez, V. S. Durga, M. Meenakshi and S. Gowsalya, "Detecting Phishing Websites using Machine Learning Algorithm," *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2023, pp. 270-275, doi: 10.1109/ICCMC56507.2023.10083999.

[2] D. He, X. Lv, S. Zhu, S. Chan and K. -K. R. Choo, "A Method for Detecting Phishing Websites Based on Tiny-

Bert Stacking," in *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2236-2243, 15Jan.15, 2024, doi: 10.1109/JIOT.2023.3292171.

[3] L. Tang and Q. H. Mahmoud, "A Deep Learning-Based Framework for Phishing Website Detection," in *IEEE Access*, vol. 10, pp.1509-1521, 2022, doi: 10.1109/ACCESS.2021.3137636.

[4] M. Aljabri *et al*., "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," in *IEEE Access*, vol. 10, pp. 121395-121417,2022, doi:10.1109/ACCESS.2022.3222307

[5] B. Yu, F. Tang, D. Ergu, R. Zeng, B. Ma and F. Liu, "Efficient Classification of Malicious URLs: M-BERT a Modified BERT Variant for Enhanced Semantic Understanding," in *IEEE Access*, vol. 12, pp. 13453-13468, 2024, doi: 10.1109/ACCESS.2024.3357095.

[6] J. Yuan, G. Chen, S. Tian and X. Pei, "Malicious URL Detection Based on a Parallel Neural Joint Model," in *IEEE Access*, vol. 9, pp. 9464-9472, 2021, doi: 10.1109/ACCESS.2021.3049625.

[7] Y. Liang, Q. Wang, K. Xiong, X. Zheng, Z. Yu and D. Zeng, "Robust Detection of Malicious URLs With Self-Paced Wide & Deep Learning," in *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 717-730, 1 March-April 2022, doi: 10.1109/TDSC.2021.3121388.

[8] E. Nowroozi, Abhishek, M. Mohammadi and M. Conti, "An Adversarial Attack Analysis on Malicious Advertisement URL Detection Framework," in *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1332-1344, June 2023, doi: 10.1109/TNSM.2022.3225217.

[9] F. Abri, J. Zheng, A. S. Namin and K.S. Jones, "Markov Decision Process for Modeling Social Engineering Attacks and Finding = Optimal Attack Strategies," in *IEEE Access*, vol. 10, pp. 109949- 109968, 2022, doi: 10.1109/ACCESS.2022.3213711

[10] A. S. Rafsanjani, N. B. Kamaruddin,H. M. Rusli and M. Dabbagh, "QsecR: Secure QR Code Scanner According to a Novel Malicious URL Detection Framework," in *IEEE Access*, vol. 11, pp. 92523-92539, 2023, doi:10.1109/ACCESS.2023.3291811.