Modeling and Predicting Cyber Hacking Breaches

C.Surekha[1], P.Ankith[2], B.Suresh[3], K.Karthik[4], A.Hemanth[5], P.Sathish[6]
[1]*Associate Professor, Hyderabad institute of technology and management, Medchal, Telangana*
[2,3,4,5,6] *UG student, Hyderabad institute of technology and management, Medchal, Telangana*

***Abstract*—The paper offers a practical method for detecting and classifying malicious URLs using a machine learning model that utilizes the Support Vector Machine (SVM) algorithm. The system determines whether URLs are malicious or safe by analyzing a dataset that contains known malicious indicators associated with different types of cyberattacks, including phishing, man-in-the-middle, and SQL injection. Real-world flag data is used to support the research and it was tested on a diverse set of URLs. By storing and analyzing the results, statistical insights are gained, which allows for visualization of attack distributions. By offering an automated, data-driven framework, the study helps cybersecurity research by identifying and mitigating web-based threats.**

I. INTRODUCTION

As technology becomes more digital, the threat of cyberattacks is on the rise as malicious actors exploit technological vulnerabilities and human error. Malicious URLs, which often impersonate legitimate web services or exploit browser vulnerabilities, are one of the most frequent attack vectors used to compromise users. The use of these deceptive links can result in phishing, malware downloads, or unauthorized access to sensitive data. The ability to detect threats in real-time has become a crucial element of cybersecurity strategies. The research addresses the challenge by proposing a machine learning-based system that is specifically designed to categorize and detect malicious URLs through analysis. The system's goal is to distinguish between web links that are benign and harmful based on patterns and attack signatures in URLs. A customized dataset is utilized has been developed to reflect real-world attack scenarios, enhancing the system's learning accuracy.

The algorithm that was selected, Support Vector Machine (SVM), is acknowledged for its effectiveness in classification tasks, particularly when working with high-dimensional feature spaces. The creation of optimal decision boundaries by SVM ensures a stable separation between malicious and safe URLs. Visual analytics are utilized in this study to provide a better understanding of attack types and their distributions, in addition to classification.

Precision, recall, and overall accuracy metrics are used to evaluate the system's performance, making sure it is practical and viable. Moreover, the following metrics are utilized: The model is created with scalability and adaptability, making it capable of integrating new attack patterns as they emerge. The implementation focus is not only on technical effectiveness, but also on usability, with an intuitive interface designed for security analysts. Early detection is emphasized, making it possible for users and organizations to take preventative measures before damage occurs.

The system continuously improves its predictive capabilities by learning from new data using machine learning. The method suggested highlights the increasing significance of intelligent systems in cybersecurity.

II. LITERATURE SURVEY

Statistical analyses of data breach patterns and historical cyberattack trends have been the main focus of previous cybersecurity studies. Maillart and Sornette conducted an in-depth examination of the statistical properties of identity thefts in the United States, revealing heavy-tailed distributions and shifts in the frequency of breach incidents over time. In the same way, Wheatley et al. analyzed large-scale organizational breaches, identifying the temporal independence of incident frequencies in U.S.-based firms and a rising trend among non-U.S. organizations. organizations. While these contributions have greatly improved our understanding of breach dynamics and their economic impact, they are mainly focused on providing retrospective insights and do not actively prevent or detect ongoing threats.

These approaches lack integration with real-time threat detection methodologies, particularly those that

utilize machine learning for predictive analysis, which is a notable limitation. This gap is bridged in the present study by the introduction of a new supervised learning framework designed to detect and categorize malicious URLs using Support Vector Machines (SVM).

Static threat indicators are transformed into actionable intelligence by this study's incorporation of structured threat intelligence, such as specific URL parameters and associated attack types, into a dynamic prediction model. It leverages curated datasets mapping URL features to known attack categories, such as phishing, SQL injection, and cross-site scripting, to train the SVM classifier. The process for processing incoming URLs, extracting relevant features, and classifying them as either safe or associated with specific attack vectors is done automatically with this approach. It not only enhances early threat detection, but it also provides a solution that is scalable and adaptive mechanism for handling evolving attack techniques.

Instead of focusing on trends after breaches occur, this method prioritizes proactive mitigation instead of statistical models. Structured threat data and machine learning are combined in the study to create a predictive and automated threat detection engine, which contributes significantly to cybersecurity defenses.

III. SYSTEM ANALYSIS

EXISTING SYSTEM

The purpose of this study is to investigate the question of whether data breaches caused by cyber-attacks increase, decrease, or stabilize? By providing a clear answer to this question, we will be better able to understand the overall situation of cyber threats. This question was not answered by previous studies. Previous studies did not answer this question. Specifically, the dataset was analyzed only for the time span from 2000 to 2008 and does not necessarily contain the breach incidents that are caused by cyber-attacks; the dataset analyzed in is more recent, but contains two kinds of incidents: negligent breaches (i.e., incidents caused by lost, discarded, stolen devices and other reasons) and malicious breaching.
Due to a higher proportion of human errors than cyber-attacks, negligent breaches are not included in the present study. the malicious breaches studied in contain four sub-categories: hacking, insider, payment card fraud, and unknown, this study will focus on the hacking sub-category (called hacking breach dataset there after), while noting that the other three sub-categories are interesting on their own and should be analyzed separately.
Researchers have recently begun to model data breaches. Maillart and Sornette examined the statistical characterization of personal identity losses in the US between 2000 and 2008. Edwards found that from 2000 to July 2006, there was a dramatic rise in breach incidents, but they remained steady thereafter. et al.examined a A dataset comprised of 2,253 breach incidents that spanned more than a decade (2005 to 2015). They found that the size or frequency of data breaches has not increased over the years. et al.analyzed a Dataset that corresponds to organizational breach incidents, i.e. those that was discovered that over 50,000 records were breached between 2000 and 2015. The frequency of large breach incidents occurring to US firms is unaffected by time, but the frequency of such incidents occurring to non-US firms is on the rise.

PROPOSED SYSTEM

In our study, we propose an intelligent system that utilizes the Support Vector Machine algorithm to categorize URLs as safe or malicious. When a malicious URL is detected, the system further categorizes it into specific types of cyberattacks. The foundation. Two datasets, malicious.json and URLS.json, contain the system, with malicious.json mapping specific flag terms to known attack types and URLS.json including a collection of real-world URLs for testing purposes. Each input URL is tokenized by the model using regular expression techniques, separating it into discrete elements or flags.
Tokens extracted from malicious.json are compared against the flag database. The corresponding attack type is assigned a weight if a token matches a known malicious pattern. The SVM model determines whether the URL is safe or malicious by aggregating the weights across all attack types. The classification result will be chosen based on the attack type with the highest score if malicious. Based on the prediction outcome, all URLs are categorised and stored in either malware.csv or unmalware.csv. The system calculates the frequency and distribution of various attack types through statistical analysis of detected threats to make them easier to interpret. The Matplotlib library is used to generate bar charts that visually represent these results, which are exported to a csv file.

The graph highlights the percentage of each attack category, giving an understanding of the prominence of specific threats. Additionally, a Tkinter-based graphical user interface was created to permit users to test URLs and view analysis results in an interactive manner. To estimate the precision of the threat detection mechanism, a custom gradient descent method applies to stored weights to calculate the model's accuracy.

By using this comprehensive setup, cyber threat awareness and response capabilities can be enhanced by automated classification, real-time feedback, and intuitive visualization.

IV. IMPLEMENTATION

Step 1: System Initialization

The SupportVectorMachine class is initialized in Python.Prepares the environment by checking for required output folders and initializing lists for attack types and their corresponding flags. approve or unapprove users based on them.

Step 2: Load Threat Data (malicious.json)

The system reads from malicious.json, which contains mappings of different cyberattack types to unique URL flags (e.g., x25, tcpchecksum, SYN). These mappings are essential for the model to understand which features in a URL suggest a certain type of cyber threat.

Step 3: Load Test Data (URLS.json)

The model reads real URLs from URLS.json. These URLs serve as inputs to test whether they are malicious and if so, to identify the type of attack.

Step 4: Feature Extraction

URLs are broken down into tokens (words, characters, numbers) using regular expressions. Each token is compared with known attack flags. Matching tokens increase the “weight” or score for the corresponding attack type.

Step 5: Threat Detection Logic

For each URL, a weight vector is formed. If any value is non-zero, the URL is marked as malicious. Distinguishes between safe and unsafe URLs based on detected threat signatures.

Step 6: Attack Classification

The attack type with the highest weight is selected using a simple max-score logic. Labels the malicious URL with a specific threat type (e.g., SQL injection, Phishing).

Step 7: Result Storage
The results are stored in two files:

- malware.csv – Malicious URLs with their attack type
- unmalware.csv – Safe URLs.
- Maintains logs of threat analysis for review and auditing.

Step 8: Analysis and Visualization

The system calculates the percentage occurrence of each attack type from malware.csv and plots a bar graph. Visualizes which attack types are most prevalent, providing insight for cybersecurity strategy.

ALGORITHM:

SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. However, it is not recommended for use in regression tasks. This is mostly used for classification problems. The algorithm involves plotting each data item as a point in a n-dimensional space, with each feature representing the value of a particular coordinate. After that, it is executed. To achieve classification, the hyper-plane must be found that effectively distinguishes between the two classes. Support vectors are simply the coordinates of individual observations. The Support Vector Machine is the frontier that is most effective in separating the two classes (hyperplane/line). formally, a support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional maargin), since in

general the larger the margin the lower the generalization error of the classifier.Whereas The original problem can be expressed in a finite-dimensional space, but it is common for the sets to separate to not be linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space.

SYSTEM ARCHITECTURE

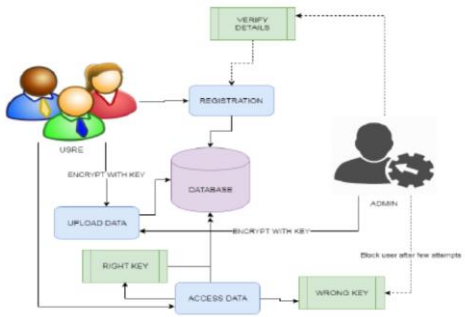

Figure1: System Arctitecture

V. RESULT

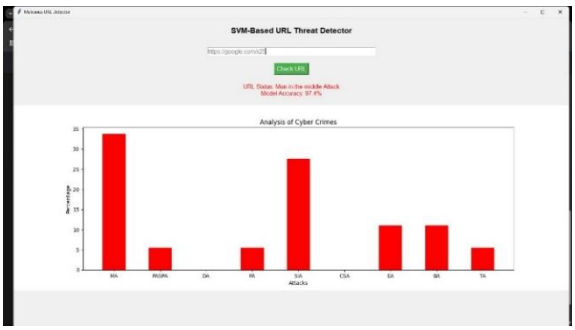

Figure 2: Output 1

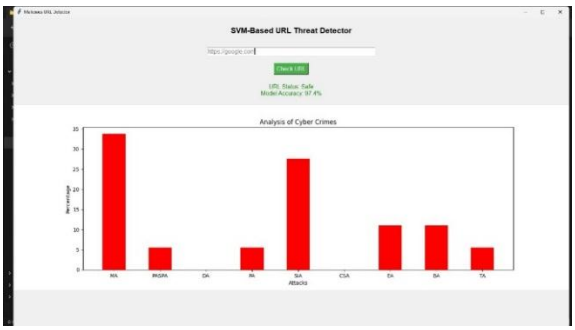

Figure 3: Output 2

VI. CONCLUSION AND FUTURE WORK

This study illustrates the effectiveness of utilizing Support Vector Machines to detect malicious URLs using structured threat patterns. Not only can the model detect threats, but it can also categorize them, which is valuable in cybersecurity forensics and prevention. Our future plan involves enhancing the model with real-time URL monitoring, deep learning models for improved accuracy, and integration with browser extensions or cybersecurity tools. We also plan to expand the dataset with more diverse samples and adapt the model for multilingual URL structures.

REFERENCES

[1] P. R. Clearinghouse. Privacy Rights Clearinghouse's Chronology of Data Breaches. Accessed: Nov. 2017. [Online]. Available: https://www.privacyrights.org/data-breaches

[2] ITR Center. Data Breaches Increase 40 Percent in 2016, Finds New Report From Identity Theft Resource Center and CyberScout. Accessed: Nov. 2017. [Online]. Available: http://www.idtheftcenter.org/ 2016databreaches.html

[3] C. R. Center. Cybersecurity Incidents. Accessed: Nov. 2017. [Online]. Available: https://www.opm.gov/cybersecurity/cyberse curity-incidents

[4] IBM Security. Accessed: Nov. 2017. [Online]. Available: https://www.ibm.com/security/data-breach/index.html

[5] NetDiligence. The 2016 Cyber Claims Study. Accessed: Nov. 2017. [Online]. Available: https://netdiligence.com/wp-content/uploads/2016/ 10/P02_NetDiligence-2016-Cyber-Claims- Study-ONLINE.pdf

[6] M. Eling and W. Schnell, "What do we know about cyber risk and cyber risk insurance?" J. Risk Finance, vol. 17, no. 5, pp. 474–491, 2016.

[7] T. Maillart and D. Sornette, "Heavy- tailed distribution of cyber-risks," Eur. Phys. J. B, vol. 75, no. 3, pp. 357–364, 2010.

[8] R. B. Security. Datalossdb. Accessed: Nov. 2017. [Online]. Available: https://blog.datalossdb.org

[9] B. Edwards, S. Hofmeyr, and S. Forrest, "Hype and heavy tails: A closer look at data breaches," J. Cybersecur., vol. 2, no. 1, pp. 3–14, 2016.

[10] S. Wheatley, T. Maillart, and D. Sornette, "The extreme risk of personal data breaches and the erosion of privacy," Eur. Phys. J. B, vol. 89, no. 1, p. 7, 2016.

[11] P. Embrechts, C. Klüppelberg, and T. Mikosch, Modelling Extremal Events: For Insurance

and Finance, vol. 33. Berlin, Germany: Springer-Verlag, 2013.

[12] R. Böhme and G. Kataria, "Models and measures for correlation in cyber- insurance," in Proc. Workshop Econ. Inf. Secur. (WEIS), 2006, pp. 1–26.

[13] H. Herath and T. Herath, "Copula- based actuarial model for pricing cyber- insurance policies," Insurance Markets Companies: Anal. Actuarial Comput., vol. 2, no. 1, pp. 7–20, 2011.

[14] A. Mukhopadhyay, S. Chatterjee, D. Saha, A. Mahanti, and S. K. Sadhukhan, "Cyber-risk decision models: To insure it or not?" Decision Support Syst., vol. 56, pp. 11–26, Dec. 2013.

[15] M. Xu and L. Hua. (2017). Cybersecurity Insurance: Modeling and Pricing. [Online]. Available: https://www.soa.org/research-reports/ 2017/cybersecurity-insurance

[16] M. Xu, L. Hua, and S. Xu, "A vine copula model for predicting the effectiveness of cyber defense early- warning," Technometrics, vol. 59, no. 4, pp. 508–520, 201