

Hybrid Machine Learning and Deep Learning Model for Malicious URL Detection

Shanthosh .S^{*1}, Dr. G.Rosline Nesa kumari^{*2}

^{*1}Assistant Professor, ^{*2}Professor

Department of computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai – 600073, Tamil Nadu, India.

Abstract: The World Wide Web has grown to be a vital component of our daily lives for the exchange of information and the sharing of knowledge. It facilitates timely, quick, and simple information transactions. Theft of identification and fraud involving identity are two forms of cybercrime in which hackers and malicious users take the personal data of authorized users in an attempt to perpetrate fraud or deception for financial gain. Because rogue URLs host unwanted material (email spam, scams, brute force flaws, etc.) to trick gullible people into falling for schemes (money loss, private information theft, malicious software deployment, etc.). Systems should be quick, accurate, and able to identify new malicious content in order to detect such crimes. Traditionally, blacklists are used primarily for this detection. Blacklists, however, are not all-inclusive and are unable to identify recently created dangerous URLs. In an effort to increase the breadth of malicious URL detectors, machine learning techniques have been the focus of more study in recent years. In order to determine whether a URL is good or poor, employ a straightforward technique in this work, which contrasts with SVM and LR, two additional methods.

Keywords: Detection of malicious URLs, machine learning, CNN, SVM, Logistic Regression, Cybersecurity, Phishing Detection, URL Threat Classification.

1. INTRODUCTION

Many studies have been conducted on preventing people from visiting dangerous websites in an effort to reduce online crimes. The worldwide address for documents and other resources on the World Wide Web is called the Uniform Resource Locator, or simply URL. Cybersecurity is often threatened by rogue URLs, sometimes known as harmful websites. Various unsolicited content, such as spam and phishing, is hosted by hostile websites or malicious URLs in order to initiate assaults. A variety of frauds, involving financial loss and the theft of private information (credit cards, identity, etc.), target unsuspecting people who visit these websites. The following are common forms of attacks that use malicious URLs: spam, social engineering, and phishing [1].

According to data from Google, up to 9,500 harmful websites are blocked on average every day. A serious risk to Web application security is the presence of these rogue websites. In order to identify malicious URLs, researchers and practitioners have worked hard to provide effective solutions.

The most popular technique used by numerous antivirus groups to identify dangerous URLs is the black-list method. More specifically, black URL lists are just a database of web addresses that have been shown to be dangerous in the past. This approach is very rapid and easy to use because WHOISust requires a little inquiry latency. It is additionally logical that the false-positive rate would be incredibly low with this approach. Having said that, it is almost difficult to maintain an exhaustive list of hazardous URLs, especially with the daily creation of new Websites. By obfuscating URLs to look authentic, attackers employ inventive methods to get over blacklists and trick consumers. All of these attempt to conceal the website's malevolent aim by disguising the harmful URL. Once users click the URLs and they seem authentic, an assault can be initiated. This is often accomplished by inserting malicious code into JavaScript. In order to evade detection by building signatures, attackers often try to disguise the code in question. Therefore, blacklisting techniques have significant drawbacks, and it seems nearly effortless to get around them, particularly since blacklists aren't useful for predicting new URLs. The building of an automated program that can swiftly and precisely differentiate newly emerging dangerous websites from URLs and other sizable, typical web pages is therefore an important issue that has to be resolved. The classification of attack types is helpful because it enables us to respond appropriately and implement relevant and efficient countermeasures against the threat. For instance, should react right away to malware infection yet comfortably overlook spam. The rest of the article is organized as follows. Section 2 displays related work. Section 3: Classification Techniques. Section 4 provides details on the Experiments. A summary of the results and conclusions will be given in Section 5.

2. RELATED WORKS

Numerous studies have been conducted both domestically and internationally to classify harmful URLs, incorporating the use of multi-layer malicious web page detection techniques [4], dynamic attack techniques for detection [3], and deep learning techniques [2], among others. provide a machine-learning technique for the automatic identification of obfuscated JavaScript. [6] suggest a method for identifying these URLs just by their lexical characteristics, making it possible to notify the user before the page is

actually retrieved. In [2], a new advanced deep learning approach to detecting harmful code written in JavaScript is introduced. Testing results show that, in the best-case scenario, it can achieve as much as 95% precision with a percentage of false positives of a value below 4.2%. The detection effectiveness is higher than that of the conventional neural network approach. [8] is the first to present access relations and possesses the qualities of self-learning and feedback. [9] On the basis of domains' historical data, an anomaly domain identification technique was put out. The suggested algorithm employed domain lifespan, WHOIS information changes, information integrity, IP changes, IP-sharing domains, TTL metrics, etc., based on statistical distinctions between malicious and legitimate territory heritage data. The primary parameters and tangible depictions of the characteristics for categorization were provided. And the suggested algorithm built an SVM classifier for identifying anomaly domains based on this. The algorithm's strong detection accuracy for unknown domains is demonstrated by feature analysis and experimental results; it is particularly well-suited for identifying hostile domains that have a long lifespan. Most current methods are feature-based and unable to identify dynamic attacks. In order to launch a malicious attack, the attacker typically employs an input form, active content, and the @ sign embedded in the URL to recognize this assault. [10] A proposed technique is called Behavior-based Malicious URL Finder (BMUF). It examines the URL's behavior. The behavior of the URL is modeled into different states using the state changeover flowchart according to the FSM. The state transition from the initial to the final state is used for classification. This technique determines if the URL is functioning deliberately or lawfully based on the user's responses. It accurately identifies the kind of URL. It [16] investigated intelligent threat analysis and fault resilience in cloud banking microservices. It [17] explored reinforcement learning-based cyber deception for attack surface reduction. It [18] examined anomaly-driven threat detection and fault tolerance, offering relevant cybersecurity foundations for developing intelligent malicious URL detection systems. It [19] explored federated data modeling for secure LLM deployment in cloud-native architectures, emphasizing privacy and secure AI infrastructure. It [20] investigated blockchain and AI integration for financial traceability and cyber-resilience, providing broader insights into trustworthy AI and security-oriented system design.

Figure 1 shows the suggested system's architecture. The components include the CNN classifier, feature extraction, results, blacklist, IP registry, and the World Wide Web.

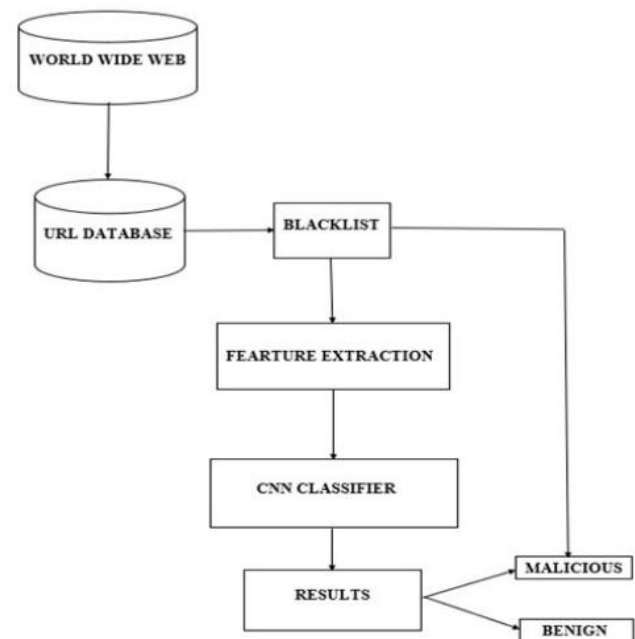


FIGURE 1: OUTLINE OF THE PROCEDURE

Figure 1 shows a description of our framework. The URL is used to serve as a database-specific feed in the above process. There are then two scenarios when the URL is entered into the blacklist: Initially, if the URL is already on our blacklist, it will be considered malicious. Second, for the analysis, the URL's feature extraction is taken out. The classifier produces either benign or malevolent outputs. The remainder of this section will provide an explanation of each step in our methodology.

3. CLASSIFICATION TECHNIQUES

An overview of the various classification methods is given in this section. Only a handful of the numerous categorization algorithms will be discussed here. Our method uses the logistic regression model standard (LR) and the standardized support vector machine (SVM) method, along with convolutional neural networks, also called CNN techniques, to classify the URL. In this subsection, we describe the many categorization types that use our method.

3.1 NETWORKS OF CONVOLUTIONAL COGNITIVE SYSTEMS

Convolutional neural systems (CNNs) are composed of transistors that learn to optimize themselves, much like traditional ANNs. Each neuron still receives an input and performs an operation (such as a product that is accompanied by a non-linear deterministic calculation) as the basis of countless ANNs. The network will keep using an individual evaluative assessment system (strength) from the input raw image vectors to generate the final classification score output. The last layer will contain class-related loss functions, and all of the common tips and methods developed for traditional ANNs remain viable. The sole notable difference between CNNs and traditional ANNs is that their main use is in the field of visual feature recognition. This improves the

network's adaptability for image-focused tasks by allowing us to embed image-specific features into the architecture and significantly reducing the number of parameters required to build and configure the framework. The inability of conventional ANN models to handle the computational complexity needed to process image input is one of their biggest drawbacks. The majority of ANN types may make use of popular machine learning benchmarking information sets, including the MNIST repository of numbers that have been handwritten, because of its comparatively tiny image dimensionality of only 28×28 . Most ANNs can handle the 784 weights ($28 \times 28 \times 1$) that this set of data will offer an individual neuron in the initial invisible region. Remember that MNIST is only adjusted to black as well as white variables. The quantity of variables on an individual first-layer neuron increases tremendously to 12,288 when a bigger 64×64 vibrant graphic source is taken into consideration. When you consider seriously the fact that the network has to be much bigger to accommodate this input scale compared to the network used to classify color-normalized MNIST digits, the drawbacks to utilizing such algorithms become evident.

3.2 VECTOR MACHINE SUPPORT

Prediction and subgroups of automated learning techniques that include methods for supervised learning like support vector machines. It is provided with an ensemble of training information to essentially classify it. A collection of data for training is made available for it to categorize. Data must often be divided into training and testing sets for a classification task. The goal of support vector machines (SVM) is to create a model that uses the training data for forecasting the desired outcomes of the test information. [13]. The SVM approach is unaffected by the constraints of limited samples and data complexity. Several current investigations have shown that support vector machines, or SVMs, may frequently outperform alternative data categorization algorithms in relation to precision in classification.. Text classification, handwriting, digit and tone identification, has been used for a variety of practical issues, spanning data categorization, experimental gene expression data manipulation, and picture classification and recognizing objects [14]. SVM is a machine learning-based algorithm to recognize malware [15].

3.3 FEATURE - BASED BINARY CLASSIFICATION

The variable that depends on a model using logistic regression is multimodal. A variable that is dependent on another can represent consequences involving either pass or fail, winning/losing, alive or deceased, or ill or healthy when it is binary, which means indicating it is limited to having just two potential values—"0" and "1." This is covered by logistic regression. When there are more than two result categories for the dependent variable, multinomial logistic regression or ordinal logistic regression may be used to evaluate the data. A qualitative response/discrete choice paradigm is exemplified by logistic regression in economics language.

4 INVESTIGATIONS

4.1 SET OF DATA

Figure 1 The information set used in this research may be found at this GitHub repository. Initially, a specially designed crawler was created to explore websites with dangerous URLs in order to gather malicious links from other domains. However, manual crawling was not necessary for this investigation because an existing dataset was used. Seventy-five,643 of the approximately 420,464 URLs in this dataset are classed as malicious; the other URLs are either clean or benign.

For the purpose of identifying phishing website addresses, a technique known as Convolutional Neural Network, or CNN for short, was employed. The first step in the procedure was gathering and preprocessing URL data. Each URL was fed into the model, which used learnt patterns and features to classify it as either benign or malicious.

4.2 BLACKLIST

Blacklisting, essentially retains a database of web addresses that have been identified to be threatening, is a prevalent and conventional method for tracking down URLs that are potentially dangerous. Each time that a new Address is accessed, an information base inquiry takes place. If a website address is determined to be malicious and added to the blacklist, if there is no warning, it is presumed to be safe. The process of blacklisting is not able to maintain a comprehensive list of all potentially dangerous URLs due to the rapidity with which fresh website addresses may be formed on a daily basis, making it hard to detect new threats [11]. Despite a number of issues, due to its simplicity and efficiency, being blacklisted is considered to be among the most commonly utilized methods used by many antivirus software products currently [12]. Common threats are discovered, and based on their behavior, this threat vector is assigned a signature. However, only a small number of typical risks may be addressed by such strategies. As previously stated, using blacklists is a simple method of identifying fraudulent URLs. Additionally, they examined how effective these features were in comparison to other features and found that while blacklist features by themselves performed worse than other features, when combined with other features, the overall performance improved.

4.3 EXTRACTION OF FEATURES

To detect fraudulent URLs in this area, employ two contrasting feature categories: word2vec aforementioned features and term prevalence-inverse document frequency features.

4.3.1 FEATURES OF WORD2VEC

Text is processed using a neural network with two layers named Word2vec. A collection of vectors—vectors of characteristics for each word in the collection—is what is delivered after receiving corpus text as input. Although Word2vec, as the name suggests, is not a deep neural network in the conventional sense, it converts text into a

statistic in numbers that deep networks can comprehend. Using deep learning is used to create a distributed version of Word2vec for Java as well as Scala which employs GPUs and a Spark framework. We have 26 letters that have been translated into space using word2vec in the area related to the vector's properties. A vector measuring one by one hundred in space is used to represent each letter.

4.3.2 Attributes of Term Frequency—Contrary to Document Resonance.

Frequency-inverse frequency of documents (tf-idf), a numerical parameter used in information retrieval, shows just how important a term is in respect to a document inside a repository as well as a standard collection.[1]. It is often used as an element of weight in text mining, user modeling, as well as information extraction. The tf-idf value, which increases in proportion to a word's occurrence frequency in the text, can occasionally be offset by the word's frequency in the corpus to account for the fact that selected words emerge more commonly compared to elsewhere. In modern times, tf-idf, for example, is one of the most used keyword weighting methods; for instance, 83% of a form of text recommendation algorithms in the internet-based library sector incorporate it. [2] Search engine optimization frequently employ different iterations of the tf-idf scaling algorithm in order to determine and evaluate a piece of content's relevancy to an individual's demand. TF-IDF stop-word filtering can be effectively applied to a variety of subject areas, such as categorization and text summarization. One of the shortest scoring formulae is calculated by summing the tf-idf for every question term; this basic model is the basis for many more complex ranking algorithms.

5. RESULTS AND ANALYSIS

We tested 75643 dangerous URLs and 344821 benign URLs. With a reliability score of exceeding 95 percent, our method has identified dangerous URLs in this algorithm.

5.1 OBTENDS

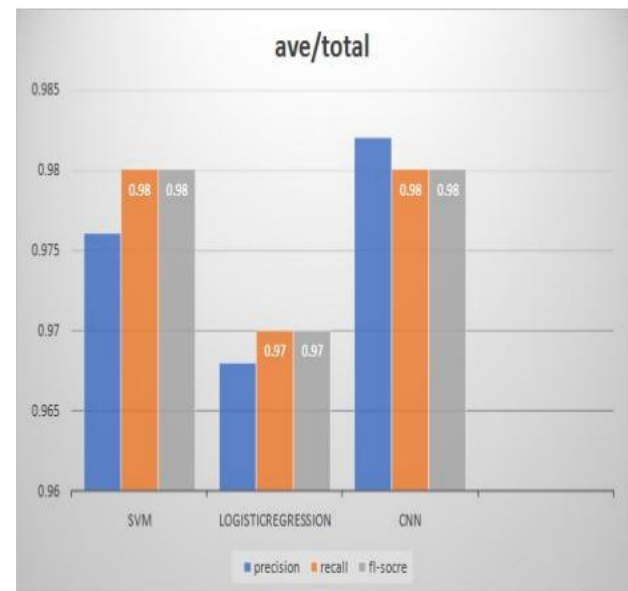


Figure 2: characteristics of various algorithm representations in the identification of dangerous URLs.

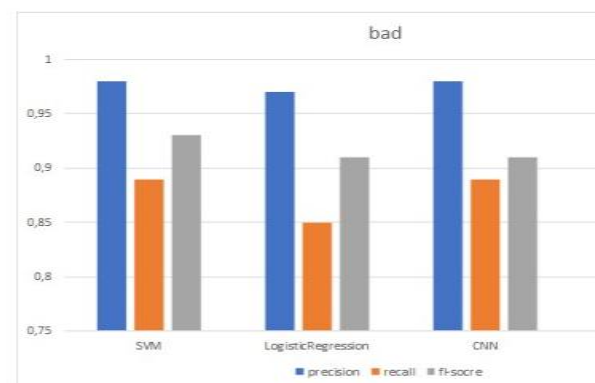


Figure 3: Several algorithms are used to identify malicious URLs.

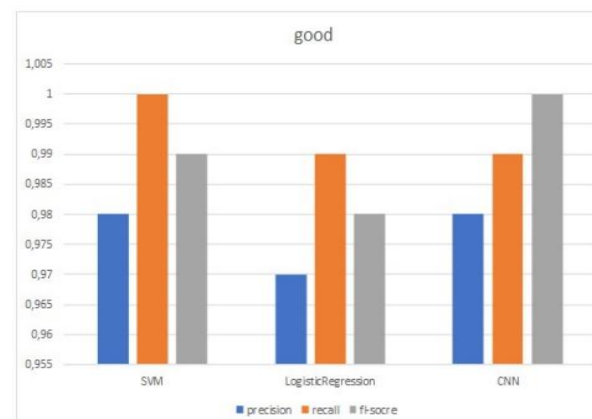


Figure 4: Several methods are used to identify excellent URLs.

6. SUMMARY

For many cybersecurity applications, malicious URL detection is essential, and deep learning techniques are undoubtedly a viable avenue. This work discusses the CNN methodology focused on the word2vac feature, logistic regression, as well as the support vector algorithm strategy

based on term prevalence–inverse file rate. The three characteristics of SVM, logical regression, and CNN—precision, recall, and fl-socre—can be compared to draw a conclusion. The following three column tables demonstrate how the logical regression method uses the term frequency–inverse document frequency of SVM. The precision, recall, and fl-socre of SVM are all marginally greater than those of the logical regression methodology. Based on the inverse document frequency and term frequency, the SVM approach is compatible with the Word2vac-based convolution neural network.

REFERENCES

- [1] Y. Wang, W.-d. Cai, and P.-c. Wei, “A deep learning approach for detecting malicious javascript code,” *Security and Communication Networks*, 2016.
- [2] JAnjali B. Sayamber ,Arati M. Dixit , “Malicious URL Detection and Identification”, *International Journal of Computer Applications* (0975 – 8887) Volume 99 – No.17, August 2014.
- [3] D. R. Patil and J. Patil, “Survey on malicious web pages detection techniques,” *International Journal of u-and e-Service, Science and Technology*, vol. 8, no. 5, pp. 195–206, 2015.
- [4] Usha Narra, Corrado Aaron Visaggio, Mark Stamp, Thomas H. Austin, “Clustering versus SVM for malware detection”, *Springer, Journal of Computer Virology and Hacking Techniques* 10/2015.
- [5] R. K. Nepali and Y. Wang, “You look suspicious!/: Leveraging visible attributes to classify malicious short urls on twitter,” in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 2648–2655.
- [6] M.D.Zeiler,“Adadelat:anadaptivelearningratemethod,”*arXi vpreprintarXiv:1212.5701*, 2012.
- [7] L. Xu, Z. Zhan, S. Xu, and K. Ye, “Cross-layer detection of malicious websites,” in *Proceedings of the third ACM conference on Data and application security and privacy*. ACM, 2013, pp. 141–152.
- [8] B.-I. Kim, C.-T. Im, and H.-C. Jung, “Suspicious malicious web site detection with strength analysis of a javascript obfuscation,” *International Journal of Advanced Science and Technology*, vol. 26, pp. 19–32, 2011.
- [9] E. Sorio, A. Bartoli, and E. Medvet, “Detection of hidden fraudulent urls within trusted sites using lexical features,” in *Availability, Reliability and Security (ARES)*, 2013 Eighth International Conference on. IEEE, 2013, pp. 242–247.
- [10]Liu Aijiang,HuangChanghui and Hu Guangjun,”Detection Method of Trojan’s Control Domain Based on Improved Neural Network Algorithm,”*China Academic Journal Electronic Publishing House*,2014.
- [11]SHA Hong-zhou,ZHOUGHOUZhou,LIU Qing-yun and QIN Peng,“Light-weight self-learning approach for URL classification”,*Journal on Communications*,2014.
- [12]YUAN Fu-xiang ,LIU Fen-lin ,LU Bin and GONG Dao-fu ,“Anomaly domains detection algorithm based on historical data”*Journal on Communications*,2016.
- [13]Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi,”Malicious URL Detection using Machine Learning: A Survey”,2017.
- [14]S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, “An empirical analysis of phishing blacklists,” in *Proceedings of Sixth Conference on Email and Anti-Spam (CEAS)*, 2009.
- [15]S. Sinha, M. Bailey, and F. Jahanian, “Shades of grey: On the effectiveness of reputation-based “blacklists”,” in *Malicious and Unwanted Software*, 2008. *MALWARE* 2008. 3rd International Conference on. IEEE, 2008, pp. 57–64.
- [16]A. Budaraju and J. D. R. Pesaramilli, “Intelligent Threat Analysis and Fault Resilience Mechanism for Cloud Banking Microservices,” in *Innovative Computing and Communications, ICICC 2026, Lecture Notes in Networks and Systems*, vol. 2037. Springer, Cham, 2026. doi: [10.1007/978-3-032-30008-9_7]
- [17]J. Pesaramilli and T. Gudisa, “Real-Time Attack Surface Reduction in Cloud Infrastructures Using Reinforcement Learning-Driven Cyber Deception Strategies,” 2025. doi: [10.1109/ICONSTEM65670.2025.11374717]
- [18]J. David and A. Budaraju, “Self-Healing Microservices for Cloud Banking with Anomaly-Driven Threat Detection and Fault Tolerance,” in *Proceedings of the International Conference on Computing Systems and Intelligent Applications (ComSIA 2026)*, *Lecture Notes in Networks and Systems*, vol. 2054, Springer, 2026, pp. 524–541.
- [19]A. Kumar Jonnalagadda, B. Madupati, R. Varma Vegesna, and S. Kumar Vududala, “Federated Data Modeling for LLM Deployment in Secure Cloud-Native Architectures,” in *2025 International Conference on Computing Technologies & Data Communication (ICCTDC)*, Hassan, India, 2025, pp. 1–8, doi: [10.1109/ICCTDC64446.2025.11158782]
- [20]B. Madupati, A. K. Jonnalagadda, S. K. Vududala, and R. V. Vegesna, “Blockchain and AI Fusion for Financial Traceability and Cyber-Resilience in Industry 5.0 Environments,” in *2026 Innovations in Machine, Engineering, and Digital Conference (IMED)*, Kota Kinabalu, Malaysia, 2026, pp. 1–7, doi: [10.1109/IMED68921.2026.1148499]