Journal of Posthumanism

2025
Volume: 5, No: 5, pp. 412–427
ISSN: 2634-3576 (Print) | ISSN 2634-3584 (Online)
posthumanism.co.uk

DOI: https://doi.org/10.63332/joph.v5i5.1352

Real-Time Detection of Command-and-Control Communications Using Deep Learning Models

Nouf Aljammaz[1], Suliman Mohamed Fati[2], Mamdouh Alenezi[3]

Abstract

Increasingly advanced cyber threats pose a challenge for cybersecurity professionals, and C2 communications detection and prevention remain an extremely critical issue. Polymorphic malware and encrypted channels support modern adversaries in stealthy control of compromised systems. Redundant signature-based detection cannot be effective in those cases. Therefore, in this paper, we present a novel framework based on deep learning and real-time classification for malicious C2 traffic detection. More specifically, an MLP model is trained with a custom-designed dataset of network traffic to efficiently discriminate between legitimate traffic and allegedly malicious C2 packets. In addition to the MLP, there is also a real-time classification system based on behavioral analysis of SSL certificates and Nmap script outputs in order to reveal Metasploit and Cobalt Strike threat types. Extensive testing of self-collected data validates the excellent performance of the detection innovation with 99% detection rate of C2 threats and 99.9% correct classification in specific frameworks. Behavioral assessments and deep learning come together to form a powerful and scalable defense against a new breed of cyber threat.

Keywords: *Command and Control (C2) Detection, Deep Learning in Cybersecurity, RealTime Threat Classification, Network Traffic Analysis.*

Introduction

In the progress of cybersecurity from the concept of sabotaging cognitive functions to falsifying the information about the events that took place, well-known examples keep emerging, such as (Zaid & Garai, 2024; Zeadally et al., 2020). It is one of the eminent avian targets for any detection or prevention treatment with regard to Command or Control (C2) communications, the other side to this malware-driven intrusion (Ghafir et al., 2018; Caviglione et al., 2020). This type of communication allows an adversary to organize and maintain control over a compromised system, allowing that system to carry out malicious activities (Ashfaq et al. 2022; Bastiaansen et al., 2020; Eisenberg et al., 2018). The increasing rate and complexity of such attacks invoke greater and more urgent demand to develop countermeasures (Ferdous et al., 2023; Obi et al., 2024). The customary signature-based detection techniques, which rely on signature or predefined patterns, have been found to be impotent against modern malware (Malik et al., 2023; Torres et al., 2023). Increased encryption has concealed malicious activity, making the situation worse, as has improved polymorphic malware, which is continuously mutated to circumvent detection (Akhtar & Feng, 2022; Aslan & Samet, 2020).

[1] College of Computer and Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia, Email: 222420834@psu.edu.sa
[2] College of Computer and Information Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia, Email: sgaber@psu.edu.sa (Corresponding Author).
[3] The Saudi Technology and Security Comprehensive Control Company, Saudi Arabia, Email: malenezi@psu.edu.sa

Modern malware, indeed, shows changes in the behavior and evasive strategies that have only complicated the job of malware detection based only on classical signatures (De Gaspari et al., 2022; Geng et al., 2024) and exemplify the meaning of evasiveness itself-, invasive behaviours, looking for newer solutions that could adapt to a dynamic threat landscape. Consequently, efforts are now directed to using deep learning and real-time classification for the purpose of the making of C2 communications detection. Deep learning has shown to be the most suitable part of machine learning that allows for the successful analysis of complex patterns and anomalies, real-time classifications, so that malicious actions can be captured as opposed in a continuous classification. It is the convergence of this that can possibly change the world of cyber by obtaining a dynamic and scalable solution to the ever-evolving threat of cyber.

Recent breakthroughs in deep learning have enabled the development of advanced models, capable of making an effective distinction between genuine and malicious network traffic. Their application in the detection of C2 communications has shown some promising results with several evidence-based studies demonstrating the power of deep learning towards the identification of threats. However, most of these studies have mainly concentrated on the analysis of offline data, sidelining what is really critical in real-time detection and classification. Thus, to realistically counter modern cyber threats, one needs to establish solutions for analyzing network traffic in real-time whilst identifying and classifying the malicious activity with the least latency.

Cyber threats have grown astronomically in the past years, increasing in sophistication among attack vectors along with the dissemination and proliferation of advanced malware. What gives current malware the most concerning edge over their predecessors would be the use of Command-and-Control (C2) systems that allow adversaries to access and take control of the compromised networks remotely for malicious activities such as data exfiltration, ransomware deployment, or espionage. Such covert communication channels increasingly endanger organizations by evading traditional security countermeasures, thereby demonstrating the importance of embedding detection mechanisms with much more adaptability. Multiple protocols are used in C2 systems, such as HTTP or DNS, along with encrypted channels: this helps them exploit the legitimate traffic to evade detection. Encryption protocol deployments like Transport Layer Security (TLS) act as an added burden to C2 activity detection, as it encrypts the contents of the communication packet. Therefore, the existing security measures, such as signature-based and rule-based systems, often fail against modern polymorphic and evasive malware. This ever-changing threat landscape emphasizes the need for advanced methodologies that would support behavioral analysis and real-time detection.

HTTP servers were being acknowledged as an excellent platform of preferred means of web communication employed by a broad spectrum of threat carriers to disguise their entire activities over the internet. The technicalities of HTTP itself, together with its wide acceptance (which very few firewalls normally block), add as an entanglement to the identification of malicious communications amidst the high volume of HTTP traffic. And so, both Botnets and Post-exploitation tools have their C2 servers propagate on HTTP, available to a vast majority of machines. Usually, an infected host picks particular HTTP GET requests from a predefined destination C2 server and executes any planned tasks as mentioned in the C2 server's response (Shah, 2004).

As distinctive signatories thereunder, detection of such malicious C2 servers has practically gained paramount significance for teams gearing up for the better defense of any sort against named adversaries. Detection of C2 communications remains the most significant defense to protect networks and minimize damage resulting from cyberattacks, including theft of data, propagation of malware, and remote control of affected systems.

Historically, signature-based methods for identifying C2 activity have involved drawing up predefined patterns or signatures from historical threat intelligence (Ghafir et al., 2018). Security experts studied various malicious behaviors and turned those into unique identifiers or fingerprints (Novo & Morla, 2020). Security systems, such as antivirus software and intrusion detection systems, implement these signatures to match them against activities involved in scanning network traffic or files. While useful with past threats, these approaches have some disadvantages since they are signature-dependent and may produce false positives or negatives. Signature databases require regular manual updating to keep pace with the threats emerging. Nonetheless, signature-based methods in traditional format have often fallen short of C2 detection for various reasons: they work only against threats that are already known and have trouble assessing novel or evolving C2 tactics that do not have established signatures. Additionally, detection based on signatures often falsely identifies benign activities as threats or misses more opportunistic ones, sometimes called polymorphic or obfuscating C2 traffic. Static by nature, signatures do not adapt to novel attacks that evolve quickly; thus, there is a strong need for more advanced detection schemes, potentially behaviorally based (Khan et al., 2019; Ghafir et al., 2018).

An advanced technique for C2 activity detection is deep learning which mainly uses the differentiating power in analyzing data pattern-the recognition of very complicated patterns. This normally involves training deep neural network models on large datasets controlled for all normal network behavior instances and C2 communication instances (Catillo et al., 2023). Important features are extracted, including network traffic pattern and packet content, for representation. The neural network architecture is thus carefully designed to include input layers, hidden layers to learn the representations, and an output layer for normal or C2 classification. The training process refines the model's internal weights via supervised learning and attempts to minimize the deviation between predicted and actual labels. After training, the model has the capability to perform inference on the unseen data by inspecting incoming network traffic or communication patterns to differentiate normal activities from C2 ones. Because they learn continuously and therefore can adapt to emerging threats, deep learning models are well-suited to address the dynamics and changes in various C2 tactics (Catillo et al., 2023). The self-generated dataset was used for this research because deep learning in C2 detection hinges on the quality and representativeness of training data, the choice of features, and the architecture of the neural network.

This paper discusses an in-depth paradigm shift through different sets of explanations on efficient use of deep learning, real-time classification, and new framework architecture to detect camouflage command and control (C2) communication communication. This framework is built on deep learning from Multi-Layer Perceptron (MLP) using real-time classification in such a way that it seeks only to distinguish malicious C2 packets from legitimate ones. The MLP model was custom trained in a dataset generated from network traffic, while the behavioral-based real-time classification component uses SSL certificate patterns and outputs of Nmap scripts to identify Metasploit and Cobalt Strike frameworks' related wrongdoings. Integrating the two major techniques, this research focuses on providing a start-to-finish solution against advanced persistent threats with an elitist and highly scalable framework in detection and prevention of C2 communication.

This paper is structured as follows: Section 1 describes the introduction of C2 in the cyber world and the role of deep learning in detection C2 and real-time C2 framework classification. In Section 2, the literature for C2 is given. Section 3 explains the proposed detection and classification model. In Section 4, model evaluation is presented with a comparison to showcase the efficacy. Section 5 presents the self-generated dataset. In Section 6, the experimental analysis is given with graphs for results. Finally, the paper ends with Section 7 presenting the conclusion and future work.

Literature Review

The Command and Control or C2 system is among the modern digital attack strategies that empower the enemy to collaborate with malicious attacks over invaded networks and endpoints (Leal et al., 2019). Research works have continued to explore, understand, detect and mitigate against C2 communications, with considerable effort from the community towards understanding the increasing sophistication in C2 systems. This section brings together the important findings in the literature, from the history of C2 systems through fundamental network concepts dependent on their ways of operation, to upcoming innovations towards both unencrypted and encrypted C2 detection. The importance of machine and deep learning techniques in amplifying the role of real-time detection to define an effective defense in future against identifying malicious activities is emphasized.

Evolution of C2 Systems and Fundamental Networking Concepts

The ubiquitous early C2 systems were simple to detect and disrupt because of their reliance on simple protocols and fixed infrastructures; however, modern C2 frameworks utilize sophisticated techniques for evading detection like dynamic IP address allocation schemes, covert communication channels, and domain generation algorithms (DGAs) (Gomes et al., 2024). Such instruments and techniques are playing effective roles in hindering detection by masking malicious traffic within the legitimate traffic flows while using encryption to conceal the danger traffic content. To understand how C2 mechanisms work in real-life environments, it is natural to understand certain core concepts in networking. The OSI and TCP/IP models suggest a multi-layered approach, but the interest lies mainly in the Network (IP) and Transport (TCP/UDP) layers for C2 detection. While IP addresses can be likened to mailing addresses ensuring a unique identification of the endpoint, the transfer of data by either TCP or UDP implies reliable and connectionless, respectively (Al-Hisnawi & Ahmadi, 2016; Kuerbis & Mueller, 2020). Enabling packet header, payload, and signature thereby more sophisticated detection methods, Deep Packet Inspection (DPI) represents some of the basic detection methods truly at work.

Unencrypted C2 Detection

Traditionally, research focusing on infected C2 detection in unencrypted formats has concentrated mainly on the application-layer protocols such as HTTP. These security tools can also identify using source and destination ports, signatures of traffic, as well as content patterns, malicious flows that may use HTTP(S) as a cover channel. For instance, an attacker selects HTTP GET or POST requests for issuing commands, making them easy to detect with pattern-based detection. Some examples are BitProb, Bodhish and Hubballi et al. (2020), which use probabilistic bit signatures to classify flows of traffic. In addition, there are also many port-based techniques, DPI, as well as machine learning algorithms, which have been extensively adopted for application-based analysis. They employ anti-spyware signatures in the NGFWs to identify dynamic C2 operations that employed real-time changing IP addresses and looked legitimate domains (Salat et al., 2023). The NIDS employs a combination of anomaly detection, misuse detection, and specification-based analysis, in order to provide protection against known threats. Still, they face false positives and static signatures problems (Ghorbani et al., 2009).

Significant research has explored how adversaries blend C2 communications with benign HTTP flows Al-Hakimi and Bax, 2020), prompting passive identification methods based on machine learning and feature extraction. Observing subtle abnormalities in HTTP header order, size distributions, and frequency has proven useful for distinguishing malicious from legitimate traffic, but these methods do not readily extend to encrypted scenarios.

Encrypted C2 Detection

The transition to TLS channels has increased much more the complexity in the detection of C2 activities already. Signature-based methods as well as DPI as approaches that are effective for plaintext traffic lose complete visibility with encrypted payloads. Investigators have shifted their focus then to the analysis of TLS metadata such as certificate particulars, key lengths, and anomalies in handshakes to expose potentially malicious connections (Anderson et al., 2018). Beyond the need for TLS inspection are flow-based techniques statistically based on packet size, inter-arrival time, and number of retransmissions that have already been shown to be important in identifying suspicious talking patterns even in adversarial environments (Novo & Morla, 2020). SSL-decryption approaches at network perimeters will restore visibility of contents but create huge overheads in resources and infringe on user privacy (Baldini et al., 2020; Rajasoundaran et al., 2024). Scarce still are blacklisting suspicious IPs or domains, while adversaries keep changing infrastructures or DGAs stay ahead of static lists (Salat et al., 2023).

Behavior-Based Detection Using Machine and Deep Learning

However, many researchers have gone to the extent of using behavior-based detection approaches, particularly machine and deep learning, to solve the problems associated with static and signature-based systems. By learning a model from traces of all benign and malicious traffic on a monitored network, the systems will then learn a way to detect subtle characteristics of C2 communications (Shao et al., 2021; Shafiq et al., 2020; Tuan et al., 2020). The typical machine learning methods-Support Vector Machines, Decision Trees, Random Forests-have shown remarkable detection rates on well-known data sets like KDD'99 and N-BaIoT. In this sense, deep learning methods somewhat enhance the techniques for detection by automatically extracting and learning complex representation of features. CNNs and LSTMs have been used to boost detection rates for botnet detection with their performance being over 94% (Parra et al. 2020). Anomalies can be detected by deep-autoencoders, which contain normal traffic patterns, at an accuracy of 99.7% (Apostol et al., 2021). All of these methods enable continuous learning, so they can be updated with new or evolved C2 tactics. Recently, several hybrid models combined rulebased heuristics with machine learning algorithms. As an integration of complexity of domain knowledge with data-driven insights, Vidhun and Kannimoola (2024) introduced a Random Forest-based behavioral filtering method for proactive C-cubed defense against threat options.

Real-Time Detection

Timely responses are the key, although post-hoc analyses prove to be successful in understanding the nature of the C2-driven attack. ElasticNet Regression Models (ENetRM) have shown that computations with low computational overhead could result in builds that accurately distinguish malicious and benign traffic along with high precision, recall, and F1 scores in real-time settings (Srinivasan & Deepalakshmi, 2023; Hussain et al.,2024). Similarly, BotDet Ghafir et al. (2018) also align the modular detection technology dealing with partial visibility and encryption and achieve a particular 82.3% true positive rate of 13.6%. Yet another burgeoning paradigm is formed by the self-supervised or semi-supervised that keep on adjusting in almost real-time. For example, Self-Supervised Intrusion Detection (SSID) frameworks attained detection accuracy at 99.83%, which is adaptable to shifting adversary behavior (Nakıp & Gelenbe, 2023). Much as they promise in terms of responses like real-time systems, they still suffer from all sorts of drawbacks, including those related to encryption, the computational expense, and false positives.

Summary of Key Findings

In general, the literature emphasizes the intricacy of C2 detection within modern networking environments. Indeed, while traffic analysis on the unencrypted plane and DPI techniques still work well, a shifting trend toward TLS-encrypted channels begs for backward methods to focus on metadata, network flow statistics, and behavioral indicators. Machine and deep learning models have greatly improved the detection rates against various scenarios, with a strong support of domain knowledge and feature engineering or selection. Nevertheless, real-time detection remains one major hurdle. A trade-off must be maintained between very high detection accuracy and latency, computation overhead, and the rate of false outside alarms. Therefore, the present study tries to corroborate these findings by putting forward the idea of combining deep learning models with behavioral analytics for on-the-fly detection and classification of malicious C2 frameworks toward establishing a more robust and adaptive scheme for network defense.

Materials and Methods

The proposed method will consist of a deep-learning detection model and a real-time classification framework. The detection model examines network traffic to detect normal and malicious packets. Once traffic has been identified as malicious, it will then be classified for further analysis. The methodology for detecting and classifying Command-and-Control (C2) communication using deep learning involves a few steps and is illustrated in Figure 1:

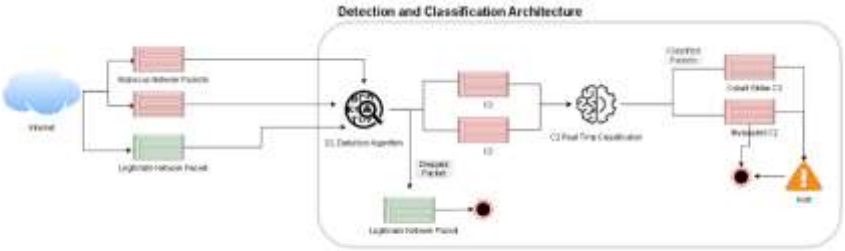
Detection and Classification Architecture
Internet
Malicious Network Packets
Legitimate Network Packet
DL Detection Algorithm
Dropped Packet
Legitimate Network Packet
C2
C2
C2 Real-Time Classification
Classified Packets
Cobalt Strike C2
Metasploit C2
Alert

Figure 1. Proposed Detection and Classification Architecture

Deep Learning Detection

The deep learning architecture that has been used in this research is Multi-Layer Perceptron (MLP). Speaking within the framework of Command and Control (C2) detection, MLP model trains itself upon many network traffic packets such that each packet has to be classified either as legitimate or that of a C2 malicious packet. A preliminary stage where cleanliness is assured of the data is done, whereby an integrity check of the dataset is performed mostly by identifying missing values showcased in the DataFrame. The steps are summarized in the following action:

Feature Extraction and Labeling: Relevant features for the study, such as ’port,’ ’Country_name,’ ’asn,’ ’isp,’ and ’organization,’ are selected from the dataset. Corresponding labels, called ’ labels,’ are then identified, providing the foundation for supervised learning. These labels indicate whether the network activity is benign or malicious. By extracting these labels, we set the stage for the supervised learning task, where the model will be trained to predict the labels of new, unseen data based on the patterns it has learned from the labeled training data.

Data Splitting: The dataset is divided into training and testing subsets using the train_test_split function. A test size of 20% is specified, with a fixed random state (42) to ensure reproducibility.

Preprocessing: The dataset contains both numerical and categorical features. Specifically, ’port’ is treated as a numerical feature, while ’Country_name,’ ’asn,’ ’isp,’ and ’organization’ are categorical features. Preprocessing steps are applied to these features using specialized pipelines: numerical features are scaled using StandardScaler, and categorical features are encoded using OneHotEncoder. These transformations are integrated into a ColumnTransformer to streamline their application.

Model Definition: The MLP model is defined using the MLPClassifier, with a single hidden layer consisting of 100 neurons. The model is configured to run for a maximum of 1,000 iterations, with a fixed random state (42) to maintain consistency across runs.

Pipeline Construction: A complete pipeline is constructed using the Pipeline class, which incorporates both the preprocessing steps and the MLP classifier. This modular design ensures the consistency and reproducibility of the workflow, as the same sequence of operations is applied during both training and testing phases.

Model Training: The pipeline is trained on the training dataset using the *fit* method. The preprocessing steps are applied during training, followed by fitting the MLP classifier to the processed data.

Prediction: The trained model is then used to predict the previously unseen test dataset.

The performance of the model is evaluated using a range of metrics, including accuracy, precision, recall, and F1-score for each class. These metrics comprehensively assess the model’s effectiveness in distinguishing between legitimate and malicious network traffic.

Real-Time Classification

This methodology follows a behavioral analysis approach by interpreting the behaviors shown by SSL certificates and Nmap script outputs. The basis of this approach is that diverse patterns are usually demonstrated by different Command and Control (C2) frameworks, and behavior recognition is therefore a useful mechanism for accurate classification. The classification approach described in the paper focuses on the detection of C2 frameworks using distinct patterns from two popular ones, namely, Metasploit and Cobalt Strike. This is done through the means of SSL certificate inspection and Nmap script execution that serve to derive features indicative of each of the frameworks. The analysis begins with the observation of the SSL certificate for a connection in question. The main interest is put into the Common Name (CN) field where anything like “MetasploitSelfSignedCA” would definitely point toward the

Metasploit framework. This fine-grained examination of SSL certificates increases classification accuracy through the identification of Metasploit C2 instances based on SSL signature attributes. If the SSL certificate does not demonstrate a characteristic specific to Metasploit, the methodology conducts a further inspection using the Nmap tool. A notable approach within this methodology involves running the grab_beacon_config_old script against the intended target IP address and port, with the parsing of any returned output. Indicators such as "BeaconType:" being present in the output are taken note of. This keyword-based view emphasizes built-in configurations that characterize Cobalt Strike C2 instances. If none of the SSL Certificate or Nmap Script Output matches any of the Metasploit or Cobalt Strike pattern signatures, the investigated packet would be considered as "others". Regardless, it should act as a safety net, covering all other possibilities and avoiding false positives. This classification methodology embraces both the analysis of SSL certificates and Nmap scripts, enabling it to recognize distinct C2 frameworks. It effectively differentiates Metasploit from Cobalt Strike by means of behavioral characteristics and configurations that are peculiar to each, thereby including any frameworks unidentified hitherto. The more structured nature that this approach uses enhances its precision for real-time classification adaptation.

For example, if the Nmap script output has not been previously grouped, the packet is classified into "other frameworks" and flagged first, before allowing for the more detailed manual examination done through Metasploit or Cobalt Strike signature matching, focusing on the groups where most of the possible matches have already been found.

Model Evaluation

The proposed deep-learning model is evaluated, this time very stringently, using the N-BaIoT dataset as its benchmark. This dataset was developed especially for evaluating IDS in an IoT environment with a complete collection of network activities. The results are tediously summarized in Table 1: greatly boosting credibility and respect for the suggested method compared to the results obtained by Castillo et al. (2023). The dataset from N-BaIoT subsequently serves as a quality basis on which to assess the detection capabilities of the proposed model. As practical performance measures for comparing models, namely, those proposed in this paper and those seen by Castillo et al., are values for F1 score, recall, and precision. These performance metrics exhibit the underscore for which a model could turn positive; false positives get minimized in their own right, thus contributing to the complete evaluation of the model's predictive efficacy and trustworthiness. Table 1 provides comparisons of performance results across different devices of the N-BaIoT dataset. The proposed model demonstrates an even better performance in every device, showing almost 100% for F1 scores, recall, and precision. These results prove the model's robustness and dependability in differentiating malicious traffic from benign activities. Recall scores were considerably high, which highlights the ability of the model to find true positives effectively; the precision score also indicated there are very few false positives.

Device No	**F1**	**Recall**	**Precision**	**F1**	**Recall**	**Precision**
1	1.0000	1.0000	1.0000	0.8861	0.7955	1.0000
2	0.9976	0.9955	0.9998	0.5192	0.3506	1.0000
3	1.0000	1.0000	1.0000	0.8616	0.7568	0.9999
4	1.0000	1.0000	1.0000	0.8796	0.7851	0.9999

5	1.0000	1.0000	1.0000	0.8423	0.7276	1.0000
6	1.0000	1.0000	1.0000	0.8486	0.7371	0.9999
7	1.0000	1.0000	1.0000	0.8621	0.7577	0.9999
8	1.0000	1.0000	1.0000	0.5239	0.3549	1.0000
9	0.9995	0.9993	0.9997	0.8661	0.7639	1.0000

Table 1. Models' Performance Comparison Proposed Model Baldini et al. (2020) 's Model

The model proposed by Catillo et al. (2023) performs with less performance variability. The F1 scores, recall, and precision values of its performance demonstrated large variation across the devices. In some cases, specifically in terms of F1 score and recall values considerably lower than those of the proposed one, Catillo et al. (2023) 's model performed worse. For instance, it was demonstrated that Device 2 had an F1 score of 0.5192 and an associated recall of 0.3506 under the scrutiny of Catillo et al. (2023) compared to an F1 score of 0.9976 and a recall of 0.9955 under the proposed approach. This downward trend was observed in other devices as well, showing that Catillo et al. (2023) had difficulty providing consistent performance. The consistently near-perfect scores achieved by the proposed model in all metrics strongly suggest a good trade-off between precision and recall, critical to reducing false alarms and enhancing detection. These findings suggest that the model can be applied in real-world IoT environments, where reliable detection of malicious activities is requisite for ensuring security and operational stability. The results suggest that the proposed model is significantly superior to that of Catillo et al. (2023) in terms of predictive accuracy and reliability overall. This improvement in performance signifies the approach's capability to address some of the peculiarities of IoT security. This makes a strong case for its deployment in real-time detection systems.

Dataset

The data set of interest for this research was harvested from Shodan, a search engine for internet-connected devices. While Shodan offers a large-scale repository of publicly accessible network servers and services, it does act as an almost irreplaceable resource for conducting research in various facets of cybersecurity. Data collection from the Shodan platform was carried out programmatically in Python via the Shodan API, providing a handy interface for interfacing with the search engine. Recognition of Command and Control (C2) servers in Shodan had to dig deep into certain malware networking behaviors and the specific distinguishing characteristics of C2 servers themselves. Searching for these was centered on two common frameworks: Cobalt Strike and Metasploit.

Cobalt Strike Identification

Cobalt Strike is a legitimate penetration testing tool frequently repurposed by threat actors to establish C2 communications. Several techniques were employed to detect Cobalt Strike servers within Shodan's dataset:

- **Product Name Match**: Queries were constructed based on the product name, Cobalt Strike Beacon.
- **SSL Certificate Serial Number**: Searches targeted SSL certificates with a specific serial number, such as 146473198.
- **Hash Matching**: Queries filtered results based on known hashes associated with Cobalt Strike, focusing on port 50050.

- **JARM Fingerprinting**: SSL JARM fingerprints were utilized to identify distinct server profiles.

- **Common Name Analysis**: SSL certificates containing the common name foren. Zik was flagged as a potential Cobalt Strike instance.

Metasploit Identification

Metasploit, another prominent penetration testing framework, was similarly scrutinized to identify C2 infrastructure. The search methodology targeted specific SSL certificate characteristics:

- **SSL String Matching**: Searches identified certificates containing the string Metasploit-SelfSignedCA.

Dataset Composition

The resulting dataset comprises a substantial collection of 682,864 network records from Shodan. These records encapsulate diverse features essential for analyzing and characterizing network traffic. Key features included in the dataset are:

- IP addresses.
- Ports.
- Country names.
- Autonomous System Numbers (ASN).
- Internet Service Providers (ISPs).
- Organizational information.

The dataset encompasses various server types: Command and Control (C2), FTP, SSH, Telnet, Mail, DNS, Web, VPN, Windows, and Linux. This diversity ensures a comprehensive representation of network environments, aiding in developing and evaluating deep learning models for real-time detection of malicious C2 communications.

Results

An experiment was designed to test the deep learning model's effectiveness for real-time detection of C2 communications. The tests were conducted under two popular C2 frameworks - Cobalt Strike and Metasploit. Model performance was assessed based on various metrics such as precision (P), recall (R), accuracy, F1-score, among others, which elaborated categorically on the model's classification capacity. True positives (TP) and true negatives (TN) exemplify the instances that have been correctly classified while misclassification errors consist of false positives (FP) and false negatives (FN). Recall more precisely predicts the value of TP over the total actual positives (TP + FN), whereas precision is given as the value of TP over total predicted positives (TP + FP). The false positive rate (FPR) represents the last important measured variable indicating the amount of identified false positives versus the total number of instances being negative in the corpus being tested. The F1 score, a balanced metric, is the harmonic mean of both precision and recall. These metrics are derived through the following formulas:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1\ score = 2\ \text{x} \frac{PxR}{P + R}$$

The results for C2 detection tasks are summarized in Table 2. The model demonstrated high performance across all evaluation metrics. For detecting non-C2 activities (Class 0), the model achieved a precision of 0.95, a recall of 0.95, and an F1-score of 0.95. C2-related activities (Class 1) exhibited even stronger performance, with a precision of 0.99, a recall of 1.00, and an F1-score of 1.00. The overall accuracy of the model was measured at 99.15%.

Metric	**Non-C2 (Class 0)**	**C2 (Class 1)**	**Macro Avg**	**Weighted Avg**
Precision	0.95	0.99	0.97	0.99
Recall	0.95	1.00	0.97	0.99
F1-Score	0.95	1.00	0.97	0.99
Accuracy	0.9915			

Table 2. Proposed Model's Result

An analysis of detected C2 activities revealed that 61.8% of instances were associated with the Cobalt Strike framework, highlighting its significant presence as a potential threat. Furthermore, 12.7% of the instances corresponded to activities linked with the Metasploit framework. Interestingly, 25.5% of the detected instances were classified under "Other C2 Frameworks," indicating a diverse range of network behaviors outside the scope of the defined Cobalt Strike and Metasploit categories. This distribution is illustrated in Figure 2.

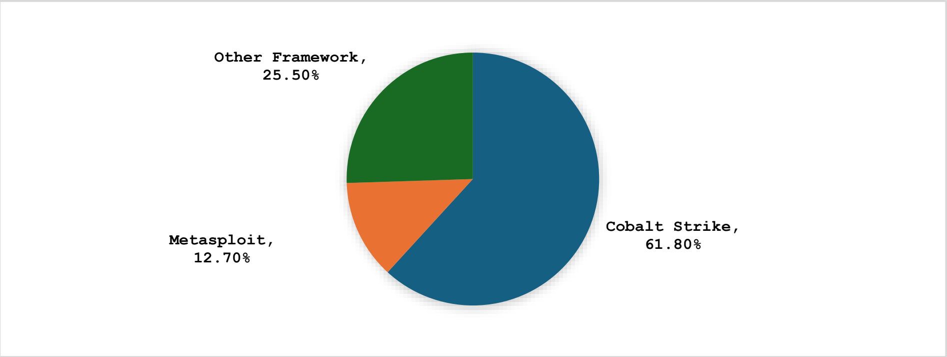

Figure 2. Classification Result

The observed high performance of the model in terms of precision, recall, and F1-score underscores its capability for effective real-time detection of C2 communications. Additionally, detecting diverse C2 frameworks suggests the model's robustness in identifying emerging threats, making it well-suited for deployment in dynamic and evolving cybersecurity environments.

Discussion

At 99.15% overall accuracy, the exception accrual to the detection model suggests it recorded impressive effectiveness beyond detection of C2 communications, yet the algorithm lacks the ability to flag suspicious C2 communications.. Precision values recorded were extremely high, with 95% for non-C2 (class 0) and 99% for C2 (class 1) instances. The prediction such that a model predicts non-C2 or C2 is right about 95% and 99% of the times, respectively. Such precision emphasizes the robustness of the model against false positives which is very important in cybersecurity applications since false alarms can bring inefficiencies and large costs. Moreover, the recall metrics indicated an outstanding performance by the model - especially in C2 (class 1), where it achieved 100% recall, signifying it correctly identified all instances. This means that the model detected every C2 instance without overlooking any, covering malicious activities comprehensively. The model was also able to exclude benign traffic correctly as for non-C2 instances; the recall value was also high. The F1 score, which is a harmonic mean of precision and recall, has also emphasized much on the model's effectiveness with 95% for non-C2 and absolutely perfect 100% for C2 (see Figure 3). On a collective scale, these metrics indicate the model's capability to strike a balance between precision and recall.

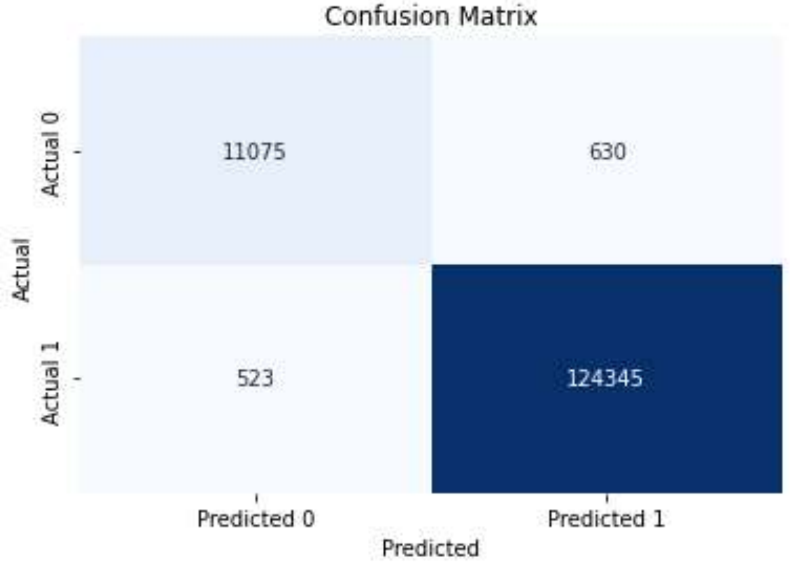

Figure 3. Result Confusion Matrix

Further evaluation of the model performance was made on the ROC curve, revealing its position near the upper-left corner, which indicates greater sensitivity and much fewer false positives. An AUC value of 0.99 shows that the model was exceptionally able to separate cases of positives (C2) and negatives (non-C2) (see Figure 4). This almost perfect AUC value further strengthens the model's viability for usage in real time where correct classification is very important. The results further reiterate the ability of the model to predict both classes accurately even when faced with inherent imbalanced class distributions. The macro and weighted averages consistently validated the overall performance and highlighted balanced model capabilities across a wide range of evaluation metrics. The capability of the model in a powerful fashion was also observed in the pinpointing of specific instances concerning various C2 frameworks, especially Cobalt Strike and Metasploit. As a whole, these frameworks find application in APTs and red teaming, which makes their detections critical to proactively defending. Strong precision, recall, and accuracy values mean that the model provides security teams with a reliable instrument for real-time monitoring and offensive operations mitigation, accomplishing the detection and classification of malicious activity concerning the C2 communication with maximum efficiency.

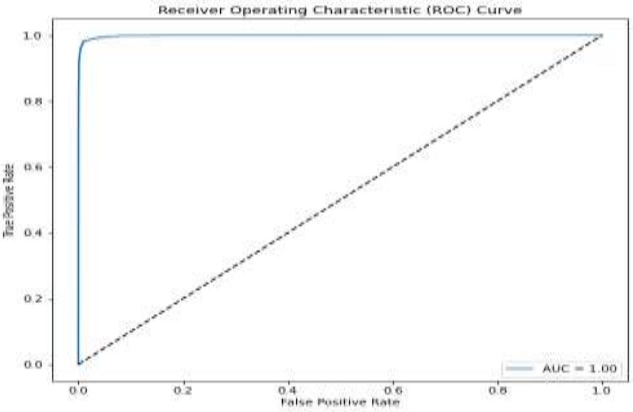

Figure 4. Receiver Operating Characteristic (ROC) Curve

The results presented in this study demonstrate the practical viability of leveraging deep learning models for real-time detection of C2 communications. Future work will enhance the model's interpretability, enabling analysts to gain insights into the decision-making process and improving resilience against adversarial evasion techniques.

Conclusion

The research presented here deals with the current problems related to Command and Control (C2) communications in modern cybersecurity, thereby requiring improved detection methods to meet increasingly sophisticated cyber challenges. The traditional signature-based methods that were previously effective have now become inadequately supportive against polymorphic malware and encrypted communications. This research thus proposes a new framework that integrates deep learning with real-time classification mechanisms to improve threat detection capabilities.

The Multi-Layer Perceptron (MLP)-based deep learning model that is proposed proves to be a high performer, recording 99% accuracy in identifying malicious C2 packets floating within network traffic. Another area in which the real-time classification mechanism found in this framework excels is in classifying patterns associated with top C2 frameworks, Cobalt Strike, and Metasploit, with accuracy scores as impressive as 99.9%. Such results prove the usefulness of applying deep learning techniques along with behavioral analysis to meet and handle the new evolving evasion strategies that are used by modern malware.

The key aspects of this study are custom dataset creation, thorough model performance evaluation, and validations based on real-world traffic simulation. It is a dynamic and scalable framework that detects malicious activity in real time as it focuses on SSL certificate patterns, extracting data using outputs from Nmap script. Such advances have been made in the transformation of static methods of detecting into proactive defensive capabilities, which stand to be far much better than the conventional static detection models. The findings presented in this paper demonstrate the need for future research in machine learning-based data-driven approaches in cybersecurity. However, as the threat landscape is continuously changing, ongoing research is needed to refine detection models and improve their abilities. Future work should also continue to explore other machine learning architectures, integrate anomaly detection techniques, and extend the dataset scope to emerging C2 frameworks and tactics. This research has made a major contribution toward building resilient and robust systems for cybersecurity. Bridging deep learning and real-time threat analysis will make strides in building scalable, intelligent, and proactive defense mechanisms. Collaborative efforts within the cybersecurity community will be important to sustain progress and address future challenges, ensuring robust protection against increasingly complex cyberattacks.

Author Contributions: Conceptualization, methodology, validation, formal analysis, investigation, and visualization, N.J., M.E., and S.M.F. Software and writing—original draft preparation, N.J. Writing review and editing, M.E., S.M.F. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) for this publication.

References

Akhtar, M. S., & Feng, T. (2022). Malware analysis and detection using machine learning algorithms. Symmetry, 14, 2304.

Al-Hakimi, S., & Bax, F. (2020). Hunting for malicious infrastructure using big data. SNE Master Research Projects 2020, 2021.

Al-Hisnawi, M., & Ahmadi, M. (2016). Deep packet inspection using a quotient filter. IEEE Communications Letters, 20, 2217–2220.

Anderson, B., Paul, S., & McGrew, D. (2018). Deciphering malware's use of TLS (without decryption). Journal of Computer Virology and Hacking Techniques, 14, 195–211.

Apostol, I., Preda, M., Nila, C., & Bica, I. (2021). IoT botnet anomaly detection using unsupervised deep learning. Electronics, 10, 1876.

Ashfaq, T., Khalid, R., Yahaya, A. S., Aslam, S., Azar, A. T., Alkhalifah, T., & Tounsi, M. (2022). An intelligent automated system for detecting malicious vehicles in intelligent transportation systems. Sensors, 22(17), 6318.

Aslan, Ö. A., & Samet, R. (2020). A comprehensive review of malware detection approaches. IEEE Access, 8, 6249–6271.

Azab, A., Khasawneh, M., Alrabaee, S., Choo, K. K. R., & Sarsour, M. (2024). Network traffic classification: Techniques, datasets, and challenges. Digital Communications and Networks, 10, 676–692.

Baldini, G., Hernandez-Ramos, J. L., Nowak, S., Neisse, R., & Nowak, M. (2020). Mitigation of privacy threats due to encrypted traffic analysis through a policy-based framework and MUD profiles. Symmetry, 12, 1576.

Bastiaansen, H., van der Geest, J., van den Broek, C., Kudla, T., Isenor, A., Webb, S., ... & Masini, A.

(2020). Federated control of distributed multi-partner cloud resources for adaptive C2 in disadvantaged networks. IEEE Communications Magazine, 58, 21–27.

Catillo, M., Pecchia, A., & Villano, U. (2023). A deep learning method for lightweight and cross-device IoT botnet detection. Applied Sciences, 13, 837.

Caviglione, L., Choraș, M., Corona, I., Janicki, A., Mazurczyk, W., Pawlicki, M., & Wasielewska, K. (2020). Tight arms race: Overview of current malware threats and trends in their detection. IEEE Access, 9, 5371–5396.

De Gaspari, F., Hitaj, D., Pagnotta, G., De Carli, L., & Mancini, L. V. (2022). Evading behavioral classifiers: A comprehensive analysis on evading ransomware detection techniques. Neural Computing and Applications, 34, 12077–12096.

Eisenberg, D. A., Alderson, D. L., Kitsak, M., Ganin, A., & Linkov, I. (2018). Network foundation for command and control (C2) systems: Literature review. IEEE Access, 6, 68782–68794.

Ferdous, J., Islam, R., Mahboubi, A., & Islam, M. Z. (2023). A state-of-the-art review of malware attack trends and defense mechanisms. IEEE Access.

Geng, J., Wang, J., Fang, Z., Zhou, Y., Wu, D., & Ge, W. (2024). A survey of strategy-driven evasion methods for PE malware: Transformation, concealment, and attack. Computers & Security, 137, 103595.

Ghafir, I., Prenosil, V., Hammoudeh, M., Baker, T., Jabbar, S., Khalid, S., & Jaf, S. (2018). Botdet: A system for real-time botnet command and control traffic detection. IEEE Access, 6, 38947–38958.

Ghorbani, A. A., Lu, W., & Tavallaee, M. (2009). Network intrusion detection and prevention: Concepts and techniques (Vol. 47). Springer Science & Business Media.

Gomes, J. E. C., Ehlert, R. R., Boesche, R. M., Santosde Lima, V., Stocchero, J. M., Barone, D. A., ... & de Araujo Fernandes, R. Q. (2024). Surveying emerging network approaches for military command and control systems. ACM Computing Surveys, 56, 1–38.

Hubballi, N., Swarnkar, M., & Conti, M. (2020). BitProb: Probabilistic bit signatures for accurate application identification. IEEE Transactions on Network and Service Management, 17, 1730–1741.

Hussain, S., He, J., Zhu, N., Mughal, F. R., Hussain, M. I., Algarni, A. D., ... & Ateya, A. A. (2024). An adaptive intrusion detection system for WSN using reinforcement learning and deep classification. Arabian Journal for Science and Engineering, 1-15.

Khan, A. Y., Latif, R., Latif, S., Tahir, S., Batool, G., & Saba, T. (2019). Malicious insider attack detection in IoTs using data analytics. IEEE Access, 8, 11743-11753.

Kuerbis, B., & Mueller, M. (2020). The hidden standards war: Economic factors affecting IPv6 deployment. Digital Policy, Regulation and Governance, 22, 333–361.

Leal, G. M., Zacarias, I., Stocchero, J. M., & De Freitas, E. P. (2019). Empowering command and control through information-centric networking and software-defined networking. IEEE Communications Magazine, 57, 48–55.

Malik, M. I., Ibrahim, A., Hannay, P., & Sikos, L. F. (2023). Developing resilient cyber-physical systems: A review of state-of-the-art malware detection approaches, gaps, and future directions. Computers, 12, 79.

Nakıp, M., & Gelenbe, E. (2023). Online self-supervised learning in machine learning intrusion detection for the Internet of Things. arXiv preprint arXiv:2306.13030.

Novo, C., & Morla, R. (2020). Flow-based detection and proxy-based evasion of encrypted malware C2 traffic. In Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security (pp. 83–91).

Obi, O. C., Akagha, O. V., Dawodu, S. O., Anyanwu, A. C., Onwusinkwue, S., & Ahmad, I. A. I. (2024). Comprehensive review on cybersecurity: Modern threats and advanced defense strategies. Computer Science & IT Research Journal, 5, 293–310.

Parra, G. D. L. T., Rad, P., Choo, K. K. R., & Beebe, N. (2020). Detecting Internet of Things attacks using distributed deep learning. Journal of Network and Computer Applications, 163, 102662.

Rajasoundaran, S., Sivakumar, S., Devaraju, S., Pasha, M. J., & Lloret, J. (2024). A deep experimental analysis of energy-proficient firewall policies and security practices for resource-limited wireless networks. Security and Privacy, e450.

Salat, L., Davis, M., & Khan, N. (2023). DNS tunnelling, exfiltration and detection over cloud environments. Sensors, 23, 2760.

Shafiq, M., Tian, Z., Bashir, A. K., Du, X., & Guizani, M. (2020). CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. IEEE Internet of Things Journal, 8, 3242–3254.

Shah, S. (2004). An introduction to HTTP fingerprinting. Net-Square Solutions, 1–21.

Shao, Z., Yuan, S., & Wang, Y. (2021). Adaptive online learning for IoT botnet detection. Information Sciences, 574, 84–95.

Shi, W. C., & Sun, H. M. (2020). DeepBot: A time-based botnet detection with deep learning. Soft Computing, 24, 16605–16616.

Srinivasan, S., & Deepalakshmi, P. (2023). ENetRM: ElasticNet regression model-based malicious cyber-attacks prediction in a real-time server. Measurement: Sensors, 25, 100654.

Torres, M., Álvarez, R., & Cazorla, M. (2023). A malware detection approach based on feature engineering and behavior analysis. IEEE Access.

Tuan, T. A., Long, H. V., Son, L. H., Kumar, R., Priyadarshini, I., & Son, N. T. K. (2020). Performance evaluation of Botnet DDoS attack detection using machine learning. Evolutionary Intelligence, 13, 283–294.

Tzagkarakis, C., Petroulakis, N., & Ioannidis, S. (2019). Botnet attack detection at the IoT edge based on sparse representation. In Proceedings of the 2019 Global IoT Summit (GIoTS) (pp. 1–6). IEEE.

Vidhun, K., & Kannimoola, J. M. (2024). Guarding against command and control (C2) agents utilizing real-world applications for communication channels. In Proceedings of the 2024 5th International Conference for Emerging Technology (INCET) (pp. 1–5). IEEE.

Yu, Y., Yan, H., Ma, Y., Zhou, H., & Guan, H. (2020). DeepHTTP: Anomalous HTTP traffic detection and malicious pattern mining based on deep learning. In Proceedings of the Cyber Security: 17th China Annual Conference, CNCERT 2020 (pp. 141–161). Springer.

Zaid, T., & Garai, S. (2024). Emerging trends in cybersecurity: A holistic view on current threats, assessing solutions, and pioneering new frontiers. Blockchain in Healthcare Today, 7.

Zeadally, S., Adi, E., Baig, Z., & Khan, I. A. (2020). Harnessing artificial intelligence capabilities to improve cybersecurity. IEEE Access, 5408, 23817–23837.