Feature extraction for anomaly detection in industrial control systems

Silvio Russo[1,*], Claudio Zanasi[1,*] and Isabella Marasco[1,*]

[1]*Department of Computer Science and Engineering, University of Bologna, Italy*

Abstract

The threat landscape for industrial systems is in rapid evolution, with cyber-attacks becoming increasingly sophisticated, targeted, and motivated. This situation should raise many concerns because of the growing interconnection of industrial control systems with the Internet, as well as the proliferation of cyber-physical systems and the Industrial Internet of Things. In these scenarios, an accurate detection of attacks is of utmost importance. The swiftness with which the environment of security risks in IoT and industrial systems is a cause for concern, given the rising complexity, specificity, and determination of cyber-attacks. This issue becomes particularly problematic due to the expanding integration of industrial control systems with the Internet and the widespread adoption of cyber-physical systems. In this work, we introduce a novel methodology for improving the Feature Extraction process. The solution shows versatility, operating not only as a standalone tool for identifying network attacks but, more significantly, as a valuable tool for pre-processing raw packet data tailored for integration with artificial intelligence models.

The proposed solution was developed with an emphasis on addressing the specific cybersecurity needs of the industrial sector. This approach is driven by the imperative requirements of the industrial landscape, where safeguarding critical systems against cyber threats is of paramount importance. Furthermore, our system was tested on an industrial dataset that demonstrates the applicability and efficacy of our solution within the peculiar context of industrial environments. The outcomes of these tests contribute to the validation of our approach.

Keywords

Anomaly Detection, Control Systems, Feature Extraction

1. Introduction

The analysis of network traffic through machine learning models is becoming necessary for the early detection of suspicious activities. Cyber-physical systems (CPS) constitute a network of interconnected devices facilitating seamless information exchange among tangible IoT devices. These types of devices are used in different contexts like medical devices, autonomous vehicles, industrial automatons, wearable, and urban smart infrastructures, and can be remotely configured and managed. With their progressive adoption, these devices can monitor and access huge amounts of critical and even sensitive data. Moreover, the proliferation of these devices in the industries expands the surface area for cyber-attacks with consequent higher risks of data leak, ransomware and sabotage operations. In certain industrial contexts and critical infrastructures, the severity of cyber threats increases due to the potential consequences of an

ITASEC 2024: The Italian Conference on CyberSecurity, April 08–12, 2024, Salerno, IT

*Corresponding author.

silvio.russo3@unibo.it (S. Russo); claudio.zanasi4@unibo.it (C. Zanasi); isabella.marasco4@unibo.it (I. Marasco)

attack on Industrial Control Systems (ICS) [1] and the challenges associated with implementing modern security measures for these systems.

Current cybersecurity research and products primarily focus on methods and techniques for the information technology (IT) environment. However, the increasing integration of industrial control systems with IT systems, the proliferation of CPS, Industrial Internet of Things (IIoT) solutions, and the rising frequency of attacks on industrial systems necessitate a shift in research objectives towards addressing the unique challenges emerging from industrial settings.

Common datasets used for evaluating network intrusion systems (NIDS), such as KDDCUPP99 [2] and NSL-KDD [3], are old and not applicable for industrial systems. For this reason, we consider the recent CIC Modbus 2023 dataset [4] and use it for the extraction of various features, including Flow Features, Basic Features, Content Features, Time Features, and Labelled Features. This comprehensive set allows a solid foundation for network traffic analysis specifically tailored for industrial environment. The feature extraction rules, implemented in the Bro-code scripting language, represent the core of this paper and a necessary step for the effective implementation of a classification neural network that takes a pcap file as its input and provides immediate evaluation of malicious or legitimate industrial traffic. Network Intrusion Detection Systems (NIDS) based on AI models assume a pivotal role in safeguarding network infrastructure [5]. In this context, the Feature Extractor (FE) components stand out as fundamental elements that contribute to the overall efficacy of the detection phase. The Feature Extractor serves as a critical component that is responsible for extracting relevant features and patterns from network traffic data. Its role is to analyze the incoming data streams, identify anomalies, and extract key information that can be indicative of potential security threats or malicious activities. This process involves scrutinizing various aspects of the network traffic, including packet headers, payload contents, communication patterns, and other pertinent attributes.

The motivation behind this research stems from a critical issue observed in real-world scenarios: NIDS frequently generate an excessive number of false positives. This persistent problem significantly undermines the effectiveness of these systems in practical settings outside of controlled research environments. One of the primary causes we have identified is the poor quality of data that classification models receive as input, due to ineffective Feature Extraction processes. Therefore, with this research, we aim to take a first step towards a reliable and effective feature extraction process that can extract truly relevant information for detection purposes, thereby maximizing the efficacy of attack classification models without sacrificing system performance.

Differently from other solutions that can be found in literature mostly based on artificial intelligence (AI) [6] techniques, our solution is based on a deterministic feature extraction process able to guarantee the quality of the extracted data to improve the training process. The paper aims to provide a macroscopic view of the functioning of NIDS, with a particular focus on the Feature Extraction process, demonstrating how they can be customized and integrated by implementing a neural network to enhance their accuracy.

The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 describes the methodology and the proposed solution. Section 4 discusses the used dataset and the identified attacks. Section 5 presents details about the prototype and the experimental tests. Section 6 summarizes the main conclusions and outlines future work.

2. Related Work

The problem of IoT[7] and industrial security [8] has become more relevant because these components are crucial for the security and safety of cyber-physical environments. Several works[9, 10] analyze the critical situation in which the cybersecurity of IoT/OT word, proposing new solutions. These works examine the challenges and vulnerabilities present in the evolving interplay between the IoT and OT. In response to the identified risks, researchers have put forth innovative solutions[11, 12] aimed at enhancing the overall security posture of this interconnected ecosystem.

The work [12] introduces a dynamic and adaptable security paradigm, aiming to redefine traditional security models by embracing the evolving nature of contemporary digital environments.

Several works have analyzed the problem of feature extraction from Network traffic, the main goal of these works is to determine the best solution to extract relevant information from traffic data to improve the capacity to identify attacks or anomalies. The adopted solutions are very different, leveraging a wide type of different techniques, from old-style script-based solutions to AI-based ones. In [13] the author presents and examines the UNSW-NB15 data set creation, comparing it with other datasets like e KDDCUP 99[2] and NSL-KDD[3]. To benchmark and compare the different datasets the authors propose a solution based on a FE composed of two different software, Zeek [14] and Argus[15]. The extracted features identify relevant information to have a comprehensive knowledge of the network behaviour, starting from this work we improve the feature extraction process by extracting a different set of features, as described in section3, and developing a set of scripts to automatic labelling process, also identifying specific types of attacks.

In [6] the authors evaluate and compare the efficacy of different Deep Learning (DL) models in identifying attack vectors against three Shallow Learning models: Deep Feed Forward, Convolution Neural Network, Recurrent Neural Network, Decision Trees, Logistic Regression, and Naive Bayes. This work analyzes three FE techniques that have been evaluated on different datasets, in particular, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Auto-encoder (AE) are investigated for their impact on three benchmark datasets: UNSW-NB15, ToN-IoT, and CSE-CIC-IDS2018.

As we have done in this work the authors leverage the output of the FE as input for the Machine Learning models trying to improve the performances of the NIDS, in particular for PCA and AE, several dimensions (1,2,3,4,5,10,20 and 30) are selected trying to find the optimal number. The problem is that the FE process based on these algorithms makes it difficult to determine the quality of the extracted data, and this has an impact on the performance of the classifier.

In our work, we emphasize the importance of a deterministic FE process to ensure the quality and reliability of extracted data. While PCA and AE in the cited paper attempt to optimize performance by varying the number of dimensions, our solution focuses on maximizing the efficacy of the FE process without compromising classifier performance. This approach addresses the challenge of determining the quality of extracted data, which directly impacts classifier performance.

Furthermore, our research takes into account the practical constraints of real-world environ-

ments, particularly the crucial aspect of execution time

Moreover, in a real environment, the execution time is crucial to have an effective solution to detect and prevent cyber-attacks. In this sense, the suggested FE algorithm requires significantly more resources than the solution presented in our work.

3. Feature extraction

An important part of this work is the FE designed to work with industrial data to build quality datasets that can be used to train machine learning model to detect anomalous activities. Today most papers use artificial networks of different types (AE, PCA, RNN ecc...) to extract relevant features to improve the efficiency and the effectiveness of other machine learning models that play the role of NIDS. This approach introduces a significant computational complexity that is acceptable in research but is useless in real contexts where NIDS must analyze huge numbers of packets per second.

The proposed solution leverages Zeek [14] to implement an effective and fast FE. It is an open-source network security monitoring tool developed by researchers at Lawrence Berkeley National Laboratory. It is able of actively capturing, indexing, and analyzing real-time network traffic. Its capabilities include the passive monitoring of network traffic to extract information about protocols, connections, and data transfer. It can identify various protocols like HTTP, DNS, FTP, and others, showcasing a focus on behavioural analysis for anomaly detection. The tool supports signature-based detection for known threats, generating detailed logs that facilitate forensic analysis and incident investigation. More importantly, being an open-source project its extensibility, allows users to create custom scripts and plugins tailored to their specific security requirements. The scripting language associated with Zeek allows us to create customized scripts for effective network data analysis and processing. This tool is widely used in cybersecurity to augment network security by providing visibility into network traffic and identifying potential threats in real time.

The proposed solution will be tested on the "CIC Modbus 2023" [4]. The extracted features are diverse, in this way, we can identify all the relevant information that allows us to analyze the connection. They are categorized into different groups, each focusing on specific aspects of communication:

- Flow Feature: These features are designed to identify a connection uniquely.
- Basic Feature: Offering general information about the connection, these features contribute to building a basic understanding.
- Content Feature: Analyzing packet content, typically of TCP type, these features help identify specific communication characteristics.
- Time Features: Focusing on packet timing within a connection.

Building upon the foundation laid out in the referenced work [13], we have expanded the set of features to enhance the quality of the extracted information. Our goal is to acquire relevant data that can significantly enhance the subsequent training of the machine learning model.

In particular, we extract new features to closely monitor the retransmitted packets that occur during communication sessions, we delve into the analysis of the time intervals between

Feature Name	Description
Label	0 for normal and 1 for attack records
s_retrans	Source segments retransmitted (TCP)
d_retrans	Destination segments retransmitted (TCP)
m_int_s	For each connection, the mean interval between two packets (Source - in mSec)
m_int_d	For each connection, the mean interval between two packets (Destination - in mSec)
http_post	No. of flows that have the method Post in HTTP service
user_ftp	User FTP if requested
pwd_ftp	Password FTP if captured
uid	A unique identifier of the connection
local_orig	If the connection is originated locally, this value will be T. If it was originated remotely, it will be F
local_resp	If the connection is responded to locally, this value will be T. If it was responded to remotely, it will be F
history	Records the state history of connections as a string of letters
tunnel_parents	If this connection was over a tunnel, indicate the **uid** values for any encapsulating parent connections used over the lifetime of this inner connection
orig_bytes	The number of payload bytes the originator sent
resp_bytes	The number of payload bytes the responder sent

Table 1
Extracted features

successive packets for each connection, providing valuable insights into the dynamics of data transfer. Additionally, we employ a quantitative approach to evaluate flows by scrutinizing the usage of the POST method in the HTTP service, integrating also data related to FTP usernames and passwords.

Furthermore, a unique identifier has been introduced for each connection, two boolean fields have been added to differentiate whether a connection originated locally or remotely, information particularly relevant in industrial networks to identify attacks. In industrial settings, it is crucial to isolate the networks from the external world. This involves minimizing remote communications as much as possible, as they can serve as potential access points for attackers, putting at risk the security of the entire infrastructure and posing risks to the safety of workers.

Also, specific information for tunnelling and the connection state history has been included and documented as a string.

The new extracted features are described in Table 1

The solution is based on different scripts for extracting the features, BASH automation was used to automatize the feature extraction.

Several data structures have been employed to facilitate the management of blocks of information in the logs. These structures include:

- Info: stores information to be logged for each identified connection in the analyzed pcap files;
- FlowFeatures: preserves details of the addresses involved in a connection and the protocol used;
- eachPackets: collects information described in the reference document for each packet;

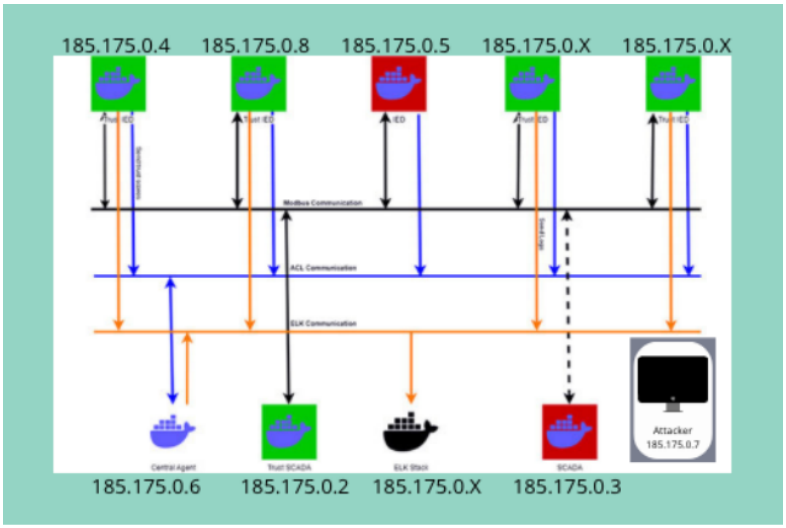

Figure 1: Simulated network architecture to create CIC Modbus Database

- each TCP Conn: specific to TCP connections, stores relevant information;
- infoAllC: summarizes the total number of HTTP flows using the Get and Post methods.

Regarding log management, crucial for the usability of the extracted feature, it was decided to group similar functionalities to minimize the number of generated log files and enhance their management.

4. Threat and anomaly detection

The dataset used is diverse and includes various types of network traffic. It is generated by capturing pcap files that simulate both legitimate and malicious network traffic within a fictional network. The dataset is divided into two parts: the 'Dataset Attack' and the 'Dataset Benign'.

Within the simulated network architecture (Figure 1), the SecureIEDs identified as IED1A (185.175.0.4) and IED4C (185.175.0.8) are considered secure. Similarly, the Secure SCADA HMI (185.175.0.3) is included among the devices considered secure. However, there is also an insecure IED (185.175.0.5) and an insecure SCADA HMI (185.175.0.3), making them potentially vulnerable.

A crucial element of the architecture is the Central Agent (185.175.0.6), which receives the so-called *detection score* from agents present in each secure device. Simultaneously, there is an Attacker (185.175.0.7), responsible for the attacks visible in the "Dataset Attack".

The provided dataset necessitates a preprocessing phase before being suitable for input into machine learning models. The FE, discussed in Section 3, facilitates the preparation of the data for the subsequent training phase. Given that the resulting data is unlabeled, the most immediate application is an unsupervised algorithm to discern the intrinsic characteristics of the traffic and perform an anomaly detection task. First, we used a K-Means clustering algorithm to perform an initial analysis of the data. Then we used the Isolation Forest algorithm to perform anomaly detection.

Unsupervised algorithms excel at detecting anomalies in comparison to normal traffic. However, for the accurate detection of malicious attacks, particularly those employing hiding techniques, a supervised learning approach may prove more suitable. Labelling the dataset becomes crucial for effectively training the models to distinguish real attacks from normal traffic. Therefore, we implemented an automatic labelling process based on various heuristics. Using these scripts, we comprehensively labelled all samples from the initial dataset to create a new dataset suitable for training a machine learning model. Following this, we trained a neural network on the refined dataset to build an automatic threat detection system.

The neural network adopts at its core is a Long Short-Term Memory (LSTM) layer with 128 units, making it particularly effective for analyzing sequential data, such as the available traffic logs. Following the LSTM is a dense layer with a ReLU activation function, succeeded by a dropout layer for regularization to reduce the risk of overfitting. Lastly, there is a dense layer with a sigmoid activation function to produce the final binary classification output.

The network can only distinguish between benign and malicious samples, as we did not include the specific type of attack in the labels. This decision was made to enhance the generalization capabilities of the network, allowing the model to attempt recognition of attacks not explicitly present in the training data.

5. Results

The use of Zeek to perform feature extraction from the pcap files of the "CIC Modbus 2023" dataset, exploiting its scripting capability, proved to be a good solution due to the reduced computational time required compared to machine learning models and the possibility of obtaining a deterministic solution, while also allowing a detailed analysis of the different connection types.

To evaluate the capability of the proposed method, we applied machine learning techniques (K-means and Isolation Forest) on features extracted with Zeek to identify unusual patterns and behaviours in network traffic. Figure 2 shows the results obtained applying K-means to the *analyzer.log* file containing the information related to network protocols extracted by Zeek. We can observe the presence of five clusters, where cluster 0 (green) is the densest, indicating a prevalent traffic type. Clusters 1 (blue) and 4 (purple), which contain fewer instances and are more compact, represent less frequent categories of network traffic. Clusters 2 (red) and 3 (yellow), the former consisting of a single instance and the latter of only two, could indicate the presence of outliers or the presence of a new type of traffic or anomalies.

The results from the *conn.log* file, containing traffic data and the subsequent analysis, is depicted in Figure 3, unveil the existence of three distinct clusters. Cluster 2 (green) stands out as the most cohesive, characterized by lower internal variance, representing standard traffic. In contrast, both cluster 0 (red) and cluster 1 (blue) exhibit greater dispersion and lower density, suggesting increased variability and heterogeneity in the type of traffic.

The results of applying Isolation Forest to detect anomalies in the data, is reported in the Figure 4. It shows inliers, which represent observations that fit the general pattern of the data and are classified as normal activity and outlier observations, highlighted in black, that deviate significantly from the norm and may indicate anomalies or suspicious activity.

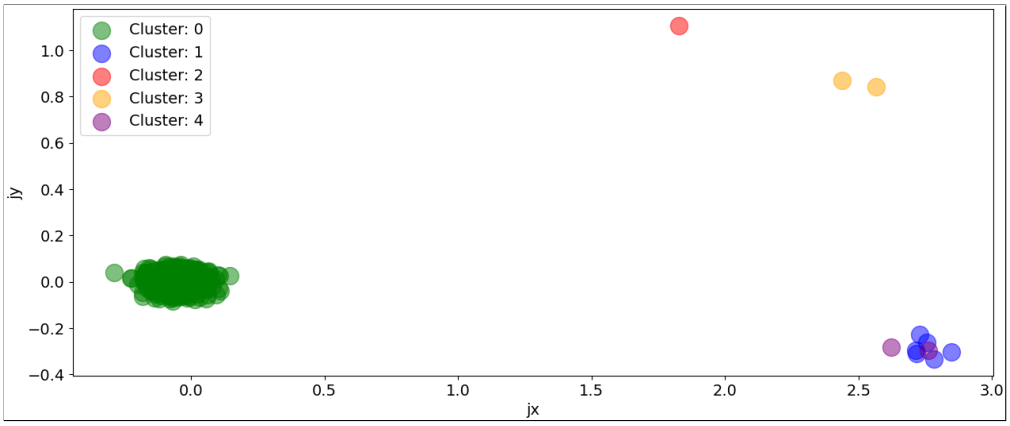

Figure 2: Clustering results of analyzer.log

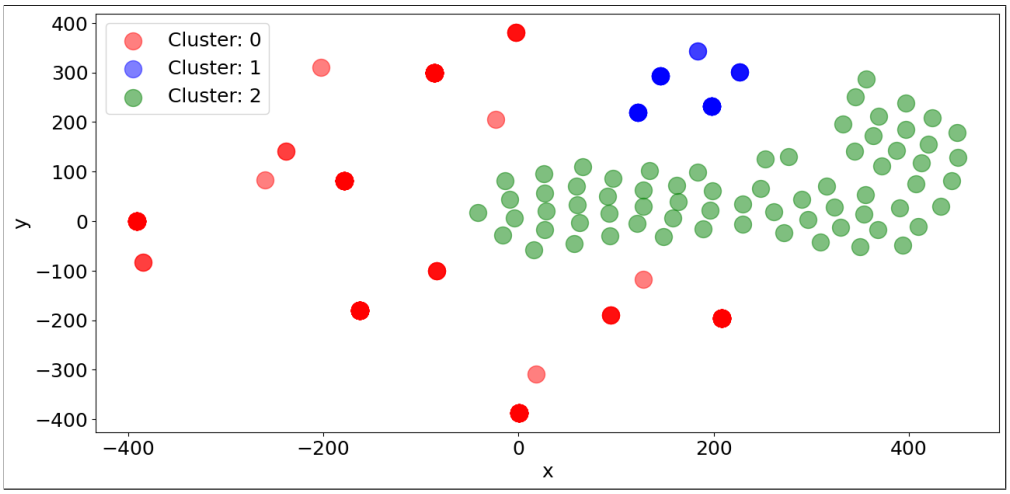

Figure 3: Clustering results of conn.log

Finally, by training the Artificial Neural Network model on the labelled dataset generated with Zeek, we evaluated its capability to detect the presence of attacks in network traffic. Table 2 presents the sample distribution in the training dataset, categorized into normal traffic and attacks.

We partitioned the data into a standard 70% - 30% split for training and testing purposes. Using the neural network described in section 4 we achieved an accuracy of 84.21% in correctly detecting malicious samples from the normal traffic.

6. Conclusions

The use of machine learning approach for feature extraction is common, although its high computational requirements make it impractical for real-world applications. In this paper, we

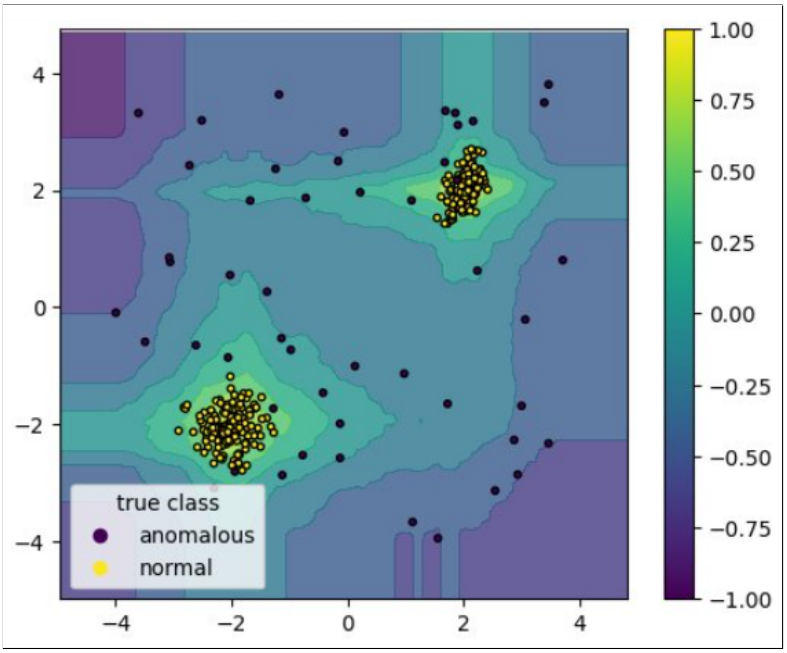

Figure 4: Visualization of inliers and outliers

Measure	Benign	Malicious
N° of flows	4,559,770	9,055,995
Src bytes	1,478,356,764	3,179,361,798
Dst bytes	1,128,175,330	2,368,174,501
Src pkts	26,006,278	49,821,269
Dst pkts	20,094,581	38,436,078
TCP	4,550,004	9,042,823
UDP	8,589	11,964
ICMP	1,177	1,208
Other	60	80

Table 2
Training dataset size

propose the use of Zeek for feature extraction to improve the speed and effectiveness, and machine learning techniques for the analysis of the extracted features. The dataset used is "CIC Modbus 2023", which contains traffic data from an industrial network.

The effectiveness of the proposal was confirmed by the results of the detailed analysis performed on the different types of connections. The K-means and Isolation Forest algorithms were able to identify patterns within the network traffic and detect possible anomalies when additional features were used. Furthermore, the neural network achieved an accuracy of 84.21%, demonstrating its capability to distinguish between malicious and benign network traffic using the dataset generated by Zeek. This research emphasizes the potential of combining traditional tools with machine learning methods to enhance the detection of potential malicious activity.

Acknowledgments

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] M. Benmalek, Ransomware on cyber-physical systems: Taxonomies, case studies, security gaps, and open challenges, Internet of Things and Cyber-Physical Systems (2024).

[2] I. University of California, Kdd cup 1999 data, 1999. URL: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[3] T. et al., Nsl-kdd dataset, 2009. URL: https://www.unb.ca/cic/datasets/nsl.html.

[4] K. Boakye-Boateng, A. A. Ghorbani, A. Lashkari, Securing substations with trust, risk posture, and multi-agent systems: A comprehensive approach, in: 2023 20th Annual International Conference on Privacy, Security and Trust (PST), IEEE Computer Society, Los Alamitos, CA, USA, 2023, pp. 1–12. URL: https://doi.ieeecomputersociety.org/10.1109/PST58708.2023.10320154. doi:10.1109/PST58708.2023.10320154.

[5] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, F. Ahmad, Network intrusion detection system: A systematic study of machine learning and deep learning approaches, Transactions on Emerging Telecommunications Technologies 32 (2021) e4150.

[6] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, M. Portmann, Feature extraction for machine learning-based intrusion detection in iot networks, Digital Communications and Networks (2022). URL: https://www.sciencedirect.com/science/article/pii/S2352864822001754. doi:https://doi.org/10.1016/j.dcan.2022.08.012.

[7] nozominetworks, What it needs to know about ot/iot security threats in 2020, 2020. URL: https://www.nozominetworks.com/blog/what-it-needs-to-know-about-ot-io-security-threats-in-2020.

[8] T. Micro, State of ot security in 2022: Big survey key insights, 2022. URL: https://www.trendmicro.com/en_nl/research/22/f/state-of-ot-security-2022.html.

[9] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, J. Lopez, A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services, IEEE Communications Surveys & Tutorials 20 (2018) 3453–3495. doi:10.1109/COMST.2018.2855563.

[10] M. A. Khan, K. Salah, Iot security: Review, blockchain solutions, and open challenges, Future Generation Computer Systems 82 (2018) 395–411. URL: https://www.sciencedirect.com/science/article/pii/S0167739X17315765. doi:https://doi.org/10.1016/j.future.2017.11.022.

[11] S. Li, M. Iqbal, N. Saxena, Future industry internet of things with zero-trust security, Information Systems Frontiers (2022) 1–14.

[12] C. Zanasi, S. Russo, M. Colajanni, Flexible zero trust architecture for the cybersecurity of industrial iot infrastructures, Ad Hoc Networks 156 (2024) 103414. URL: https://www.sciencedirect.com/science/article/pii/S1570870524000258. doi:https://doi.org/10.1016/j.adhoc.2024.103414.

[13] N. Moustafa, J. Slay, Unsw-nb15: a comprehensive data set for network intrusion detection

systems (unsw-nb15 network data set), in: 2015 Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6. doi:10.1109/MilCIS.2015.7348942.
[14] I. C. S. I. (ICSI), Zeek, 2024. URL: https://zeek.org/.
[15] Argus, 2024. URL: https://openargus.org/.