Enhancing IoT Security: Intrusion Detection System Using Machine Learning for Cross-Environment Attack Identification

Priscilla Kyei Danso, P.A.C. Abisheka, Sagor Sikdar, G M Tasnim Alam
Department of Computer Science
Stony Brook University, New York, USA
{priscilla.danso, abisheka.pitumpe, sagor.sikdar, gmtasnim.alam}@stonybrook.edu

***Abstract*—The heterogeneity of connectivity and communication protocols, coupled with the sheer volume of data generated by Internet of Things (IoT) devices, creates a complex landscape that is challenging to secure. The diversity of devices and communication protocols within IoT networks complicates the task of identifying compromised devices, particularly as the number of connected devices increases. While similar research has been conducted in the past, this study aims to address the challenge of misclassifications when training and testing IDS models in different environments. Our primary objective is to generate robust features that can excel in training and testing environments distinct from one another. By doing so, we aim to enhance the security of IoT ecosystems and contribute to the development of more reliable intrusion detection mechanisms. The CICIoT2023 and MedIoTBot datasets will be used in the evaluation of our study. Our findings indicate that .....**

***Index Terms*—Intrusion Detection, Transferability, Machine Learning, Cross-Environment, IoT Security**

I. Introduction

Internet of Things (IoT) devices have seamlessly integrated into our daily lives. From instructing virtual assistants like Alexa to relay tomorrow's weather forecast, to remotely securing smart doors from our office when we forget to lock our homes, and even having iRobot autonomously vacuum our living spaces—coupled with automobiles equipped with built-in sensors to alert us of low tire pressure [9]. Undoubtedly, IoT devices have significantly enhanced convenience and efficiency in various aspects of our lives. However, this convenience comes at the cost of security and privacy.

The Internet of Things, commonly defined by the Internet Engineering Task Force (IETF) as the, *"network of physical objects, or 'things,' embedded with electronics, software, sensors, actuators, and connectivity to enable objects to exchange data with the manufacturer, operator, and/or other connected devices"* [28], has seen exponential growth. Statista reports estimates that between 2021 and 2025, global deployment of IoT devices is projected to reach 30.9 billion units, a significant increase from the 13.8 billion deployed in 2021 [8]. This pervasive connectivity is creating a more interlinked world than ever before.

The extensive array of interconnected IoT devices, while offering immense functionality, also introduces numerous vulnerabilities and potential attack vectors that can compromise their operations. Cyberattacks targeting vulnerable IoT devices have had severe repercussions. Notably, the attack on the Dyn DNS provider in October 2016 by the Mirai botnet remains one of the most substantial Distributed Denial of Service (DDoS) events involving IoT devices, resulting in the outage of Dyn's DNS servers [2]. The aftermath of the Mirai cyberattack saw these compromised devices transformed into "botnets" or "zombie armies" [6], causing significant service disruptions on major platforms like Amazon, CNN, GitHub, Netflix, PayPal, and Twitter for several hours [10].

The prevalent approach in tackling these issues involves the use of Intrusion Detection Systems (IDS), especially anomaly-based IDS. While rule-based IDS has limitations in predicting solely known attacks, rendering them inadequate in countering zero-day attacks, anomaly-based IDS functions by observing sequences of incoming events during non-attack periods. It constructs a model based on the system's regular behavior and identifies anomalies through a similarity index between normal and abnormal observations [11].

Although anomaly detection in IoT using machine learning has garnered significant attention, an underexplored area lies in the challenge of training and testing models in different environments. This challenge, often referred to as domain adaptation or transfer learning, is particularly pertinent in IoT scenarios. Given the diverse deployment of devices in various environments, the data distribution during testing may not consistently align with the distribution of the training data.

This study proposes an Intrusion Detection System (IDS) that utilizes Machine Learning to uniquely identify various attack types within IoT networks and develops a methodology for distinguishing network traffic of the trained model in one domain and using the trained model for anomaly detection in another domain. Our primary objective is to demonstrate and mitigate the issue highlighted by Haque et al. [17] by generating robust features that can excel in both training and testing environments, distinct from one another. By doing so, we aim to enhance the security of IoT ecosystems and contribute to the development of more reliable intrusion detection mechanisms. To the best of our knowledge, this is the first research endeavor to tackle anomaly detection with the underlying concept of domain adaptation. The main contributions of this article can be summarized as follows.

- We present a domain-adaptive anomaly-based IDS able to differentiate the behavior of incoming IoT traffic and predict the type of anomaly.
- We extract robust features to be employed in our study. Due to the overly unique approach of our study, the only way our model is going to excel in an environment where the model makes a prediction in our target domain based on the performance of the model in the source domain requires efficient features.
- We evaluate our approach in two publicly available datasets. With the CICIoT acting as our source domain and the MedBIoT acting as our target domain.
- We explore different Machine Learning approaches in our study to achieve the best results in our domain-adaptive anomaly-based IDS predictions.

II. Literature Review

A. Machine Learning for Anomaly Detection

In the field of IoT anomaly detection, challenges primarily revolve around the scarcity of labeled data resulting from irregular anomalies and the intricacy of acquiring genuine system data. There exists a notable gap in formalizing the process from data collection to model development and validation in real-world settings. Current methods [12] typically involve training models on normal behavior and identifying anomalies. However, more precise techniques are imperative for dealing with complex, real-world datasets. The availability of suitable and regularly updated datasets, encompassing a range of normal and abnormal behaviors, is pivotal for training and validating real-time anomaly detection methods. Nevertheless, existing datasets often suffer from labeling errors and inadequacies for real-time detection. To counter this, new datasets should emulate realistic scenarios and include ground truth data to enhance their credibility for testing new anomaly detection systems. Developing a model for anomaly detection proves challenging due to data complexity, encompassing issues such as imbalanced datasets, unexpected noises, and data redundancy. Effective data curation approaches are crucial for extracting valuable information and knowledge from such datasets.

A plethora of machine learning anomaly detection techniques for data streams have emerged across various sectors including manufacturing, finance, military, healthcare, and the Internet of Things. These techniques aim to identify anomalies in real-time data streams. Various methods, such as Cumulative Local Outlier Factor (C-LOF), AutoCloud, and a multi-kernel approach, address different facets of anomaly detection but are hindered by limitations like high time complexity and issues related to evolving data.

On the other hand, approaches like xStream focus on feature evolution in data streams and present promising results. However, certain challenges, such as evolving clusters, remain unaddressed. Furthermore, methods designed for anomaly detection in specific domains, such as air traffic control and road safety, have been developed with varying degrees of success. Hierarchical Temporary Memory (HTM) algorithms have been utilized for real-time anomaly detection in resource utilization and network metrics. Although promising in detecting anomalies, these methods require enhancements in accuracy and latency.

The MuDi-Stream approach has been introduced in the context of evolving multi-density data streams. It preserves summary information and applies density-based clustering algorithms. However, its performance is impacted by the number of empty grids and high-dimensional data. Fast anomaly detection algorithms utilizing Extreme Learning Machine (ELM) have been developed for aviation datasets. However, there is a lack of comparison to state-of-the-art techniques. Another approach, AMAD, focuses on dynamic anomaly detection in evolving attributed networks and incorporates attribute selection. However, it does not address challenges related to high dimensionality.

Recent advancements in anomaly detection leverage deep learning techniques. Notable methods include KPI-TSAD, addressing imbalanced classification in non-stationary time series, Streaming Autoencoder (SA) for evolving data streams, OFAT for accurate long-range forecasting in high-dimensional, feature-evolving time series, e-SREBOM for real-time malware detection, ISTL for real-time video surveillance, and an e-SREBOM variant for efficient malware classification. Although promising in various contexts, some may have limitations such as reliance on labeled data, scalability issues, or a lack of real-time interactivity.

The choice of an anomaly detection technique relies on the nature of the analyzed data, especially when handling data streams, prevalent in the Internet of Things (IoT). Data streams are characterized by continuous and evolving patterns over time, necessitating precise and real-time anomaly detection methods. Various learning algorithms have been proposed for data stream anomaly detection, including methods like C-LOF [47], AutoCloud [5], TEDA Clustering, evolving spiking neural networks, KPI-TSAD [35], HTM [47], ensemble neural networks, multiple kernel learning, xStream [26], CEDAS, MuDi-Stream [1], Long Short-Term Memory, and density-based clustering.

Anomaly detection techniques can be categorized into supervised, semi-supervised, and unsupervised approaches, each serving distinct purposes. Supervised methods involve creating rules for predicting anomalies, such as the use of SVM for real-time alerts in smart transportation or Naive Bayes for intrusion detection in smart homes. Semi-supervised techniques classify anomalies within standard data, utilizing methods like Bayesian networks in healthcare or reinforcement learning in intelligent transportation. Unsupervised methods handle unlabeled data without separate training and testing, employing clustering-based detection like K-means clustering or Gaussian mixture models (GMM) in various domains such as transportation and healthcare for anomaly detection.

B. Domain adaptation or transfer learning for Anomaly Detection

Recent advancements in securing the Industrial Internet of Things (IIoT) and Network Intrusion Detection (NID) have prompted an upsurge in research dedicated to innovative machine learning techniques. Notably, the exploration of Malware traffic classification (MTC) methods, exemplified by ConvLaddernet and KT-ConvLaddernet [19], has proven effective but demands careful consideration due to their high complexity. The shift towards deeper architectures like VGG networks emphasizes the crucial need to balance complexity with computational efficiency.

Addressing the challenge of limited labeled samples in NID, the prevalent use of transfer learning has led to the proposal of adversarial Domain Adaptation (DA) [3]. This approach enables the training of Deep Learning (DL) classification models for NID with minimal labeled data, empowering organizations to identify new attack families by leveraging existing datasets. Experimental results showcase the superiority of the adversarial DA approach over traditional methods, including transfer learning via fine-tuning, in both homogeneous and heterogeneous DA scenarios.

Looking forward, future research avenues are suggested in the literature. Exploring semi-supervised and unsupervised DA scenarios where labeled data in the target dataset varies presents intriguing possibilities. Additionally, investigating the efficacy of adversarial DA approaches in multi-class classification, and discerning specific attack categories, merits further attention. The evaluation of more sophisticated Generative Adversarial Network (GAN) architectures, such as Wasserstein GANs, could further enhance the robustness of these methods.

Simultaneously, the literature underscores the need for practical considerations in deploying these models. Complex models like VGG networks may necessitate significant computing resources, requiring thoughtful simplification without compromising accuracy. Moreover, the reliance on pre-trained models, a common theme in transfer learning and DA poses challenges in practical applications. Privacy concerns stemming from sharing source datasets advocate for privacy-preserving DA techniques, especially in scenarios akin to federated learning.

In summary, while the proposed MTC methods and adversarial DA approaches show promising strides in enhancing IIoT security and NID, the literature highlights the significance of addressing challenges related to model complexity, practical implementation, and privacy preservation. Future research endeavors should strive for solutions that strike a balance between sophistication and applicability, advancing the field toward secure, efficient, and widely applicable solutions. In our approach, we aim to mitigate these issues and leverage publicly available datasets to generate robust features.

In a paper entitled “A Systematic Review of Data-Driven Attack Detection Trends in IoT” [16], the authors underscore the importance of identifying distinctive features within IoT traffic that distinguish it from conventional networks. They also emphasize the challenges posed by datasets generated using a small number of IoT devices, including issues like overfitting that could lead to inaccurate results. The authors stress the significance of exploring the connections between attacks occurring at various architectural layers of IoT systems, which can significantly inform the design of more accurate and targeted Intrusion Detection Systems (IDSs).

Currently available datasets, except for MedBIoT and CI-CIDS2023, which are considered large, can lead models to misclassify benign traffic as harmful and vice versa. This issue highlights the paramount concern in developing IDS models that possess ample data for correct classifications.

III. Background

A. Intrusion Detection

Intrusion Detection Systems (IDS) are a critical security component that plays a crucial role in safeguarding computer networks and systems from unauthorized access, malicious activities, and potential security breaches. The team at NIST [38] defines Intrusion detection as *the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices.* IDS is therefore defined as a system designed to continuously monitor network and system activities, detect and respond to security threats, and help protect against unauthorized access, data breaches, and other cyberattacks. The two notable Intrusion detection methodologies categories are classified into Signature-based Detection (SD), and Anomaly-based Detection (AD). Signature-based sometimes known as rule-based [11] technique involves comparing network traffic or system behavior against known attack patterns or signatures. If a match is found, the IDS raises an alert. The shortcoming of this approach is that only known attacks in the signature-based database can be detected and hence cannot protect the system against unknown attacks (zero-day attacks). An anomaly is when behavior deviates from the expected norm. A profile defines the typical behaviors obtained by monitoring regular network activities. Anomaly-based IDS involves comparing observed events with these normal profiles to identify attacks [25]. The assumption for this technique is that malicious behavior differs from normal user behavior [21]. Machine Learning has been an effective approach employed in this technique. Hence, in our work on cross-environment attack identification, we employ the Anomaly-based IDS using Machine Learning.

B. Transfer Learning

Conventional machine learning involves training and testing data sharing both the input feature space and the same data distribution. If there is a dissimilarity in data distribution between the training and test data, the performance of a predictive model can deteriorate [40]. Transfer learning aims to enhance the learning process in the target domain by harnessing knowledge acquired from the source domain. [43]. For the cross-environment attack identification, transductive transfer

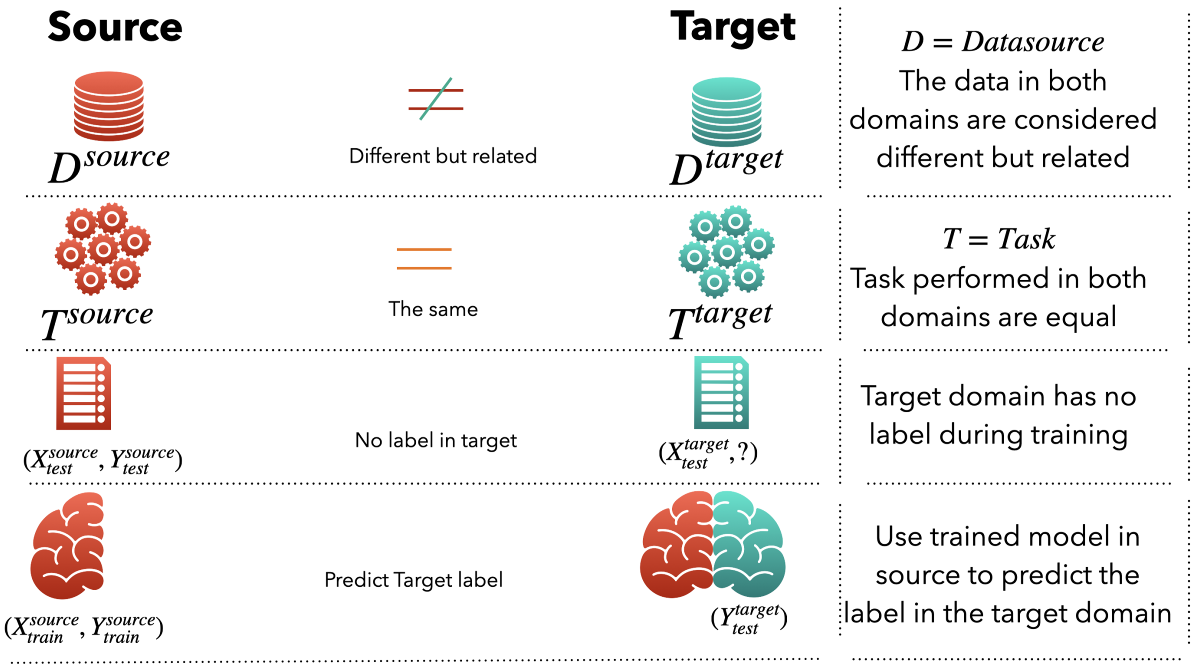

Source
Target
D^{source}
Different but related
D^{target}
D = Datasource
The data in both domains are considered different but related
T^{source}
The same
T^{target}
T = Task
Task performed in both domains are equal
$(X_{test}^{source}, Y_{test}^{source})$
No label in target
$(X_{test}^{target}, ?)$
Target domain has no label during training
$(X_{train}^{source}, Y_{train}^{source})$
Predict Target label
(Y_{test}^{target})
Use trained model in source to predict the label in the target domain

Fig. 1: Transductive Transfer Learning for Cross-environment Attack Identification.

learning is used. From figure 1 transductive transfer learning assumes the following; the data source in the source and target domain being different but related, the tasks performed being equal, and having labeled source and no labeled target domain data [33]. Ultimately, the data can be derived from two distinct distributions, D^{source} and D^{target}. The goal of this learning method is to assign label Y_{test}^{target} to test data X_{test}^{target} drawn from D^{source} which is typically derived from D^{target}, with training data (X_{train}^{source} , Y_{train}^{source}) [7].

IV. The Overall Proposed Framework

This study proposes a cross-environment approach based on the underlying concept of transductive transfer learning by employing the use of several supervised ML techniques to train a model in one lab (source) and transfer the knowledge of the trained model to another lab (target) to enhance the learning process. In this section, an overview of the proposed attack identification framework is analyzed. A detailed explanation of the feature selection and classifier modeling methods used is provided. Figure 2 depicts the proposed attack identification framework. The framework is divided into four modules: data acquisition, feature generation, model training, and knowledge transfer of the model. The following is a summary of each module:

- Data Acquisition: This study will use multiple publicly available datasets to analyze our work. The raw pcap files are acquired and due to the large size of the pcap files, they need to be split into smaller chunks before any preprocessing can commence.
- Feature generation: During this stage, we use a publicly available script that converts the pcaps file to CSV files. The converted CSV files are then prepared for our model training. It is worth noting that the feature set derived in both datasets acting as different domains has to be the same.
- Model training: This study employs a supervised machine learning technique to profile and identify the different types of attacks in the network. Several ML models are investigated to determine the most effective method for our approach.
- Knowledge transfer of model: This research employs a cross-environment approach for attack identification based on the concept of transductive transfer learning. To test and predict the target label that corresponds to the attack type, the knowledge from the trained model in the source domain is transferred to the target domain.

A. Data Acquisition

To evaluate the idea of cross-environment attack identification we are proposing, we need a desirable dataset to achieve this goal. Initially, we had resolved to two datasets; the CICIoT2023 and MedBIoT datasets. However, we realized as much as there were attacks that were common in both datasets (Mirai and Benign) they weren't enough for for our study. We had to look for other IoT-related datasets. We stumbled on IoT23 dataset [1] [14] and the IoT Network Intrusion dataset [2] [20]. Table I shows the table structure of the dataset that they benign and attack types and where they were acquired from for the source and target domain. We collected the raw

[1] https://www.stratosphereips.org/datasets-iot23
[2] https://sites.google.com/view/iot-network-intrusion-dataset/home

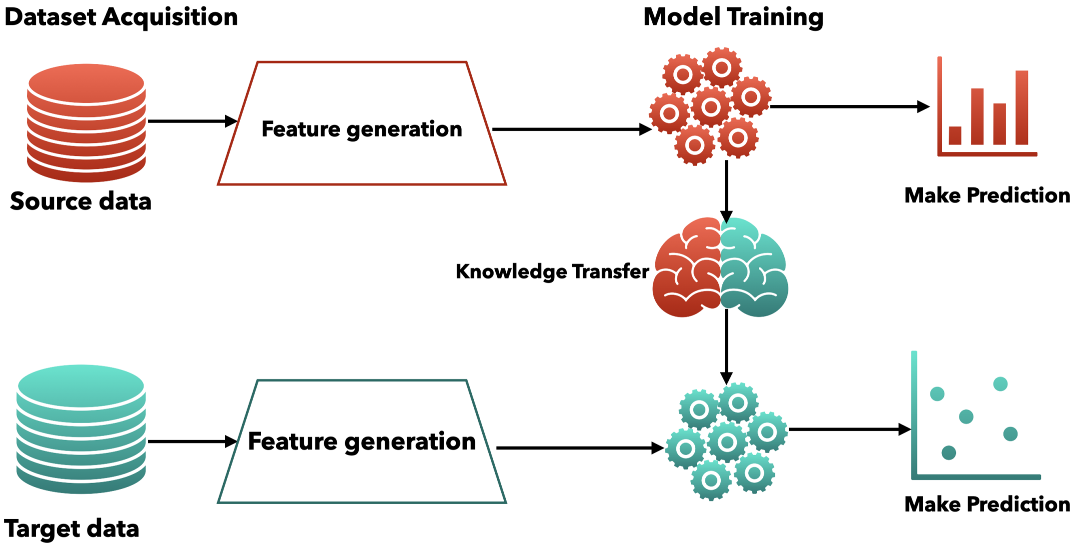


Fig. 2: Approach of Project.

pcap files from these datasets and stored them in the respective folders “source_domain” and “target_domain”

The attack types used in this experiment are as follows.

- Benign - A benign cyber activity or entity refers to actions or elements that are non-malicious or pose no threat to a system or network. It encompasses normal, routine, or harmless activities within cyberspace that do not aim to cause harm or disruption.
- Mirai - Mirai is a type of malware that targets Internet of Things (IoT) devices. It functions by exploiting weak security measures in these devices, infecting them to form a botnet. The botnet is then utilized to launch large-scale DDoS attacks, causing service disruptions or network outages.
- DoS - DoS Syn Flood (Denial-of-Service SYN Flood) is a type of cyber attack that targets the initiation of a TCP (Transmission Control Protocol) handshake. It involves overwhelming a server with a flood of connection requests (SYN packets), but the attacker doesn’t complete the handshake, leaving the server waiting for confirmation. This flood of half-open connections exhausts the server’s resources, rendering it unable to accept legitimate connections, causing service disruption or a complete denial of service to legitimate users.
- Torii - Torii is a sophisticated and stealthy IoT botnet malware that targets various architectures, including IoT devices and enterprise servers. It’s capable of exploiting numerous vulnerabilities, providing attackers with remote control, data exfiltration, and the ability to run arbitrary commands on infected devices, posing a serious threat to network security and privacy.

B. Feature generation

Using the publicly available CIC feature extraction script which converts pcaps to CVS, we converted all pcap files into CSV. A total of 62 features were extracted from each pcap file. Table II shows the list of features extracted for our study. We have listed the features we retrieved after running the CIC feature generation script. To learn about what each feature means [31] is a step in a great direction. It is worth mentioning that to achieve the most optimum performance by any ML classifier, applying techniques such as data sampling, feature selection, and removing highly correlated features, among others are key. However, we will use these techniques to improve the results of our work but such techniques may not be practical in a real-world setting where there is a continuous dataset generation. Additionally, applying any form of technique needs to happen in both the source and target domains.

TABLE I: Benign and attack types and where they were acquired from for each domain

	Dataset	
Label	**Source**	**Target**
Benign	CICIoT2023 [31]	IoT2023 [14]
DoS (Syn flood)	CICIoT2023 [31]	IoT Network Intrusion [20]
Mirai (UDP Plain)	CICIoT2023 [31]	IoT Network Intrusion [20]
Torii	IoT2023 [14]	MedBIoT [15]

C. Model Training

We aim to train and test our cross-environment attack identification on K-nearest neighbor [22], [34], Support Vector machines [18], [27], [42], Naive Bayes [37], [44], Logistic regression [24], [32], [46], Decision tree [29], [41], Random forest [4], Adaboost [39], Extra Tree Classifier [45], and Gradient Boosting [30] algorithm. The goal is to determine which classifier performs better and adapts to a new environment.

D. Knowledge transfer of model

In this research, we collected IoT traffic data from various laboratories located in different countries, as indicated in

Time	Flow duration	Header Length	Protocol Type	Duration
Rate	Source Rate	Destination Rate	Fin flag number	Syn flag number
Rst flag number	Psh flag number	Ack flag number	Urg flag number	Ece flag number
Cwr flag number	Ack count	Syn count	Fin count	Urg count
Rst count	Max duration	Min duration	Sum duration	Average duration
std_duration	CoAP	HTTP	HTTPS	DNS
Telnet	SMTP	SSH	IRC	TCP
UDP	DHCP	ARP	ICMP	IGMP
IPv	LLC	Tot sum	Min	Max
AVG	Std	Tot size	IAT	Number
MAC	Magnitue	Radius	Covariance	Variance
Weight	DS status	Fragments	Sequence number	Protocol Version
Flow idle time	Flow active time			

TABLE II: Features extracted from pcaps used in evaluation

Table I. The purpose of this dataset is to validate the proposed cross-environment attack identification approach. It's worth noting that each of these datasets corresponds to a distinct network configuration, comprising different devices and tools employed for packet capture. To maintain consistency, we ensured that attacks used in both domains were at least the same. For instance, if the source domain utilized a DoS Syn flood attack, we also applied the DoS Syn flood in the target domain, thereby preserving the data source's inherent characteristics in the context of transductive transfer learning. It's essential to emphasize that our research may be subject to various privacy and security compliance regulations, depending on the locations of the different laboratories, such as GDPR in the EU and FTC in the US. Currently, the dataset we utilized originates from Canada [31], Estonia [15], and the Czech Republic [14].

In this stage of the study, the aim is to assign attack-type labels to the target feature sets using the input variables from each of the target sources and the pre-trained model from the source domain. Consequently, we employ the pre-trained model from the source domain for testing, rather than retraining the model in the target domain. We evaluate the performance of classifiers when trained on data from the source domain and subsequently apply the trained model for testing in the target domain.

V. Dataset Description

A. *CICIDS2023 [31]*

This is a novel and extensive IoT attack dataset to foster the development of security analytics applications in real IoT operations. To accomplish this, 33 attacks are executed in an IoT topology composed of 105 devices. These attacks are classified into seven categories, namely DDoS, DoS, Recon, Web-based, Brute Force, Spoofing, and Mirai. Finally, all attacks are executed by malicious IoT devices targeting other IoT devices.

This research introduces a significant contribution in the form of a realistic IoT attack dataset, featuring a diverse topology with actual IoT devices acting as both attackers and victims. The study documents and collects data from 33 attacks across 7 classes against IoT devices, demonstrating their reproducibility. Additionally, the research evaluates the effectiveness of machine and deep learning algorithms in classifying and detecting malicious or benign IoT network traffic using the CICIoT2023 dataset.

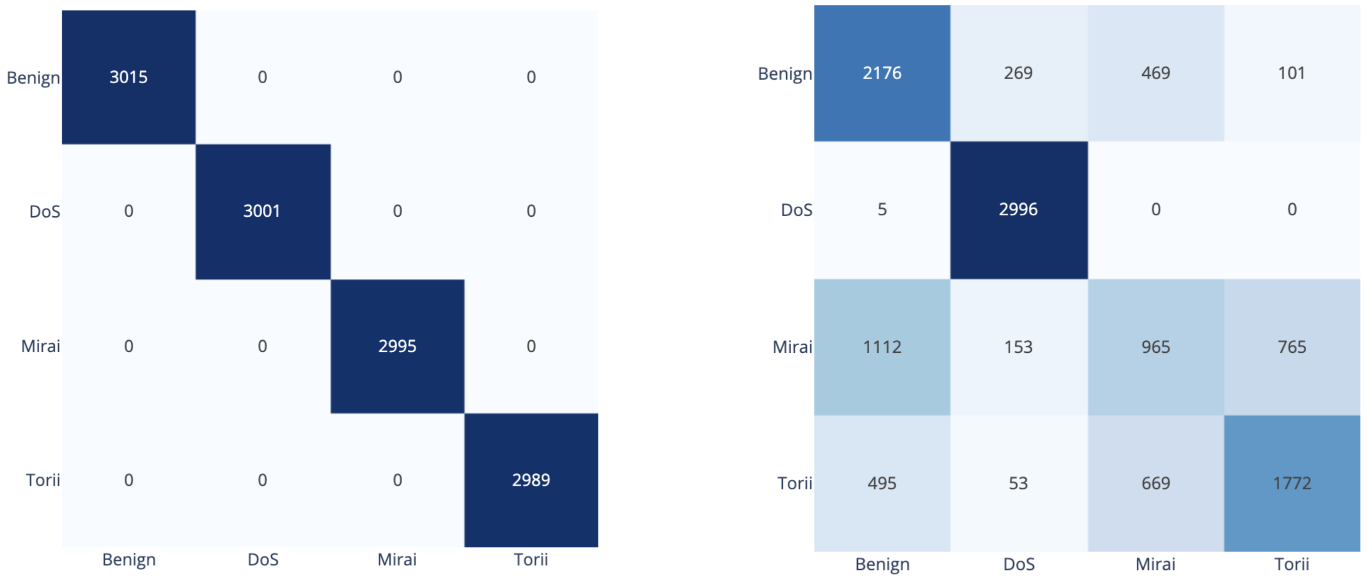
The main dataset directory, named CICIoT2023, is organized into four subdirectories. The PCAP directory contains original traffic captured during attacks in .pcap files. The CSV directory holds features extracted from the original files for Machine Learning (ML) evaluation in .csv format. The Example subdirectory provides a Jupyter notebook demonstrating how the dataset can be used to train and evaluate ML models for attack detection and classification. The Supplementary material subdirectory includes source code and tool descriptions used in collecting and processing the attack data, employing tools like Mergecap, PySpark, TCPDump, and DPKT for tasks such as merging files, handling data, splitting pcap files, and feature extraction.

B. *MedIoTBoT [15]*

The experimental setup for this research, conducted as part of Jorge Alberto Medina Galindo's master's thesis, aimed to address the lack of a comprehensive dataset for IoT botnet detection. The dataset is unique in combining real and emulated IoT devices within a medium-sized network, consisting of 83 devices. Notably, the inclusion of actual malware deployment, specifically Mirai, BashLite, and Torii, contributes real-world botnet network data. The dataset is labeled, and categorized by traffic source (normal or malware), facilitating easy data labeling and feature extraction from raw pcap files. The primary focus is on the early stages of botnet deployment, specifically spreading and command-and-control (C&C) communication.

The research emphasizes the dataset's suitability for both general IoT botnet research and intrusion detection system applications. It supports various machine learning approaches, including supervised learning (binary and multi-class classification) and unsupervised learning (anomaly and outlier detection), as demonstrated in published papers. The network comprises a combination of real devices (Sonoff Tasmota smart switch, TPLink smart switch, and TPLink light bulb) and emulated devices (Lock, Switch, Fan, Light) within a medium-sized network.

The dataset is provided in two main formats: bulk pcap files for each data source type (legitimate, Mirai, BashLite, and



(a) Confusion Matrix of Random Forest in the source domain

(b) Confusion Matrix of Support Vector Classifier in the source domain

Fig. 3: Confusion matrix of Random Forest and SVC in the source domain

Torii) and fine-grained pcap files for each data source, botnet phase, and device type. For instance, mirai_mal_CC_lock.pcap corresponds to Mirai botnet malware data during C&C communication for lock devices. Each pcap file is labeled as either malicious or legitimate/benign traffic, with bulk pcap files specifying the malware deployed and traffic type, making it a valuable resource for researchers and practitioners in IoT botnet detection and intrusion detection systems.

C. IoT2023 [14]

The IoT-23 dataset, originating from the Stratosphere Laboratory at the AIC group, FEL, CTU University, Czech Republic, is a pivotal contribution to IoT security. Initially published in January 2020 and funded by Avast Software, Prague, this dataset encompasses network traffic data collected from IoT devices spanning from 2018 to 2019. Its primary objective is to furnish researchers with labeled instances of both IoT malware infections and benign IoT traffic, serving as a valuable resource for developing and assessing machine learning algorithms tailored for IoT security.

Comprising twenty-three distinct captures (scenarios), the dataset includes twenty captures from infected IoT devices, each distinctly labeled with the specific malware sample executed. Additionally, it contains three captures of network traffic from real benign IoT devices: a Philips HUE smart LED lamp, an Amazon Echo home intelligent personal assistant, and a Somfy smart door lock.

The scenarios, executed within a controlled network environment with unrestricted internet connections, aimed to mimic real IoT device behaviors. Malware samples were executed in a Raspberry Pi in the malicious scenarios, using various protocols and carrying out diverse actions.

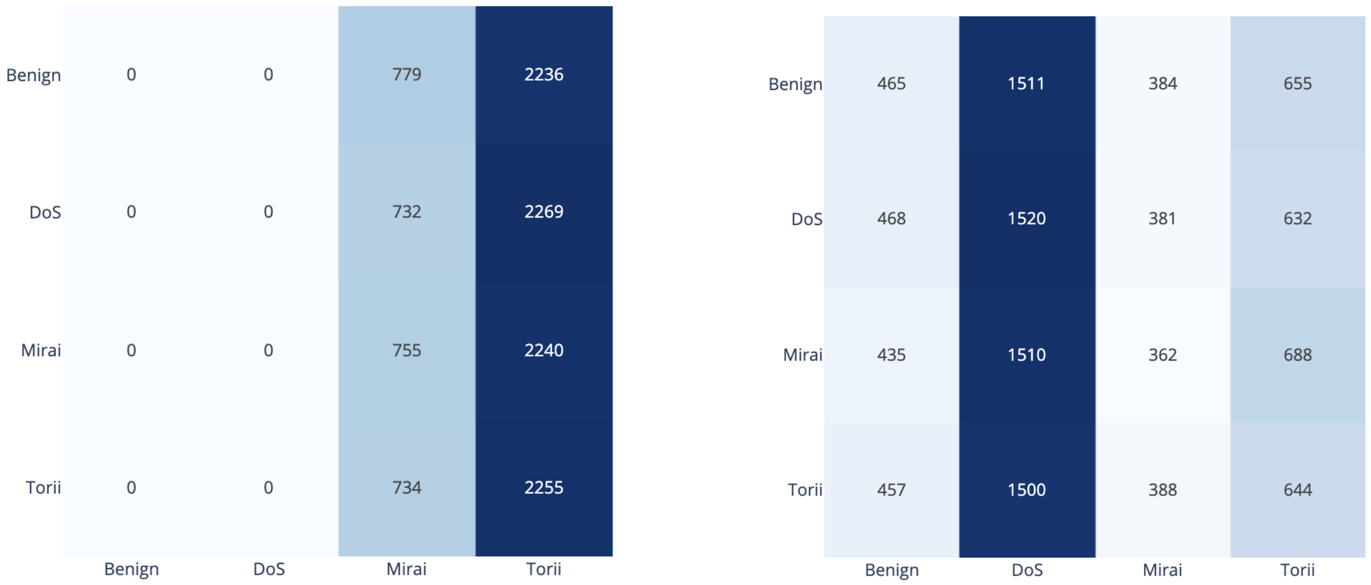
In the process of assigning behavioral labels to the captured network traffic flows, researchers developed a script named Flaber. This custom-made Flaber script was designed to automate the labeling process. It compared flow data in the conn.log file with predefined rules, generating and assigning corresponding labels based on the defined criteria, thereby streamlining the categorization and labeling of the flows in the dataset.

D. IoT Network Intrusion [20]

The IoTID20 dataset is a result of a testbed that integrates a combination of IoT devices and interconnected structures. In this setup, a typical smart home environment was established, featuring the SKT NGU smart home device and an EZVIZ Wi-Fi camera. These devices played a pivotal role in generating the IoTID20 dataset. Visualized in Figure 2, the testbed showcases these two IoT devices connected to a smart home Wi-Fi router. Additionally, other devices such as laptops, tablets, and smartphones are linked to this same router. Notably, the SKT NGU and EZVIZ Wi-Fi cameras act as IoT victim devices, while all other devices within the testbed operate as the attacking devices.

The CICflowmeter application [23] was used to extract network features from these Pcap files, resulting in a dataset presented in CSV format. The dataset comprises 80 network features and three label features - binary, category, and subcategory. The IoTID20 dataset's development led to the final version comprising 83 network features and three label features.

The IoTID20 dataset's development led to the final version comprising 83 network features and three label features. This dataset accurately replicates modern trends observed in



(a) Confusion Matrix of Random Forest in the target domain

(b) Confusion Matrix of Support Vector Classifier in the target domain

Fig. 4: Confusion matrix of Random Forest and SVC in the target domain

IoT network communication, offering a valuable snapshot of current practices within this domain. Additionally, it stands out as one of the limited publicly available IoT intrusion detection datasets, providing researchers and analysts with a publicly accessible resource for evaluating and understanding IoT-based security threats and intrusion patterns

VI. Methodology

A. Overcoming Limitations of Classical Statistical Machine

In the quest to develop effective models for a dataset comprising four distinct classes, initial attempts were made using classical statistical machine learning approaches. However, these methods faced significant challenges in terms of overfitting and underfitting. Overfitting occurs since some of these models are too complex and learn the noise and details in the training data to the extent that it undeniably negatively impacts the performance on new data. Underfitting, on the other hand, happens when the rest of the models are too simple to capture the underlying pattern in the data, leading to poor performance both on training and new data. These issues rendered the classical approaches technically unusable for the task at hand, necessitating a shift towards more sophisticated and adaptable machine learning techniques, i.e deep learning based techniques.

B. Leveraging a Large Source Domain Dataset

To tackle the prevalent challenges, attention was redirected towards the development of LSTM (Long Short-Term Memory) and LSTM+CNN (Convolutional Neural Network) models. These models underwent training using an extensive, varied dataset obtained from the source domain, encompassing four distinct categories. The considerable size and diversity of this dataset played a crucial role in equipping the models with the ability to precisely recognize and assimilate the unique characteristics of each category. This thorough training strategy enabled the models to gain a deep insight into the data, a key factor for efficient categorization. It also helped in addressing the common issues of overfitting and underfitting, which are often drawbacks of traditional methods.

C. Domain Shift

The shift to an unfamiliar testing domain introduced the issue of domain shift, where the testing domain's data significantly differs from that of the training (source) domain. Principal Component Analysis (PCA), a statistical method for reducing data dimensionality, was employed to address this challenge. PCA works by transforming data into a new set of coordinates, prioritizing the directions where data variation is maximal. The first coordinate, or the first principal component, captures the most significant variance, with subsequent components representing decreasing variances. This technique is especially beneficial for simplifying complex, high-dimensional data, while preserving its critical aspects. In this scenario, PCA was instrumental in condensing the testing domain's data to its essential elements - the principal components, which represent the data's main variance directions. This was a crucial step in minimizing the discrepancies between the source and testing domains. By concentrating on these principal components, the models could more effectively grasp and adapt to the principal features and variations of the new domain, thereby efficiently managing the domain shift. Moreover, this adaptation opened the door to incorporating few-shot learning techniques. Few-shot learning involves training models to understand and adapt to new tasks or data with minimal examples.

D. Incorporating Few-Shot Learning and Ablation Study

The adaptation process further included a few-shot learning approach. Here, a specific number of samples, denoted as *k* for each class, were carefully selected from the testing domain for retraining the models. These k samples, chosen based on their alignment with the principal components identified through PCA, ensured that the retraining process was focused and efficient, addressing the most critical aspects of the domain shift.

Additionally, to optimize the few-shot learning process, an ablation study was conducted. This study systematically varied the number of samples (k) to identify the optimal quantity that would yield the best performance for the classification task. The goal was to determine the smallest number of samples that would still ensure high classification accuracy, thereby making the retraining process as efficient as possible.

In summary, this integrated approach of employing advanced machine learning models, utilizing PCA for domain shift, and optimizing few-shot learning through an ablation study, offers a robust solution to the challenges initially faced by classical statistical methods. This strategy ensures that the models are highly adaptable and perform well across different domains, effectively addressing the issues of overfitting and underfitting that were present in the classical approaches.

VII. Experimental Results and Analysis

A. Experiments

This section outlines the resources used for the experimentation and evaluates the data collection, data sampling, or reduction technique used in this research work as well as the feature selection technique. Additionally, we describe our approach in the training and testing of the machine learning models and the performance metrics used on the training dataset.

B. Experimental Setup

We utilize, via SSH, a remote machine linked to four discrete NVIDIA GeForce RTX 2080 TI GPUs with 11GB of VRAM each to power our use of LLMs. Standard Python machine-learning libraries , for example PyTorch, alongside data- processing and math libraries such as Pandas and NumPy allow us to collect results.

1) Data collection: The CICIDS2023[3] [31] , MedIoTBoT [15], IoT2023 [14], IoT Network Intrusion [20] were the different sources we acquired our data from. For some of the sources, one attack was retrieved and used either in the source or target domain. The collection of features created for the source and target domains are identical.

2) Training and Testing of the ML methods: Scikit learn[4], a Python library, is used in the implementation of the IDS. To determine which machine learning technique performs the best, ten machine learning models are utilized to train and test the dataset. The ten machine learning classifiers were evaluated using a 7:3 split, which involves training on 70% of the data that was randomly chosen and testing on the remaining 30%.

3) Performance measuring metrics: In this section, we outline five benchmark performance indicators for assessing how well our classifiers performed on the dataset with the training and test data taken from the set. Macro-averaging is employed for the overall performance of precision, recall, and F1-score. For datasets with training and testing data drawn from a different distribution (lab), a different metric is used to evaluate the device identification performance.

Confusion matrix is a visual depiction of the model's performance in predicting the various classes.

Accuracy is the proportion of accurate predictions to all observations.

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

Precision is calculated as the number of correctly identified device types by the number of correctly or incorrectly predicted device types by the classifier.

$$Precision = \frac{TP}{TP + FP}$$

Detection Rate shows how many actual device types the classifier classified properly.

$$Recall = \frac{TP}{TP + FN}$$

F1-score is a geometric average of precision and recall.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

C. Results and Analysis

Using the performance metrics described in Section VII-B3, and drawing on the machine learning models discussed in Section IV-C, our experimental analysis of cross-environment attack identification, as detailed in Section V, is presented in Table III. The table unmistakably illustrates a phenomenon observed by previous studies [13], [36] - machine learning models tend to excel when training and testing data originate from the same distribution. However, even a slight variation in the dataset has a detrimental impact on performance, as evident in Table III. Notably, all but one of the five boosting techniques used achieved accuracy, precision, recall, and F1-scores of less than 100% in the source domain. Regrettably, this exceptional performance in the source domain does not translate when the dataset varies slightly for testing. In the source domain, despite the dominance of these techniques, their performance during cross-environment attack identification is relatively poor compared to other models.

Several factors contribute to the current results, one of the most prominent being the differences among labs, network configurations, devices, tools used during experiments, and network packet capture processes. The Canadian Institute for Cybersecurity (CIC) collected data from around 105 IoT devices, and the benign dataset employed in the source domain could belong to any of these devices. These devices engage in various activities and actions during network packet capture. In contrast, the benign dataset used in the target domain comes from just three devices - a lamp, a speaker, and a smart door lock. Beyond device differences, the geographical separation between CIC in Canada and the target domain lab in the Czech Republic, combined with the smaller sample space in the target domain, are additional factors contributing to the contrasting results between the source and target domains. Figure 3a, Figure 3b, Figure 4a and Figure 4b show the confusion matrix of Random Forest (RF) and Support Vector classifier (SVC) in source and target domain

[3] https://www.unb.ca/cic/datasets/iotdataset-2023.html

[4] https://scikit-learn.org/stable/

	Accuracy (%)		Precision (%)		Recall (%)		F1-Score (%)	
	Source	Target	Source	Target	Source	Target	Source	Target
Algorithms								
KNN	64	25	64	25	64	25	63	24
SVC	66	25	64	25	66	25	64	24
NaiveBayes	75	25	79	25	75	25	75	24
LogisticRegression	68	25	67	25	68	25	67	24
DecisionTree	100	25	100	13	100	25	100	16
RandomForest	100	25	100	13	100	25	100	16
GradientBoosting	100	25	100	13	100	25	100	16
ExtraTree	100	25	100	13	100	25	100	16
AdaBoost	75	25	62	13	75	25	67	16

TABLE III: Cross-environment attack identification results using 10 ML algorithms and their respective Accuracy, Precision, Recall and F1-Score results

respectively. It can be seen that random forest in the target domain misclassified all the Benign and DoS as belonging to Mirai or Torri but correctly predicted these different target variables in the source domain. SVC in the target domain outperformed the RF while RF in the source domain outperformed SVC.

While the ultimate objective is to enhance cross-environment attack identification performance, the practicality and efficiency of applying performance-enhancing techniques to machine learning models in real-world IoT environments may be limited. Nevertheless, for experimental purposes, we intend to explore techniques such as data sampling, feature selection, regularization, and, significantly, the utilization of deep learning for attack identification.

D. Comparative Analysis of LSTM and Hybrid LSTM-CNN

In our experimental setup, we focused on evaluating the performance of two machine learning models, namely LSTM (Long Short-Term Memory) and a hybrid LSTM-CNN (Convolutional Neural Network), on a dataset comprising four distinct classes. The goal was to ascertain the effectiveness of each model in accurately classifying these classes. The results of these experiments are presented in the table IV.

Model	Accuracy
LSTM	35.71%
Hybrid LSTM-CNN	36.89%

TABLE IV: Accuracy of LSTM and Hybrid LSTM-CNN models on a four-class dataset

These results, especially relevant for the scenario where k=5, demonstrate the performance of the models under the few-shot learning paradigm. The LSTM model achieved an accuracy of 35.71%, while the hybrid LSTM-CNN model showed a slightly higher accuracy of 36.89%. This performance difference, though modest, is insightful when considering the constraints of few-shot learning, where models have limited data (only four samples) to adapt to the testing domain.The hybrid LSTM-CNN model combines the sequential data processing power of LSTM with the spatial feature extraction strength of CNN. This combination is potentially more effective in handling the complexities of a dataset with multiple classes, where both temporal and spatial features of this network dataset can be crucial for accurate classification. Additionally, the slightly higher accuracy of the hybrid model suggests that it might be better equipped to distinguish between the varied features of the four classes. The CNN component in the hybrid model can extract nuanced spatial patterns, which, when combined with LSTM's sequence processing, provides a comprehensive approach to class differentiation.

In our study, we conducted an ablation experiment to evaluate the performance of the LSTM-CNN (Long Short-Term Memory - Convolutional Neural Network) model with different values of k in a few-shot learning setting. Specifically, we tested with k = [3, 4, 5], and measured the accuracy for each of these scenarios.

k-Value	Accuracy
3	35.17%
4	35.60%
5	36.89%

TABLE V: Accuracy of the LSTM-CNN model for varying k-values in few-shot learning

This trend indicates a positive correlation between the number of samples in few-shot learning and the accuracy of the LSTM-CNN model. With the increase in k, the model is provided with more samples for learning, which appears to enhance its ability to adapt to the dataset's characteristics and improve its generalization capabilities, thereby increasing classification accuracy.

The highest accuracy observed at k=5 suggests that this is the optimal number of samples for few-shot learning in this specific experimental setup. This finding underscores the importance of carefully selecting the number of training samples in few-shot learning scenarios, particularly for complex models like LSTM-CNN where the balance between training data and the risk of overfitting is crucial.

The marked increase in accuracy compared to random guessing underscores the effectiveness of the LSTM-CNN architecture in handling the complexities of multi-class datasets, especially in a few-shot learning framework. This performance boost highlights the model's capability to make informed predictions rather than relying on chance, thus confirming its suitability and efficiency for classification tasks in multi-class scenarios.

VIII. Future Work

Building upon the insights gained from our current research, there are several avenues for future exploration and enhancement of IoT intrusion detection systems. The following areas represent potential directions for future work:

A. Enhanced Classification Models

To further improve classification results, future research could focus on refining and fine-tuning deep learning models. Investigating advanced architectures, optimizing hyperparameters, and exploring novel training techniques may contribute to increased accuracy and robustness in intrusion detection.

B. Ensemble Learning and Boosting Techniques

Employing ensemble learning methods, such as boosting, could prove beneficial in extracting specific information from individual models. Combining the strengths of multiple models may enhance overall detection capabilities and mitigate the weaknesses inherent in any single model. This approach could provide a more comprehensive and nuanced understanding of diverse IoT intrusion scenarios.

C. Real-World Experiments in Simulated Environments

While our study utilized diverse datasets for experimentation, the transition to real-world IoT networks with simulated attacks is a crucial step for validating the practical applicability of our enhanced intrusion detection system. Conducting experiments in environments that closely mimic real-world conditions will provide valuable insights into the system's performance, adaptability, and scalability.

D. Integration of Dynamic Learning Mechanisms

Investigating dynamic learning mechanisms that can adapt in real-time to evolving threats is paramount. Continuous monitoring of network dynamics and incorporating mechanisms for model adaptation could significantly enhance the system's ability to detect emerging and previously unseen attack patterns.

E. Consideration of Resource Constraints

Recognizing the resource-intensive nature of deep learning models, future work should explore strategies to optimize model architectures for deployment in resource-constrained IoT devices. This involves striking a balance between model complexity and efficiency to ensure practical applicability in IoT environments with limited computational resources.

IX. Conclusion

In this research, we conducted a comprehensive investigation into the efficacy of machine learning models for IoT intrusion detection across diverse datasets and attack types. Initially, our exploration involved the application of statistical machine learning models to the source and target domains. However, the outcomes were far from satisfactory, with only a 25% accuracy achieved on the target domain. This emphasized the challenges posed by the complexity and variability of IoT data.

Recognizing the limitations of traditional statistical approaches, we transitioned to a deep learning paradigm, leveraging the power of neural networks. By applying a deep learning approach to the source domain and fine-tuning a pretrained model for enhanced domain adaptation, we observed a remarkable improvement in our baseline results. The achieved enhancement underscores the potential success of employing deep learning techniques for IoT intrusion detection, especially when provided with ample resources for training.

Our findings highlight the importance of adapting to the dynamic nature of IoT environments, where traditional models may fall short. Deep learning's capacity to capture intricate patterns and representations in data offers a promising avenue for achieving robust intrusion detection in these complex scenarios. While our current results showcase significant improvement, it is worth noting that further advancements can be achieved with increased resources for training and model refinement.

As we look toward the future, continued research and development in the realm of deep learning for IoT intrusion detection hold the promise of even greater accuracy and reliability. In conclusion, our research demonstrates the potential of deep learning as a formidable tool in enhancing IoT intrusion detection systems. The journey from traditional statistical models to deep learning frameworks has illuminated a path forward, encouraging further exploration and innovation in this critical domain of cybersecurity.

Acknowledgment

We would like to thank Dr. Amir Rahmati for his constant support throughout this semester.

References

[1] Amineh Amini, Hadi Saboohi, Tutut Herawan, and Teh Ying Wah. Mudi-stream: A multi density clustering algorithm for evolving data stream. *Journal of Network and Computer Applications*, 59:370–385, 2016.

[2] Kishore Angrishi. Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. *arXiv preprint arXiv:1702.03681*, 2017.

[3] Elisa Bertino Ankush Singla and Dinesh Verma. Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation. In *ACM Conference on Computer and Communications Security ASIA*, 2020.

[4] Mariana Belgiu and Lucian Drăguţ. Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, 114:24–31, 2016.

[5] Clauber Gomes Bezerra, Bruno Sielly Jales Costa, Luiz Affonso Guedes, and Plamen Parvanov Angelov. An evolving approach to data streams clustering based on typicality and eccentricity data analytics. *Information Sciences*, 518:13–28, 2020.

[6] Nadia Chaabouni, Mohamed Mosbah, Akka Zemmari, Cyrille Sauvignac, and Parvez Faruki. Network intrusion detection for iot security based on learning techniques. *IEEE Communications Surveys Tutorials*, 21(3):2671–2701, 2019.

[7] Priscilla Kyei Danso, Sajjad Dadkhah, Euclides Carlos Pinto Neto, Alireza Zohourian, Heather Molyneaux, Rongxing Lu, and Ali A. Ghorbani. Transferability of machine learning algorithm for iot device profiling and identification. *IEEE Internet of Things Journal*, pages 1–1, 2023.

[8] Statista Research Department. Internet of things (iot) and non-iot active device connections worldwide from 2010 to 2025 (in billions) [graph]. 2020.

[9] Nomusa Nomhle Dlamini and Kevin Johnston. The use, benefits and challenges of using the internet of things (iot) in retail businesses: A literature review. In *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pages 430–436, 2016.

[10] Rohan Doshi, Noah Apthorpe, and Nick Feamster. Machine learning ddos detection for consumer internet of things devices. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 29–35, 2018.

[11] Mojtaba Eskandari, Zaffar Haider Janjua, Massimo Vecchio, and Fabio Antonelli. Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices. *IEEE Internet of Things Journal*, 7(8):6882–6897, 2020.

[12] Muhammad Fahim and Alberto Sillitti. Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review. *IEEE Access*, 7:81664–81681, 2019.

[13] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A brief review of domain adaptation. In Robert Stahlbock, Gary M. Weiss, Mahmoud Abou-Nasr, Cheng-Ying Yang, Hamid R. Arabnia, and Leonidas Deligiannidis, editors, *Advances in Data Science and Information Engineering*, pages 877–894, Cham, 2021. Springer International Publishing.

[14] Sebastián García, Agustin Parmisano, and María José Erquiaga. Iot-23: A labeled dataset with malicious and benign iot network traffic. 2020.

[15] Alejandro Guerra-Manzanares, Jorge Medina-Galindo, Hayretdin Bahsi, and Sven Nõmm. Medbiot: Generation of an iot botnet dataset in a medium-sized iot network. In *International Conference on Information Systems Security and Privacy*, 2020.

[16] S. Haque, F. El-Moussa, N. Komninos, and R. Muttukrishnan. A systematic review of data-driven attack detection trends in iot. 23(16):7191, 2023.

[17] Safwana Haque, Fadi El-Moussa, Nikos Komninos, and Rajarajan Muttukrishnan. A systematic review of data-driven attack detection trends in iot. *Sensors*, 23(16), 2023.

[18] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[19] Bamidele Adebisi Jinhui Ning, Yu Wang and Haris Gacanin. Malware traffic classification using domain adaptation and ladder network for secure industrial internet of things. 9(18), 2022.

[20] Hyunjae Kang, Dong Hyun Ahn, Gyung Min Lee, Jeong Do Yoo, Kyung Ho Park, and Huy Kang Kim. Iot network intrusion dataset, 2019.

[21] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):20, Jul 2019.

[22] Oliver Kramer and Oliver Kramer. K-nearest neighbors. *Dimensionality reduction with unsupervised nearest neighbors*, pages 13–23, 2013.

[23] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani. Characterization of tor traffic using time based features. In *ICISSP 2017 – Proceedings of 3rd International Conference Information System Security and Privacy*, pages 253–262, January 2017.

[24] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[25] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24, 2013.

[26] Emaad Manzoor, Hemank Lamba, and Leman Akoglu. xstream: Outlier detection in feature-evolving data streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1963–1972, 2018.

[27] David Meyer and FT Wien. Support vector machines. *The Interface to libsvm in package e1071*, 28(20):597, 2015.

[28] Roberto Minerva, Abyi Biru, and Domenico Rotondi. Towards a definition of the internet of things (iot). *IEEE Internet Initiative*, 1(1):1–86, 2015.

[29] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.

[30] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[31] Euclides Carlos Pinto Neto, Sajjad Dadkhah, Raphael Ferreira, Alireza Zohourian, Rongxing Lu, and Ali A. Ghorbani. Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. *Sensors*, 23(13), 2023.

[32] Todd G Nick and Kathleen M Campbell. Logistic regression. *Topics in biostatistics*, pages 273–301, 2007.

[33] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[34] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[35] Juan Qiu, Qingfeng Du, and Chongshu Qian. Kpi-tsad: A time-series anomaly detector for kpi monitoring in cloud applications. *Symmetry*, 11(11):1350, 2019.

[36] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence. *Dataset Shift in Machine Learning*. Neural Information Processing series. MIT Press, 2008.

[37] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[38] Karen Scarfone, Peter Mell, et al. Guide to intrusion detection and prevention systems (idps). *NIST special publication*, 800(2007):94, 2007.

[39] Robert E Schapire. Explaining adaboost. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, pages 37–52. Springer, 2013.

[40] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[41] Yan-Yan Song and LU Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

[42] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[43] Lisa Torrey and Jude Shavlik. *Transfer Learning*, pages 242–264. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. IGI Global, Hershey, PA, USA, 2010.

[44] Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. Naïve bayes. *Encyclopedia of machine learning*, 15(1):713–714, 2010.

[45] Sanford Weisberg. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.

[46] Raymond E Wright. Logistic regression. 1995.

[47] Kangqing Yu, Wei Shi, and Nicola Santoro. Designing a streaming algorithm for outlier detection in data mining—an incremental approach. *Sensors*, 20(5):1261, 2020.