2023 IEEE International Conference on Edge Computing and Communications (EDGE) | 979-8-3503-0483-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/EDGE60047.2023.00063

Using Machine Learning for Detection and Classification of Cyber Attacks in Edge IoT

Elena Becker*, Maanak Gupta†, and Kshitiz Aryal‡
Department of Computer Science
Tennessee Tech University, Cookeville, TN, USA
*ebecker42@tntech.edu, †mgupta@tntech.edu, ‡karyal42@tntech.edu

***Abstract*—Internet of Things (IoT) devices are omnipresent due to their ease of use and level of connectivity. Because of wide deployment, IoT network traffic security is a large issue, especially as the devices become more common at the edge of the connected ecosystem. In general, low-powered IoT devices themselves are not inherently secure, so tailored security mechanisms are needed to make the ecosystem secure. The incorporation of the cloud also adds new security issues with the cloud service provider (CSP). In addition, several smart applications necessitate deploying edge-based infrastructure due to their real-time computation and communication requirements, while also having the ability to detect and mitigate different cyber attacks and remain light-weight. In this paper, we propose a machine learning-based approach to detect and classify different edge IoT network traffic driven cyber attacks, and evaluate their strengths and weaknesses. Particularly, we will compare eleven machine learning models to determine the best security agent trained for attack detection and classification on an edge IoT cyber security dataset with fourteen different attacks. We also provide experimental evaluation and analysis of our work, followed by our conclusion.**

***Index Terms*—Internet of Things, Edge IoT, Machine Learning, Attack Detection and Classification**

I. Introduction

The world of Internet of Things (IoT) is expanding fast, and IoT-related security concerns are becoming increasingly more important. The domain of IoT includes physical technological devices that contain various processing abilities, sensors, software, and connections to other tools and systems. Experts predict that by 2025, there will be approximately 40 billion connected IoT devices around the world, compared to about 19 billion in 2022 [1].

Cloud IoT involves cloud computing, which includes on-demand access to computational resources, data storage, tools, and internet applications hosted at a remote data center by a cloud service provider (CSP) [2]. It allows for lower IT costs, improved efficiency, and greater scalability [2]. Appropriate security of IoT devices and their CSP(s) is crucial to the continuation of the technology and security of data as a whole, as IoT devices are becoming more prevalent in our day-to-day lives, and hackers take any opportunity they can to steal or modify sensitive data. Unfortunately, cloud computing involves broader attack surfaces, leading to more ways for hackers to remotely access information while it is in transit or at rest, effectively decreasing the security of the data. This is also due to the data being controlled and accessed by a third party, the CSP. Cloud IoT is also not always able to perform calculations in real time, so sectors where time matters most struggle. Because of these vulnerabilities, and a growing need for more real-time data computation and communication, IoT systems are being connected to more ”local” edge devices.

Edge IoT, a part of IoT generally not involving the cloud, is made up of IoT devices that process data as close to the source device as possible, like with smart homes and glucose monitors. This allows for faster and more reliable services with lower latency, eliminating the time issue with cloud IoT. Edge IoT devices consist of physical hardware that is located at the edge of a network and can collect, process, and execute data in almost real-time with limited assistance from the cloud [17]. They are able to do this because they contain a greater amount of memory, processing power, and computing resources than normal IoT devices. Edge IoT is generally used where time matters most, whether this be for making more money in a business or saving someone’s life.

In general, Edge computing requires nodes that are near the end user, while cloud computing can be accessed from virtually anywhere. Because of this, the response time (latency) of edge is extremely low in comparison to possible hundreds of milliseconds for cloud computing. Processing for edge computing occurs at the edge nodes, while cloud processing occurs remotely. Cloud storage can be extremely large, however, edge computing storage is much more constrained. The operation environment for cloud computing is decided upon by the cloud operator. On the other hand, the edge computing environment is chosen by the customer. Edge IoT’s computing capability is much lower than that of cloud IoT, as the cloud has far more resources. Therefore, both cloud and edge computing have their uses. The cloud is for large tasks that do not need low latency, while edge computing is for tasks that require low latency, but do not necessarily need a lot of computational power in comparison.

Because IoT devices are being moved to the edge, there is a greater focus on their security at the edge layer itself. Even without the connection to the cloud, edge IoT devices possess vulnerabilities that need to be protected against. Some of the common attacks that edge IoT devices are still vulnerable to include: denial-of-service (DoS)/distributed denial-of-service (DDoS) attacks, information gathering attacks, man-in-the-middle (MITM) attacks, injection attacks, and malware attacks. To assist in the security of these edge IoT-driven

2767-9918/23/$31.00 ©2023 IEEE
DOI 10.1109/EDGE60047.2023.00063

ecosystems, attack detection and classification must be used and deployed to identify when an attack takes place and the specific type of cyber attack that is carried out. Furthermore, machine learning (ML) and neural network techniques can be used to train a model to recognize and classify different types of attacks, which can lead to a more accurate, timely prevention and response to growing cyber attacks at the edge.

In this paper, we review popular methods of securing IoT systems with both edge and cloud architecture. After analyzing the advantages and disadvantages of both architectures, we propose an intersection of using edge IoT network data and ML models for cyber attacks detection and classification. We chose a public edge IoT dataset, referred Edge-IIoTset Cyber Security Dataset of IoT & IIoT [18], with fourteen types of attacks, along with normal data, to compare eleven ML models and find the best ones for detecting the majority of the Edge IoT attacks.

The key contributions of this paper are:

1) We demonstrate that the network traffic data extracted from edge IoT devices can be used to secure edge IoT systems against attacks using machine learning.
2) We compare the performance of eleven machine learning algorithms in detecting and classifying fourteen different types of attacks.
3) We conjecture that machine learning models AdaBoost (AB), Random Forest (RF), and Decision Tree (DT) are successful in detecting and classifying the majority of attacks with high accuracy.

The rest of the paper is organized as follows: Section II covers an overview of related work and highlights limitations of existing approaches. Section III discusses the threat model and IoT architecture, different types of attacks detected and classified, problem definition, and proposed solution methodology. Section IV presents the implementation and testing procedure of our proposed approach, as well as results and discussion. Lastly, Section V covers the conclusion, a summary of findings, and future work.

II. Related Work

Figure 1 shows the categorization of the research problems and solution methodologies covered in the literature reviewed in this paper. These works are organized first based on if they were a survey or they covered a specific aspect of security. Further, they were separated into types of security methods: blockchain, edge computing, machine learning, and fog computing. Works in [7], [36]–[39], discusses multiple environments where solutions have been developed for cloud IoT, including smart cities, healthcare, and smart homes. Some of these include intrusion detection systems, access control models and architectures, the use of secure communication protocols, multi-factor authentication, predictive analysis of users activities, and identity-based encryption. These works bring various security issues to attention that pertain to cloud IoT, as a lot of data from IoT devices is stored in some form of cloud. Some of these problems include data security at rest and in transit, data loss, breaches, and data integrity.

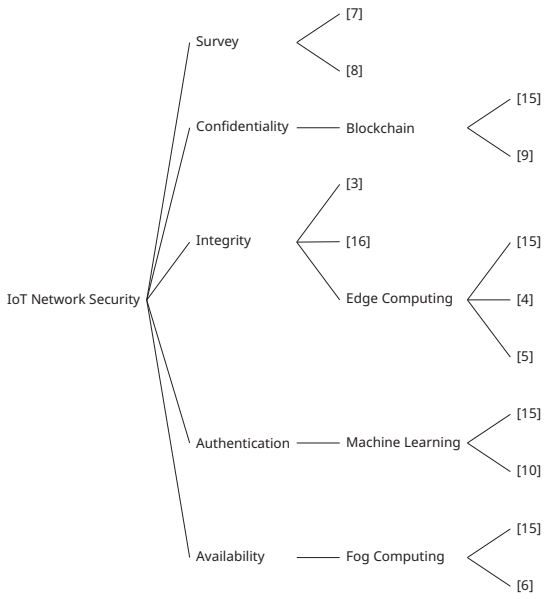

Fig. 1. IoT Network Security Tree Diagram

Authors in [8], discuss several cloud IoT security challenges. The first is data privacy, to which they propose a "User-driven Privacy Enforcement for Cloud-based Services in the IoT (UPECSI)", which utilizes model-driven privacy, interaction with user, and privacy enforcement points as its main components. The model-driven privacy focuses on integrating privacy within the cloud service itself, while users from various levels of access are offered interaction to understand their requirements and create a transparent ecosystem. Additionally, the privacy enforcement points, which reside in the IoT gateway, make sure user-defined policies are followed while data is outsourced to the cloud. Another approach they addressed was developers meeting the privacy requirements of the IoT device in the design phase before it is even launched. This approach entails privacy by design. The second security challenge defined is authentication and confidentiality, and this paper proposes a two-way authentication security scheme for IoT systems based on the Datagram Transport Layer Security (DTLS) protocol, which is placed between the transport and application layer. This scheme is backed by the RSA and designed for the 6LoWPANs. The third challenge is access control, for which this paper spotlights a method that identifies data holders and data collectors as the two types of actors. Data holders and collectors only receive data that they are required to and nothing more. Likewise, a data collector is also responsible for authenticating the data holders and points of origin. They highlight a cipher-text policy-based encryption approach for IoT data storage and secure access in the cloud. Their method also reduces storage overhead of public keys, and a user access control list (UACL) is created

to support authorization access per user. This paper promotes heterogeneity-supportive hybrid models to meet organizational needs for trust, privacy, context, knowledge, and not just subjects, objects, and the interactions between them.

A. Confidentiality

1) Blockchain: In [15], authors proposed IoT using blockchain, which is defined as data security with a distributed, decentralized, shared ledger where each entry is coupled with the previous using cryptographic hash keys. With the presence of these keys in each block, it makes it effectively too time-consuming for attackers to modify the data in the blocks. This paper also highlights the benefits of using blockchain with IoT devices. Data coming from IoT devices can be stored securely in blockchain, where the data can be encrypted using a hash key that is verified by miners. The use of blockchain can also help to prevent spoofing attacks and data loss through the use of its secure blocks. It can additionally prevent unauthorized access through the use of asymmetric encryption, where both public and private keys are used.

Work in [9] highlights that cloud includes security concerns like data protection, confidentiality, and data integrity, but software defined networking (SDN), with the addition of blockchain, can be used to solve those problems. SDN virtualizes the network by separating the data plane, where data traffic flows, and the control plane, where the network is managed. SDN allows for dynamic control of network throughput, individualizing device connection attributes, and enforcing individual network security policies. However, SDN has vulnerabilities to distributed denial of service (DDoS) attacks when it is merged with cloud storage. Blockchain can be added to combat this issue and enforce confidentiality in the network. The combination of cloud, SDN, and blockchain is termed ”Block-SDoTCloud”, where a distributed secure blockchain is created with a cloud storage environment based on SDN. The authors propose their solution architecture with five layers: perception layer, infrastructure layer/IoT networks, SDN layer, blockchain layer, and cloud layer. The perception layer can comprehend data from the real world and interact with the IoT application environment. It collects data in real-time and then passes it along with some form of identification. The infrastructure layer/IoT networks includes devices, like routers, switches, smart TVs, etc., that transmit data via SDN gateways. The SDN layer includes the gateways, southbound API, dynamic controllers, northbound API, and applications plane. It allows for mobility management, smart optimization, switching, routing, load balancing, reliability, and network monitoring. The blockchain layer includes the ledger of connected entries using cryptographic hash keys. It provides access control, confidentiality, tamper-resistance, and security to the data stored in each block. The cloud layer includes the cloud storage and secure transmission to and from it. With the addition of SDN and blockchain, this layer is effectively made more secure. Block-SDoTCloud model shows better throughput in comparison to OpenFlow-based SDN, and also provides protection against DDoS attacks.

B. Integrity

Work in [3], addresses the issue of security and trust complications being transferred from IoT devices to the cloud environment. This paper proposes the trust assessment framework for Cloud IoT. It includes the security-based trust assessment model, which covers various security features such as risk management, information security, and physical security. Another work [16], explores the trust issues of dynamic IoT cloud computing. The authors include the usage of both vertical and horizontal computing structures in the “extended IoT cloud”, where IoT devices, edge, fog, and cloud are integrated in a layered infrastructure. This paper includes a framework with a vertical IoT cloud for trustworthy computing. For dynamic IoT networks, authors created a policy definition model and advanced access control, called Resource and Role hierarchy Based Access Control (RRBAC).

1) Edge Computing: In [15], edge computing is described as where a small edge server is located between the user and the cloud or fog to process data closer to the user. This helps to keep the data local, making it less vulnerable to data tampering in transit. Additionally, edge computing has less data compliance issues. Data cleaning and aggregating can also be done at edge nodes, which allows for lower bandwidth if data needs to be sent somewhere. As discussed in [4], collecting large amounts of data from IoT devices and storing it in one core cloud infrastructure is not scalable in the long term. This paper proposes the “IoT-aware multilayer (packet/optical) transport software defined networking and edge/cloud orchestration architecture,” which dynamically distributes IoT edge processing based on the network resource state. It also contains a congestion avoidance mechanism and IoT-traffic control. The authors also utilize SDN-enabled containers to assess an edge node and integrate with the “IoT-aware SDN and cloud orchestration platform,”.

As discussed in [5], IoT devices are connected to the cloud through Message Queuing Telemetry Transport (MQTT). IoT uses WiFi, Bluetooth, cellular networks, and Ethernet. Radio frequency identification (RFID) allows wireless communication between devices. Scheduling tasks and load balancing is an issue in general networking, and it is just as much of an issue in edge computing. Because edge relies on databases, they are vulnerable to the same concerns. As highlighted, because edge devices are usually personal, trust and authentication is of importance. Edge devices are also concerned with privacy, digital security, and physical security.

C. Authentication

1) Machine Learning: The paper, [15], defines ML as training algorithms to detect anomalies or unwanted activity to increase security. This can be used in IoT devices to protect against different types of attacks. For instance, by using a multi-layer perceptron (MLP) discriminator, IoT networks can prevent denial of service (DoS) attacks, and by utilizing a Q-learning-based offloading strategy, like Dyna-Q, or non-parametric Bayesian techniques, an IoT network can be protected against eavesdropping. Additionally, privacy-preserving

	Domain			Machine Learning Models Used														Attacks Classified																		
Paper	Edge IoT Environment	Traditional Host-Based	Cloud IoT Environment	DNN	Random Forest	XGBoost	Naïve Bayes	Logistic	CART	RIPPER	Decision Tree	SVM	AdaBoost	KNN	ANN	BayesNet	DDoS UDP	DDoS ICMP	DDoS HTTP	DDoS TCP	SQL Injection	Uploading	Vulnerability Scanner	Password	Backdoor	Ransomware	XSS	Port Scanning	Fingerprinting	MITM	Scavenging	Spoofing	DoS	Fuzzing
Ferrag et al. 2022 [22]	✓			✓	✓						✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Sachdeva et al. 2022 [28]			✓		✓		✓								✓		✓	✓		✓														
Shakeele et al. 2021 [23]			✓					✓																						✓				
Elhoseny et al. 2021 [25]			✓	✓																														
Selim et al. 2021 [31]			✓				✓	✓	✓			✓		✓																	✓	✓		
Qadir et al. 2020 [19]		✓					✓				✓	✓		✓	✓														✓					
Bhandari et al. 2020 [29]		✓																																
Narwal et al. 2020 [33]		✓	✓																															
Soltani et al. 2019 [30]		✓																																
Lovanshi et al. 2019 [32]		✓	✓																															
Mohammad 2018 [21]		✓													✓																			
Babun et al. 2018 [24]			✓																															
Koroniotis et al. 2017 [26]			✓				✓				✓				✓						✓		✓		✓			✓			✓		✓	✓
Tallón-Ballesteros et al. 2014 [20]		✓							✓	✓	✓	✓		✓	✓	✓																		
Marturana et al. 2012 [27]			✓																															
Our Approach	✓			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				

TABLE I
DIFFERENCES AMONG RELATED WORKS WITH RESPECT TO ARCHITECTURE DOMAIN, MODELS, AND ATTACKS CLASSIFIED

scientific computations (PPSCs) can prevent privacy leakage on IoT devices, and digital fingerprinting can be adjusted to work with IoT devices with support vector machines (SVMs) and artificial neural networks (ANNs) to identify, track, and monitor the devices accurately. Authors in [10] propose a deep neural network (DNN) for IoT network security using an autoencoder neural network (AENN). By keeping in mind the causative, exploratory, and priority violation attacks, as well as the behavior of evasion, the AENN is able to predict intermediate attacks. Their network also circumvents high data analysis time and data overfitting by indicating if input samples cannot be used, and it works in an unsupervised environment, where unknown attacks can be predicted using learned parameters. By examining the network traffic before transmitting data, the AENN is able to prevent intermediate attacks early, as most cause the network to appear busy. Moreover, the AENN uses three phases to detect intermediate attacks: sensing, transmission, and feedback. They also utilized back-propagation learning to lessen deviations and minimize the false alarm rate. The proposed solution increased security while minimizing latency and decreased the number of intermediate attacks. Their AENN had above a 97% accuracy for detecting no attack, a jamming attack, data poisoning, and a priority violation attack.

D. Limitations of Literature

The literature explored show that there are many limitations in the current security of IoT ecosystems. Many of them propose solutions involving blockchain, fog computing, machine learning, or edge computing, but each of these have their own set of problems as well. Blockchain, for one, isn't scalable, is vulnerable to phishing, routing, and Sybil attacks, and generally has poor endpoint security [11]. Fog computing has issues in authentication, trust, privacy, and security [12]. Machine learning is vulnerable to adversarial attacks, data poisoning, online system manipulation, transfer learning attacks, data confidentiality and trustworthiness, reproducibility, and overfitting [13]. Lastly, edge computing has problems with data storage, backup, and protection, password risks and authentication, and perimeter defense [14].

Several works [19]–[23], [25], [26], [28], [32], [33] in the literature have demonstrated the use of machine learning-based solutions to detect and classify cyber attacks. However, most of them are either applied on a different domain, only detected selected attacks, or used a selected ML algorithm without comparing the performance of different models, as shown in Table I. This table shows (see the last row in red) how our approach is different in terms of domain and models used, as none of them cover both the edge IoT environment, as many models, or as many attacks as ours does.

This paper aims to address the following limitations:

1) Unlike traditional host-based approaches in [19]- [21], [30], [32], and [33], and a cloud environment in [23]- [28] and [31]- [33], we aim to focus on edge IoT systems to detect and classify cyber attacks.
2) Unlike the experiments done in [19], [20], [22], [23], [25], [26], [28], and [31], we use eleven different machine learning models to compare their performance.
3) Unlike the attack classification analysis done in [19], [23], [26], [28], and [31], we cover fourteen different attacks, as detailed in the next section.

III. SYSTEM ARCHITECTURE AND PROPOSED METHODOLOGY

A. Multi-layered IoT Architecture

Figure 2 displays a threat system model for an IoT system with edge and cloud layers. It depicts that a user sometimes interacts with both an edge and cloud user interface to use an IoT device. The edge interface connects the user to various

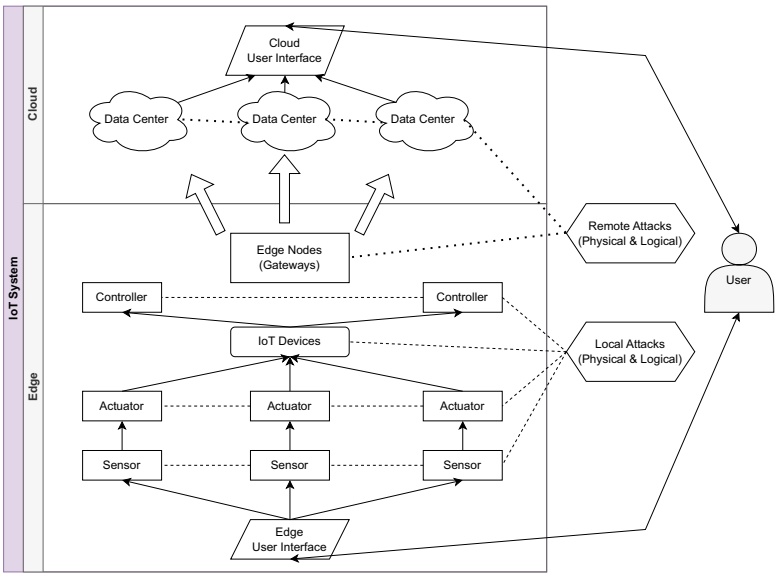

Fig. 2. Multi-layered IoT Architecture and Threat Model

sensors, actuators, and then eventually an IoT device and device controller. If the data collection, computation, and execution can be done all locally, then an edge node, or gateway, does not need to be used, and the system stays at the edge layer. At this layer, both physical and logical local attacks can be performed by malicious actors. However, if an edge IoT device cannot perform all of the computations locally, an edge node is used to send the data to a cloud data center, which assists in larger computations and executions. This brings in the cloud layer of the system, wherein more physical and logical remote attacks can occur. As more edge IoT driven use-cases are getting developed, it is important to deploy security mechanisms at the edge of the ecosystem.

B. Problem Definition and Threat Model

IoT devices are not inherently secure. Most IoT devices are small and are only made to collect data and send it out to the cloud. This means that data transfer is one of the most important parts of IoT to secure. Edge devices are different in that they are made to execute as many computations as possible locally, so they effectively remove the potential attack surfaces dealing with the cloud. This is why we chose to use an edge IoT dataset, as they are more secure without the connection to the cloud. Still, edge devices have their own vulnerabilities that attackers can exploit, so attack detection and classification can be carried out to know how to better secure them.

Our goal in this paper is to detect and classify the following cyber attacks orchestrated at the IoT edge layer [22].

1) *DDoS_UDP:* This is a DDoS attack where a large quantity of User Datagram Protocol (UDP) packets are transmitted to an edge server to overload it and prevent it from processing and responding to legitimate requests. Attackers will typically impersonate the UDP packets' source IP address.
2) *DDoS_ICMP:* This is a DDoS attack where the attacker floods the IoT device with Internet Control Message Protocol (ICMP) echo queries, or pings, to make it unable to respond to legitimate queries.
3) *Ransomware:* This is a type of malware attack where the attacker takes the IoT device or files hostage by restricting access and then demands a ransom to restore access.
4) *DDoS_HTTP:* This is a type of DDoS attack where the attacker floods the IoT server with Hypertext Transfer Protocol (HTTP) queries and causes it to overload.
5) *SQL_injection:* This is a type of injection attack where a Structured Query Language (SQL) query is modified by an injected query fragment that the attacker puts in, causing the attacker to gain access to the target IoT database and be able to alter the data .
6) *Uploading:* This is a type of injection attack where an attacker uploads a malware program file into a web server associated with an IoT system to gain administrative privileges.
7) *DDoS_TCP:* This is a type of DDoS attack where the attacker floods the IoT network with requests for Transmission Control Protocol (TCP) connections faster than it can handle, effectively disabling it.
8) *Backdoor:* This is a type of malware attack where the attacker takes advantage of vulnerabilities in the IoT system to gain unauthorized remote access. This then lets the attacker manage the IoT system's files, install software, and monitor the system as a whole.
9) *Vulnerability_scanner:* This is a type of information

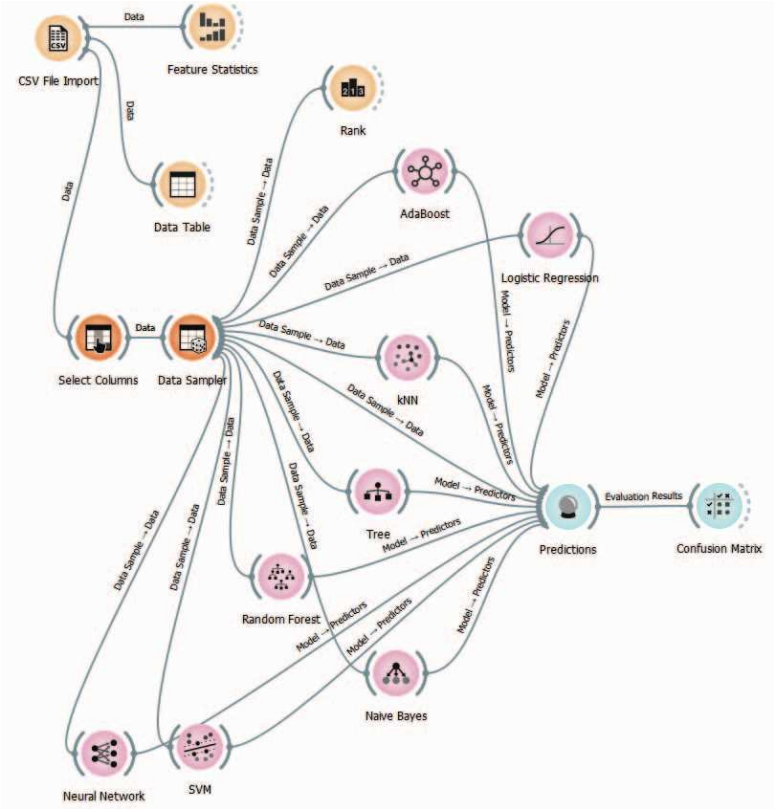

Fig. 3. System Architecture Diagram Developed in Orange

gathering attack where the IoT network is automatically scanned for environmental, internal, or external vulnerabilities.

10) *Port_Scanning:* This is a type of information gathering attack where IoT device ports are automatically scanned to find which are open or closed, and what security protocol they use if they have one.
11) *Cross-Site Scripting (XSS):* This is a type of injection attack where an attacker injects malicious scripts into a website associated with the IoT device, so they can gain access to vulnerable data.
12) *Password:* This is a type of malware attack where an attacker tries to find the password to an IoT device or system through successive attempts.
13) *Man-in-the-middle (MITM):* This is a type of attack that alters or hinders the communication between two legitimate points in the IoT network.
14) *Fingerprinting:* This is a type of information gathering attack where an attacker identifies the operating system (OS) of the target device to determine system vulnerabilities and map the remote network.

C. Proposed Approach

The proposed solution methodology combines edge computing and machine learning to identify IoT attacks and create a more secure method of transferring data for IoT devices. Edge computing was chosen specifically because many IoT devices that can perform quick computations locally with minimal cloud involvement are exploding in popularity, as they are more secure and have better processing speed.

TABLE II
CNN HEAD MODEL

Layer Number	Type	Activation	Number of Neurons
1	Dense	ReLU	256
2	Dense	ReLU	128
3	Dense	Softmax	15

The system architecture for our proposed approach, as designed by Orange [35], is shown in Figure 3. The specific types of detection and classification machine learning models that were compared for best performance were: Random Forest (RF), XGBoost (XGB), Gaussian Naive Bayes (GNB), Convolutional Neural Network (CNN) with a ResNet50 base [34] and a custom head model - as shown in Table II, Classification and Regression Tree (CART), AdaBoost (AB), K-Nearest Neighbors (KNN), Multilayer perceptron (MLP) - a type of Artificial Neural Network (ANN), Support Vector Machine (SVM), Logistic Regression (LR), and Decision Tree

TABLE III
MODEL HYPERPARAMETERS

Model	Hyperparameters
RF	num_trees = 10 num_attr_each_split = sqrt(70) smallest_subset_split = 5
XGB	objective = multi:softprob n_estimators = 100
GNB	n/a
CNN	optimizer = Adam learning_rate = 0.001 loss = categorical_crossentropy
CART	ccp_alpha = 0 criterion = gini min_samples_leaf = 1 min_samples_split = 2 splitter = best
AB	num_estimators = 50 learning_rate = 1.0 classif_algorithm = SAMME.R
KNN	num_neighbors = 5 metric = Euclidean weight = Uniform
ANN	neurons_per_hidden_layer = 100 activation = ReLU solver = Adam regularization = 0.0001 max_iter = 400 replicable _training = true
SVM	C = 1.0 kernel = RBF g = auto num_tolerance = 0.0010 iter_limit = 100
LR	regularization = L2 strength = 0.110
DT	induce_binary_tree = true min_num_instances_in_leaves = 2 smallest_subset_split = 5 max_tree_depth = 100 stop_when_maj_reaches (%) = 95

(DT). The hyperparameters used for these models are outlined in Table III. The Edge-IIoTset Cyber Security Dataset of IoT & IIoT was used for experimentation [18]. This dataset contains both normal and attack IoT network traffic data.

IV. EXPERIMENT

A. Implementation and Testing

To preprocess the data, each CSV file was first uploaded to Google Colaboratory, consisting of fourteen attacks and thirteen types of normal IoT network traffic data. The attacks included were: DDoS_UDP, DDoS_ICMP, Ransomware, DDoS_HTTP, SQL_injection, Uploading, DDoS_TCP, Backdoor, Vulnerability_scanner, Port_Scanning, Cross-Site Scripting (XSS), Password, Man-in-the-middle (MITM), and Fingerprinting. The normal data consisted of sensors that detected the following: sound, temperature, humidity, ultrasonic waves, water level, pH level, soil moisture, heart rate, flame level, servo motor speed, stepper motor speed, DC motor speed, and IR receiver frequencies. Next, the CSV files were converted to data frames, and random samples were taken from each of them before combining them all into one data frame. We

```
Normal                    24101
DDoS_UDP                  14498
DDoS_ICMP                 13096
DDoS_HTTP                 10495
SQL_injection             10282
DDoS_TCP                  10247
Uploading                 10214
Vulnerability_scanner     10062
Password                   9972
Backdoor                   9865
Ransomware                 9689
XSS                        9543
Port_Scanning              8921
Fingerprinting              853
MITM                        358
Name: Attack_type, dtype: int64
```

Fig. 4. Preprocessed Class Numbers

		#	Gain ratio
1	C http.request.method-0.0	2	1.000
2	C mqtt.topic-0.0	2	1.000
3	C mqtt.protoname-0.0	2	1.000
4	C mqtt.conack.flags-0.0	2	1.000
5	C http.referer-0.0	2	1.000
6	C http.request.version-0.0	2	1.000
7	C dns.qry.name.len-0.0	2	0.997
8	C dns.qry.name.len-0	2	0.997
9	C http.referer-0	2	0.982
10	C mqtt.topic-0	2	0.925
11	C mqtt.protoname-0	2	0.923
12	C mqtt.conack.flags-0	2	0.922
13	C dns.qry.name.len-1.0	2	0.861
14	C http.request.version-0	2	0.826
15	C http.request.method-0	2	0.826
16	N udp.time_delta		0.784
17	N udp.stream		0.747
18	N icmp.checksum		0.745
19	N icmp.seq_le		0.674
20	N dns.qry.qu		0.672
21	N http.content_length		0.598
22	C tcp.flags.ack	2	0.491
23	N tcp.flags		0.473
24	C http.request.version-HTTP/1.1	2	0.451
25	C http.request.method-GET	2	0.421
26	N tcp.ack		0.410
27	C http.request.version-HTTP/1.0	2	0.404
28	N dns.qry.name		0.400
29	C http.response	2	0.389
30	N tcp.seq		0.383

Fig. 5. Top 30 Data Features - (C) Categorical and (N) Numerical

eradicated features that were either unnecessary for attack detection and classification - like full URLs or messages -

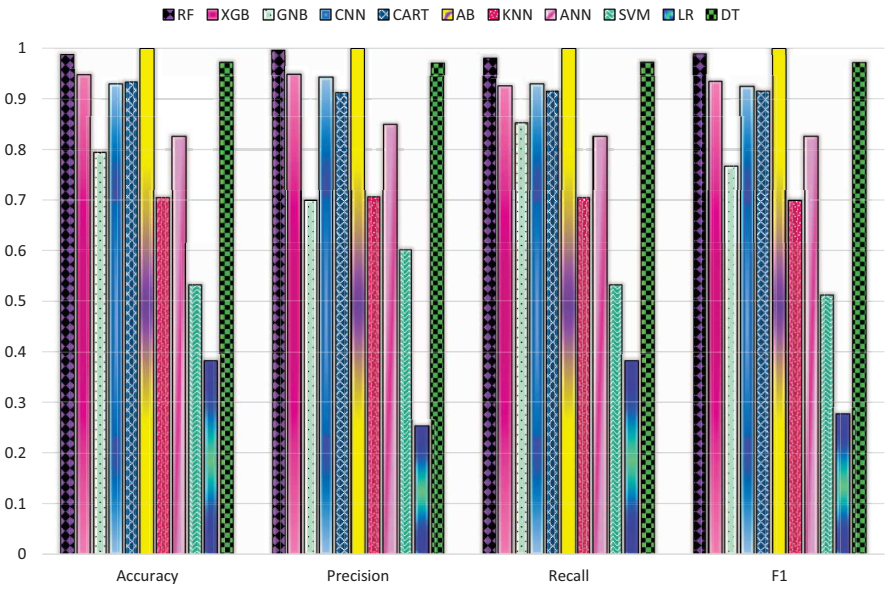

Fig. 6. Model Results Comparison

or had all zeroes or null data. Figure 4 shows the normal data and attack numbers after preprocessing the data. This ended up with the data having a total of 70 unique features, with the top 30 being shown in Figure 5. The top features were found through the ranking in Orange and feature importance functions in the RF and XGB models. After the data was cleaned, we conducted a random grid search to determine the hyperparameters - shown in Table III. Then the data was split into 70% training, 10% validation, and 20% testing. To assess, we used the Scikit-Learn library in Python on Google Colaboratory Pro (Colab) with 25.5GB for system RAM and 166.8GB of disk space for five of the models, and the machine learning tool, Orange, for eight of them. The RF and GNB models were done in both Colab and Orange, so the best-performing one was chosen for the results. Furthermore, the models were evaluated based on accuracy, precision, recall, and F1 scores. The equations below include shorthand for the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

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

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

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

$$F1\ Score\ = \frac{2TP}{2TP + FP + FN}$$

TABLE IV
DETAILED MODEL RESULTS

Model	Accuracy	Precision	Recall	F1
RF	0.984	0.993	0.979	0.986
XGB	0.945	0.949	0.925	0.933
GNB	0.792	0.698	0.850	0.766
CNN	0.928	0.941	0.928	0.923
CART	0.931	0.910	0.913	0.912
AB	0.999	0.999	0.999	0.999
KNN	0.703	0.704	0.703	0.698
ANN	0.825	0.848	0.825	0.825
SVM	0.531	0.600	0.531	0.511
LR	0.380	0.251	0.380	0.275
DT	0.971	0.970	0.971	0.970

B. Results

Figure 6 shows the statistical evaluation results for each model, and Table IV presents a more detailed version. Figures 7, 8, and 9 in Appendix A display the confusion matrices for the AB, RF, and DT models, respectively.

C. Discussion

Each of the models performed relatively well in identifying the Normal data in the dataset, except for the LR model, which had an accuracy of 21.2%. However, the attack data identifications were more varied. The AB model produced the best results for the dataset overall with an average F1 score and accuracy of 99.9%. The RF model came in second with an average F1 score of 98.6% and an accuracy of 98.4%. The DT model was a close third with an average 97.0% F1 score and 97.1% accuracy. The rest of the models achieved

similar results above an average of 70.0% for the F1 score and accuracy, except for the KNN, SVM, and LR models. The LR model was the worst-performing model, with an average F1 score of 27.5% and an accuracy of 38.0%. The KNN model achieved mediocre results due to the scale of the data, and the SVM model didn't perform well due to the target classes overlapping in some areas and the dataset being fairly big. The reason for the LR model's poor performance was mainly due to its issue with discrete values, as it normally handles continuous values better. In addition, the classification problem deals with higher dimensions that are not best suited for SVM and LR.

As for the specific classes, the Fingerprinting attacks were among the highest misclassified attacks across all models. The AB model performed the best on them, with a 99.3% accuracy for correct classifications, while the RF model had 87.3% and the DT model a 16.0%. The Uploading attacks were another of the highest misclassified, with the AB model having an accuracy of 99.7%, the RF model a 96.3%, and the DT model a 92.0%. In contrast, the DDoS_UDP and MITM attacks were among the highest correctly classified attacks across all of the models, with over half of them scoring 100.0% accuracy.

It was noted that different models performed better or worse on different attacks, regardless of the overall accuracy recorded. Additionally, our proposal was limited to analyzing static IoT network traffic data, rather than live data, and using only one dataset. In all, an AB model is the best-performing choice out of the eleven for edge IoT device attack detection and classification of the attacks in this dataset.

V. Conclusion and Future Work

IoT is a rapidly growing sector in the technology field, and there are a lot of security concerns that come along with this growth. IoT devices are everywhere and affect everyone, whether someone is exercising in the park with their smartwatch, working at a smart industrial plant, or cooking in their smart kitchen. These devices constantly give and receive huge amounts of sensitive data, so their security is a definite concern. The addition of cloud and cloud service providers (CSPs) adds even more security vulnerabilities due to the level of connectivity that the nature of a cloud network provides. Multi-tenancy, misconfiguration, and CSP trust are all common issues that can lead to security issues with IoT devices in a cloud environment.

After looking over each of these papers, we are even more aware of the issues that cloud IoT, specifically, brings to the realm of technology. Sensitive data is more at risk of exposure due to the distance it travels between the IoT device and the cloud server. Proper network-layer protocols and encryption must be used to ensure secure data in transit. Additionally, there is a loss of control when a client begins to use a CSP to manage and access their IoT device data. In most cases, the provider has a large amount of control over the data storage and even the software used for the device, so a lot of the system transactions happen behind the scenes and away from the client's eyes. This brings in the concept of provider trust, in which the client has to trust their provider to handle their information with the best and most up-to-date security procedures, so that the data is not modified, deleted, or stolen. Even without adding the cloud, many issues exist within IoT networks, like confidentiality, integrity, authentication, availability, and scalability. And, with edge IoT devices especially, the security impact can be disastrous. With all of these issues, many of the papers reviewed proposed framework solutions to mitigate some vulnerabilities in and out of the cloud for IoT devices. Unfortunately, there will never be a catch-all solution to the problems of cloud and IoT, due to the nature of the devices and the cloud environment itself, but the frameworks shown are helpful ways to protect sensitive information, nonetheless.

In this work, we compared eleven different machine learning models with edge IoT network traffic data, including both diverse normal data and attacks, as a way to combine the aspects of machine learning and the edge to better secure the data. The machine learning models were used to assist in determining the best attack detection and classification model for fourteen different attacks. Through our experiments, we determined that an AdaBoost model is the best-performing model choice out of the eleven tested for the dataset used.

For future work, the results obtained in this paper will be fed through several digital forensics tools to determine the best one for attack reconstruction with anti-forensics in mind. Additional future work could include using multiple datsets and live data analysis.

Acknowledgements

This work is partially supported by the DoD Cybersecurity Scholarship H98230-22-1-0248 and the NSF grant 2025682.

References

[1] "IoT Security Statistics (2022): What You Should Know," Intersog, 03-Dec-2021. [Online]. Available: https://intersog.com/blog/iot-security-statistics/.

[2] S. Vennam, "What is Cloud Computing?," IBM Cloud Learn Hub, 18-Aug-2020. [Online]. Available: https://www.ibm.com/cloud/learn/cloud-computing.

[3] X. Li, Q. Wang, X. Lan, X. Chen, N. Zhang and D. Chen, "Enhancing Cloud-Based IoT Security Through Trustworthy Cloud Service: An Integration of Security and Reputation Approach," in IEEE Access, vol. 7, pp. 9368-9383, 2019, doi: 10.1109/ACCESS.2018.2890432.

[4] R. Muñoz et al., "Integration of IoT, Transport SDN, and Edge/Cloud Computing for Dynamic Distribution of IoT Analytics and Efficient Use of Network Resources," in Journal of Lightwave Technology, vol. 36, no. 7, pp. 1420-1428, 1 April1, 2018, doi: 10.1109/JLT.2018.2800660.

[5] S. Naveen and M. R. Kounte, "Key Technologies and challenges in IoT Edge Computing," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, pp. 61-65, doi: 10.1109/I-SMAC47947.2019.9032541.

[6] M. Aleisa, A. A. Hussein, F. Alsubaei and F. T. Sheldon, "Performance Analysis of Two Cloud-Based IoT Implementations: Empirical Study," 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 2020, pp. 276-280, doi: 10.1109/CSCloud-EdgeCom49738.2020.00055.

[7] N. Almolhis, A. M. Alashjaee, S. Duraibi, F. Alqahtani and A. N. Moussa, "The Security Issues in IoT - Cloud: A Review," 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), 2020, pp. 191-196, doi: 10.1109/CSPA48992.2020.9068693.

[8] I. Mohiuddin and A. Almogren, "Security Challenges and Strategies for the IoT in Cloud Computing," 2020 11th International Conference on Information and Communication Systems (ICICS), 2020, pp. 367-372, doi: 10.1109/ICICS49469.2020.239563.

[9] A. Rahman, M. J. Islam, M. Saikat Islam Khan, S. Kabir, A. I. Pritom and M. Razaul Karim, "Block-SDoTCloud: Enhancing Security of Cloud Storage through Blockchain-based SDN in IoT Network," 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), 2020, pp. 1-6, doi: 10.1109/STI50764.2020.9350419.

[10] Ali, Saif Mohammed, Elameer, Amer S. and Jaber, and Mustafa Musa, "IoT network security using autoencoder deep neural network and channel access algorithm," Journal of Intelligent Systems, vol. 31, no. 1, 2022, pp. 95-103. https://doi.org/10.1515/jisys-2021-0173

[11] M. Shah, "5 blockchain security issues and how to prevent them," Fast Company, 15-Feb-2022. [Online]. Available: https://www.fastcompany.com/90722111/5-blockchain-security-issues-and-how-to-prevent-them#: :text=ROUTING%20ATTACKS,transmitted%20to%20internet%20service%20providers.

[12] B. Curtis, "5 challenges that come with fog computing," YourTechDiet, 07-Dec-2021. [Online]. Available: https://yourtechdiet.com/blogs/fog-computing-issues/.

[13] G. McGraw, R. Bonett, V. Shepardson and H. Figueroa, "The Top 10 Risks of Machine Learning Security" in Computer, vol. 53, no. 06, pp. 57-61, 2020. doi: 10.1109/MC.2020.2984868 Available: https://doi.ieeecomputersociety.org/10.1109/MC.2020.2984868

[14] T. Nolle, "Edge computing security risks and how to overcome them," IoT Agenda, 08-Nov-2021. [Online]. Available: https://www.techtarget.com/iotagenda/tip/Edge-computing-security-risks-and-how-to-overcome-them.

[15] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," in IEEE Access, vol. 7, pp. 82721-82743, 2019, doi: 10.1109/ACCESS.2019.2924045.

[16] I. Yen, F. Bastani, N. Solanki, Y. Huang and S. Hwang, "Trustworthy Computing in the Dynamic IoT Cloud," 2018 IEEE International Conference on Information Reuse and Integration (IRI), 2018, pp. 411-418, doi: 10.1109/IRI.2018.00067.

[17] "What is Iot edge computing?," Red Hat - We make open source technologies for the enterprise, 29-Jul-2022. [Online]. Available: https://www.redhat.com/en/topics/edge-computing/iot-edge-computing-need-to-work-together.

[18] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," in IEEE Access, vol. 10, pp. 40281-40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[19] A. M. Qadir and A. Varol, "The Role of Machine Learning in Digital Forensics," 2020 8th International Symposium on Digital Forensics and Security (ISDFS), 2020, pp. 1-5, doi: 10.1109/ISDFS49300.2020.9116298.

[20] A. J. Tallón-Ballesteros and J. C. Riquelme, "Data mining methods applied to a digital forensics task for supervised machine learning," in Computational Intelligence in Digital Forensics: Forensic Investigation and Applications, Springer, 2014, pp. 413–428.

[21] R. M. Mohammad, "A Neural Network based Digital Forensics Classification," 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), 2018, pp. 1-7, doi: 10.1109/AICCSA.2018.8612868.

[22] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," in IEEE Access, vol. 10, pp. 40281-40306, 2022, doi: 10.1109/ACCESS.2022.3165809.

[23] P. M. Shakeel, S. Baskar, H. Fouad, G. Manogaran, V. Saravanan, and C. E. Montenegro-Marin, "Internet of things forensic data analysis using machine learning to identify roots of data scavenging," Future Generation Computer Systems, vol. 115, pp. 756–768, 2021, doi: https://doi.org/10.1016/j.future.2020.10.001.

[24] L. Babun, A. K. Sikder, A. Acar, and A. S. Uluagac, "Iotdots: A digital forensics framework for smart environments," arXiv preprint arXiv:1809.00745, 2018.

[25] M. Elhoseny, M. M. Selim, and K. Shankar, "Optimal deep learning based convolution neural network for digital forensics face sketch synthesis in internet of things (IoT)," International Journal of Machine Learning and Cybernetics, vol. 12, no. 11, pp. 3249–3260, 2021.

[26] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques," in International Conference on Mobile Networks and Management, 2017, pp. 30–44.

[27] F. Marturana, G. Me and S. Tacconi, "A Case Study on Digital Forensics in the Cloud," 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2012, pp. 111-116, doi: 10.1109/CyberC.2012.26.

[28] S. Sachdeva and A. Ali, "Machine learning with digital forensics for attack classification in cloud network environment," International Journal of System Assurance Engineering and Management, vol. 13, no. 1, pp. 156–165, 2022.

[29] S. Bhandari and V. Jusas, "An Abstraction Based Approach for Reconstruction of TimeLine in Digital Forensics," Symmetry, vol. 12, no. 1, 2020, doi: 10.3390/sym12010104.

[30] S. Soltani and S. A. H. Seno, "A formal model for event reconstruction in digital forensic investigation," Digital Investigation, vol. 30, pp. 148–160, 2019, doi: https://doi.org/10.1016/j.diin.2019.07.006.

[31] I. Selim Gamal Eldin et al, "Anomaly events classification and detection system in critical industrial internet of things infrastructure using machine learning algorithms," Multimedia Tools Appl, vol. 80, (8), pp. 12619-12640, 2021. Available: https://ezproxy.tntech.edu/login?url=https://www.proquest.com/scholarly-journals/anomaly-events-classification-detection-system/docview/2513419623/se-2. DOI: https://doi.org/10.1007/s11042-020-10354-1.

[32] Lovanshi, M., Bansal, P. (2019). Comparative Study of Digital Forensic Tools. In: Shukla, R.K., Agrawal, J., Sharma, S., Singh Tomer, G. (eds) Data, Engineering and Applications. Springer, Singapore. https://doi.org/10.1007/978-981-13-6351-1_15

[33] B. Narwal and Nimisha Goel, "A Walkthrough of Digital Forensics and its Tools," 2020. [Online]. Available: https://www.researchgate.net/profile/Bhawna-Narwal/publication/339674292_A_Walkthrough_of_Digital_Forensics_and_its_Tools/links/5eeb2e5092851ce9e7ec93a9/A-Walkthrough-of-Digital-Forensics-and-its-Tools.pdf.

[34] A. H. Anselmoo, Q. S. Zhu, H. Jin, T. Lee, and F. Chollet, "Keras Documentation: Resnet and RESNETV2," Keras, 24-Aug-2022. [Online]. Available: https://keras.io/api/applications/resnet/#resnet50-function.

[35] Demsar J, Curk T, Erjavec A, Gorup C, Hocevar T, Milutinovic M, Mozina M, Polajnar M, Toplak M, Staric A, Stajdohar M, Umek L, Zagar L, Zbontar J, Zitnik M, Zupan B 2013 Orange: Data Mining Toolbox in Python, Journal of Machine Learning Research 14Aug: 2349-2353.

[36] Gupta, M., Benson, J., Patwa, F. & Sandhu, R. Dynamic groups and attribute-based access control for next-generation smart cars. *Proceedings Of The Ninth ACM Conference On Data And Application Security And Privacy*. pp. 61-72 (2019)

[37] Gupta, M., Benson, J., Patwa, F. & Sandhu, R. Secure V2V and V2I Communication in Intelligent Transportation using Cloudlets. *IEEE Transactions On Services Computing*. (2020)

[38] Gupta, M. & Sandhu, R. Towards Activity-Centric Access Control for Smart Collaborative Ecosystems. *Proceedings Of The ACM Symposium On Access Control Models And Technologies (SACMAT - 2021)*. (2021)

[39] Cathey, G., Benson, J., Gupta, M. & Sandhu, R. Edge Centric Secure Data Sharing with Digital Twins in Smart Ecosystems. *IEEE International Conference On Trust, Privacy And Security In Intelligent Systems, And Applications*. (2021)

APPENDIX A
CONFUSION MATRICES FOR TOP PERFORMING MODELS

		Predicted Backdoor	DDoS_HTTP	DDoS_ICMP	DDoS_TCP	DDoS_UDP	Fingerprinting	MITM	Normal	Password	Port_Scanning	Ransomware	SQL_injection	Uploading	Vulnerability_scanner	XSS	∑
Actual	Backdoor	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	6888
	DDoS_HTTP	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	7383
	DDoS_ICMP	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	9194
	DDoS_TCP	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	7149
	DDoS_UDP	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10035
	Fingerprinting	0.3 %	0.0 %	0.0 %	0.0 %	0.0 %	99.3 %	0.0 %	0.0 %	0.0 %	0.2 %	0.2 %	0.0 %	0.0 %	0.0 %	0.0 %	599
	MITM	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	235
	Normal	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	16888
	Password	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	6904
	Port_Scanning	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	6269
	Ransomware	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	99.9 %	0.0 %	0.0 %	0.0 %	0.0 %	6835
	SQL_injection	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.1 %	0.0 %	0.0 %	99.9 %	0.0 %	0.0 %	0.0 %	7210
	Uploading	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.2 %	0.0 %	0.0 %	0.1 %	99.7 %	0.0 %	0.0 %	7170
	Vulnerability_scanner	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	7142
	XSS	0.0 %	0.2 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	99.7 %	6637
	∑	6894	7400	9193	7149	10035	598	235	16888	6922	6271	6829	7211	7151	7142	6620	106538

Fig. 7. AB Model Confusion Matrix

		Predicted Backdoor	DDoS_HTTP	DDoS_ICMP	DDoS_TCP	DDoS_UDP	Fingerprinting	MITM	Normal	Password	Port_Scanning	Ransomware	SQL_injection	Uploading	Vulnerability_scanner	XSS	∑
Actual	Backdoor	98.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.1 %	0.0 %	0.0 %	0.0 %	1.0 %	0.9 %	0.0 %	0.0 %	0.0 %	0.0 %	6888
	DDoS_HTTP	0.0 %	98.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.1 %	1.9 %	7383
	DDoS_ICMP	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	9194
	DDoS_TCP	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	7149
	DDoS_UDP	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10035
	Fingerprinting	1.7 %	0.0 %	0.0 %	0.0 %	0.0 %	87.3 %	0.0 %	0.0 %	0.0 %	6.2 %	4.8 %	0.0 %	0.0 %	0.0 %	0.0 %	599
	MITM	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	235
	Normal	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	16888
	Password	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	95.3 %	0.0 %	0.0 %	1.9 %	2.8 %	0.0 %	0.0 %	6904
	Port_Scanning	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	6269
	Ransomware	0.7 %	0.0 %	0.0 %	0.0 %	0.0 %	0.1 %	0.0 %	0.0 %	0.0 %	1.3 %	98.0 %	0.0 %	0.0 %	0.0 %	0.0 %	6835
	SQL_injection	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1.8 %	0.0 %	0.0 %	95.7 %	2.5 %	0.0 %	0.0 %	7210
	Uploading	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1.7 %	0.0 %	0.0 %	2.0 %	96.3 %	0.0 %	0.0 %	7170
	Vulnerability_scanner	0.0 %	1.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	98.2 %	0.8 %	7142
	XSS	0.0 %	3.1 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.1 %	96.9 %	6637
	∑	6807	7514	9193	7149	10035	536	235	16888	6832	6458	6791	7179	7273	7022	6626	106538

Fig. 8. RF Model Confusion Matrix

		Predicted Backdoor	DDoS_HTTP	DDoS_ICMP	DDoS_TCP	DDoS_UDP	Fingerprinting	MITM	Normal	Password	Port_Scanning	Ransomware	SQL_injection	Uploading	Vulnerability_scanner	XSS	∑
Actual	Backdoor	98.3 %	0.0 %	0.0 %	0.0 %	0.0 %	0.1 %	0.0 %	0.0 %	0.0 %	0.6 %	0.9 %	0.0 %	0.0 %	0.0 %	0.0 %	6888
	DDoS_HTTP	0.0 %	97.7 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.2 %	2.0 %	7383
	DDoS_ICMP	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	9194
	DDoS_TCP	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	7149
	DDoS_UDP	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	10035
	Fingerprinting	5.0 %	0.0 %	68.6 %	0.0 %	0.0 %	16.0 %	0.0 %	0.0 %	0.0 %	6.0 %	4.3 %	0.0 %	0.0 %	0.0 %	0.0 %	599
	MITM	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	235
	Normal	0.0 %	0.0 %	0.0 %	0.0 %	0.2 %	0.0 %	0.0 %	99.8 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	16888
	Password	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	95.8 %	0.0 %	0.0 %	1.4 %	2.8 %	0.0 %	0.0 %	6904
	Port_Scanning	0.7 %	0.0 %	0.0 %	0.0 %	0.0 %	0.4 %	0.0 %	0.0 %	0.0 %	98.8 %	0.1 %	0.0 %	0.0 %	0.0 %	0.0 %	6269
	Ransomware	1.3 %	0.0 %	0.0 %	0.0 %	0.0 %	0.4 %	0.0 %	0.0 %	0.0 %	1.5 %	96.8 %	0.0 %	0.0 %	0.0 %	0.0 %	6835
	SQL_injection	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	4.2 %	0.0 %	0.0 %	93.9 %	1.8 %	0.0 %	0.0 %	7210
	Uploading	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	4.1 %	0.0 %	0.0 %	3.9 %	92.0 %	0.0 %	0.0 %	7170
	Vulnerability_scanner	0.0 %	1.4 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	96.4 %	2.1 %	7142
	XSS	0.0 %	4.6 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1.2 %	94.2 %	6637
	∑	6931	7621	9612	7147	10061	156	235	16860	7212	6373	6717	7150	6922	6984	6557	106538

Fig. 9. DT Model Confusion Matrix