Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.038043
Article

Mirai Botnet Attack Detection in Low-Scale Network Traffic

Ebu Yusuf GÜVEN and Zeynep GÜRKAŞ-AYDIN*

Department of Computer Engineering, Istanbul University-Cerrahpasa, Istanbul, Turkey
*Corresponding Author: Zeynep GÜRKAŞ-AYDIN. Email: zeynepg@iuc.edu.tr
Received: 24 November 2022; Accepted: 16 January 2023

Abstract: The Internet of Things (IoT) has aided in the development of new products and services. Due to the heterogeneity of IoT items and networks, traditional techniques cannot identify network risks. Rule-based solutions make it challenging to secure and manage IoT devices and services due to their diversity. While the use of artificial intelligence eliminates the need to define rules, the training and retraining processes require additional processing power. This study proposes a methodology for analyzing constrained devices in IoT environments. We examined the relationship between different sized samples from the Kitsune dataset to simulate the Mirai attack on IoT devices. The training and retraining stages for the Mirai attack were also evaluated for accuracy. Various approaches are evaluated in smaller sample sizes to minimize training time on low-resource devices. Cross-validation was used to avoid overfitting classification methods during the learning process. We used the Bootstrapping technique to generate 1000, 10000, and 100000 samples to examine the performance metrics of different-sized variations of the dataset. In this study, we demonstrated that a sample size of 10000 is sufficient for 99,56% accuracy and learning in the detection of Mirai attacks in IoT devices.

Keywords: Mirai; internet of things; low-scale traffic; machine learning; intrusion detection

1 Introduction

Internet of Things (IoT) ecosystem consists of connected devices designed with low power consumption, low-cost processors, and memory source limited to only performing several dedicated activities. IoT products and services appear in various fields like wearable technologies, smart vehicles, houses, factories, and cities. These products and services reveal several privacy and security problems because they cannot execute traditional security procedures due to inadequate facilities. As a consequence of the insufficient software security protection because of the constrained resources, this intelligent world became the attackers' first target. Because of smart devices' vulnerabilities, governments and end-users are concerned that cyber attackers can transform them into cyber weapons. The probability of causing harm to people using intelligent systems, which can control actual objects like smart door locks, increases the concerns of the end-user. While this technology is becoming more widely used, it is also becoming more vulnerable to cyber-attacks [1].

IoT confronts the danger of being taken over and exploited by the attackers besides its provided facilities. Attackers are capable of transforming innocent devices into weapons. It is possible to carry out a Distributed Denial of Service (DDoS) attack anywhere globally through smart devices seized by the attacker. Mirai is a DDOS attack method that makes use of the default credential. Mirai malware was first detected in August 2016 by a research group [2]. Mirai malware can turn Linux-running network smart devices into remotely controlled bots as a part of a botnet dedicated for large-scale network attacks. It firstly targets online consumer devices like Internet Protocol (IP) cameras and house routers. Mirai was responsible for some of the largest and most destructive DDoS attacks in recent years.

In September 2016, the Mirai Botnet's attackers launched their first attacks against cybersecurity writer Brian Krebs' website and French web hosting company OVH Groupe SAS (OHV) [3]. In October 2016, the attack in the United States of America targeting the DYN company, one of the most significant DDoS attacks in information technologies (IT) history, was carried out by hacking widely used home and business-type IP cameras and redirecting traffic to DYN'S Domain Name Service (DNS) servers, resulting in millions of dollars in losses. A massive DDoS attack to date stemmed from simple weaknesses in IoT devices, prompting governments to adopt legal cybersecurity frameworks. Consumers were also worried about the law draft for which the owners of the devices used in these attacks are responsible.

Mirai malware-infected devices could be quickly detected and blocked at the source of the local area network if the network had an anomaly-based intrusion detection system for IoT devices. Network intrusion detection systems are available to detect attacks from devices on the network. There are three main types of intrusion detection systems for network security infrastructures: network intrusion systems (NIDS), host-based intrusion detection systems (HIDS), and distributed intrusion detection systems (DIDS) [4]. These systems aim to detect malicious activities like intrusions of service traffic (Denial of Service-DoS) and port scans or attack attempts to connected devices by monitoring the network traffic [5]. Besides controlling the network traffic, NIDS can extract valuable data from outgoing and ongoing attacks on the local traffic. So far, researchers have designed rule-based and machine learning [5], fuzzy clustering theory [6], artificial neural networks [7–9] based NIDSs to detect suspicious connections. IoT has brought specific problems in attack detection and prevention compared to traditional networks [10,11]. The rule definition is required when considering the task and function, as IoT devices can behave differently in rule-based NIDS systems [12]. Also, analysis and control rules must be checked and updated whenever a new device is attached to the system. In machine learning-based NIDSs, the learning task requires labeled communication data [13]. IoT applications' non-generalizable communication data from the various and limited heterogeneous network structures makes implementing supervised methods difficult. It is possible to detect these attacks by intrusion detection systems designed for IoT.

Labeling data vectors belonging to different classes is possible using Artificial Intelligence. Attack detection is a dual or multi-class classification problem [14,15]. It determines the network traffic behavior as normal or abnormal and detects the attack type of abnormal behavior [7]. The main motivation of attack detection is to increase the accuracy of classifiers' analysis. The majority of research employs data mining and artificial intelligence techniques in the design of NIDS. The majority of research utilizes resource-consuming data mining and artificial intelligence techniques to design effective IDSs. However, it is not preferred to train artificial intelligence models on smart devices. Within the scope of the study, we provided faster model training by utilizing less processing power to generate small samples from a larger dataset using the Bootstrapping method. The network traffic generated by the Mirai malware is classified using artificial intelligence techniques on the Kitsune

dataset shared in 2018 [16], which was extracted from real smart devices. On the personal computer, we calculated learning time and performance metrics by creating sub datasets from the Mirai attack in the Kitsune dataset. The objective was to determine the optimal amount of resources and working hours by comparing various performance metrics. Using Artificial Neural Networks (ANN), Support Vector Machines (SVM), and K-Nearest Neighbor (K-NN) algorithms, we detected the Mirai attack in the Kitsune dataset and compared performance metrics. Our study also offers an opportunity for model training and testing on smart objects in the future.

In this study, the second section discusses similar studies and existing work, the third section explains the Mirai malware in-depth, the fourth section refers to the methodology used, and the fifth section describes the Kitsune Dataset utilized by the proposed system. Finally, the article concludes with the results of the study and the conclusion.

2 Related Work

Many protocols that contribute to the maintenance of the Internet infrastructure introduce numerous protocol security risks. Due to the difficulty of updating all devices and infrastructure to a completely secure protocol, low-cost and practicable intrusion detection systems (IDS) and intrusion prevention systems (IPS) are developed as middleware solutions instead. The first intrusion detection system designed in the 1980s by the United States National Security Agency (NSA) security employee James Anderson was inspecting user access logs, file access logs, and system event logs to detect attacks [17]. The system developed by Dorothy E. Denning and Peter G. Neumann was introduced as "Intrusion Detection Expert System" and detected anomalies by statistical analysis. This system is accepted as the basis of current modern intrusion detection systems [18].

Snort is a widely used, rule-oriented, open-source network intrusion detection and prevention system [19]. Due to its deficiency in detecting new intrusions, Snort IDS preprocessors integrate a learning algorithm such as an ANN to detect recent attacks [20]. It is supported by artificial intelligence to reduce the updating and cost, besides traditional methods' insufficiency for even regular traffic. Due to the difficulty of detecting intrusions in heterogeneous networks, it aims to implement different Machine Learning (ML) algorithms in WEKA tools to analyze the detection performance for DDoS attacks using the most recent CICDDoS2019 datasets [21]. This study included six distinct machine learning methods, including K-Nearest Neighbors (K-NN), Super Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR). In the provided evaluation, the Decision Tree (DT) and Random Forest (RF) algorithms achieved the highest level of accuracy, 99% and 99%, respectively. The researchers aim to develop low-cost, adaptive intrusion detection systems capable of detecting intrusions in real-time. As an alternative, various artificial intelligence methods have been used for autonomous intrusion detection to reduce human intervention. Machine learning-based techniques applied to IDSs are widely developed since researchers admit detecting network intrusions as the typical dual classification problem as normal and abnormal traffic.

On the other hand, IDS must recognize non-linear relationships between regular anomalies (non-attack) and anomalies. The artificial neural network stands out considering its capability of learning and modeling complex relationships despite its calculation charge [22]. Most preferred techniques on IDS applications are ANNs and other ML techniques. Additionally, researchers have employed techniques such as Data Mining [23], Fuzzy Logic [24], and Probabilistic Logic [25] in various studies. As the IoT network comprises millions of devices that are connected to the network, it is susceptible to a variety of security risks, particularly denial-of-service attacks. There are numerous intrusion

detection systems available for IoT networks; nonetheless, accuracy detection remains a significant issue. Bedine Kerim offers an ensemble IDS for IoT networks [26]. Compared to IDSs that use Naive Bayesian and Random Forest classifiers, experimental results show the highest accuracy performance of 99.8% for all specified characteristics. The Mirai Botnet, a malware that turns networked consumer devices into a botnet to conduct DDoS attacks, is one of the most destructive cyberattacks on IoT networks. In [27,28], Mirai is presented as one of the most significant recent DDoS attacks via Internet of Things roBots (IoTBots). Using machine learning-based technologies to enhance the IoT network's detection capacity is a viable strategy.

Tushir et al. [29] propose a way to identify the Scan, Acknowledge (ACK) Flooding, Synchronize (SYN) Flooding, UDP Flooding, and UDPplain Mirai Botnet attacks on IoT networks using ML techniques, comparing several ML techniques (KNN, SVM, and LR). The suggested technique was evaluated using a real-world IoT traffic dataset, achieving a detection accuracy of 99% for the Mirai Botnet. Tushir et al. [29] aim to analyze and explain the Mirai code and create a low-cost simulation environment to aid in the dynamic analysis of Mirai. They perform controlled Denial-of-Service attacks while monitoring resource usage on exploited and victim IoT devices with limited resources. Das et al. [30] attempted to discover IoT bots infected by Mirai in their research. For Mirai detection, the packet traffic generated by IoT devices at a given moment is evaluated. Additionally, the suggested approach aids in identifying the unique signature of Mirai and similar malware. To examine bots and botnets, researchers have utilized honeypots. Tolijan Trajanovski et al. propose and evaluate the IoT botnet detection and analysis (IoT-BDA) framework of honeypots for automatic capture, analysis, identification and reporting of IoT botnets [31].

IoT malware has shown a significant rise in recent years. According to statistics, the number of IoT malware families is steadily expanding. Mirai, Bashlite, Tsunami, Hide and Seek, BrickerBot, Luabot, and Hajime are just a few examples of malware families that target IoT devices specifically [32]. Kitsune Network Attack Dataset is a compilation of nine network attack datasets taken from either an IP-based commercial surveillance system or a network containing IoT devices. Each dataset consists of millions of network packets and various cyber-attacks. It is also applicable to Mirai botnet attacks. Mert Nakip and his colleague [33] use a method for detecting Mirai Botnet attacks based on a Dense Random Neural Network (Dense RNN). Experiments on a public dataset reveal that this method's performance is extremely close to that of an offline-trained neural network model. Abdullah Alabdulatif et al. give a comprehensive investigation into the selection of the best machine learning model (tree algorithms such as Simple Tree, Medium Tree, Coarse Tree, RUSBoosted, and Bagged Tree) for Kitsune. The winning method for detecting Mirai botnet malware attacks has been determined to be Coarse Tree. Satyanegara et al. [34] used the Kitsune Network Attack Dataset (ARP MitM Ettercap) in their research. They used two combinations of deep learning methods, which are Convolutional Neural Network-Multilayer Perceptron (CNN-MLP) and Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). CNN-MLP has a higher average accuracy rate than CNN-LSTM (99.67% *vs*. 99.57%). Abu Al-Haija and his friend [35] use the distilled Kitsune-2018 and Network Security Laboratory-Knowledge Discovery in Databases (NSL-KDD) datasets, which include real-world IoT network traffic under attack. They used six different machine-learning methods (Ensemble Boosted Trees (EBT), Ensemble Subspace kNN (ESK), Ensemble RUSBoosted Trees (ERT), Shallow Neural Network (SNN), Bilayered Neural Network (BNN), and Logistic Regression Kernel (LRK).) that belong to ensemble learning, neural networks, and kernel methods Standard machine-learning metrics are used to measure accuracy, error rates, and inference speed. Empirical investigation shows ensemble methods have superior accuracy and lower error rates than neural network and kernel methods.

Using technologies based on ML to improve the IoT network's detection capacity is an efficient strategy [36]. In addition to machine learning techniques, deep learning is also among the widely used techniques. Psathas et al. [37] tries to solve network security issues with the introduced hybrid intrusion detection system COREM^2. COREM^2 effectively identifies nine cyberattacks. It consists of a 2-D Convolutional Neural Network (CNN), a recurrent neural network with LSTM layers, and a multilayer perceptron. The COREM^2 was evaluated against the timely Kitsune Network Attack Dataset and achieved 98,64% and 98,92% accuracy, respectively. Haq et al. developed an IoT IDS using CNN for Enhanced Data rates for GSM Evolution (EDGE) Computing and calculated the accuracy for the NSL-KDD dataset to be 99.34% for binary classification and 99.13% for multiclass classification [38]. Anwer et al. [39] proposed hybrid DL driven approach to detect the attacks, one is Cuda Deep Neural Network Long Short-Term Memory (CuDNNLSTM) and another is LSTM on the Kitsune dataset. CuDNNLSTM outperforms LSTMs, which has 99,79% accuracy on a 6GB dataset. Haq and his friend developed two models, DNNBoT1 and DNNBoT2, with an accuracy of 90.54% and 91.24%, respectively, using Deep Neural Network (DNN) on the N-BaIoT Dataset [40].

Researchers attempt to detect cyber-attacks using artificial intelligence algorithms and log and network packets compiled from various experimental and real-world data. In our study, we focused on the Mirai attack targeting IoT devices. On a personal computer, we evaluated the learning performance of our proposed ANN-based model using samples of varying sizes. We compared performance metrics and the length of learning time for datasets of various sizes. As in the related studies, we also compared the results of ANN, K-NN, and SVM models using the entire Mirai attack in the Kitsune dataset to compare the performance metrics of various models. In particular, Cross Validation has been applied to avoid overfitting for our models. It is a groundbreaking effort to train and validate artificial intelligence models on devices with limited resources.

3 Technical Background

Mirai attack is a large-scale denial of service attack against smart devices using default credentials. It is necessary to classify between regular network packets and attack packets to detect cyber-attacks via network traffic. Kitsune converted the network packets collected during the Mirai attack into a dataset with many features. We conducted model training for samples of different sizes using artificial intelligence methods. We preferred ANN, SVM, and KNN supervised learning methods as classifiers.

3.1 Mirai Malware

Mirai is a malware that targets the insecure Telnet service using IoT devices such as IP cameras or smart home devices to aggregate botnets for DDoS attacks [41]. It has spread over billions of IoT devices manufactured without considering basic security requirements. Mirai's primary objective was to seize manufactured devices with default usernames and passwords. Afterward, infected devices are used for DDoS attacks.

Mirai botnet was first discovered by a malware "white hat" research group in August 2016; only one month later, an unprecedented DDoS attack on the "Kreb on Security" blog occurred [3]. Then, in October 2016, Mirai initiated a significant DDoS attack on Dyn company, the DNS provider of big companies like Netflix and Twitter. Also, Mirai was used to tear down Liberia's entire network offline [42]. Due to IoT device authorization weaknesses, the Mirai attack spread quickly; attackers shared Mirai source code 2 and started to develop other variants. Some of these featured variations have been named Satori [42], Okiru [43], and OMG [44] botnets. Ikiru is the first botnet targeting the ARC processors, the second most popular CPU core [42]. Satori botnet scans open Ethereum mining

facilities and occupies a considerable amount of cryptocurrency [43]. Finally, the OMG botnet turns infected IoT devices into proxy servers to protect attackers' identities [45].

Detecting intrusions by intelligent systems which can learn from threats' behavior patterns is more desirable than using rule-based security systems. Therefore, some Mirai analyses were presented in [46–48] to understand how malware works. Fig. 1 shows Mirai's typical central botnet structures. It consists of three parts: Control Command Server (CNC) server, Bot, and Loader. First, the attackers control CNC servers to control bot resources. Second, bots are functional parts responsible for finding vulnerable devices and starting massive DDoS attacks. Finally, the loader injects malware into newly exploited devices [49]. The infected device scans IP addresses on ports 23 and 2323 to detect other vulnerable devices while waiting for a command from the CNC server. Devices send their login, username, password, IP number, and port information on the CNC server and register to a separate report server. Then, the loader is notified to install malicious code for infection. When the attacker directs to initiate an attack, infected devices wait for a message from the CNC server. Each infected device strikes targets when the attacker commands to start an attack.

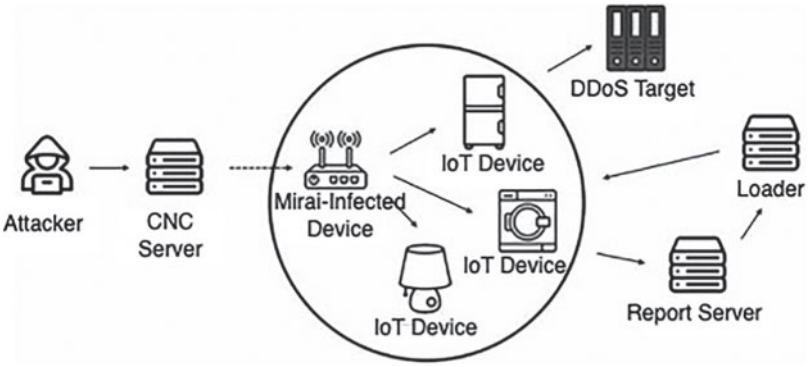

Figure 1: Structure of Mirai Botnet [43]

3.2 Artificial Neural Networks

Artificial Neural Network (ANN) is an information processing technology based on the biological nervous system. ANN imitation of biological neuron cells [7]. It is the digital representation of the synaptic connection between nerve cells. Neurons connect in a variety of ways to form networks. These networks are capable of discovering and learning about the relationships between data [8]. ANN mathematically simulates synaptic connections between neurons. In other words, learning takes place by adjusting the synaptic connections to adjust the outputs in response to the inputs. A primary ANN neuron consists of inputs, weights, transfer function, activation function, and output [50]. Weights initially consist of random values and change during the learning process to establish a relationship between the input and output values. The transfer function refers to the mathematical calculation between the input and weight values from different neurons [7]. Finally, the activation function (such as Sigmoid, RELU, TanH) is calculated with a mathematical function that determines the neuron's activation according to the threshold value of the signal transmitted from the transfer function [9]. Each input to the transfer function and the assigned weights represents the information flow.

Artificial Neural Network has two different learning types: supervised and unsupervised learning. In this study, supervised learning has been preferred. Multi-Layer Perceptron (MLP) is an ANN type using supervised learning procedures. MLP was used to detect offline analysis-based intrusions in

[51]. MLP was also used to detect intrusion in-network data, comparing its performance with Self Organizing Maps (SOM). SOM is a type of ANN using unsupervised learning to produce a low-dimensional, discrete representation of the input field of training samples called "Map" [52]. The feedforward neural network, including MLP architecture, is used in this work. Artificial Neurons are used in every neuron of input, output, and hidden layers. Thus, the ANN model consists of an input layer where the selected inputs are given, hidden layers, and output layers.

3.3 Support Vector Machine

The Support Vector Machine (SVM) method defines a repeatable hyperplane between classes [53]. Learning methods generally aim to classify each sample correctly according to its characteristics. This situation causes memorizing training data rather than learning models, especially for too-fit training data, and the classifier does not generalize adequately. The SVM algorithm maximizes the ability to generalize by evaluating all samples within the classes in the training set, separating them with a surface that maximizes the margin between them [54]. The process of training the SVM decision function is to maximize the margin between the support vectors of both class tags. Although SVM is most commonly preferred linearly, it does not have to be linear. Linear SVM problems vary in complexity depending on the number of features used. Whereas the hyperplane is simply a line for two properties, it corresponds to a two-dimensional plane for three properties. Assuming that the properties we use for SVM are linearly separable, we can efficiently draw a flat hyperplane (called a linear classifier) on the graph of the properties that separate the two labels of the respective class.

3.4 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a nonparametric statistical method used for classification and regression [55]. KNN uses a vector space model to classify samples with similar properties. KNN can be used to compare unknown class instances to known class instances to determine their possible classification. Besides being simple and effective, it is suitable for incremental learning. It is used in many areas, such as clustering, big data, and multi-label learning [56]. The classical KNN algorithm is highly complex in terms of time and space, and it is difficult to determine the k value [57]. If k is too small, the interference sensitivity increases, and the classification accuracy decreases. If k is too large and the dataset is imbalanced (imbalanced dataset), noisy samples will be chosen as the closest neighbors and adversely affect classification performance [58]. Since the KNN classification parameter is straightforward, the similarity of the features selected in learning to the rest of the class directly affects the classification performance. There are also risks of incorrect classification for outlier values of types.

3.5 Cross-Validation

Cross-validation is a resampling method to prevent overfitting, frequently encountered in methods such as classification, clustering, regression, and prediction [59]. The dataset is divided into parts, and a part of it is used for model testing, excluding the learning process. Typically, a larger training dataset is required to generalize across all cases. It can be evaluated with new data for model verification. When it is impossible to provide new data, resampling methods can be used for model validation. In most studies, this method divides the dataset "Single hold-out random subsampling," a type of Cross-Validation, into 90%–10% or 70%–30% training-test sets. Another sub-method, k-fold cross-validation, divides the dataset into k parts, with one part being tested and the remaining part being used for training. The performance is calculated using the arithmetic mean of the k iterations.

3.6 Bootstrapping

Resampling methods for detecting network base cyber-attacks improve statistical-predictive analysis in studies with small datasets or no sample balance between classes [60]. Bootstrapping (or Bootstrap) is a technique for evaluating the accuracy of estimators such as resampling and machine learning. Bootstrapping is mainly used for variance and bias estimation. Bootstrap generally relies on substitution and random sampling. In addition, Bootstrap provides accuracy metrics such as variance, prediction error, bias, and confidence intervals. This technique estimates the distribution of almost any statistic using random sampling methods. Bootstrap estimates the properties of an estimator (such as its variance) by sampling these properties from an approximate distribution. They are often used as an alternative to statistical inference based on the assumption of a parametric model when this assumption is doubtful or when parametric inference is impossible or requires complex formulas to calculate standard errors [61].

4 Proposed System

Network intrusion detection systems use rule-based methods with low resource requirements and behavior-based methods with high resource requirements. Using artificial intelligence methods to improve training and retraining accuracy results in inefficiency in terms of memory, processing power, and runtime when dealing with large datasets. Traditional behavior-based systems classify network traffic with the artificial intelligence model created in the flow shown in Fig. 2. The dataset's quality (interclass balance, number of rows, number of features, normalization, and feature selection) directly affects the model's accuracy.

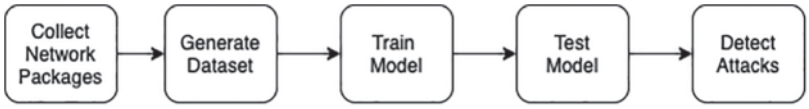

Figure 2: Traditional network attack systems that use artificial intelligence

This study proposes a methodology for analyzing constrained devices in IoT environments. The flowchart of the proposed methodology is depicted in Fig. 3. We used cross-validation to prevent the overfitting of classification methods during learning. In addition, we created 1000, 10000, and 100000 samples with the Bootstrapping technique to see the performance metrics of different-sized variations of the dataset.

It is widely accepted that performance metrics improve with increasing dataset size. While this exponentially increases the time required to create artificial intelligence models, it has a negligible effect on model accuracy. Furthermore, it is impossible to implement behavior-based methods on commonly used resource-constrained devices. With the sampling studies to be carried out on the training set, we aimed to develop a model that is as fast as the definition of rule-based systems and has the accepted accuracy levels of behavior-based systems. This enables the development of a behavior analysis model compatible with IoT systems, requires minimal resources, and provides high performance and accuracy. We developed a behavior-based network intrusion detection model on low-resource devices by determining the optimum value of resource requirement, learning time, and performance metrics. Contrary to traditional methods, as seen in Fig. 3, we trained models on various samples and then compared their accuracy. Then, based on the resource requirement and performance metrics, the path that gives the desired criteria is selected in the Select Model stage and run on the

device with limited resources. Thus, rather than wasting time developing a small, high-quality sample, the training time for the model is reduced.

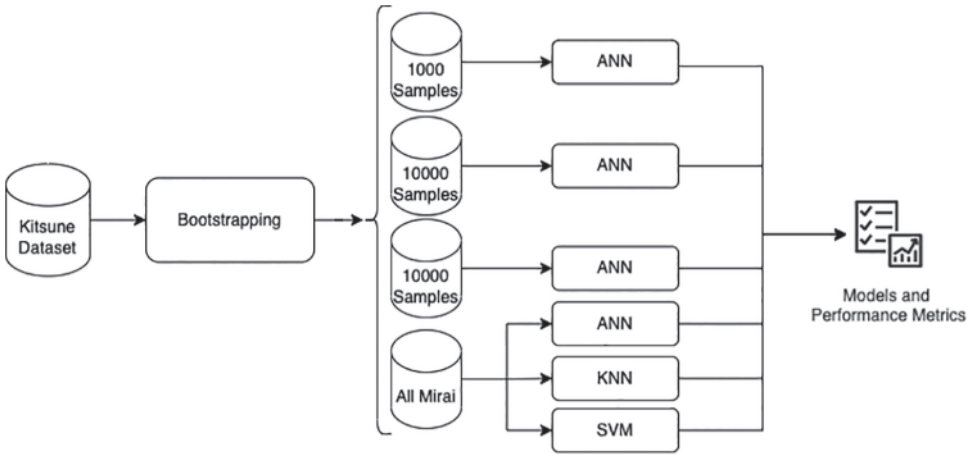

Figure 3: Flowchart of the proposed analysis methodology

4.1 Implementation

ANN, KNN, and SVM Mirai attack detection models were trained over The Kitsune dataset. Then, we compared the performance metrics of artificial intelligence models trained on samples of different sizes taken from the dataset with the Bootstrapping method and, at first, created sub-datasets of various sizes from the heritage dataset with the bootstrapping technique. Then, we evaluated training, testing, and validation methods, especially ANN, KNN, and SVM methods.

ANN has three hidden input and output layers in this study. With 115 features in the input layer, two output results are obtained. In our ANN model, there are 3 hidden layers: the first hidden layer contains 6 neurons, the second hidden layer contains 9 neurons, and the third hidden layer contains 5 neurons. Neurons in each layer are directly connected to neurons in the previous and next layers. The Sigmoid function was utilized as the activation function. The learning rate for the ANN algorithm was set to 0.01, and the algorithm ran for 20 iterations. Fig. 4 depicts the ANN model developed to detect the Mirai attack vector. According to the obtained results, it is observed that ANN is highly accurate at learning and detecting Mirai attacks.

For the KNN method, the k value is selected as 5, while the other parameters are selected as default. For SVM, we used default hyperparameter that means $C = 1.0$, *kernel* = *rbf* (*radial basis function*) and *gamma* = *auto* among other parameters. To avoid method overfitting, we used k-fold cross-validation with k is 10. Finally, we created confusion matrices with Python visualization libraries. We used Python to optimize the Behavior-Based IoT IDS system.

While three different ANN models were trained on different-sized datasets, ANN, K-NN, and SVM models were trained on the entire Mirai dataset. We implemented artificial intelligence models on the CPU using 16 GB of RAM and a seventh-generation Intel i7 processor. 20 epochs were utilized to train ANN models. We also analyzed the trained models in terms of their time complexity. During the training period, the 1k-row-ANN-Model required 47 s, the 10k-row-ANN-Model required 515 s, and the 100k-row-ANN-Model required 5405 s. The total time required by the 764137-row ANN-Model, the 178233-s SVM model, and the 184520-s KNN model for the entire dataset was 83643 s.

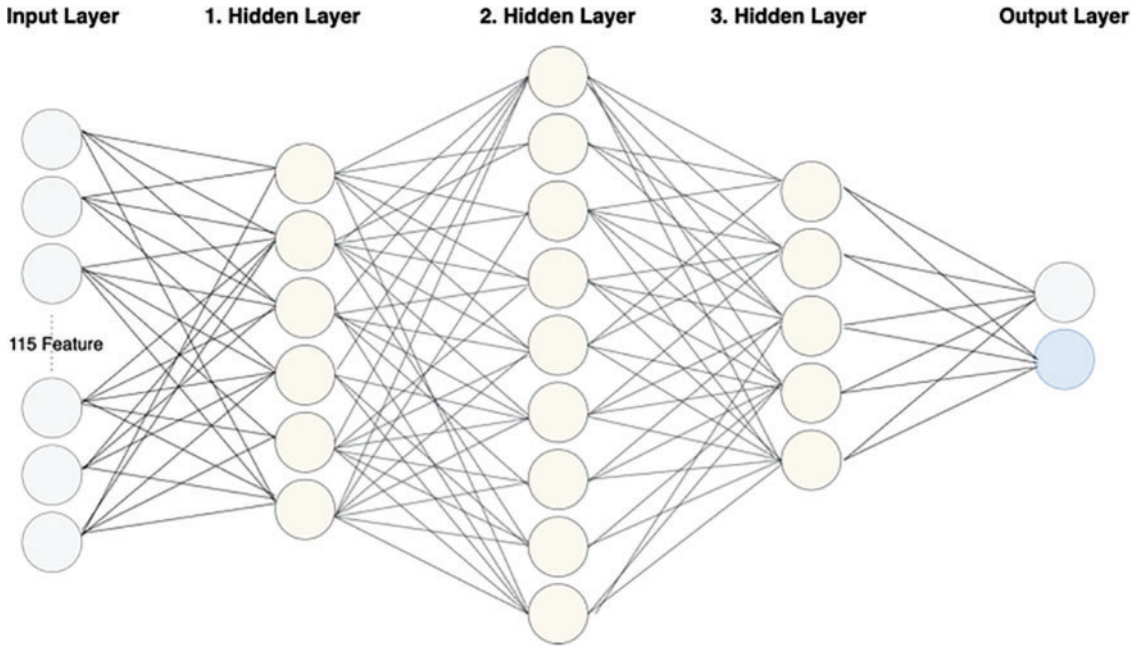

Figure 4: ANN structure used in intrusion detection

4.2 Dataset

Intrusion detection systems perform higher accuracy and performance from large datasets extracted from actual network traffic patterns. In our study, the dataset shared by Mirsky et al. was used [16]. Yisroel et al. published the Kitsune NIDS framework together with the Kitsune dataset. Instead, we utilized the Kitsune outputs for training and evaluating supervised learning methods using sub-datasets generated via Bootstrapping sampling operations.

In this study, we examined the Mirai attack. We used Kitsune Surveillance Network Intrusion Datasets, which include nine different IoT attack datasets. The Kitsune dataset was created in a laboratory environment and was used for this study. The dataset consists of network packages extracted from attacks that are using IoT devices. For example, Address Resolution Protocol Man in the Middle (ARP MitM), Simple Service Discovery Protocol (SSDP) Flood, Operating System (OS) Scan, SYN Flooding, Fuzzing, Video Injection, Secure Socket Layer (SSL) Renegotiation, and Mirai are included in the Kitsune Dataset. Our primary focus was on infecting an IoT network using the Mirai attack.

Mirsky et al. publish a Kitsune Dataset from various attack network traffic delivered to IoT devices in the laboratory using a mechanism called Kitnet [16]. Kitnet creates the features observed in the dataset from raw network packets and finds abnormalities using unsupervised learning. It produced the Kitsune Dataset, which has 115 features identified by analyzing raw network packets for eight separate attacks, including Kitnet Mirai and SYN Flooding. Only the Mirai attack was deployed out of the eight distinct attacks employed in our study. On the labeled dataset, optimization research was undertaken to construct an artificial intelligence model with limited samples on IoT devices using supervised learning.

5 Results and Discussion

The developed model for detecting Mirai attacks is based on artificial intelligence and machine learning techniques such as ANN, SVM, and K-NN. In this experimental work, a supervised learning process is executed using Mirai attack data in the Kitsune dataset. The performance metrics of intrusion detection with ANN are compared for different sized datasets. An ANN is trained with four different sized datasets separately. Also, ANN, K-NN, and SVM are compared for the complete dataset. The ANN algorithm's execution time, accuracy, and precision rate are compared to machine learning techniques SVM and K-NN, which are widely used in IDSs.

There are 764137 samples in the Kitsune dataset for Mirai, and nearly 125000 of them contain attack traffic. The ANN classification's confusion matrix, which is trained on 1000 samples resampled with Bootstrapping technique, is shown in Fig. 5. For 1000 samples, ANN's training and testing processes lasted 47 s in our setup with 99% accuracy and 99,95% precision. The confusion matrix of the ANN classification is trained on 10000 samples depicted in Fig. 6. ANN's training and testing processes lasted 8 min and 35 s in our setup with 99,56% accuracy and 99,95% precision. The increase of 10 times in the number of samples also improved the accuracy and precision rates.

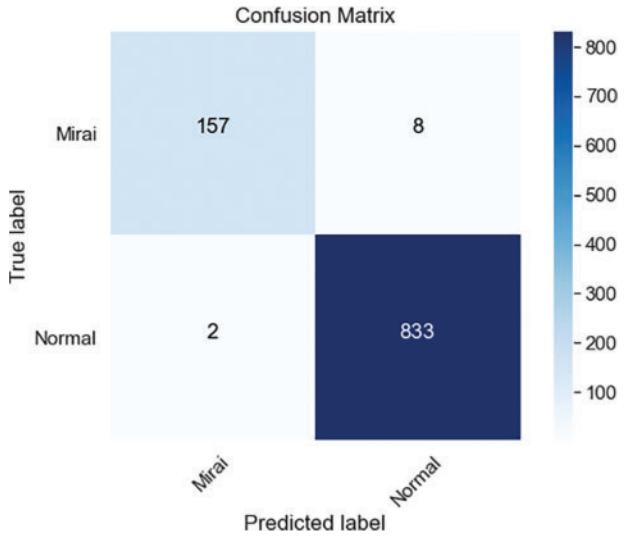

Figure 5: ANN confusion matrices for 1000 samples

The ANN classification's confusion matrix, which is trained on 100000 samples resampled with Bootstrapping technique, is shown in Fig. 7. Similarly, increasing the number of samples in the dataset improved the accuracy to 99,67% while lowering the precision to 99,82%. ANN's training and testing processes lasted 1 h, 30 min, and 5 s in our setup. As the last, in Fig. 8, the ANN classification's confusion matrix, which is trained on all 764137 samples of the complete Mirai dataset, is shown. Compared to the 100000 sampled dataset, accuracy has been increased to 99,85% and precision to 99,97% for the whole dataset. ANN's training and testing processes lasted 23 h, 27 min, and 33 s in our setup. We developed a Mirai Attack detection model with various error rates using artificial neural network, SVM, and KNN techniques on dataset samples of varying sizes. Since Mirai attacks target IoT devices, detecting intrusions in IoT ecosystems should be possible using low-cost and limited resources. Therefore, it must be processed using smaller datasets on devices incapable of processing large datasets or that take longer to process. The study shows the relationship between performance metrics and dataset size.

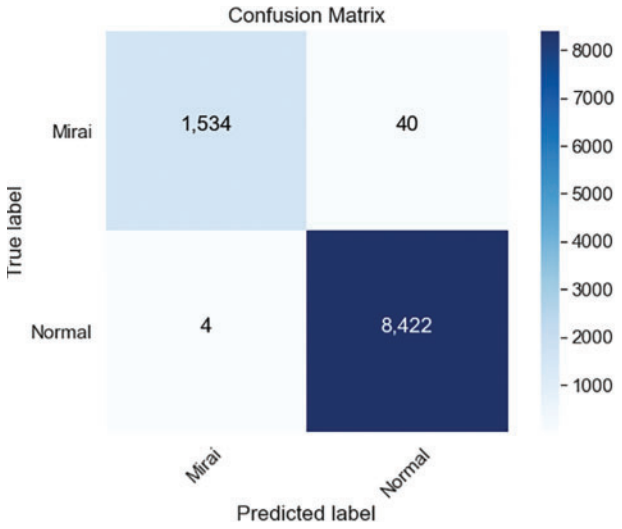

Figure 6: ANN confusion matrices for 10000 samples

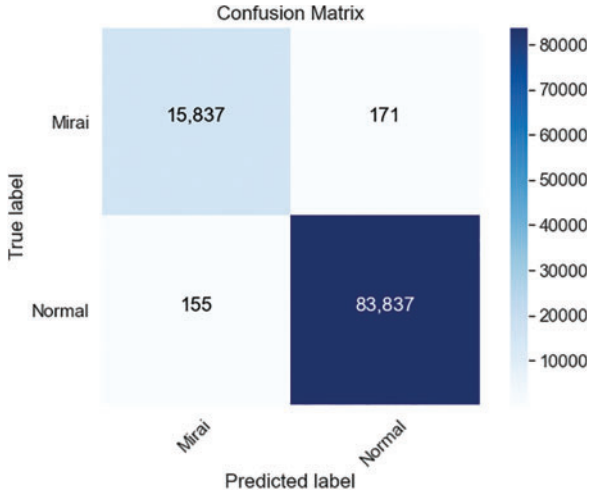

Figure 7: ANN confusion matrices for 100000 samples

Additionally, we summarized the accuracy and precision of the results obtained for four different sample sets in Fig. 9. Increasing the sample size also improves accuracy and precision, although this increase is not consistent. Fig. 9 shows the ANN's execution time with different sized datasets. In the first three datasets, while the sample size increased to ten times more extensive, the learning time improved 11 times. However, when all samples in the dataset were used, the size of the dataset increased eight times, and the learning time increased 15 times. It demonstrates that there is no linear relationship between the size of the dataset and the time required to execute it.

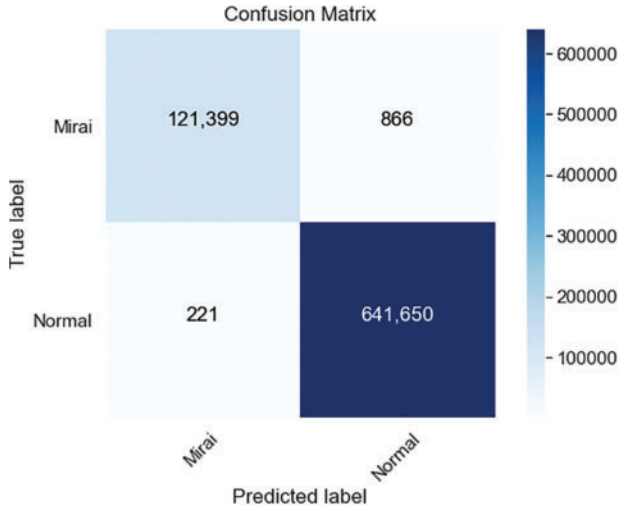

Figure 8: ANN confusion matrices for complete Mirai dataset

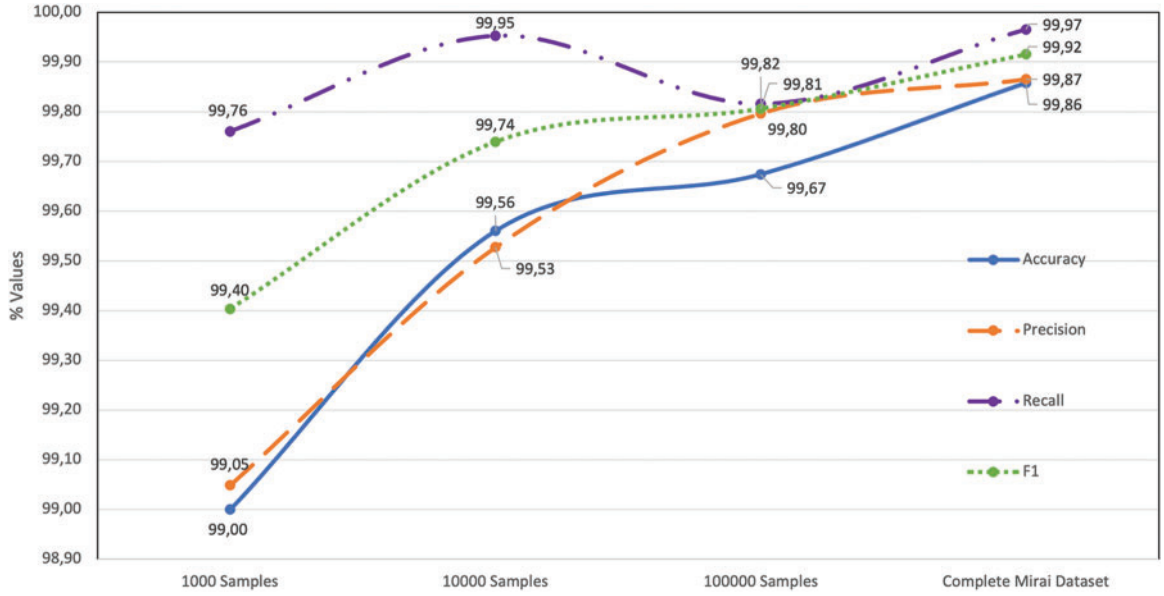

Figure 9: Performance metrics for ANN datasets of different sizes

In order to observe for the ANN algorithm Accuracy and Precision, it is seen that while passing from 1000 sample datasets to 10000 sample datasets, it increases by about 0.5%, F1-Score by 0.3%, and recall by 0.19%. Contrary to other metrics, recall decreased from a 10000 samples dataset to a 100000 samples dataset. We observe that the increase in the number of "False Negatives" records decreases the recall value due to the excessive increase in the number of "False Labels" records. We observe that the best performance metrics are obtained when the whole dataset is used. As a result, the ANN algorithm has an accuracy rate of 10000 datasets 99,56%, 100000 datasets 99,67%, and complete datasets 99,86%, using the sample extracted using Bootstrapping method. As a result, when the low

accuracy rate difference (0.3%) is accepted, it is determined that intrusion detection can be performed with small well-prepared samples that require less learning time.

ANN is compared to widely used machine learning algorithms SVM and K-NN. SVM algorithm follows the ANN performance with a 99,55% accuracy rate. On the other hand, SVM resulted in a precision rate of 100%. The confusion matrix of SVM is shown in Fig. 10. KNN performs the lowest accuracy rate of 93,51% and precision rate of 93,90% among all algorithms when the complete dataset is used. The confusion matrix of KNN is shown in Fig. 11.

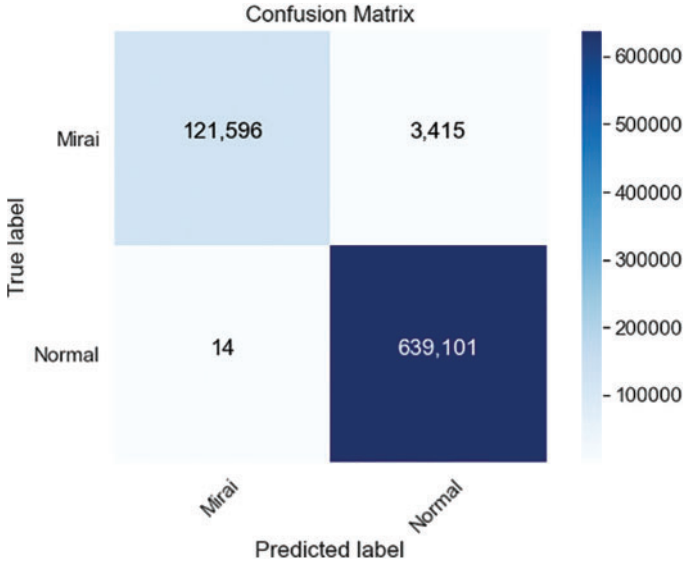

Figure 10: Confusion matrices for SVM models

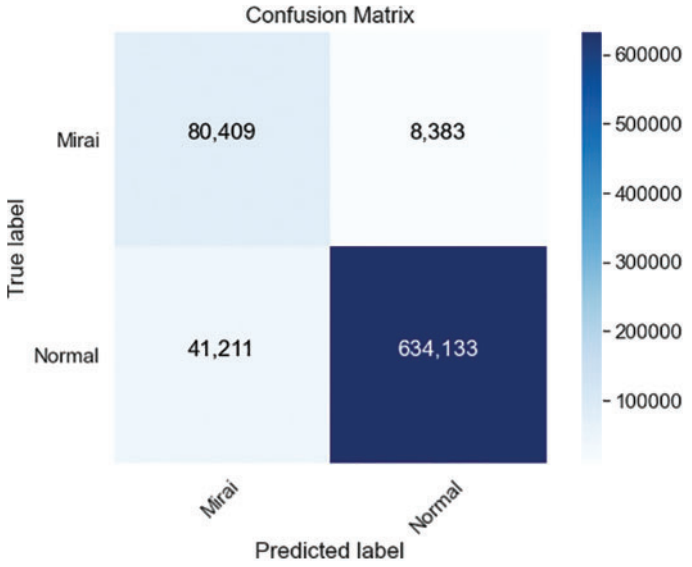

Figure 11: Confusion matrices for K-NN models

When algorithms are compared, ANN has the highest recall of 99,86% and an accuracy rate of 99,85%. SVM's precision rate is 100%, while ANN's precision rate is 99,97%. Also, ANN outperforms twice the other algorithms in terms of learning time. In our implementation, the learning time for ANN is 84453 s, for SVM it is 178233 s, and for KNN it is 184520 s. In addition to the outcomes of our study with the Kitsune data set, Table 1 compares the outcomes of several significant studies conducted with artificial intelligence algorithms utilizing diverse data sets and models. At the end of the table, all obtained accuracy, precision, and recall values for SVM, KNN, and ANN algorithms when cross-validated on Mirai attack data from the Kitsune dataset are also presented.

Table 1: Comparison of artificial intelligence-based studies on different datasets and results of our proposed model

Reference	Attack Type	Dataset	Approach	Performance			
				Accuracy	Precision	Recall	F1-Score
[21]	DDoS	CICDDoS2019	SVM	0.86	0.86	0.87	0.85
			K-NN	0.98	0.99	0.99	0.99
			DT	0.99	0.99	0.99	0.99
			NB	0.45	0.66	0.54	0.38
			RF	0.99	0.99	0.99	0.99
			LR	0.98	0.99	0.98	0.99
[27]	Mirai Botnet	Kitsune	AA-Dense RNN	0.9984	-	-	-
			Lasso	0.9978	-	-	-
			K-NN	0.9979	-	-	-
[34]	Man in the Middle (MitM)	Kitsune	CNN-MLP	0.9974	1.0	1.0	1.0
			CNN-LSTM	0.9944	0.99	0.99	0.99
[37]	Cyber-attacks (Including Mirai)	Kitsune	2-D CNN with LSTM-RNN	0.9973	0.9765	–	0.9801
[39]	Not defined	Kitsune	CuDNNLSTM	0.997949	0.997568	0.997249	0.997324
Our study	Mirai Botnet	Kitsune	ANN	0.9985	0.9997	0.99,86	0.9985
			K-NN	0.9955	1.00	0.99,47	0.9955
			SVM	0.9351	0.9390	0.9870	0.9351

6 Conclusion and Future Work

The Mirai attack vector starts a new era in botnet attacks regarding the method and attack size. Despite the limited resources of the devices Mirai exploited, they could carry out large-scale DDoS attacks thanks to their quantity. Furthermore, it demonstrated how Mirai malware could turn harmless IoT devices into weapons. An IDS that can apply in smart homes and workplaces has been developed, as it is the least costly method to prevent attacks carried out via IoT devices at their source. We show that the Mirai attack can be detected with artificial intelligence models trained on a personal computer using the Kitsune dataset prepared in the laboratory environment.

In order to observe the performance metrics during the training process, we subsampled using the bootstrapping method and applied it to the ANN algorithm with cross-validation. Performance metrics generally get better as the sample gets larger. In the experimental process, although the number of dataset records increased by 40 times, we observed an accuracy increase of 0.3%. When the low

accuracy rate difference is accepted, it is determined that intrusion detection can be performed with small well-prepared samples that require less learning time. With these results, it is observed that when developing intrusion detection systems that require high accuracy and a short execution time, ANN can result in increased performance metrics when samples are extracted from datasets.

An intrusion detection system is designed to distinguish Mirai attacks from regular traffic using supervised learning techniques. For complete Mirai dataset involving cross-validation, SVM, KNN, and ANN algorithms were compared, with ANN achieving a 99,85% accuracy rate. While the learning performance of ANN was twice as good as other algorithms, the slowest learning algorithm was KNN. In addition, the precision of the SVM algorithm performed better than other algorithms on the entire Mirai dataset.

Detection of the Mirai attack was carried out using constrained resources in order to show that the training process can be carried out on a personal computer. Afterward, it is planned to carry out attacks such as SYN DoS and SSDP Flood in the Kitsune dataset, where IoT devices and traditional devices are exposed. It is also planned to implement the training process directly on more than one raspberry pi for different data sets. It is aimed to demonstrate that model training and model testing will be done on IoT devices as distributed. In case the training process takes a long time and IoT devices cause performance problems such as high battery consumption, alternative processes such as cloud-based model training and transferring the verification of the trained model to IoT devices are also planned. It is a pioneering effort to train and validate artificial intelligence models on resource-constrained devices. In our study, it is aimed to provide cost-effective and high-accuracy attack detection for smart homes and smart offices. In the continuation of the study, it is planned to design a distributed IoT IDS system that detects it by focusing on a single attack.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Rahman and R. Rahmani, "Enabling distributed intelligence assisted future internet of things controller (fitc)," *Applied Computing and Informatics*, vol. 14, no. 1, pp. 73–87, 2018.

[2] R. H. Hsu, J. Lee, T. Q. Quek and J. C. Chen, "Reconfigurable security: Edge-computing-based framework for IoT," *IEEE Network*, vol. 32, no. 5, pp. 92–99, 2018.

[3] B. Krebs, "*Krebs on Security Hit with Record DDoS*, 2016. [Online]. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[4] G. Leander, C. Paar, A. Poschmann and K. Schramm, "New lightweight des variants," in *Int. Workshop on Fast Software Encryption*, Luxembourg, Luxembourg, Springer, pp. 196–210, 2007.

[5] D. Moon, H. Im, I. Kim and J. H. Park, "Dtb-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing apt attacks," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 2881–2895, 2017.

[6] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," *Computer Communications*, vol. 30, no. 10, pp. 2201–2212, 2007.

[7] A. A. Anitha and L. Arockiam, "ANNIDS: Artificial neural network based intrusion detection system for Internet of Things," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 11, pp. 2583–2588, 2019.

[8] Y. Hamid, F. A. Shah and M. Sugumaran, "Wavelet neural network model for network intrusion detection system," *International Journal of Information Technology*, vol. 11, no. 2, pp. 251–263, 2019.
[9] S. A. Khanday, H. Fatima and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks," *Expert Systems with Applications*, vol. 215, no. 8, pp. 119330, 2023.
[10] A. Sajid, H. Abbas and K. Saleem, "Cloud-assisted IoT-based scada systems security: A review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.
[11] J. Zhou, Z. Cao, X. Dong and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.
[12] Z. Wang, "The applications of deep learning on traffic identification," *BlackHat USA*, vol. 24, no. 11, pp. 1–10, 2015.
[13] R. R. Reddy, Y. Ramadevi and K. N. Sunitha, "Effective discriminant function for intrusion detection using svm," in *2016 Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, India, IEEE, pp. 1148–1153, 2016.
[14] L. Celardo and M. G. Everett, "Network text analysis: A two-way classification approach," *International Journal of Information Management*, vol. 51, no. 5, pp. 102009, 2020.
[15] R. A. Welikala, M. M. Fraz, J. Dehmeshki, A. Hoppe, V. Tah *et al.,* "Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy," *Computerized Medical Imaging and Graphics*, vol. 43, no. Suppl. 1, pp. 64–77, 2015.
[16] Y. Mirsky, T. Doitshman, Y. Elovici and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proc. 2018 Network and Distributed System Security Symp. (NDSS 2018)*, San Diego, California, USA, 2018.
[17] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report, James P. Anderson Company*, 1980.
[18] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.
[19] A. Cholakoska, M. Shushlevska, Z. Todorov and D. Efnusheva, "Analysis of machine learning classification techniques for anomaly detection with NSL-KDD data set," *Proceedings of the Computational Methods in Systems and Software (CoMeSySo 2021)*, vol. 2, pp. 258–267, 2021.
[20] R. Coulter and L. Pan, "Intelligent agents defending for an IoT world: A review," *Computers & Security*, vol. 73, no. 8, pp. 439–458, 2018.
[21] R. J. Alzahrani and A. Alzahrani, "Security analysis of DDoS attacks using machine learning algorithms in networks traffic," *Electronics*, vol. 10, no. 23, pp. 2919, 2021.
[22] D. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik *et al.,* "Deep neural network based anomaly detection in internet of things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, pp. e4121, 2021.
[23] S. Kavitha, M. Hanumanthappa and B. N. Kalavathi, "Intrusion detection of internet of things botnet attacks using data mining technique," in *Rising Threats in Expert Applications and Solutions*, vol. 434. Singapore: Springer, pp. 159–165, 2022.
[24] A. Dallali, T. Omrani and B. C. Rhaimi, *Fusion of artificial neural networks by fuzzy logic based attack detection method Easy Chair Preprint No: 7023*, 2021. [Online]. Available: https://easychair.org/publications/preprint/svJt
[25] M. A. Khan, M. M. Nasralla, M. M. Umar, S. Khan and N. Choudhury, "An efficient multilevel probabilistic model for abnormal traffic detection in wireless sensor networks," *Sensors*, vol. 22, no. 2, pp. 410, 2022.
[26] B. Kerim, "Securing IoT network against DDoS attacks using multi-agent ids," *Journal of Physics: Conference Series*, vol. 1898, pp. 012033, IOP Publishing, 2021.
[27] M. Nakip and E. Gelenbe, "Mirai botnet attack detection with auto-associative dense random neural network," in *2021 IEEE Global Communications Conf. (GLOBECOM)*, Madrid, Spain, IEEE, pp. 1–6, 2021.

[28] S. G. Abbas, F. Hashmat, G. A. Shah and K. Zafar, "Generic signature development for IoT botnet families," *Forensic Science International: Digital Investigation*, vol. 38, pp. 301224, 2021.
[29] B. Tushir, H. Sehgal, R. Nair, B. Dezfouli and Y. Liu, *The impact of DoS attacks on resource-constrained IoT devices: A study on the Mirai attack*, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2104.09041
[30] S. Das, P. Amritha and K. Praveen, "Detection and prevention of Mirai attack," in *Soft Computing and Signal Processing*, vol. 1, Singapore: Springer, pp. 79–88, 2021.
[31] T. Trajanovski and N. Zhang, "An automated and comprehensive framework for IoT botnet detection and analysis (IoT-BDA)," *IEEE Access*, vol. 9, pp. 124360–124383, 2021.
[32] R. Nath and H. V. Nath, "Critical analysis of the layered and systematic approaches for understanding IoT security threats and challenges," *Computers and Electrical Engineering*, vol. 100, no. 2, pp. 107997, 2022.
[33] M. Nakip and E. Gelenbe, "Botnet attack detection with incremental online learning," in *EuroCybersec 2021: Security in Computer and Information Sciences, Int. ISCIS Security Workshop*, pp. 51–60, Nice, France, Springer, 2022.
[34] H. H. Satyanegara and K. Ramli, "Implementation of CNN-MLP and CNN-LSTM for MITM attack detection system," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 6, no. 3, pp. 387–396, 2022.
[35] Q. Abu Al-Haija and A. Al-Badawi, "Attack-aware IoT network traffic routing leveraging ensemble learning," *Sensors*, vol. 22, no. 1, pp. 241, 2021.
[36] A. Gaurav, B. B. Gupta and P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system," *Enterprise Information Systems*, vol. 11917, no. 1, pp. 1–25, 2022.
[37] A. P. Psathas, L. Iliadis, A. Papaleonidas and D. Bountas, "Corem2 project: A beginning to end approach for cyber intrusion detection," *Neural Computing and Applications*, vol. 34, no. 22, pp. 1–20, 2022.
[38] M. Anul Haq, M. Abdul Rahim Khan and T. AL-Harbi, "Development of PCCNN-based network intrusion detection system for edge computing," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1769–1788, 2022.
[39] M. Anwer, G. Ahmed, A. Akhunzada and S. Siddiqui, "Intrusion detection using deep learning," in *2021 Int. Conf. on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Mauritius, IEEE, pp. 1–6, 2021.
[40] M. Anul Haq and M. Abdul Rahim Khan, "Dnnbot: Deep neural network-based botnet detection and classification," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1729–1750, 2022.
[41] L. E. S. Jaramillo, "Malware detection and mitigation techniques: Lessons learned from Mirai DDoS attack," *Journal of Information Systems Engineering & Management*, vol. 3, no. 3, pp. 19, 2018.
[42] Y. Xu, H. Koide, D. V. Vargas and K. Sakurai, "Tracing Mirai malware in networked system," in *2018 Sixth Int. Symp. on Computing and Networking Workshops (CANDARW)*, Takayama, Japan, IEEE, pp. 534–538, 2018.
[43] M. Kumar, *New Mirai okiru botnet targets devices, running widely-used arc processors*, 2018. [Online]. Available: https://thehackernews.com/2018/01/mirai-okiru-arc-botnet.html
[44] J. Manuel, R. Joven and D. Durando, *OMG: Mirai-based bot turns IoT devices into proxy servers*, 2018. [Online]. Available: https://www.fortinet.com/blog/threat-research/omg--mirai-based-bot-turns-iot-devices-into-proxy-servers
[45] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein *et al.,* "Understanding the Mirai botnet," in *26th USENIX Security Symp. (USENIX Security 17)*, Vancouver, BC, Canada, pp. 1093–1110, 2017.
[46] H. Sinanović and S. Mrdovic, "Analysis of Mirai malicious software," in *2017 25th Int. Conf. on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, IEEE, pp. 1–5, 2017.
[47] S. G. Abbas, F. Hashmat, G. A. Shah and K. Zafar, "Generic signature development for IoT botnet families," *Forensic Science International: Digital Investigation*, vol. 38, pp. 301224, 2021.
[48] G. L. Nguyen, B. Dumba, Q. -D. Ngo, H. -V. Le and T. N. Nguyen, "A collaborative approach to early detection of IoT botnet," *Computers & Electrical Engineering*, vol. 97, pp. 107525, 2021.

[49] C. D. McDermott, A. V. Petrovski and F. Majdani, "Towards situational awareness of botnet activity in the internet of things," in *2018 Int. Conf. on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, Scotland, United Kingdom, pp. 1–8, 2018.
[50] M. Choraś and M. Pawlicki, "Intrusion detection approach based on optimised artificial neural network," *Neurocomputing*, vol. 452, pp. 705–715, 2021.
[51] B. S. Khater, A. W. Abdul Wahab, M. Y. I. Idris, M. A. Hussain, A. A. Ibrahim *et al.,* "Classifier performance evaluation for lightweight ids using fog computing in IoT security," *Electronics*, vol. 10, no. 14, pp. 1633, 2021.
[52] D. B. Dasari, G. Edamadaka, C. S. Chowdary and M. Sobhana, "Anomaly-based network intrusion detection with ensemble classifiers and meta-heuristic scale (ECMHS) in traffic flow streams," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–28, 2020.
[53] D. A. Pisner and D. M. Schnyer, "Chapter 6: Support vector machine," in *Machine Learning: Methods and Applications to Brain Disorders*, London: Academic Press, pp. 101–121, 2020.
[54] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, no. 7, pp. 189–215, 2020.
[55] R. Wazirali, "An improved intrusion detection system based on KNN hyper-parameter tuning and cross-validation," *Arabian Journal for Science and Engineering*, vol. 45, no. 12, pp. 10859–10873, 2020.
[56] A. J. Gallego, J. Calvo-Zaragoza, J. J. Valero-Mas and J. R. Rico-Juan, "Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation," *Pattern Recognition*, vol. 74, no. 1, pp. 531–543, 2018.
[57] W. Li, Y. Chen and Y. Song, "Boosted k-nearest neighbor classifiers based on fuzzy granules," *Knowledge-Based Systems*, vol. 195, no. 2, pp. 105606, 2020.
[58] Z. Pan, Y. Wang and Y. Pan, "A new locally adaptive k-nearest neighbor algorithm based on discrimination class," *Knowledge-Based Systems*, vol. 204, pp. 106185, 2020.
[59] D. Berrar, "Cross-validation," *Encyclopedia of Bioinformatics and Computational Biology*, vol. 1, pp. 542–545, 2019.
[60] D. Hongle, Z. Yan and K. Gang, "A selective ensemble learning algorithm for imbalanced dataset," *Journal of Ambient Intelligence and Humanized Computing*, vol. 45, no. 1, pp. 1–10, 2021.
[61] G. Oh, J. E. Lee and J. C. Ye, "Unpaired MR motion artifact deep learning using outlier-rejecting bootstrap aggregation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 11, pp. 3125–3139, 2021.