A Data Collection System from the RPL Routing Protocol for Detecting Distributed Denial of Service Attacks in IoT Networks

Anastasiia Nicheporuk[a], Andrii Nicheporuk[a], Andrzej Kwiecien[b] , Serhii Posonskyi[a] and Galina Radelchuk[a]

[a] *Khmelnytskyi National University, Institutska str., 11, Khmelnytskyi, 29016, Ukraine*
[b] *Silesian University of Technology, Akademicka str., 2A, Gliwice, Poland*

Abstract

The work presents a data collection system from the RPL routing protocol for detecting distributed denial-of-service attacks in Internet of Things (IoT) networks operating on the basis of the 6LoWPAN and RPL protocols. The system consists of three modules: a data gathering module, a classification module and a detection module. The main feature of the data collection module was that data collection was provided by several sniffers installed in the network and with subsequent aggregation of the collected data. For the implementation of the classification module, research was carried out on the method of support vector machines (SVM) and a multilayer perceptron (MLP). The detection module was used to broadcast a message about the abnormal behavior to the rest of the IoT network nodes, containing the ID of the compromised node and the path to it.

To evaluate the efficiency of the proposed system that is based on the data collected by the data gathering module, a number of experiments were conducted. To obtain the data set for the experiments, an infrastructure based on the Ubuntu operating system and the Cooja simulator was deployed, which allowed to simulate the RPL network. Based on the operation of the deployed network, network traffic was collected that corresponded to both legitimate traffic and traffic during a black hole attack. The total number of test data was 24,023 samples. According to the research results, it was established that the SVM-based model demonstrated better performance level, in particular, the accuracy of detecting denial-of-service attacks was 89.6%, while the rate of false positives was 6%.

Keywords

RPL protocol, denial of service attack, anomaly detection, black hole attack, Contiki, sniffer

1. Introduction

To date, the concept of the Internet of Things demonstrates a steady trend towards development in the field of information technology, primarily due to the spread of wireless sensor networks, the acceleration of the transition to IPv6 addressing, the use of cloud computing and the development of machine learning, the development of new methods of data access in computer systems [1]. The Internet of Things connects devices in a computer network and allows them to collect, analyze, process and transmit data to other things connected to each other through software, applications or technical devices. However, the heterogeneity of the environment and the wireless way data exchange make IoT networks potential targets for attackers.

Among the main security threats of IoT networks are denial-of-service (DoS) attacks and distributed denial-of-service (DDoS) attacks. This type of attack leads to the loss of access to the

IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine
EMAIL: eldess06@gmail.com (A. Nicheporuk); andrey.nicheporuk@gmail.com (A. Nicheporuk); andrzej.kwiecien@polsl.pl; (A. Kwiecien); p.s.f@ukr.net (S. Posonskyi); gal_2015@ukr.net (G. Radelchuk);
ORCID: 0000-0001-5366-5792 (A. Nicheporuk); 0000-0002-7230-9475 (A. Nicheporuk); 0000-0003-1447-3303 (A. Kwiecien); 0000-0002-4697-7699 (S. Posonskyi); 0000-0002-9728-4390 (G. Radelchuk)

device or the resources it offers. Attackers implement a wide range of different attack methods, but the most common of them consists in bombarding the system with a huge amount of unnecessary data in order to fill the available bandwidth of the target network or its computing power [2]. Another variant of the influence on the IoT network is the redirection of packets or their rejection [3]. This type of attack is especially acute in IoT networks due to the nature of the implementation of routing algorithms, which involve the use of fully connected topologies and the transmission of data from the source to the receiver through a chain of intermediate nodes [4]. In general, this type of attack results in legitimate users losing access to resources or devices. The use of various obfuscation techniques, which are used, for example, in metamorphic viruses [5-8] makes the situation even worse for detection systems. As for the purpose of implementing such attacks, it can be different, starting from the creation of bot networks [9] to receive a monetary reward, and ending with the satisfaction of one's own ambitions.

Today, traditional denial-of-service attack detection approaches do not meet current security requirements. The existing methods and means do not allow to fully resist the constantly growing threats. Therefore, development of new methods for detecting denial of service attacks on the infrastructure of the Internet of Things is an urgent task.

2. Related works

The problem of detecting distributed denial-of-service attacks is receiving considerable attention. To date, the most well-known approaches to detecting DDoS attacks in Internet of Things networks are methods based on machine learning, statistical algorithms, time-series algorithms [10] and based on the involvement of software-configured networks.

In [11], an approach to detecting distributed denial-of-service attacks based on the construction of a topological structure of traffic data with the involvement of graph theory is proposed. Network traffic data is rearranged in the form of a directed graph. As edges in the graph, the authors use information about connections between nodes, frequency, flow duration, and other information from network characteristics of traffic. The PCA method was used to reduce the dimensionality of the data. The detection process is implemented using fuzzy C-means classification.

The authors of the paper [12] proposed an approach to detecting DDoS attacks in Internet of Things networks, which involves converting network traffic into an image form with the subsequent involvement of a residual neural network model. The results of the proposed method show high detection results for binary classification (99%) and 87% for multi-class classification.

The authors of [13] approached the solution of the security problem in IoT networks by optimizing and reducing the size of input data and investigated the problem of extracting a subset of the most relevant functions from network traffic. A cost-effective model for cleaning and preparing raw data before dimensionality reduction is proposed. A hybrid method of feature selection based on the measure of mutual information, analysis of variance (ANOVA), chi-square method, decision tree algorithm was proposed.

The authors of the [14] proposed a statistical detection method based on continuous ranked probability score (CRPS) and exponential smoothing for effective detection of denial of service (DoS) and DDoS attacks. The authors use CRPS to quantify the difference between the new observation and the normal traffic distribution. To check the effectiveness of the proposed solution, the authors conducted a number of experiments on three sets of data. The presented solution demonstrated a fairly high efficiency index, however, with low network traffic, the efficiency of the proposed solution deteriorates.

Another statistical approach to detecting distributed denial of service attacks is presented in [15]. The authors investigated the manifestations of malicious activity in smart home networks. To detect active attacks, the authors used VPN technology together with the Snort intrusion detection system.

In [10], a method for detecting distributed denial-of-service attacks based on time-series analysis was proposed. The authors consider a meta time-series similarity profile representing time series data. The Euclidean metric was used to find the difference between time series data.

The authors of [16] presented a way to detect DDoS attacks in Internet of Things networks based on the involvement of low-cost machine learning algorithms and data obtained from network traffic based on flows and protocols. In that work, some limited features of IoT network behavior were

considered, such as the calculation of endpoints and the time required to pass from one packet to another (time intervals between packets), packet size, bandwidth, and others. KNN, KDTree algorithm, SVM with the linear kernel (LSVM), DT using Gini impurity scores, RF using Gini impurity scores, NN were used as classifiers for detecting attacks. The paper claims that the proposed methods can identify DDoS attacks on local IoT devices working together with home routers and other intermediate network modules.

The use of blockchain presents another viable solution for mitigating denial-of-service attacks on IoT networks, as proposed by the authors in [17]. Specifically, the authors suggest the use of an Ethereum blockchain model to detect and prevent DDoS attacks on IoT systems. Furthermore, the proposed system can address issues related to individual points of failure, privacy, and security in IoT systems. The authors advocate for a decentralized platform, as opposed to centralized system solutions, to prevent DDoS attacks on IoT devices at the application level through device authentication and verification.

3. Architecture of the data collection system from the RPL routing protocol for detecting distributed denial of service attacks in IoT networks

The task of data gathering in Internet of Things networks is one of the directions of the reverse engineering process and can be implemented in order to perform two main functions: to analyze the collected data in order to improve the efficiency of interaction between devices in the network or to carry out network diagnostics for troubleshooting. In turn, one of the main directions of network diagnostics is the analysis of network traffic data for the purpose of detecting malicious activity or the impact of cyberattacks. This makes it possible to implement one of the main requirements for the infrastructure of the Internet of Things – ensuring its operational security from the point of view of the ability to resist the influence of malicious software and cyberattacks. This paper presents a data collection system from the RPL routing protocol for detecting distributed denial of service attacks in Internet of Things networks operating on the basis of the 6LoWPAN and RPL protocols. The proposed system consists of three main modules (Fig.1):

- Data gathering module (DGM);
- Classification module (CM);
- Mitigation and detection module (MDM).

The DGM can be considered as an interphase module, as it is involved in two phases of system operation: pre-training and post-training. The CM and MDM are part of the post-training phase and are responsible for detecting attacks and forming countermeasures. In addition, traffic monitoring, data classification, and isolation of malicious nodes take place at this stage. A generalized structural scheme of the data collection system from the RPL routing protocol for detecting distributed denial of service attacks in IoT networks is shown in Fig. 1.

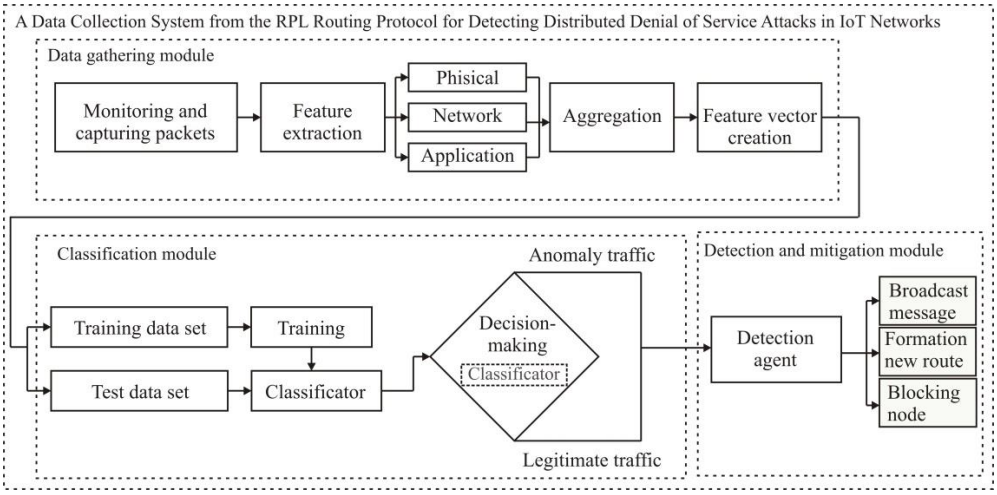

Figure 1: Architecture of the data collection system from the RPL routing protocol for detecting distributed denial of service attacks in IoT networks

3.1. Data gathering module

Before detecting any malicious activity, it is necessary to obtain features from the network that would allow to identify the occurrence of anomalies. For this purpose, the system uses a data gathering module. The primary objective of this module is to gather information within an actual or simulated Internet of Things network that employs the 6LoWPAN and RPL protocols. It is important to mention that the suggested data gathering system based on the RPL routing protocol for detecting distributed denial of service attacks in IoT networks is not exclusive to these protocols, and can potentially be adapted and expanded to encompass other data exchange protocols in the realm of Internet of Things networks.

In this system, it is proposed to use features from three logical levels: physical, network, and application levels. The processing of physical layer features, in particular, such as received and transmitted dBm signals at the MAC level, is related to physical layer jamming attacks, which pursue the goal of disrupting the physical connection between nodes in the network. As a result of processing the packets of the physical level, will be get the received signal strength indicator RSSI (f^{p}_{RSSI}), the value of the received signal dBm (f^{p}_{RdBM}), the value of the transmitted signal dBm (f^{p}_{TdBM}).

Obtaining features of the packets in network level is important given the specifics of the operation of many known denial-of-service attacks (for example, selective packet forwarding attacks and black hole attacks). From packages of this level, the following characteristics are obtained: link quality indicator LQI (f^{n}_{LQI}), mean value of the expected transmission count ETX (f^{n}_{ETX}), number of DIO messages (f^{n}_{NDIO}), number of DIS messages (f^{n}_{NDIS}) and changing the node's RPL rank (f^{n}_{LRPL}).

This module gathers data that pertains to the application level, including information such as the node power level and temperature. Application-level features can be obtained by programming nodes to calculate power consumption and other related functions [7].

The application layer is the connection between the network and the application software. In this study, such characteristics as the average (f^{a}_{MeCP}) and modal (f^{a}_{MoCP}) value of power consumption as well as the node ID (f^{a}_{NID}) are obtained from application-level packets. Table 1 shows the features obtained by the data gathering module from packets of the physical, network, and application layers in the IoT network.

Table 1
A set of features gathered from IoT packets

Features	Description
f^{p}_{RSSI}	average value received signal strength indicator
f^{p}_{RdBM}	average value of the received signal dBm
f^{p}_{TdBM}	average value of the transmitted signal dBm
f^{n}_{LQI}	value of link quality indicator
f^{n}_{ETX}	average value of the expected transmission count
f^{n}_{NDIO}	number of DIO messages
f^{n}_{NDIS}	number of DIS messages
f^{n}_{LRPL}	number of node's RPL rank changing
f^{a}_{MeCP}	modal value of power consumption
f^{a}_{MoCP}	average value of power consumption
f^{a}_{NID}	node ID

Figure 2 shows the process of extracting features from a pcap file obtained from the RPL IoT network. The feature extraction process involves sequentially obtaining features from each level and saving them to the database for further processing and analyzing.

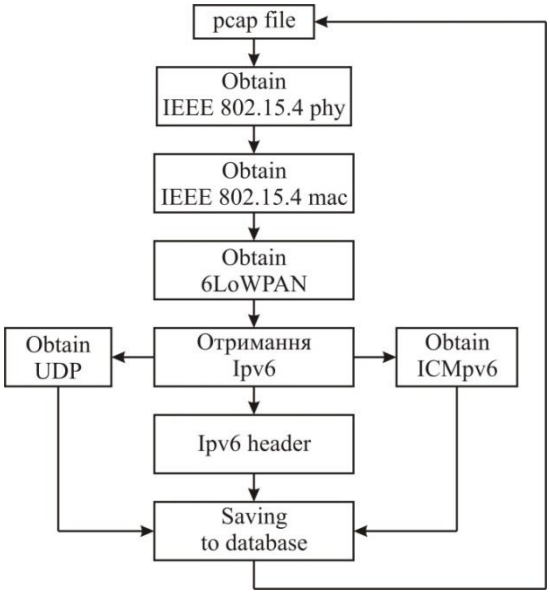

Figure 2: The process of extracting features from IoT network traffic

Also, it should be noted that before feature extraction, a time window slot for aggregating data into feature vector should be defined. This time window will be used later for quantitative values and averages.

Thus, after processing network traffic through the DGM, a dataset will be generated that is founded on the RPL and 6LoWPAN protocols. This dataset will be utilized for both training and testing machine learning algorithms, with the goal of establishing a detection model during the pre-training phase. It is worth noting that the same feature selection steps will be executed in the post-training phase, where the machine learning model created during the pre-training phase will be employed to scrutinize unknown activities in real-time.

3.2. Classification module

The machine learning algorithms will be trained and tested using the dataset generated by the DGM. At this level, the analysis of several machine learning methods is performed, and the algorithm that has the best results in terms of effectiveness of attack detection is selected. In this paper, we will use the two most common methods for this area of research, namely the support vector machine (SVM) and the multilayer perceptron (MLP) [18-20]. Figure 3 shows a schematic representation of the training and validation process of SVM for detecting cyberattacks.

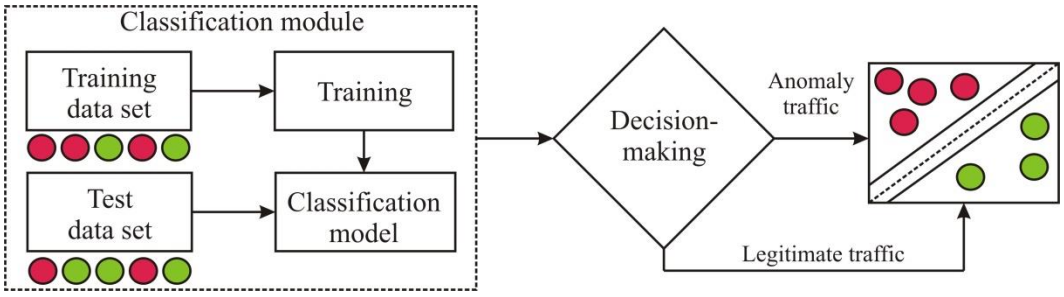

Figure 3: The process of extracting features from IoT network traffic

3.3. Mitigation and detection module

This module serves as a bridge between the local network and the distributed denial-of-service attack detection system in the Internet of Things infrastructure. It is situated on top of the sink node in the network, as all nodes are either directly connected to the sink node or are several hops away. The key purpose of this module is to transmit a message to all IoT network nodes regarding anomalous behavior, which includes the attacker's identification and the route taken by the attacker. This will enable unaffected nodes to blacklist the malicious node and refrain from communicating with it [21]. In addition, the MDM redirects the victim node and creates a new alternative route to the receiver node. After that, reconfiguration of the network topology will be performed to isolate the malicious node by establishing a new path to the receiver from the victim node. All nodes will blacklist the malicious node, and all network traffic from it will be ignored and dropped.

4. System functioning: pre-training phase and post-training phase

The function of the proposed a data collection system from the RPL routing protocol for detecting distributed denial of service attacks in IoT networks includes two phases: a pre-training phase and a post-training phase. Figure 4 shows the two phases and the associated tasks performed within each phase.

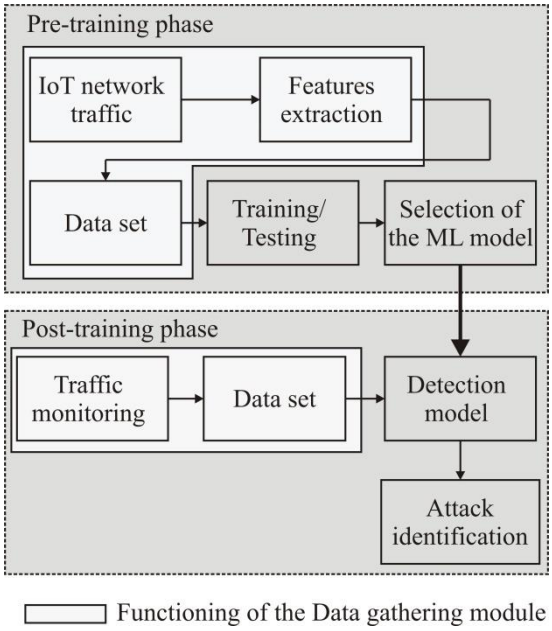

Figure 4: System functioning

4.1. Pre-training phase

In the pre-training phase, the machine learning model is trained and tested on the basis of the data collected by DGM. In this work, two machine learning algorithms will be investigated and a set of tests will be conducted to determine the most effective model. It should be noted that the processing of machine learning models is carried out on the basis of the received data of the DGM.

One could outline the following steps to describe the process of selecting the most suitable machine learning method:

1. Algorithm selection: before training the model, it is necessary to select the type of machine learning. There were two machine learning algorithms investigate: SVM and MLP. It should be noted that this set can be expanded by adding other machine learning algorithms.

2. Training/Testing: During the training phase, the selected algorithm is utilized to develop the machine learning model by feeding it with data.

3. Validation: In this step, the model is validated using a set of metrics and scores.

4. Optimization: In this step, the given model is repeated several iterations with a different set of hyperparameters. These steps are repeated until the most optimal model for the given machine learning algorithm is obtained.

After completing the aforementioned steps, two optimized machine learning models are produced. Depending on the verification stage outcomes, the superior model will be chosen and deployed in the RPL routing protocol-based data collection system to identify distributed denial of service attacks in IoT networks.

4.2. Post-training phase

The post-training phase is responsible for processing data and performing real-time activities. The operation of the data collection system from the RPL routing protocol for detecting distributed denial of service attacks in IoT networks in the post-learning phase will be presented in the form of the following sequence of stages:

1. Aggregation of traffic. This step involves collecting data from several sniffers operating in the Internet of Things network. Supporting multiple sniffers in a network is essential to ensure network scalability and attack detection coverage, especially when dealing with distributed attacks targeting multiple nodes.

In order to check the uniqueness of packets, packets are compared by timestamp. Further, if there is a match, the node identifier is checked. Thus, the data signature is defined as a pair of values <*time stamp, node identifier*>. In case the packet signature matches any received packets from other sniffers, only one instance of the packet will be included in the output queue, while the rest will be dropped. Otherwise, the packets are forwarded to the next set without any further action. This process ensures real-time avoidance of data duplication. It is essential to mention that the data acquisition procedure is executed within the time window w, dividing network traffic into *k* intervals with length *w*.

2. Features extraction. This step involves the same sequence of actions as for the pre-training phase (in offline mode), with the only exception that this process is performed in real time for Internet of Things networks.

3. Classification of attacks. Based on the optimal machine learning model obtained in the pre-training phase, anomalies in network traffic are classified.

4. Generation of results. This step generates the discovery result and creates and sends a UDP packet to the detection agent. The packet contains parameters such as node ID, timestamp, node parent, rank, and discovery result. The detection outcome is a variable that has two potential values, either true or false. If the result is false, it signifies that no attack was identified, and no additional packets will be sent to the detection agent. Conversely, if the detection outcome is true, the packet with the detection result will be transmitted to the detection agent.

It should be noted also about anomalous behavior in the network during which the post-learning phase is activated. In general, a change in the parameters of the network compared to the established indicators of these indicators by more than a given threshold of sensitivity is considered as an anomaly. The sensitivity threshold value is an empirical number specific to each network [22]. In this work, the indicators that are triggers for the activation of the post-training phase are [23-25]:

– Changing the number of DIO information messages. A given node in the DODAG tree can broadcast this message, which lets other nodes know about it. This message is used to get information about whether there are nodes that want to join the tree.

– Changing the number of DIS information messages. If there is no DIO message, and if the node wants to join the DODAG tree, it sends this control message. In this way, DIS allows to generate a search query for any DODAG.

– Changing the number of DAO information messages. That is, requests sent by a child node to the parent or root node. In this (child to parent) message, the parent is asked to allow the child to join the DODAG tree.

5. Gathering data and verifying the detection accuracy of distributed denial of service attacks

An infrastructure based on the Ubuntu operating system and the Cooja simulator was deployed to obtain a dataset for conducting experiments [26]. The main advantage of the Cooja simulator is that it can simulate a sensor node based on its real characteristics using the Java Native Interface (JNI) to execute ContikiOS and TinyOS software code. JNI provides communication between C programming code (usually this programming language is used to flash sensor nodes) and the Java virtual machine. Thus, the Cooja simulator can simulate any sensor node of the platform as closely as possible to a real sensor node operating in the Internet of Things network. A schematic representation of the studied 6LoWPAN-RPL network is shown in Fig. 5.

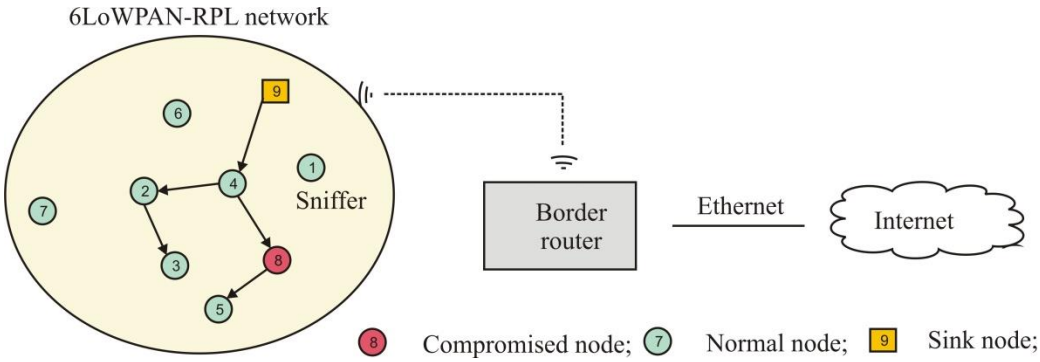

Figure 5: The modeled 6LoWPAN-RPL network

The process of gathering data for the conducting of experiments was carried out in the Cooja simulator. As a result of the simulation in the Cooja environment, the computing capabilities in the form of CPU and memory power are provided to the nodes. All parameters for the studied 6LoWPAN network are given in table 2.

Table 2
Parameters for the simulation process in Cooja

Parameter	Value
Wireless channel model	UDGM
Number of nodes	21
Routing protocol	RPL
Transport protocol	UDP
MAC protocol	CSMA + ContikiMAC
Network size	50 x 100 meters
Mote type	Zolteria Z1
Time of modeling	3 hour

The SHT21 sensor that collects temperature data was used as nodes for the simulation. The parameters of the nodes used in the simulation are given in table 3.

Table 3
The parameters of the nodes used in the simulation

Parameter	Value
Mote type	Zolertia Z1
CPU	16 bit RISC

Memory	8 Кб
Flash memory	92 Кб
Transmitter chip	CC2420
Power	3.3/5 В
Node	STH21
Wireless connection	IEEE 802.15.4, 2.4 ГГц

During the modeling network based on the RPL protocol, all transmitters (ordinary nodes and sink node) use the same type of mote – Zolertia Z1 (fig. 6). The Z1 use a second-generation low-power MSP430F2617 microcontroller with a powerful 16-bit RISC processor clocked at 16 MHz, built-in factory clock calibration, 8 KB of RAM, and 92 KB of flash memory. This device also includes a CC2420 transceiver, compatible with the IEEE 802.15.4 standard, which operates at a frequency of 2.4 GHz with an effective data transfer rate of up to 250 Kbps. The Z1 equipment provides maximum efficiency and reliability with low energy consumption [27].

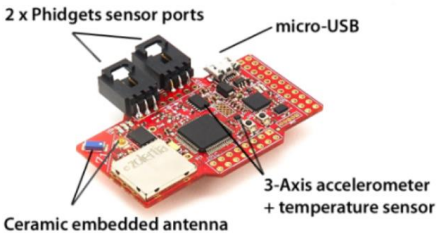

Figure 6: Zolertia Z1 wireless sensor device for temperature measurement in 6LowPAN-RPL networks

To obtain test data, a homogeneous network consisting of two types of nodes a sink and client nodes was deployed (fig. 7). The main task solved in the simulated network was temperature measurement. This task was performed by client nodes measuring and sending the temperature to the sink node at 20-second intervals. Along with this data, other service information such as RSSI level, LQI and ETX values were received in the UDP packets sent to the sink node. The sink node represented the root node, which performed not only an organizing function (maintaining the hierarchy of connections between higher education institutions in the IoT network), but also worked as a server to which data from client nodes arrived. In addition, this node was a bridge between the IoT network and the border router. The organization of the sink node and its functionality is implemented using components in Contiki OS.

All nodes in the simulated network run the modified Contiki 3.0 operating system, including the sniffer nodes. In terms of routing, the standard network stack in Contiki OS based on the RPL protocol was used. The border router is implemented on the basis of the Ubuntu 22.04 operating system, which handles all connections coming from sniffers and sink node (fig.8). The WireShark tool [28] was used to implement sniffers. Figure 8 shows the process of modeling a deployed IoT network in Cooja.

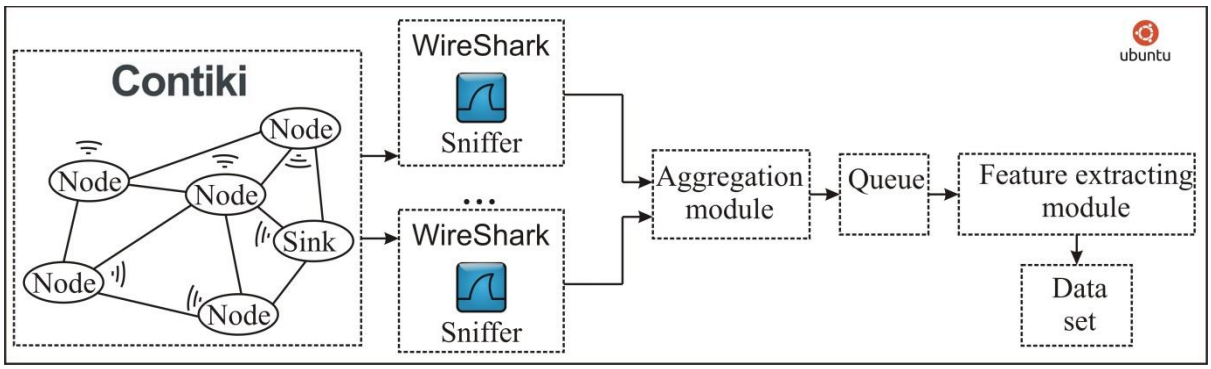

Figure 7: Functioning and implementation of the DGM in a deployed IoT network

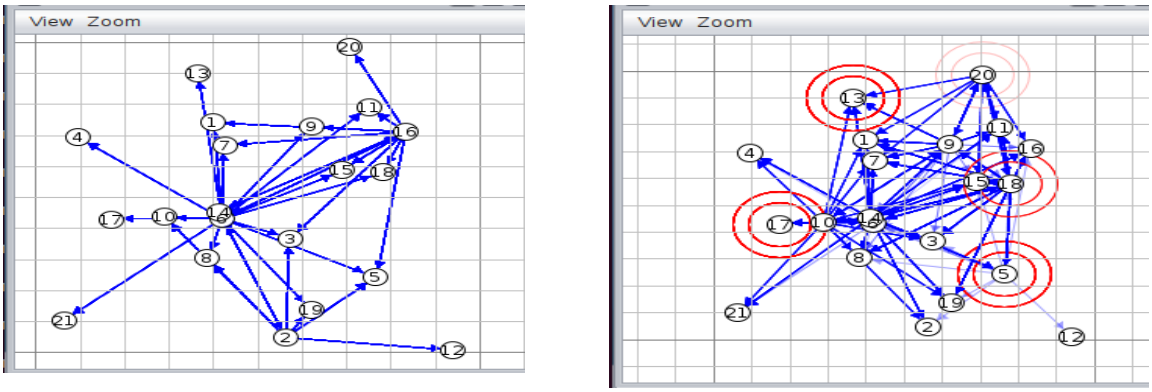

Figure 8: Process of modeling a deployed IoT network in Cooja

5.1. Implementation of black hole attack in the Internet of Things network

A black hole attack is a denial-of-service attack in which a router is supposed to retransmit packets, but instead drops them. This is usually due to a compromised router.

A malicious router can execute a selective attack by dropping packets for a specific network device during a specific time of day, omitting every nth packet or every t seconds, or a randomly chosen subset of packets (as shown in fig. 9). If a malicious router endeavors to drop all incoming packets, the attack can be rapidly recognized through common network tools like traceroute. Moreover, other routers typically exclude a malicious router from their forwarding tables if they notice that it is dropping all traffic, leading to the cessation of traffic directed to the attacker eventually. However, detecting a malicious router that drops packets during a certain time interval or every n packets is typically more challenging because some traffic still traverses through the network.

The black hole attack is commonly utilized for attacking wireless networks due to their distinct architecture from typical wired networks. A compromised host in a wireless network can deceive other nodes into believing that it has the shortest path to the intended destination, thereby redirecting all traffic to itself and allowing it to selectively drop packets at its discretion. Also, in a mobile ad-hoc network, hosts are particularly vulnerable to joint attacks: when several hosts are compromised, they can disrupt the correct operation of other hosts in the network. In this study, the RPL Attacks Framework [30] was used to implement the packet drop attack.

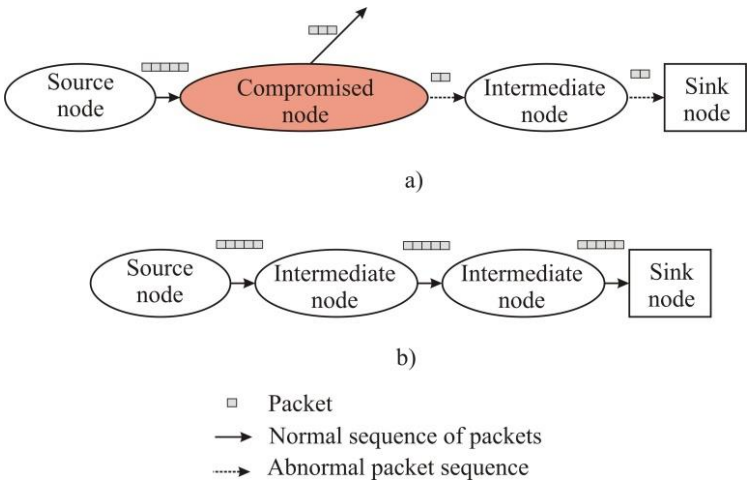

Figure 9: Schematic representation of the data transfer process in the 6LowPAN network: a) during a black hole attack; b) normal functioning

5.2. Machine learning for data collection system from the RPL routing protocol for detecting distributed denial of service attacks in IoT networks

As a result of wireless sensor network modeling, 24,023 feature vectors obtained from IEEE 802.15.4, 6LoWPAN, IPv6 and ICMPv6 packets were obtained. From the received feature vectors, 14,596 samples were assigned to the malicious traffic class, and 9,426 feature vectors were assigned to the legitimate traffic class (table 4).

Table 4

The parameters of the nodes used in the simulation

Class label	Number of samples
The number of feature vectors marked as malicious traffic	14596
The number of feature vectors marked as legitimate traffic	9426

To create a detection model in the distributed denial-of-service attack detection system, the entire data set was divided into 2 parts: a training and a test sample.

The training data set is a set of feature vectors used for the process of training and fitting the parameters of the classifier. This set is 80% of all feature vectors from both classes (i.e., 7,540 samples of legitimate traffic and 11,676 samples marked as malicious traffic). Thus, the training dataset is used to build models that are candidates for recognizing malicious activity in network traffic.

The K-cross-validation method was used to select the optimal hyperparameters for each model. This method is used to find the optimal hyperparameters of the model and to level the processes of underfitting and overtraining of the model.

To perform K-cross-validation, the entire training data set was divided into two parts: a training sample and a validation sample. The value of K was chosen to be 8. This means that out of 8 parts, the model is trained on 7 parts and the validation is done on the remaining part. This process continued iteratively until each of the 8 parts was used as a test set. At each iteration, the classifier model was evaluated using the F1 measure. Based on the results of all K classes and tests of the classifier, the average value of the F1 measure was determined.

A multilayer perceptron with backpropagation algorithm was used for the artificial neural network model. The number of hidden layers, alpha value and activation function were investigated as hyperparameters for the proposed ANN. The number of hidden layers is used to determine the number of layers between the network input and the network output and the number of neurons in each hidden layer. The alpha value is used for regularization, and defines the penalty value used to determine the size of the weights used to prevent overtraining.

According to the results of experiments to determine the optimal parameters, the optimal value of the number of hidden layers is (6, 4), the ReLU activation function, and the alpha value is 0.001.

C and Gamma values are chosen as hyperparameters for the SVM-based model. The parameter C tells the SVM how much to avoid misclassifying each example when training. For large values of C, the optimization will choose a hyperplane with a smaller margin if that hyperplane does a better job of correctly classifying all training points. Conversely, a very small value of C will force the optimizer to search for a separating hyperplane with a larger margin, even if that hyperplane misclassifies more points. A radial basis function is selected as the SVM kernel. According to the results of the experiments, the optimal hyperparameters for SVM were determined to be the value of C at the level of 1 and the gamma parameter, which is 0.001. As for MLP, the definition of hyperparameters for SVM was carried out on the basis of K-cross-validation.

5.3. Evaluation of the effectiveness of the distributed denial of service attack detection system

To determine the effectiveness of the proposed system, an experiment was conducted, which consisted in evaluating the process of detecting distributed denial-of-service attacks by two models of classifiers.

The value of Accuracy, as well as indicators of TP, TN, FP, FN were used as metrics for evaluation.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

As a null hypothesis H0, a statement was defined, which can be formulated as follows: "the network traffic sample has manifestation of abnormality and may be a black hole attack." Then the indicators TP, FP, TN and FN determine:

TP – determines the number of feature vectors marked as malicious and correctly recognized by the system as abnormal activity corresponding to a denial of service attack;

TN – determines the number of feature vectors marked as legitimate and correctly recognized by the system as normal traffic;

FP – determines the number of feature vectors marked as legitimate but erroneously recognized by the system as anomalous traffic;

FN – determines the number of feature vectors marked as malicious but mistakenly recognized by the system as normal traffic.

Table 5

Evaluation of the accuracy of the data collection system for detecting distributed denial of service attacks in IoT networks with ANNs and SVMs models

Classification model	Observation				Metric
	TP	FP	TN	FN	Accuracy
MLP	2546	264	1622	374	0,867
SVM	2686	298	1588	234	0,896

Based on the results of the conducted experiments, it can be concluded that both models of classifiers, which represent the core of the detection module in the proposed system, demonstrated a detection accuracy of more than 85%. Better results were obtained in the SVM-based model (accuracy of detection 89.6%) with a rate of false positives (first type errors) of 6% and a rate of false negatives of 4.87%. It should be noted that the model based on the MLP showed the results of errors of the first type at the level of 5.5, which is lower than the corresponding value in the SVM model. However, from the point of view of criticality for end users, second-order errors are more important, which in this experiment are better precisely in the SVM-based model.

6. Conclusion

As a result of the research the data collection system from the RPL routing protocol for detecting distributed denial-of-service attacks in IoT networks operating on the basis of the 6LoWPAN and RPL protocols. The basis of the proposed system consists of three main modules: a data gathering module, a classification module and a detection and mitigation module. The main feature of the data collection module was that data collection was provided by several sniffers installed in the network and with subsequent aggregation of the collected data. In the basis of the classification module, two machine learning algorithms were investigated. The detection module was used to broadcast a message about the abnormal behavior to the rest of the IoT network nodes, containing the identifier of the compromised node and the path to it in the network.

The purpose of the experiments was to check the accuracy of detecting distributed denial-of-service attacks on the data set received by the data gathering module. In order to obtain a data set for conducting experiments, an infrastructure based on the Ubuntu operating system and the Cooja simulator was deployed, which allowed to simulate an RPL network, the main task of which nodes was to measure the temperature and send the received value to the base station. Based on the operation of the deployed network, network traffic corresponding to both legitimate traffic and traffic affected by a denial of service attack was collected. The total number of test data was 24,023 samples. A black hole attack was used as the attack for the study. According to the results of the experiments, the SVM-based model showed the best reliability indicators, with a false positive rate of 6% and a false negative rate of 4.87%.

7. References

[1] A. Melnyk, Computer memory with parallel conflict-free sorting network-based ordered data access. Recent Patents on Computer Science, 8 1 (2015) 67–77

[2] B. Serrano, J. Fernando, W. Song, et al, A survey on DoS/DDoS attacks mathematical modelling for traditional, SDN and virtual networks. Engineering Science and Technology, an International Journal. 31 (2021). doi:10.1016/j.jestch.2021.09.011.

[3] J. Kafke, T. Viana, Call Me Maybe: Using Dynamic Protocol Switching to Mitigate Denial-of-Service Attacks on VoIP Systems, Network 2022 2(4) 545-567; doi:10.3390/network2040032

[4] Y. Al-Hadhrami, F. K. Hussain, Real time dataset generation framework for intrusion detection systems in IoT, Future Generation Computer Systems 108 (2020) 414-423. doi: 10.1016/j.future.2020.02.051

[5] O. Pomorova, O. Savenko, S. Lysenko, et al., Metamorphic Viruses Detection Technique based on the Modified Emulators, CEUR Workshop Proceedings 1614 (2016) 375–383

[6] O. Savenko, S. Lysenko, A. Nicheporuk et al., Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search, CEUR Workshop Proceedings, 1844 (2017) 555-569.

[7] O. Savenko, A. Nicheporuk, I. Hurman, S. Lysenko, Dynamic signature-based malware detection technique based on API call tracing, CEUR Workshop Proceedings 2393 (2019) 633–643

[8] O. Savenko, S. Lysenko, A. Nicheporuk, B. Savenko, Approach for the Unknown Metamorphic Virus Detection, Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Bucharest Romania, September 21–23, (2017) 71–76.

[9] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, A.Nicheporuk, A Technique for detection of bots which are using polymorphic code, Communications in Computer and Information Science 431 (2014) 265-276.

[10] M.A. Alzahrani, A.M. Alzahrani, M.S. Siddiqui, Detecting DDoS Attacks in IoT-Based Networks Using Matrix Profile, Appl. Sci. (2022) 12(16). doi: 10.3390/app12168294

[11] H. Jing, J. Wang, C.L. Chen, Detection of DDoS Attack within Industrial IoT Devices Based on Clustering and Graph Structure Features, Security and Communication Networks 2022 (2022) 1401683 doi: 10.1155/2022/1401683

[12] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad and G. A. Shah, IoT DoS and DDoS Attack Detection using ResNet, Proceedings of 2020 IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 2020, pp. 1-6. doi: 10.1109/INMIC50486.2020.9318216.

[13] L. Hong, K. Wehbi and T. H. Alsalah, Hybrid Feature Selection for Efficient Detection of DDoS Attacks in IoT, Proceedings of the 2022 6th International Conference on Deep Learning Technologies (ICDLT '22), Association for Computing Machinery, New York, NY, USA, 2022, pp. 120–127. doi: 10.1145/3556677.3556687

[14] B. Bouyeddou, B. Kadri, F. Harrou, Y. Sun, DDOS-attacks detection using an efficient measurement-based statistical mechanism, Engineering Science and Technology, an International Journal 23 4 (2020) 870-878. doi: 10.1016/j.jestch.2020.05.002
[15] B. Wibowo, M. Alaydrus, Smart Home Security Analysis Using Arduino Based Virtual Private Network, Proceedings of 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-4. doi: 10.1109/ICIC47613.2019.8985669.
[16] R. Doshi, N. Apthorpe, N. Feamster, Machine learning ddos detection for consumer internet of things devices, Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 29–35.
[17] R. F. Ibrahim, Q. A. Al-Haija, A. Ahmad, DDoS Attack Prevention for Internet of Thing Devices Using Ethereum Blockchain Technology, Sensors 22 18 (2022) 6806. doi: 10.3390/s22186806
[18] D. Nanthiya, P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja and R. S. Priya, SVM Based DDoS Attack Detection in IoT Using Iot-23 Botnet Dataset, Proceedings of 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-7, doi: 10.1109/i-PACT52855.2021.9696569.
[19] J. Ye, X. Cheng, J. Zhu, L. Song. A DDoS Attack Detection Method Based on SVM in Software Defined Network, Security and Communication Networks (2018) 1-8. doi:10.1155/2018/9804061.
[20] M. B. Farukee, M. S. Zaman Shabit, Md. R.Haque et al, DDoS Attack Detection in IoT Networks Using Deep Learning Models Combined with Random Forest as Feature Selector. Advances in Cyber Security, Penang, Malaysia, 2021, pp.118-134. doi:10.1007/978-981-33-6835-4_8.
[21] Y. Al-hadhrami F. K. Hussain, A Machine Learning Architecture Towards Detecting Denial of Service Attack in IoT. Conference on Complex, Intelligent, and Software Intensive Systems, Springer, 2019, pp.417-429.
[22] K. Fotiadou, T-H. Velivasaki, A. Voulkidis, et al, Network Traffic Anomaly Detection via Deep Learning. Information, 12 5 (2021), 215. doi: 10.3390/info12050215
[23] E. Aydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsström and M. Gidlund, A Central Intrusion Detection System for RPL-Based Industrial Internet of Things, 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), Sundsvall, Sweden, 2019, pp. 1-5, doi: 10.1109/WFCS.2019.8758024.
[24] S. Cakir, S. Toklu and N. Yalcin, RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning, in IEEE Access, vol. 8, pp. 183678-183689, 2020, doi: 10.1109/ACCESS.2020.3029191.
[25] E. Kfoury, J. Saab, P. Younes and R. Achkar, A self organizing map intrusion detection system for RPL protocol attacks", Int. J. Interdiscipl. Telecommun. Netw., 11 1 (2019) 30-43
[26] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne and T. Voigt, Cross-Level Sensor Network Simulation with COOJA, Proceedings. 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, USA, 2006, pp. 641-648. doi: 10.1109/LCN.2006.322172.
[27] B. Thébaudeau, Zolertia z1 motes, 2019, URL: https://github.com/contiki-os/contiki/wiki/Zolertia-z1-motes
[28] Wireshark, URL: https://www.wireshark.org/
[29] C. Black, S. Scott-Hayward, S. A Survey on the Verification of Adversarial Data Planes in Software-Defined Networks. Proceedings of 2021 ACM International Workshop on Software Defined Networks & Network Function Virtualization Security, New York, NY, USA, pp. 3-10. doi: 10.1145/3445968.3452092
[30] RPL Attacks Framework, URL: https://github.com/dhondta/rpl-attacks