BotNet Intrusion Detection System in Internet of Things with Developed Deep Learning

Amirabas Kabiri Zamani
Department of Computer Engineering & Information Technology, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

Amirahmad Chapnevis
Department of Computer Engineering & Information Technology, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

Abstract
The rapid growth of technology has led to the creation of computing networks. The applications of the Internet of Things are becoming more and more visible with the expansion and development of sensors and the use of a series of equipment to connect to the Internet. Of course, the growth of any network will also provide some challenges. The main challenge of IoT like any other network is its security. In the field of security, there are issues such as attack detection, authentication, encryption and the so on. One of the most important attack is cyber-attacks that disrupt the network usage. One of the most important attacks on the IoT is BotNet attack. The most important challenges of this topic include very high computational complexity, lack of comparison with previous methods, lack of scalability, high execution time, lack of review of the proposed approach in terms of accuracy to detect and classify attacks and intrusions. Using intrusion detection systems for the IoT is an important step in identifying and detecting various attacks. Therefore, an algorithm that can solve these challenges has provided a near-optimal method. Using training-based models and algorithms such as Deep Dearning-Reinforcement Learning and XGBoost learning in combination (DRL-XGBoost) models can be an interesting approach to overcoming previous weaknesses. The data of this research is Bot-IoT-2018.

Keywords: Internet of Things (IoT), Intrusion Detection System (IDS), Deep Learning (DL), XGBoost, BotNet Attack.

Introduction

The world of information technology and computers is expanding daily. This development has led to the creation of new systems with a specific type of communication. One of these communications is Machine-to-Machine (M2M). In this type of communication, as a solution to move from single-purpose devices that receive data in the form of commands from an application on the Internet that allows the device with solutions, to multi-objective and applications move towards cooperation together. M2M communication with network structures can benefit from global standardization efforts, which include a number of cases. Among these cases, the following points can be mentioned [1]:

- Establishing standards for compatibility with devices and applications without localization or with minimal localization by the appropriate device ecosystem to reduce the time required for deployment.
- Provide an ecosystem that allows programs to easily share their information and experiences.
- Provide an environment in which secure communication takes place and users' privacy and confidentiality are maintained.

In today's world, M2M solutions abound and their architecture has not changed much since the 1970s. The Franhouver Institute for Open Communication Systems defined the definition of M2M systems as communication terminals independent of human interactive communication with mainstream networks or other terminals in order to automate services. Admittedly, M2M communication network changed dramatically since the 1970s and expanded many capabilities (such as 3GPP M2M communication), but M2M architectural solutions have remained relatively stable. A device associated in the field with a software program on the main network for a specific purpose [2].
The most important issue in any network is security. Security analysis in computer networks is divided into several main categories including layers, identity and location of objects, authentication and permission, privacy, protocols and light weighted encryption, software vulnerabilities, mobile operating systems and so on. The connections between some of these principles in the analysis of security in the computer network environment are interdependent and only the field of Wireless Sensor Networks (WSNs) is a different topic because of a particular type of computer network. Security must be considered in different computer networks layers including sensor, network, platform and application which can be considered as a secure structure in computer networks. Also, according to the reference [3], security considered management as a control panel and synchronization for communications which can be in infrastructure networks, urban transmission network, data network, FTTx network, dimensional networks such as WAN and MAN, ports and sensor connections to the network [12-1].
System protection against vandalism or theft of hardware, software and data is provided as a definition of computer security and cyber security is defined as data protection in the network. According to the traditional definition of security, the topic of security is defined based on the three elements of data confidentiality, data integration and data availability. There are a number of security restrictions on computer networks that include:

- Hardware-based limitations: include energy and computational constraints, memory constraints, and equipment-resistant packaging.
- Software-based limitations: Includes limitations of embedded software and dynamic security patches when updating a piece of software.
- Network-based limitations: include mobility, scalability, multiplicity of devices, multiplicity of communication media, multi-protocol network, and dynamic network synchronization network.

There are four common classifications of attacks on computer networks: Reconnaissance attacks, access attacks, denial of service attacks, and data manipulation attacks. In Reconnaissance attacks, the attacker first examines the network systems and services provided or vulnerabilities of the computer network, and then search to detect any unauthorized entry and collection into the computer network. In most cases, these attacks act as a real access or Denial of Service (DoS) attack. Exploration attacks can be somewhat likened to a thief covering a neighborhood and infiltrating vulnerable homes (such as vacant homes, open doors or open windows). This paper presents a deep learning approach [6-9] based on XGBoost method to provide IoT intrusion detection to detect BotNet attacks.

Proposed Method

The NSL-.KDD data set is used to build a network-based intrusion detection system on the Internet. The research variables are in two categories: one is related to the types of attacks that can be detected, which is defined as Boolean, and using the Bot-IoT-2018 data set to detect intrusion into computer networks on the Internet that have attacks such as BotNet. The other category is about the efficiency of the intrusion detection algorithm, which can be expressed in terms of the duration of intrusion detection and the percentage of error in detecting intrusion and so on. MATLAB software is used to simulate the proposed approach, and the reason is that algorithms can be easily implemented in it. In order to ensure the proposed approach, several evaluation criteria will be used, including accuracy, sensitivity, mean error squares, bit error rate when preventing or detecting an attack, signal-to-noise ratio, and other methods used in other articles. In general, the proposed approach can be presented in the following phases:

- ✓ Select the appropriate data set
- ✓ Determine the type of attack and provide a model for their detection (include BotNet attack)
- ✓ Build a network environment in specific dimensions with the number of nodes
- ✓ Provide an intrusion detection system based on the proposed method, including the following steps and placing data in it.
- ✓ Normalize input data
- ✓ Data training and testing with the aim of classifying that will lead to the discovery of knowledge from the data.
- ✓ Select and extract the best features with XGBoost algorithm and data penetration detection
- ✓ Training data and testing with Deep Learning.
- ✓ Use evaluation criteria to ensure the proposed approach and compare with previous methods.

The main reasons for using the Deep Learning-Reinforcement Learning XGBoost (DRL-XGBoost) algorithm include the following:

- ✓ Has analysis with training on the process of attack and infiltration and creating rules to detect suspicious activities.
- ✓ Identify and identify additional consuming traffic that is suspicious.
- ✓ Prioritize alerts by identifying and marking alerts with lower priority.
- ✓ Identify scanned activities known at the network level and its activities.
- ✓ Analyze activities and provide reports to the team in response to network events and process monitoring by marking repetitive attack activities at the network level.
- ✓ Identify and detect unknown attacks using generalizability.

Networking in the Internet environment requires modeling. Initially, an environment is created in specific dimensions and a number of nodes are placed in it. Positioning refers to the initial deployment that is considered in two ways, either randomly or predefined. In this study, nodes are randomly located in the environment. An intrusion detection system is then provided with the XGBoost algorithm for classification. Once the nodes are set up in a networked environment on the Internet, the data set that has a series of attacks needs to be given to the intrusion detection system on the Internet. Bot-IoT-2018 dataset is normally given to the intrusion detection system, and at the time of starting the network to send and receive data through nodes, all information exchanged in the network, with the intrusion detection system based on the available data from The NSL-KDD data set is examined and tested. But there is a need for a way to penetrate the intrusion detection system to improve its performance.

The most important variables used are attacks on computer networks on the Internet, such as DoS, U2R, R2P, and DDoS. Identifying other variables of computer networks in the Internet to create its structure, including the number of nodes and servers, dimensions of the supported environment in the Internet environment, data distribution rate on the Internet and energy available throughout the network, are other important parameters.

The proposed method is based on the host-based intrusion detection system as well as the network-based intrusion detection system. The factor of host-based intrusion detection system is the factor of host-based intrusion detection system. This factor detects incompatibility in the cloud environment by deploying on the hosts by monitoring the behavior of the system files used in the hosts, network events, and system calls. It should be noted that in each host, a host-based intrusion detection factor is placed, which over time continues the learning process (described below) and maintains its efficiency.

A major difference between the network-based intrusion detection system and the host-based intrusion detection system is the location of these agents in the cloud environment. In the proposed method, these agents monitor the traffic passing through each switch. This increases the computational overhead, but also slightly overlaps the functions between network-based intrusion detection system and host-based intrusion detection system, but in fact from another perspective and level. Attempts are being made to detect infiltration, as well as to provide additional coverage to the work of agents that increase the level of system reliability. Factor performance has two main phases:

1. Determining priority levels
2. Decide on new data

In the priority level determination phase, the determinant determines what data belongs to which priority level. In the decision-making phase, the threat of new data is identified and decisions are made about their level.

The proposed system receives the data online according to the probability cycle. Then, by comparing these data with the centers of different classes, it determines the degree of importance or priority of this type of data.

Once the data is idealized and the desired features are identified, XGBoost-based classification probabilities are performed. With the help of classifiers, data is classified into classes with the same characteristics. In general, classifiers are divided into two groups of supervisors with and without supervisors. In classification with the supervisor, the data are labeled and their affiliation is eloquent. In classification without observers, donors do not have a label. In this study, using the XGBoost classification method, data are classified into three categories, which include penetrating data, healthy data, and suspicious data penetration. Therefore, the method used in this project is classification with the supervisor. For this purpose, XGBoost classification is used to determine probabilities. The XGBoost classification relation is expressed as equation (1).

$$l_r(x) = \frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{p(\omega_2)}{p(\omega_1)} = x \,\epsilon\, class \tag{1}$$

In this case, the function $p(x|\omega_1)$ indicates the conditional density of the corresponding class and $p(\omega_1)$ indicates the posterior probability of each class. Using this classifier, the pattern of lung changes is classified into three classes: infiltrated data, healthy data, and infiltrated suspicious data. The general algorithm works as follows:

- ✓ Step 1) The first step is to formulate the parameters that show the characteristics of the signals with S and the risk criteria or risk factors that are with F, which are carefully adjusted based on an expert and have equation (2).

$$P = \{x|x \in R \cup S\} \tag{2}$$

- ✓ Step 2) Collect data that deals with whether the data is intact or not. Penetration data is shown as a set D which is as equation (3).

$$\{d_1, d_2, \dots d_n\} \; where \; d_i = \{p_{i1}, p_{i2}, \dots p_{im}\} \tag{3}$$

- ✓ Step 3) the data is thus in the pre-processing stage of classification: filling in the missing values between the data and the established states, converting continuous numerical variables to discrete variables using multi-digit thresholds. Inputs to definite inputs are filtering with a multiple filtering approach and data normalization.
- ✓ Step 4) the pre-processed data in the classification is placed in a set. For each confusion matrix model that contains the actual value, the actual value is negated and collected on the floor.
- ✓ Step 5) all the data characteristics and each classified output are placed in its own area, which consists of three areas, ie three floors, which include penetrating data, healthy data and suspicious data.
- ✓ Step 6) Develop the model based on the results found to reduce the dimensions.

After extracting features and classifying with XGBoost, it is necessary to perform XGBoost to combine the two sections, as well as to determine the probabilities of network intrusion detection. After determining the nearest class and focusing on high-similarity data to the new data, we then make a decision that we consider appropriate for the new data. It is not possible to calculate the probabilities among the whole data set, but only the data in the same class is used to calculate these values. The probabilities for the XGBoost classification section are related to the rule $R = X \rightarrow Y$ such as equation (4).

$$P_{XGBoost}(R) = \frac{P(XY)}{|D|} \tag{4}$$

And the probabilities for the XGBoost classification section are in the form of equation (5).

$$P_{XGBoost}(R) = \frac{P(XY)}{P(X)} \quad (5)$$

In the above formulas, $P(.)$ is equal to the number of data from the whole set D in which both X and Y are present. In our proposed method, D is equal to the class from which the closest data was selected. But another important point that has not yet been mentioned about these rules is how to produce them and calculate the probabilities according to the type of data considered, which are discussed below.

If the properties of l_k are divided into two parts, decision and properties, then the whole set of properties can be displayed as $I_k = [f_i, \dots f_n, d_i, \dots d_m]$. In this representation, f_i represents properties such as the type of connection and d_i represents the decision and accepts the value 0 or 1. Of course, only one decision has been made for each data, so only one of the d_i to d_m can be 1. For this data, all the rules are generated in the form of equation (6).

$R_1: f_1, \dots, f_n > S_i$
$R_2: f_1, \dots, f_n \rightarrow S_i$
.
.
.
$R_{2n+1}: f_1, \dots, f_n \rightarrow S_i$
$i \in [1, m]$

(6)

As is clear from Equation (6), a decision may be made for several data. The evaluation of each law is done by calculating the probability values. In fact, the formulas given for calculating probabilities are appropriate for the data in the categories, and here the numerical data are difficult to make.

In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals.

In contrast to bagging techniques like Random Forest, in which trees are grown to their maximum extent, boosting makes use of trees with fewer splits. Such small trees, which are not very deep, are highly interpretable. Having a large number of trees might lead to overfitting. So, it is necessary to carefully choose the stopping criteria for boosting. The boosting ensemble technique consists of three simple steps:

- ✓ An initial model F_0 is defined to predict the target variable y. This model will be associated with a residual $(y - F_0)$.
- ✓ A new model h_1 is fit to the residuals from the previous step.
- ✓ Now, F_0 and h_1 are combined to give F_1 , the boosted version of F_0. The mean squared error from F_1 will be lower than that from F_0.
- ✓ The mean squared error from F_1 will be lower than that from F_0 and calculated as equation (7).

$$F_1(x) \leq F_0(x) + h_1(x) \quad (7)$$

- ✓ To improve the performance of F_1, we could model after the residuals of F_1 and create a new model F_1 as equation (8).

$$F_2(x) \leq F_1(x) + h_2(x) \quad (8)$$

- ✓ This can be done for *'m'* iterations, until residuals have been minimized as much as possible like equation (9).

$$F_m(x) \leq F_{m-1}(x) + h_m(x) \quad (9)$$

Then output of XGBoost is input of deep learning algorithm. The type of deep learning is combination of deep learning with reinforcement learning (DRL-XGBoost. Different deep-reinforcement neural network models have been implemented, the only difference being the research approach in that two neural networks are used in DRL: one executes the current Q function while the other targets the Q function. The Q function is intended as a copy of the current Q function which is Q-Learning from the family of reinforcement

learning algorithms, but works with a delayed coordination due to its presence and combination with a deep neural network. A copy is made after a certain number of training repetitions. The objective Q function is used to calculate the value of Q for the next case $(\hat{q}_t + 1)$. The purpose of this Q function is to prevent the effect of the moving target when performing a slope of more than $\left(\hat{q}_t - q_{ref}\right)^2$ and to prevent the return of q_{ref} dependence on the training network.

The algorithm begins by predicting actions using policy modes and functions. The action prediction is performed for all states of a path $(s_{\{t\}})$ and the sequence of predicted actions is generated. These predicted measures are obtained by sampling the probability distribution of measures $(\pi(a_{\{t\}}))$ provided by the policy performance. This section is considered as the probability of sample distribution. This research uses the symbol $\{T\}$ to indicate a sequence during the time steps of a path. When this symbol is used, it can have a sequence of scales, such as $r_{\{T\}}$ or a sequence of vectors such as $\pi(a_{\{T\}})$ or $\hat{a}_{\{T\}}$, because in the second case, $\pi(a)$ is the probability vector for any possible action under the current policy, and $\hat{a}_t$ is an encoded vector which assigned to the selected part, so extending them to a sequence produces vectors. The reward function creates a reward of 0/1, but in this case, it is a complete sequence of predicted actions $(\hat{a}_{\{T\}})$ and grand truth actions $(a^*_{\{T\}})$ which applied in one direction. The resulting bonus sequence $(r_{\{T\}})$ is converted to the vector of discounted bonus amounts $(R_{\{T\}})$. $(R_{\{T\}})$ is calculated by relation (10).

The policy gradient is based on a policy performance tutorial called $Q_{current}$ and using these two determines the operation that must be performed for each possible case. The policy function is performed with a simple neural network with multiple layers and ReLU activation for all layers except the last layer which has a SoftMax activation which is a possible distribution of actions or $(\pi(a))$. ReLU is a linear function that separately positively inputs, otherwise it will have zero. This has become the default activation function for many neural networks because the model they use is easier to teach and often performs better.

$$R_{\{T\}} = \left[\sum_{i=0}^{T} \lambda^i r_{t+i}, \dots, \sum_{i=T}^{T} \lambda^i r_{t+i}\right] = \left[\sum_{i=0}^{T} \lambda^i r_{t+i}, \sum_{i=1}^{T} \lambda^i r_{t+i}, \dots, \lambda^T r_{t+T}\right] \tag{10}$$

This means that each term $R_{\{T\}}$ corresponds to the decreasing amount of consecutive discount bonuses. The proposed method, due to the use of reinforcement learning model, uses the reward / punishment model that is the basis of these methods. From the vector of the discounted bonus amounts $(R_{\{T\}})$, the average of the discounted bonuses in different paths $(b_{\{T\}})$ is subdivided, and as a result, the superiority vectors of $(A_{\{T\}})$ is obtained. The vector $b_{\{T\}}$ is also called the baseline. The vector means the amount of rewards and punishments that will be counted as a set of data. Advantage values estimate how much better the expected return for a particular element of the path (s_t) is than the average expected return. This is the reason for subtracting the baseline from $R_{\{T\}}$.

The scalar product between the sequences of the vectors $\pi(a_{\{T\}})$ and $\hat{a}_{\{T\}}$‘, derives the probability of a selective action for each time step $(\pi(\hat{a}_{\{T\}}))$, because $\hat{a}_t$ is an encrypted vector. The loss used to train the neural network which is an approximation of the policy function is a type of log-loss function with the sum of the recorded paths of the probability of action performed for a particular element of the path $(\log \pi([\hat{a}_{\{T\}}]_i$ multiplied by the value of the corresponding advantage $([A_{\{T\}}]_i)$. Overfitting occurs when a model trains details and noise in training data to the extent that it negatively affects the performance of the model on new data. This means that random noise in training data are selected and trained by the model as concepts. The problem is that these concepts do not apply to new data and negatively affect the modeling ability to generalize. Overfitting is more in non-parametric and nonlinear models that have more flexibility when learning target performance. Similarly, many non-parametric machine learning algorithms also include parameters or techniques for limiting the amount of model details.

$$J(W,b) = \frac{1}{2k}\sum_{k=0}^{k}\left(\left\|x^{(i)} - \hat{x}^{(i)}\right\|\right)^2 + \frac{\lambda}{2}\sum_{l=1}^{nl-1}\sum_{i=1}^{sl}\sum_{j=1}^{sl+1}\left(W_{ij}^{(l)}\right)^2 \tag{11}$$

In this regard, nl represents the number of layers in the deep neural network and sl is the number of neurons in each input layer. By combining equation (10) and (11), a structure is presented as a combination of deep-reinforced neural network, the general equation is as (12).

$$R_{\{T\}}.J(W,b) = \left(\left|\left|x^{(i)} - \hat{x}^{(i)}\right|\right|^2 . \sum_{i=1}^{T} \lambda^i r_{t+i} + \frac{\lambda}{2}.\left(W_{ij}^{(l)}\right)^2\right. \quad (12)$$

An accurate diagnosis of any intrusion and suspicious symptoms can be provided with the help of this equations during training and testing. Also, the structure of the IoT network is an $N_x \times M_y$ environment which will be in terms of square meters in which the number of nodes or N_{user} in different environments are randomly located by devices connected to the Internet. The energy used to transmit a one-bit packet from transmitter to receiver at a distance d at the same time as intrusion detection can be defined as Equation (13).

$$E_{TX} = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & , d < d_0 \\ lE_{elec} + l\varepsilon_{mp}d^4 & , d \geq d_0 \end{cases} \quad (13)$$

In this regard, E_{elec} is the scattered energy to work with the transmitter or receiver circuit per bit, d is the transmission distance. ε_{fs} and ε_{mp} are the amplifier energy factors for open space and the multi-path dimming channel models, respectively. The intersection d_0 is the threshold distance that depends on the specific scene and the amplifying energy factors, which can be given as $d_0 = \sqrt{\varepsilon_{mp}/\varepsilon_{fs}}$. The energy used to receive one-bit data can be written as equation (14).

$$E_{Rx}(l,d) = lE_{elec} \quad (14)$$

And the energy consumed for the aggregated data is also in the form of equation (15).

$$E_{Agg}(l,d) = lE_{DA} \quad (15)$$

In this regard, E_{DA} is the energy used to send the accumulated data bit. It is necessary to balance the energy between the energies of the sensor nodes to extend the life of the network at the time of intrusion detection. Here, the additive learners do not disturb the functions created in the previous steps. Instead, they impart information of their own to bring down the errors. DRL-XGBoost is a popular implementation of gradient boosting.

Several evaluation criteria have been used in this study, including Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Signal-to-Noise Ratio (SNR), and precision criteria. The accuracy rate is a criterion expressed as a percentage, which is the most important overall result of the evaluation criteria section, which is the accuracy of the relationship and its equation is (16).

$$Accuracy = 100 \times \frac{TP + TN}{TN + TP + FN + FP} \quad (16)$$

In equation (16), TP is false positive, TN positive negative, false positive FP and false negative FN. Equation (17) shows the sensitivity expressed in percentage.

$$Sensitivity = \frac{TP}{TP + FN} \quad (17)$$

Equation (18) shows the data properties expressed in percentage.

$$Specificity = \frac{TN}{TN + FP} \quad (18)$$

Results Discussion and Simulation

The simulator used in this research is MATLAB. One of the reasons for its use is due to the simplicity of using smart methods and algorithms that have already been coded and need to be coded and modeled

according to the problem. But there are other simulators that can be used. Among these simulators, the following can be mentioned that the problem of using each of them is expressed separately (although each of them also has advantages that can be ignored):

- ✓ CloudSim: The ability to use evolutionary methods and swarm intelligence is difficult and must be coded in C ++ and called as a library in this simulator. It also requires a series of functions in the form of header files with the extension .h.
- ✓ NS-2 or NS-3: Installing these two emulators is very complicated and difficult, and with the functionality of an intrusion detection system, it requires the installation of a series of separate packages as plugins. The code must also be written in tcl and the main parts of the proposed algorithm must be written and added in C ++.
- ✓ OPNet: This simulator does not have any interesting free versions in Iran and using it is a high risk of endangering the proposed methods as an idea in a variety of networks.
- ✓ OMNet ++: A powerful emulator, but its installation is complex and requires expressions to launch, and the code of the proposed method must be written and added in C ++.

Bot-IoT-2018[1] as dataset used in this approach. First, it is necessary to define the basic parameters of the computer network on the Internet. "Table 1" shows the parameters of a computer network on the IoT.

Parameters	Magnitude
Network Scale	100x100 m2
Nodes or Users Numbers	200
Nodes or Users Distribution Rate	0.05
Nodes or Users Energy	0.5
ransmission Energy	20 Joule
Radio Range	60 m

Table 1 - Computer network parameters in the IoT

The data used in this research is Bot-IoT-2018 which has different versions. The version used in this research is the 2018 version, which has attacks such as DoS, U2R, R2P, SP and AUB, but we will use BotNet attack to detect and classify. The placement of nodes, which are also moving is done randomly in the environment, which is neighborhood and distance based on Euclidean distance, based on radio radius. This can be seen in "Figure 1".

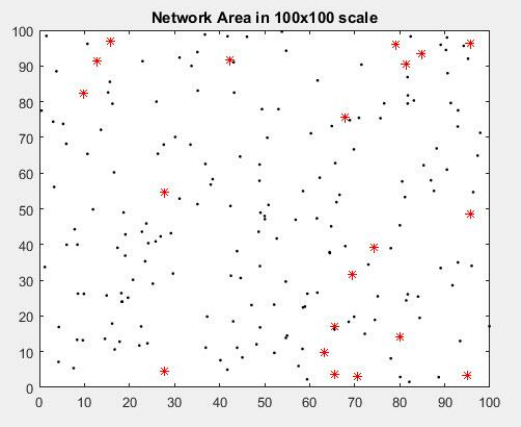

Figure (1), network dimensions and node deployment as well as DRL-XGBoost decision priority selection for packet distribution

Then, in "Figure 2", the network life is shown as a signal, which is based on different network repetitions in penetration detection. Circles indicate the peak or highest and lowest energy consumption.

[1] https://research.unsw.edu.au/projects/bot-iot-dataset

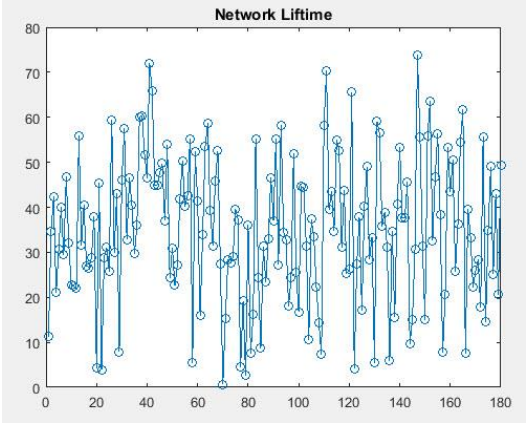

Figure (2), network life as a signal with the highest value and the lowest energy consumption

In "Figure 3", the proposed method can be applied to detect penetration and prevent it. The red graph shows this. In areas where there is a square (blue) and green circles inside it, it shows the prevention of intrusion in that area in terms of signal-to-noise ratio, which is based on network life and stability.

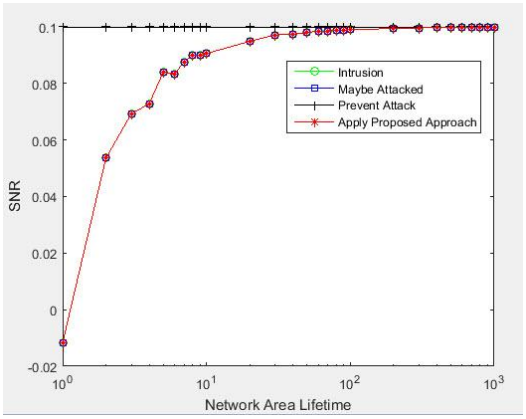

Figure (3), apply the proposed method to detect penetration and prevent it

"Figure 4" also shows the reduction of intrusions and attacks in the Internet-based computer network environment, which are detected and identified by the approach presented in Chapter Three. The red graph shows this. In areas where there is a square (blue) and inside the green circles, it shows the prevention of penetration in that area in terms of longevity and stability.

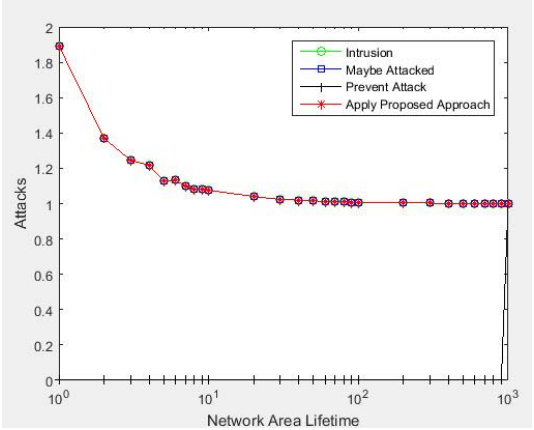

Figure (4), reduction of intrusion and attacks in the network after applying the proposed method

After implementation, the accuracy of the proposed method in detecting and preventing intrusion is 99.9940%. Also, the average error square is 0.1315. "Table 2" compares the proposed interstitial accuracy with four other studies.

Methods	Accuracy
[43]	96.032 %
[20]	89.025 %
[21]	88.700 %
[44]	97.59 %
Proposed Method	99.994 %

Table 2- Comparison in terms of accuracy between the proposed method and two other methods

According to the results, it is clear that the proposed method has a functional advantage over the other two similar and previous methods in terms of accuracy in intrusion detection.

Conclusion

Security is an important issue for the IoT as it is for other networks. In this research, IoT components are initially initialized including sensor nodes, network dimensions, and intelligent physical objects connected to the Internet. The sensor nodes are the users that will transfer the data. Data transfer will be called routing to send packets from one source to another destination. The data packet is placed at the source, at the time of transmitting on the communication channel and the IoT platform where the intrusion detection system targets data packet with trained data. Bot-IoT-2018 data used as input dataset in this research and the main attack is BotNet. This data has been trained once in the DRL-XGBoost intrusion detection system. Therefore, after transmitting the data by the source, it is placed in this area as a monitoring area. The pattern of possible attacks in user-submitted data is compared in pairs with the patterns trained in the DRL-XGBoost based intrusion detection system, and in case of any intrusion, the attack detection module is activated and the operation is performed that can temporarily stop transmitting data from source to destination. In fact, such a system ends when the operations delivers the message to the IoT. Therefore, this system does not have a meaningful termination, i.e. it may be sent in pseudo-continuous data and should be constantly checked, but logically, the termination part will be after the intrusion detection system.

References

[1] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng. Fog Computing for the Internet of Things: Security and Privacy Issues. In IEEE Internet Computing, Volume 21, No. 2, Pages 34-42, 2017.
[2] S. Yi, Z. Qin, and Q. Li. Security and privacy issues of fog computing: A survey. International Conference on Wireless Algorithms, Systems and Applications (WASA), 2015.
[3] V. L. L. Thing. IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach. 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, Pages 1-6, 2017.
[4] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. In IEEE Communications Surveys and Tutorials, Volume 18, No. 1, Pages 184-208, 2016.
[5] Guy Caspi, Introducing Deep Learning: Boosting Cybersecurity with an Artificial Brain, last accessed on July 1, 2017.
[6] Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam, Deep Learning Approach for Network Intrusion Detection System, ACM 9th EAI International Conference on Bio-inspired Information and Communications Technologies, New York,2016.
[7] Kang M. J., and Kang J. W. Intrusion Detection System using Deep Neural Networks for In-Vehicle Network Security. PLOS One, Volume 11, Issue 6, 2016.
[8] Y. Li, R. Ma, and R. Jiao. A Hybrid Malicious Code Detection Method based on Deep Learning. In International Journal of Security and Its Application, Volume 9, Pages 205-206, 2015.
[9] Yoshoua Bengiu, and Pascal Lamblin. Greedy Layer-wise Training of Deep Networks. In Advances in Neural, Nr. 1, Pages 153-163, 2007.
[10] Antonio de Souza, Cristiano, Becker Westphall, Carlos, Bobsin Machado, Renato, Bosco Mangueira Sobral, João, and dos Santos Vieira, Gustavo. Hybrid approach to intrusion detection in fog-based IoT environments. Computer Networks, Volume 180, 24 October 2020.
[11] Li, Xinghua, Hu, Zhongyuan, Xu, Mengfan, Wang, Yunwei, and Ma, Jianfeng. Transfer learning based intrusion detection scheme for Internet of vehicles. Information Sciences, Volume 547, Pages 119-135, 8 February 2021.
[12] Akbar Telikani, and Amir H. Gandomi. Cost-sensitive stacked auto-encoders for intrusion detection in the Internet of Things. Internet of Things, Available online 3 October 2019, 100122, In Press, Corrected Proof.
[13] Anderson, and James, P. Computer Security Threat Monitoring and Surveillance. 1980.
[14] E. D., Denning. An intrusion-detection model. IEEE Transactions on Software Engineering, 1987, Vol. 13, Issue 2, pp. 222–232.
[15] Farid, Dewan Md., and Rahman, Mohammad Zahidur. Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm. JOURNAL OF COMPUTERS, 2010, Vol. 5, No. 1.
[16] Smaha, Stephen E. Haysatck: An Intrusion Detection System. Tracor Applied Science, 1988, INC. Austin, Texas.
[17] Vaccaro, H. S., and Liepins, G. E. Detection of anomalous computer session activity. In Proceedings of the 1989 IEEE Symposium on Research in Security and Privacy, 1989. pp. 280–289.
[18] Heberlein, L. T., Mukherjee, B., and Levitt, K. N. Internet security monitor: An intrusion detection system for large-scale networks. In 15th National Computer Security Conference, Baltimore, MD, 1992.
[19] Lunt, T., Tamaru, A., Gilham, F., Jagannathan, R., Jalali, C., Neumann, P. G., Javitz, H. S., Valdes, A., and Garvey, T. D. A real time intrusion detection expert system (IDES). Technical report, SRI, 1992.
[20] Nima Aberomand. “Network Intrusion Detection Classification Using Optimized Probabilistic Neural Network.” Recent Advances in Computer Supported Education, Educational Technologies Series, Vol. 19, Michigan State University, pp. 108-111, 2015.
[21] Mehdi Moradi, and Mohammad Zulkernine. “A Novel Network Base System for Intrusion Detection and Classification of Attacks.” 2012.
[22] Mehdi Moradi, and Mohammad Zulkernine. “A Novel Network Base System for Intrusion Detection and Classification of Attacks: A Survey.” 2012.
[23] Madhusmita Mishra, Amrut Ranjan Jena, and Rajas Das. “A Probabilistic Neural Network Approach for Classification of Vehicle.” International Journal of Application or Innovation in Engineering & Management (IJAIEM). Vol. 2, 2013.
[24] Partha Gosh, Abhay Kumar Mandal, and Rupesh Kumar. “An Efficient Cloud Network Intrusion Detection System.” Advances in Intelligent Systems and computing, Vol. 339, pp. 91-99, 2015.
[25] Science Direct News. IoT multiplies risk of attack. 2015, Vol. 2015, Issue 5, pp. 20.

[26] Science Direct News. Major ISPs targeted in Internet of Things botnet attacks. Network Security, 2016, Vol. 2016, Issue 12, pp. 1-2.
[27] Conti, Mauro, Dehghantanha, Ali, Franke, Katrin, and Watson, Steve. *Internet of Things* Security and Forensics: Challenges and Opportunities, Future Generation Computer Systems, 2017, In press, accepted manuscript, Available online 26 July 2017.
[28] Pan, Meng-Shiuan, and Yang, Shu-Wei. A lightweight and distributed geographic multicast routing protocol for IoT applications. Computer Networks, 2017, Vol. 12, pp. 95-107.
[29] Jin, Yichao, Gormus, Sedat, Kulkarni, Parag, and Sooriyabandara, Mahesh. Content centric routing in IoT networks and its integration in RPL. Computer Communications, Internet of Things: Research challenges and Solutions, 2016, Vol. 89-90, pp. 87-104.
[30] Kharkongor, Carynthia, Chithralekha, T., and Varghese, Reena. A SDN Controller with Energy Efficient Routing in the Internet of Things (IoT). Procedia Computer Science, Twelfth International Conference on Communication Networks, ICCN 2016, August 19–21, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, Bangalore, India, 2016, Vol. 89, pp. 218-227.
[31] Krishna, G. Gautham, Krishna, G., and Bhalaji, N. Analysis of Routing Protocol for Low-power and Lossy Networks in IoT Real Time Applications. Procedia Computer Science, Fourth International Conference on Recent Trends in Computer Science & Engineering (ICRTCSE 2016), 2016, Vol. 87, pp. 270-274.
[32] Badenhop, Christopher W., Graham, Scott R., Ramsey, Benjamin W., Mullins, Barry E., and Mailloux, Logan O. The Z-Wave routing protocol and its security implications. Computers & Security, 2017, Vol. 68, pp. 112-119.
[33] Almobaideen, Wesam, Krayshan, Rand, Allan, Mamoon, and Saadeh, Maha. Internet of Things: Geographical Routing based on healthcare centers vicinity for mobile smart tourism destination. Technological Forecasting and Social Change, In Press, Corrected Proof, 2017.