Future Generation of Communication and Internet of Things (FGCIOT)

Journal homepage: https://sanad.iau.ir/journal/fgciot/

PAPER TYPE (Research paper)

Improving intrusion detection system in the internet of things using a combination of convolutional neural network and cuckoo algorithm

Ali Shahriari[1], Mohammad Hosein Davarpour[2], Mohammad Ahmadinia[1,*]

[1] Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran

[2] Department of Computer Engineering, Semnan Branch, Islamic Azad University, Semnan, Iran

Article Info

Article History:
Received: 1 January, 2024
Revised: 20 February, 2024
Accepted: 27 February, 2024

Keywords:
Internet of Things, intrusion detection, convolutional neural network, cuckoo algorithm, dimensionality reduction

*Corresponding Author's Email Address: ahmadinia@gmail.com

Abstract

The Internet of Things (IoT) refers to the connection of various devices to each other via the internet. Conceptually, the IoT can be defined as a dynamic, self-configuring network infrastructure based on standards and participatory communication protocols. The main goal of the IoT is to lead towards a better and safer community. However, one of the fundamental challenges in developing the IoT is the issue of security, and intrusion detection systems are one of the main methods to create security in the IoT. On the other hand, Convolutional Neural Network (CNN), with its specific features, is one of the best methods for analyzing network data. This network is a type of deep neural network composed of multiple layers that can ultimately reduce the dimensions of features. Additionally, the cuckoo algorithm has parameters required for configuration in the initial search, which are very few and can naturally and efficiently cope with multi-state problems. In this paper, a new method for intrusion detection in the IoT using CNN and feature selection by the cuckoo algorithm is presented. Simulation results indicate the satisfactory performance of the proposed method.

Introduction

The internet of things is a new paradigm that integrates the internet with physical objects belonging to various domains such as smart home, industrial processes, human health, environmental monitoring, and more. The presence of internet-connected devices deeply embeds in our daily activities, bringing about numerous benefits but also introducing various security challenges. Intrusion detection systems (IDS) have been vital tools for protecting networks and information systems for over two decades. However, applying traditional IDS techniques to the IoT is challenging due to its unique characteristics such as devices with limited resources, specialized protocol stacks, and specific standards. Recently, with the technological revolution and digital transformation, data security has become crucial as a significant amount of data is generated from various networks, leading to a considerable increase in network traffic. This expansion has brought about some security challenges, including various network attacks, both known and unknown. Intrusion can be described as attempts or actions to threaten a computer or network in terms of confidentiality, integrity, and/or availability [1]. IDS is the best solution due to its ability to differentiate between attacks occurring inside or outside a corporate network [2]. The primary goal of the Internet of Things is to lead to a better and safer society, where securing information in terms of general security, environmental protection, healthcare, production, etc., is essential[3]. Since IDS is one of the primary security methods used for securing the IoT, there is a tendency to use more than one technique simultaneously [4]. Various methods and techniques have been designed and implemented to protect IoT operating systems, such as firewalls, data encryption, and user authentication patterns. As intrusion detection systems are one of the primary security measures used for IoT security, there is a trend towards using multiple techniques simultaneously, as proposed by Alharbi et al. accordingly, various methods and techniques have been designed and implemented to

Doi:

protect IoT operating systems, such as firewalls, data encryption, and user authentication patterns. The classical security techniques are becoming inefficient and ineffective in addressing IoT security issues, opening the door for a new generation of intrusion detection systems that are built using machine learning and artificial neural networks [5]. The convolutional neural network is a deep neural network composed of multiple layers. Initially, in the convolution layer, a set of filters, also known as convolution kernels, is applied to the input data. Each filter operates on the input data to produce a feature map. By collecting all the generated feature maps together, the final output of the convolution layer is obtained. Then, in the pooling layer, this feature map works on all features to perform sampling, reducing the dimensions of the features. Finally, in the fully connected layer, it takes the outputs of the previous layers and transforms them into a single vector that can be input to the next layer. One of the significant problems with many feature selection algorithms is the high computational time complexity. The cuckoo algorithm has parameters required for configuration in the initial search, which are very few and can naturally and efficiently cope with multi-state problems [6].

This paper will focus on a new method for intrusion detection in the Internet of Things using Convolutional Neural Network and feature selection by the cuckoo algorithm.

Related Works

In the field of providing intrusion detection systems for the Internet of Things (IoT), various works have been done. Moudar and his colleagues proposed a method for an IoT intrusion detection system using deep neural networks in 2019. In this work, a fully automated artificial intrusion detection system for IoT security against cyber attacks has been presented. The proposed model uses a multi-layer recurrent neural network designed for the security of IoT computations very close to end-users and IoT devices. It has been validated using the NSL-KDD dataset and demonstrated using a balanced version of challenging datasets. The model's performance has been measured using various common metrics, and two additional metrics, Matthew's correlation and Cohen's kappa coefficients, have been added for more precise evaluation. The stability and resilience of the proposed model have been proven in terms of various performance metrics through experimental results and simulations. As a result, this model exhibits a high sensitivity to DoS attacks, indicating one of the prominent attacks, enhancing IoT network development besides identifying other attack categories such as Probe, R2L, and U2R in a competitive computational burden, as each record requires an average of 66 microseconds for processing. Therefore, the proposed model has the ability to function properly and efficiently in real-time environments[5]. Samira and her colleagues presented an efficient method for anomaly detection using feature selection and evolutionary neural networks in 2020. In this research, important features are selected to enhance IDS performance and create a smaller dataset to reduce execution time for attack identification from a significant network dataset. This research has designed an anomaly-based detection system by adopting the CSA fuzzy search algorithm, named the MCF fuzzy mutant, for feature selection, and the ENN evolutionary neural network for classification. The proposed search algorithm uses mutation to explore the search space more accurately, allowing candidates to escape from local minima. Additionally, the solution value is evaluated based on the objective function and FCM clustering method to provide the best results for overlapping datasets and create a fuzzy membership search domain. A proposed model is practically used for intrusion detection problems and has been validated using the NSL-KDD dataset. Experimental results show that reducing features by selecting and using the most important ones can improve execution time while increasing IDS performance and efficiency[1]. Sia Kyran and her colleagues proposed a method for building an intrusion detection system for the IoT environment using machine learning techniques in 2020. An experimental platform for simulating the IoT environment is built using the ESP8266 MCU ESP node, DHT11 sensor, and wireless router. An adversarial system is built using a laptop system that performs attack actions. Data collected from sensors include temperature, humidity, and the point of arrival, which are transferred to the ThingSpeak platform using a wireless gateway. Machine learning classifiers such as Naïve Bayes, SVM, Decision Tree, and Adaboost are employed to classify data into normal and attack classes. Thus, machine learning algorithms can be used to build IDS for IoT networks[7]. Abdollahi Darhab and his colleagues presented a system for intrusion detection in the IoT based on temporal convergence neural networks and efficient feature engineering in 2020. They identified five key design principles that should be considered when developing IDS based on deep learning for IoT. Based on these principles, a Temporal Convergence Neural Network (TCNN), a deep learning framework for IoT intrusion detection systems, which combines Convolutional Neural Network (CNN) and causality-based complexity, is designed and implemented. TCNN is combined with the Synthetic Minority Over-sampling Technique for Non-continuous SMOTE-NC to manage imbalanced datasets. Also, efficient feature engineering techniques, including feature reduction and transformation, are combined. TCNN is evaluated on the Bot IoT dataset and compared

with two common machine learning algorithms: Logistic Regression (LR) and Random Forest (RF), and two deep learning methods: LSTM and CNN. Experimental results show that TCNN achieves a good balance between effectiveness and efficiency[8]. Fal Sadikin and his colleagues presented a combined Zigbee IoT intrusion detection system using safe and efficient data collection in 2020. This research briefly involves implementing intrusion detection, covering various detection methods for identifying known attacks and potential new attack types in Zigbee IoT systems. Furthermore, this research introduces a safe and efficient method for collecting intrusion detection data on a large scale. Thus, it provides a reliable mechanism that can operate under the precise resource requirements imposed by current IoT systems. A rule-based approach is used to develop an accurate detection mechanism for known attacks, but it is also effective for anomaly detection. However, the rule-based method introduces complexity and is time-consuming to determine precise rules for accurate detection. Machine learning is specifically used to create a complex model of normal behavior used for anomaly detection, which is a much more efficient method and also takes less time. However, it potentially introduces false alarms in real intrusion detection deployment. Additionally, this research introduces a new mechanism for reporting data that increases data collection efficiency in IoT systems on a large scale[9]. Mandal and his colleagues presented a method for better security using machine learning for IoT intrusion detection systems in 2020. This research has focused on security challenges for intrusion detection in IoT systems. The primary focus is on the machine learning classification algorithm executed in the IoT system network to enhance performance in attack detection. Thus, the result of anomaly identification in various IoT networks for 500 cycles of each type with 800 bits of data transmission, to identify TCP packets with anomalies or normal transactions with a threshold of 0.499 for intrusion detection. The classification model has an accuracy of 94.57%. The classification report details include precision, recall, and F1 score between 0 and 1, with the model applied for training with a dataset and for testing with a dataset, providing more accurate results based on macro-average and weighted-average of TCP packets[10].

Proposed Method

This section discusses the proposed method in detail. The proposed method is based on deep learning and convolutional neural networks (CNNs). CNN is a specific type of neural network with multiple layers. It processes data with a network-like arrangement and then extracts important features. Input data are preprocessed in the traffic processing unit, leading to traffic data in a suitable format for processing by the convolutional network. If these connections are classified as normal, they are attacked by the intelligent intrusion detection unit. The proposed model can be implemented in the Internet of Things (IoT), which is very close to end-users and devices existing in the IoT. This model trains a convolutional neural network and increases intrusion detection capability in identifying attack/normal behavior using feature selection technique by the evolutionary algorithm.

In short, the steps of the proposed model can be shown in the following flowchart:

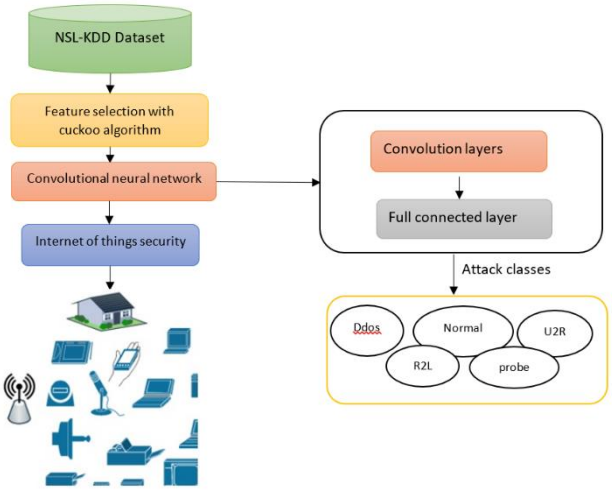

Fig. 1: Chart of the proposed method

Below, the various stages of the process are described:

A. Data Preprocessing:

Prior to training or testing, preprocessing is used to identify each feature and the value of the features. It is also used to convert symbolic feature mappings into numerical values. Instead of symbolic features like HTTP, TCP, UDP, SMTP, etc., integer values are used. All attacks will be labeled numerically to one of the five values: Label number 1 - Normal class, Label number 2 - DOS class, Label number 3 - R2L class, Label number 4 - U2R class, and Label number 5 - Probe class. You can see the attacks for each of these classes in Table 1.

Table 1: Examining the characteristics of the attack assigned to the desired class [11]

Attack Class	Sub-attacks assigned to the target attack
DOS	{ 'back' ; 'land' ; 'neptune' ; 'smurf ' ; 'teardrop' ; 'pod' ; 'apache2' ; 'udpstorm' ; 'mailbomb' ; 'processtable' }
R2L	{ 'spy' ; 'warezclient' ; 'waremaster' ; 'named' ; 'sendmail' ;'snmpgetattack' ; 'snmpguess' ; 'worm' ; 'xlock' ; 'xsnoop' } ; 'phf'

U2R	{ 'buffer_overflow' ; 'loadmodule' ; 'perl' ; 'rootkit' ; 'httptunnel' ; 'ps' ; 'sqlattack' ; 'xterm' } ; 'phf'
Probe	{ 'ipsweep' ; 'nmap' ; 'portsweep' ; 'satan' ; 'mscan' ; 'saint' ; 'i4weep' } ; 'phf'

B. Data Normalization:

Convolutional Neural Networks (CNNs) are recognized as a model for image classification; therefore, the input format for CNNs should be an image. The dataset containing the pixel matrix values of the image needs to be normalized in general. Each value in a pixel ranges from 0 to 255. The KDD-99 dataset is a network traffic dataset consisting of 41 features in a record with heterogeneous types. These types could be strings, integers, or decimals, with entirely different ranges. The feature values of the dataset are not suitable for input to the CNN, so initially, the data needs to be normalized to a new dataset containing integer values, with the normalization range from 0 to 255. However, most values in the KDD dataset are less than 122, and there are a few records of anomalies larger than 255. For data normalization, algorithm (1) for data preprocessing is proposed. If the data is a string, the value can be easily considered as an integer from 0 to 255. For example, if a string field consists of 3 different values, the new values would be normalized to 0, 128, and 255.

C. Feature Selection with CSA Algorithm

To implement the CSA algorithm as a computationally feasible algorithm, researchers used three principles: (a) Each cuckoo lays one egg at a time and randomly places it in a selected nest. (b) Optimal nests contain eggs of excellent quality and are passed on to the next generation. (c) Multiple predetermined nests exist, and the egg laid by the cuckoo is identifiable with a probability of pu ∈ (0,1). When this happens, the egg is removed or the nest is abandoned, and a new nest is created. Regarding the mentioned rules, CS is executed as follows: Individual eggs in nests represent candidate solutions. Therefore, a cuckoo can only lay one egg in a nest, while generally each nest can have multiple eggs as solutions. CS is responsible for generating new solutions and possibly replacing inappropriate solutions in the population. The quality of solutions is evaluated and resolved by the actual performance of the problem, which must be maximized. The last rule estimates the switching probability by pu as the "switching probability", determining when the worst nest is replaced by a randomly created new nest. This factor balances the two parts of the CS process, exploration and exploitation [12]. Therefore, excessive exploitation leads to premature convergence, while excessive exploration reduces the convergence rate. In generating new solutions X(t+1) for cuckoo i, the following equation has been used for flight execution.

$$x_i(t+1) = x_i(t) + \alpha \oplus \text{flight}(\lambda) \quad (1)$$

where α>0 indicates the determination of the step size based on the scale of the problem. It is often possible to use α=1. The symbol ⊕ is an internal multiplication. Often, flights follow a random motion, however, their random steps are derived from the distribution for large steps given in the equation below, which has infinite mean with infinite variance.

$$\text{flight} \sim \text{u} = t^{-\lambda} \quad (2)$$

In order to improve the efficiency of CSA to achieve better solutions, modified CSA has been used as feature selection in this research. The proposed feature selection method increases the efficiency of the CS algorithm to select the best features. In this research, we will implement a feature selection algorithm called cuckoo for network intrusion data, so that intrusion in network traffic can be detected in real time using high accuracy and real-time speed. The feature selection algorithm generally consists of two parts: feature evaluation and search method.

In the cuckoo algorithm, the solutions are continuously updated in the search space towards valuable positions. In addition, because the issue of selecting or not providing a feature is specific, a binary vector is used for the solution, so that 1 is equal to selecting the feature to create a new data set and 0 is not selected. To build this binary vector, equation 3 and 4 are used, which can provide only binary values in the network and limit new solutions to only binary values [13].

$$S\left(X_i^j(t)\right) = \frac{1}{1+e^{-X_i^j(t)}} \quad (3)$$

$$X_i^j(t+1) = \begin{cases} 1 & if\ S\left(X_i^j(t)\right) > \sigma. \\ 0 & otherwise \end{cases} \quad (4)$$

D. The objective function of the cuckoo algorithm

In the proposed method, the convolution algorithm is used as the objective function, and the detection accuracy is considered in this algorithm. So that after reducing the number of features, the neural network is trained, the results are checked with real values. The main features are sent to the convolution neural network and the neural network performs the processing operation and a result is obtained that shows the accuracy value, which is the main output value of the proposed method. To obtain the mean square error in the objective function from a set or n data, the following equation can be used (Equation 5).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (5)$$

The value of each cuckoo is calculated using this function, where $(\frac{1}{n}\sum_{i=1}^{n} \ldots)$ does the averaging operation and $(Y_i - \hat{Y}_i)^2$ is the squared value of the error of each calculates the data. So MSE is the mean square of the errors.

E. CNN convolutional neural network

KDD consists of 41 traffic attributes and one tag attribute that determines which attack each data belongs to. Once you have a new dataset, you need to render it as a pixel matrix. The input image of the convolutional neural network must be a square matrix, the most appropriate and smallest of which for the value of 41 features is a 7*7 matrix. Each record in the new dataset can be converted into a 7*7 square matrix. The last eight bits of the matrix are set to zero.

F. Convolution neural network model design:

Convolutional neural network consists of three main components: input, output and hidden layers, which are composed of convolution layers, integration layers, fully connected layers. First, convolution layers apply a certain number of convolution filters to the image. For each sub-region, the layer performs a set of mathematical operations with a specified spatial extent and step value to produce a single value in the output feature map. Convolution layers then typically apply an activation function to the output to introduce nonlinearities into the model. In the second step, the fusion layers reduce the image data extracted by the convolution layers to reduce the dimensions of the feature map to reduce the processing time. Finally, fully connected layers perform the classification of features extracted by convolution layers and downscaling by pooling layers. and determines which class the input image belongs to. The convolution layer extracts the unique features of the image while preserving the input/output and spatial information of the image, and by adding an integration layer to the convolution layer, it reduces the feature data size. Figure 2 shows the structure of the convolutional neural network model.

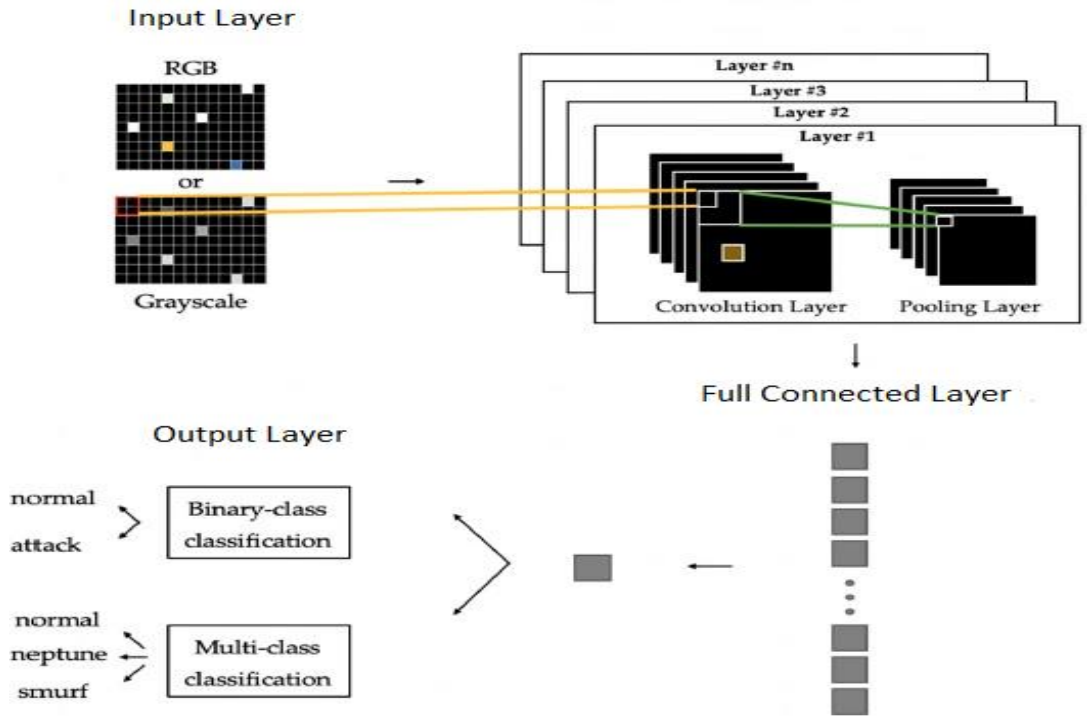

Fig. 2: Convolution neural network model design[14]

Convolutional neural network is a deep neural network that consists of several layers. The three main types of layers are as follows:

(a) Convolution layer: It applies a set of filters, which are also known as convolution kernels, to the input data. Each filter is applied to the input data to produce a feature pattern. By adding all the generated feature patterns together, the final output of the convolution layer is obtained. The main core of the CNN network is the convolution layer, which accounts for the majority of convolutional neural network calculations. Each convolution layer in the convolution neural network contains a set of filters and the output is made from the convolution between the filters and the input layer. The output of the convolutional layer is called a feature map.

Convolution layers have several convolutional kernels that are composed of kernel weights. The convolution kernel i, the weight coefficient is shown by wi, and the deviation value is shown by bi . xi-1 is the input layer of

convolution and this layer is processed by equation 6 [15]

$$x_i = f(w_i \otimes x_{i-1} + b_i) \quad (6)$$

where x_i is the convolution output, i is the convolution kernel, $\otimes$ is the convolution operation and f(x) is the activation function.

(b) Pooling layer: This layer works on the feature patterns to perform sampling, which reduces the dimensions of the feature patterns. Average pooling and maximum pooling are the most common pooling methods. Pooling layer is another important layer in convolutional neural network. The purpose of the pooling layer is to reduce the spatial size of the feature map obtained by using the convolution layer. The pooling layer has no teachable parameters. It just does a simple and effective sampling.

(c) Full layer connection: It takes the output of the previous layers and converts them into a single vector that can be the input of the next layer. One of the main uses of this layer in the convolutional network is to use it as a classifier. That is, the set of features extracted using convolutional layers will eventually become a vector. Finally, this feature vector is fed to a full connection classifier to identify the correct class. The softmax function was used to calculate the probability distribution of an N-dimensional vector. The correct calculation of the output probability helps to determine the appropriate target of the class and the input data set, and the maximum probability of the values is increased by using an exponential element. The softmax equation is shown in equation 7:

$$Q_i = \frac{e^{z_i}}{\sum_{i=1}^{M} e^{z_i}} \quad (7)$$

where i and z_i are the output of the previous layers, Q_i shows the output of the softmax function, and M is the total number of output nodes [15].

Two convolution layers are used in the design of this article. In the first convolution layer, 64 filters with size [3*3] are used. The input data for this layer are images generated from network traffic, each of which has a maximum size of [7*7]. This layer results in feature maps with size [64*7*7]. The second convolution layer uses 128 filters whose size is [3*3], thus giving feature maps of size [128*7*7].

Pooling layers are not used in the architecture of this research, because for samples that are very small in size (for example, image size 7*7), there is no need for sample dimension reduction operations.

In this article, convolutional neural network with two fully connected layers is used. The feature maps produced by convolution layer 2 are used as input for the first fully connected layer. The last fully connected layer (output layer) computes the class scores, resulting in a matrix of size [5*1*1], where each of the 5 numbers corresponds to a class (normal and attacks). In addition, the elimination parameter, d = 0.5, has been used to avoid the problem of overtraining in the convolutional neural network training stage.

Simulation and evaluation

In this section, the result of applying the stated techniques and algorithms on the NSL-KDD data set will be presented and also the simulation results will be analyzed. All stages of feature selection and classification are performed assuming that the problem of intrusion detection is five classes. At first, in order to select the appropriate number of features, the cuckoo algorithm is used. In the next step, using the selected features, the efficiency of the convolutional neural network algorithm is evaluated on the applied data set with the aim of improving classification accuracy. In this research, MATLAB software version 2017 was used for the pre-processing and simulation stages of different parts.

NSL-KDD dataset: In this article, the NSL-KDD dataset [16] will be used for model training, testing and validation. The data characteristics that represent the incoming traffic of the network system naturally include various forms of data. The NSL-KDD dataset has a total of 42 columns with a number of 148,517 records, 41 columns for the characteristics of this data and the last one column for identifying the type of attack (for example, normal behavior 1, DOS attack, Probe, U2R and R2L are 2, 3, 4 and 5 respectively) are considered. The values of the records in some columns are numerical and some columns are strings, so they should be normalized at first due to the inconsistency of these columns. For this, a number should be considered instead of any string.

The proposed model of this research is evaluated with two different settings. In the first test, the number of convolution layers in the model is changed from 1 layer to 3 layers. Then, in the second experiment, the number of layers that gives the best result is tested with the number of different training repetitions. The number of training repetitions is 100, 1000 and 1500. After that, with the settings that give the best results, the accuracy of the proposed model is compared with other researches.

Table 2: Simulation parameters

Variable	Value
The primary population of cuckoos	20
At least egg per cuckoo	2
Maximum eggs per cuckoo	4
The maximum number of iterations of the cuckoo algorithm	100
The number of cuckoos remaining in the environment	10
The number of convolution layers	2
The number of filter layers	64 و 128
Filter size	3*3

Activation function	Relu
Number of full connection layers	2
The number of neurons in the full connected layer	100
Bios	0
Output layer size	1*1*5

According to the input dataset, 70% of the dataset is used for training and 30% for testing. First, the data is pre-processed and each record is converted into a maximum 7*7 matrix as input data. Then the suitable features are selected with the cuckoo algorithm and finally the attacks are classified with the convolutional neural network.

Evaluation of the classification model is one of the most important parts that should be paid attention to in the classification. The evaluation of the classification model should be based on training and test samples. Various evaluation criteria have been presented, among the most important of which we can mention the correctness, accuracy, recall and F-measure criteria. The most important criterion for determining the efficiency of a classification algorithm is the accuracy criterion. This measure calculates the overall accuracy of a category. This measure indicates what percentage of the entire data set is correctly classified.

The two values of TN and TP are the most important values that should be maximized to maximize classification efficiency. Considering that in classification problems, there may not be a balance between the number of samples of different categories, one category may have more samples than the other category, as a result, the final model will move towards the category with the most samples. Therefore, the batch with a small number of samples will practically not have much effect on improving or not improving the performance. It can be concluded that the Accuracy criterion is not a suitable criterion in the data set that has unbalanced categories with different number of samples.

The classification error criterion is also obtained from equation (8). This relationship is exactly the opposite of the Accuracy criterion. The lowest value is equal to zero (the best efficiency) and the highest value is equal to one (the lowest efficiency).

$$ER = \frac{FN+FP}{TP+FP+FN+TN} = 1 - Accuracy \quad (8)$$

Precision measures show the percentage that are correctly classified out of all the categories assigned to that category by the classifier. In other words, it shows the classification accuracy of the category i according to all the cases where the label i was suggested by the category r for the examined sample. Recall criterion for a category, which is correctly classified among all attack categories belonging to that category. In other words, it shows the classification accuracy of category i according to the total samples with label i.

The F-measure is obtained from the combination of the Precision and Recall criteria and is used in cases where it is not possible to assign special importance to each of the two Precision and Recall criteria. The relation (9) shows how to calculate this measure.

$$F - measure = \frac{2*Precision_i*Recall_i}{Precision_i+Recall_i} \quad (9)$$

We use the experimental set to check and evaluate the classification methods. The criteria by which we evaluate the proposed method are: Recall, Accuracy, Precision and F-Measure [17].

Equation (10) shows the FPR criterion, which expresses the false alarm rate according to the negative category.

$$FPR=\frac{fp}{tn+fp} \quad (10)$$

Using the cuckoo algorithm, 23 features were selected as the most important features. The Src-byte feature is the most important parameter in the data set, which shows the number of bytes sent from the source to the destination. The selected features are shown in Table 3.

Table 3: The set of selected features

S.No	Feature Name	S.No	Feature Name
1	Duration	30	diff_srv_rate
3	Service	31	srv_diff_host_rate
4	Flag	32	dst_host_count
5	src_bytes	33	dst_host_srv_count
6	dst_bytes	34	dst_host_same_srv_rate
23	count	35	dst_host_diff_srv_rate
24	srv_count	36	dst_host_same_src_port_rate
25	serror_rate	37	dst_host_srv_diff_host_rate
26	srv_serror_rate	38	dst_host_serror_rate
27	rerror_rate	40	dst_host_rerror_rate
28	srv_rerror_rate	41	dst_host_srv_rerror_rate
29	same_srv_rate		

After selecting the feature mentioned above using the cuckoo algorithm, 56.12% of the features have been reduced, which significantly reduces the memory requirement. According to the number of selected features, the input matrix to the convolutional neural network is 5x5.

- Test results with different layers:

In the first experiment, the number of convolution layers was changed from 1 layer to 3 layers and the detection results were compared. The result of detection in this change is shown in figure (3). The detection result with one layer is 92.78% and when the complexity increases to 2 layers, the accuracy increases to 94.14%. When 3 convolution layers are used, it decreases to 91.86%.

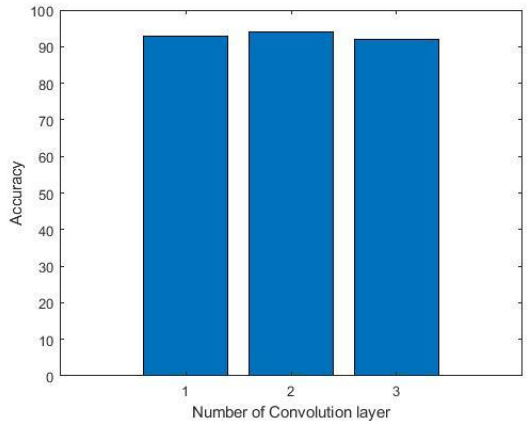

Fig. 3: Accuracy of the proposed method with different number of layers

In the second test, the amount of training repetitions is equal to 100, 1000 and 1500. The result of detecting attacks with 100 repetitions is 92.49%, 1000 repetitions is 94.14%, and 1500 repetitions is 94.28%. The best result is in 1500 repetitions of training with 94.28% percentage, which is shown in Figure (4). Therefore, if the number of training repetitions increases to 1500, the attack detection accuracy also increases.

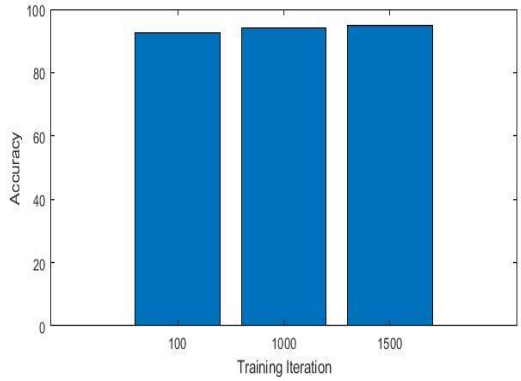

Fig. 4: Accuracy of the proposed method in different training iterations

According to the obtained results, the best result in performance is presented with 2 convolution layers and 1500 training repetitions.

In Table 4, the proposed method named CS+CNN is compared with the method presented in the article [5], which is Recurrent Neural Network (RNN).

Table 4: Evaluation of TPR, FPR criteria

FS methods		RNN	CS+CNN
Normal	TP Rate	93.14	95.91
	FP Rate	0.18	0.1
Dos	TP Rate	94.25	96.59
	FP Rate	0.08	0.05
R2L	TP Rate	94.2	95.12
	FP Rate	0.16	0.09
U2R	TP Rate	88.02	92.38
	FP Rate	0.05	0.04
Probe	TP Rate	91.27	94.09
	FP Rate	0.06	0.03
Total Accuracy		92.18	94.82

An experiment is evaluated with NSL-KDD multi-class datasets. Table 5 shows the comparative analysis of the results of RNN and CS+CNN methods, TP rate and FP rate for Dos, Probe, U2R, R2L and normal network connections and the total accuracy in the proposed research method for the set Various data has been reduced.

For the U2R attack, our proposed model achieved an accuracy of 94.82%, which is higher than the RNN method. The experimental results show that the CS+CNN method can achieve good accuracy, higher TPR and lower FPR.

- Accuracy criteria

Table 5 shows the comparison of precision and accuracy criteria and F-score. It can be seen that the proposed method has a better answer.

Table 5: Accuracy and accuracy evaluation criteria and F-measure

Evaluation	Recall	precision	F-measure
Normal	96.44	94.14	95.28
Dos	95.67	92.47	94.04
R2L	96.04	92.94	97.24

U2R	94.55	90.67	97.85
Probe	97.36	93.94	97.78

Table 6: Comparison of precision and accuracy criteria and F-measure

Algorithm	Recall	precision	F-measure
RNN	94.61	90.23	92.29
CS+CNN	96.02	92.83	94.54

Conclusion

In this article, the combination of cuckoo algorithm for feature selection and convolutional neural network is used to detect intrusion in Internet of Things network. The NSL-KDD dataset, which is the updated version of KDD CUP99, was investigated to evaluate network intrusion detection. Due to the fact that duplicate records have been removed in this collection, surely there will be no tendency towards duplicate records in those answers. In the proposed method, the classification of data set records into four attack classes and one normal class was investigated and its efficiency was evaluated. The results show that the proposed algorithm has reached a classification accuracy of 94.82%, and this indicates that the proposed algorithm has effectively trained the attacks. The combination of these two algorithms according to the proposed method resulted in high accuracy and low false alarm with an acceptable value for the network intrusion detection system. The proposed method shows better results compared to the article [5].

References

[1] SAMIRA SARVARI, NOR FAZLIDA MOHD SANI, ZURINA MOHD HANAPI, AND MOHD TAUFIK ABDULLAH, (2020), " An Efficient Anomaly Intrusion Detection Method with Feature Selection and Evolutionary Neural Network ", IEEE Access ,VOLUME 8, ,Digital Object Identifier 10.1109/ACCESS.2020.2986217.

[2] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, ``A new intrusion detection system based on fast learning network and particle swarm optimization,'' IEEE Access, vol. 6, pp. 20255_20261, 2018.

[3] Yun, M. and B. Yuxin. Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid. in Advances in Energy Engineering (ICAEE), 2010 International Conference on. 2010. IEEE.

[4] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Bose, Z. Ye, FOCUS: A Fog computing-based security system for the Internet of Things, CCNC 2018 - 2018 15th IEEE Annu. Consum. Commun. Netw. Conf. 2018-Janua (2018) 1–5. doi:10.1109/CCNC.2018.8319238.

[5] Muder Almiani, Alia AbuGhazleh, Amer Al-Rahayfeh, Saleh Atiewi and Abdul Razaque, (2019), " Deep Recurrent Neural Network for IoT Intrusion Detection System", Simulation Modelling Practice and Theory, doi: https://doi.org/10.1016/j.simpat.2019.102031.

[6] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer, (2020), " Deep Learning for Financial applications: A Survey", Preprint submitted to Applied Soft Computing, arXiv:2002.05786v1 [q-fin.ST] 9 Feb 2020.

[7] K. V. V. N. L Sai Kiran, R. N. Kamakshi Devisetty, N. Pavan Kalyan, K. Mukundini, and R. Karthi, (2020), "Building intrusion Detection System for IoT Environment using Machine Learning Techniques", Third International Conference on Computing and Network Communications, Procedia Computer Science 171 (2020) 2372–2379, DOI: 10.1016/j.procs.2020.04.257.

[8] Abdelouahid Derhab, Arwa Aldweesh, Ahmed Z. Emam and Farrukh Aslam Khan, (2020), " Intrusion Detection System for Internet of Things Based on Temporal Convolution Neural Network and Efficient Feature Engineering", Hindawi, Wireless Communications and Mobile Computing, Volume 2020, Article ID 6689134, 16 pages, https://doi.org/10.1155/2020/6689134.

[9] Fal Sadikin, Ton van Deursen, and Sandeep Kumar, (2020), " A Hybrid Zigbee IoT intrusion detection system using secure and efficient data collection", Internet of Things 12 (2020) 100306, https://doi.org/10.1016/j.iot.2020.100306.

[10] K. Mandal, M. Rajkumar, P. Ezhumalai, D. Jayakumar, and R. Yuvarani, (2020), " Improved security using machine learning for IoT intrusion detection system ", Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2020.10.187.

[11] Z. Ihsan, Yazid Idris and A.H. Abdullah, "Attribute normalization techniques and performance of intrusion classifiers: A comparative analysis", January 2013.

[12] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar,`Cuckoo search Optimization_A review,'' Mater. Today, Proc., vol. 4, no. 8, pp. 7262_7269, 2017.

[13] L. A. M. Pereira, D. Rodrigues, T. N. S. Almeida, C. C. O. Ramos, A. N. Souza, X.-S. Yang, and J. P. Papa, (2014), "A Binary Cuckoo Search and its Application for Feature Selection", Studies in Computational Intelligence, DOI: 10.1007/978-3-319-02141-6_7.

[14] Kim, J., Kim, J., Kim, H., Shim, M., & Choi, E. (2020). CNN-based network intrusion detection against denial-of-service attacks. Electronics, 9(6), 916.

[15] A. Hasan, and H. H. A, Theyazn, (2021), " Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms", Hindawi Complexity Volume 2021, Article ID 5579851, 18 pages https://doi.org/10.1155/2021/5579851.

[16] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, NRC Publications Archive (NPArC) archives des publications du CNRC (NPArC) A Detailed Analysis of the KDD CUP 99 Data Set A Detailed Analysis of the KDD CUP 99 Data Set, (2009).