Anomaly Detection using Fuzzy Q-learning Algorithm

Shahaboddin Shamshirband[1], Nor Badrul Anuar[2], Miss Laiha Mat Kiah[2], Sanjay Misra[3]

[1]Department of Computer Science, Chalous Branch, Islamic Azad University (IAU), 46615-397, Chalous, Iran, e-mail: shamshirband@um.edu.my

[2]Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia, e-mails: badrul@um.edu.my, misslaiha@um.edu.my

[3]Department of Computer Engineering, Atilim University, 06836-Incek, Ankara Turkey, e-mail: smisra@atilim.edu.tr

Abstract: Wireless networks are increasingly overwhelmed by Distributed Denial of Service (DDoS) attacks by generating flooding packets that exhaust critical computing and communication resources of a victim's mobile device within a very short period of time. This must be protected. Effective detection of DDoS attacks requires an adaptive learning classifier, with less computational complexity, and an accurate decision making to stunt such attacks. In this paper, we propose an intrusion detection system called Fuzzy Q-learning (FQL) algorithm to protect wireless nodes within the network and target nodes from DDoS attacks to identify the attack patterns and take appropriate countermeasures. The FQL algorithm was trained and tested to establish its performance by generating attacks from the NSL-KDD and CAIDA DDoS Attack datasets during the simulation experiments. Experimental results show that the proposed FQL IDS has higher accuracy of detection rate than Fuzzy Logic Controller and Q-learning algorithm alone.

Keywords: Intrusion detection; Fuzzy system; Reinforcement learning; Multi Agent System

1 Introduction

Recent advances in wireless communication and digital electronic have enabled the development of low-cost, low-power, multifunctional nodes which are small in size and which communicate with each other using radio frequencies [1]. A single node has limited capability in sensing and it is only capable of collecting data from a limited region within its range. Therefore, in order to gather useful information from an entire of Wireless Sensor Networks (WSNs), the data must be collected through the collective work of a number of sensor nodes.

The application designs for wireless sensors such as natural disaster relief [2], health monitoring [3], and hazardous events [4], afford greater flexibility in establishing communications and increase system automation, though lack in security and privacy [5]. The core weaknesses with these sensor nodes lie in the limited-resource devices, i.e. power and processing units. For this reason, vulnerability to various security threats is notably high. Meanwhile, an adversary possesses passive and active abilities. It may thus implicate sensor nodes through access to secret information such as keys stored in the compromised node in addition to the potential to eavesdrop and exhaust the sensor node resources [6]. Therefore, security is still a major design goal in WSNs.

In 2012, a report by Gartner reveals that a sophisticated class of DDoS attack sent an attack command to hundreds or even thousands of mobile agents, which then launched flooding attacks to access multiple websites [7]. Different types of DDoS attacks have been developed, which can be classified as TCP flood, UDP flood, ICMP flood, smurf, distributed reflector attack and distributed reflector attack are discussed [8]. During the distributed SYN flood attack, the compromised systems ("zombies") are led to send SYN packets with an invalid source IP address, to create an instance of a half-open connection data structure on the target server. It can be concluded that the memory stack on the victim's system is filled up and no new demands can be handled [9].

The problem of DDoS attacks has already been addressed in many studies. Fuzzy logic controller as a soft computing (SC) technique enables decision making when the values are mostly estimated or the available information is incomplete or ambiguous especially in systems that deliver tedious mathematical models [10, 11]. Fuzzy logic based detection systems are capable of calculating with availability of only ambiguous information; these systems are suitable for describing their decisions but the rules they utilize to generate decisions cannot be obtained automatically. To improve the drawbacks of unknown behavior detection, fuzzy logic combined with neural network in terms of adaptive neuro fuzzy to identify the abnormal behavior by tuning the fuzzy rules [12, 13]. The most remarkable advantages of the neuro fuzzy classifier are robustness and flexibility, but consume massive computing resources when performing fuzzy alarm correlation in large scale wireless network [14].

Reinforcement Learning [15] appears to be a greatly significant method of wireless network security due to its capability to autonomously learn new attacks via online, unsupervised learning, as well as to modify new policies without complex mathematical approaches [14]. It has been proven to be effective, especially in real time fault detection and when no prior system's behavior information is assumed. A disadvantage of reinforcement learning is the lack of memory to sustain the agent's data [16]. These limitations have been our motivation for the creating of intelligent systems where fuzzy logic systems utilized reinforcement learning algorithms to overcome the problem of memory and accuracy of detection.

To improve the accuracy of detection, Intrusion Detection System (IDS) proposed to identify the type of possible attacks [17]. Munoz et al. [18] utilized fuzzy Q-learning for congestion detection to drop packets that differs from normal features. Fuzzy Q-learning algorithms proposed by Munoz improved the accuracy of detection and consumed minimum resources, due to an increase in the high volume of traffic. The approach we have used in this study aims to design a hybrid intrusion detection system called a Fuzzy Q-Learning (FQL) to enhance the learning ability of attack detection. Our research work, fuzzy logic controller utilized fuzzy min-max strategy to provide the action selection policy. The Q-learning algorithm adjusts their parameters (i.e, state, action) based on fuzzy functions to reduce the complexity of states and action as well as speed up the decision process.

This paper will discuss how DDoS attacks launched in wireless network can be modeled through fuzzy Q-learning algorithms. The purposes of developing such models are manifold:

1) To evaluate whether resources of a given system are vulnerable to certain types of attacks.

2) To understand whether we can possibly detect DDoS, by observing fuzzy behavior of network traffic and other observable data.

3) To develop methodologies and formulate machine learning algorithms that can detect DDoS attacks in wireless network.

The remainder of this paper is organized as follows: in Section 2, we discuss related studies. In Section 3, we proposed the system model. In section 4, we describe the self-tuning scheme of the model, incorporating Fuzzy Logic Controller (FLC) with Q-learning algorithm into the IDS of a WSN. Section 5 presents simulation results. Finally, the paper concludes in Section 6.

2 Related Studies

2.1. DDoS Attack Dataset

The most significant challenge for an appraisal of a DDoS attack detection algorithm is the lack of proper public DDoS attack dataset. Since 2000, the two classes of publicly accessible datasets for IDS are Network-based IDS (NIDS) and Wireless-based IDS (WIDS) datasets.

The KDD Cup dataset was produced by processing the tcpdump portions of the 1998 DARPA Intrusion Detection System (IDS) evaluation dataset [19]. The data is not synthetic and does not reflect contemporary attacks. NSL-KDD datasets [20] were selected to mitigate the difficulties incurred by KDD'99 datasets. NSL-

KDD is significant in that it contains fewer redundant, duplicate records in the training and test phases of learning-based detection, making the evaluation process of the learning system more efficient. CAIDA dataset consists of DDoS attack dataset 2007, which can be availed by user's request. CAIDA DDoS attack dataset [21] consist of an hour of anonymized traffic traces from a DDoS attack.

2.2. Real Time Feature Extraction

Online feature extraction methods based on per flow analysis are expensive, not scalable, and thus prohibitive for large scale networks. An increase in the number of features led to better accuracy but computation of a larger number of features in real time causes more overhead and time consumption. As a result, fewer feature selection is suitable for better pattern classification in real time.

The detailed analysis on DDoS attacks, available attack tools and defense mechanisms [22] indicate that the DDoS attack has the following features.

- Source and Destination IP address and port numbers of the packets are spoofed.
- Window size, sequence number, and packet length are fixed during the attacks.
- Flags in the TCP and UDP protocols are manipulated.
- Roundtrip time is measured from the server response.
- Routing table of host or gateway is changed.
- DNS transaction IDs (reply packet) are flooded.
- HTTP requests are flooded through port 80.

Our objective is to differentiate the DDoS attack and normal traffic. In this research work, the 'duration' feature or response time has been used to identify the incomplete length time of the connection due to handshake. Most of the attacks target the victims' servers through legitimate ports such as 80, 53, 443, etc. Hence, the 'Protocol_type' feature from clients over a time window was used to monitor the legitimate port. DDoS attacks send flooding packets to victims in order to consume the resources such as memory and CPU. The "Src_bytes and Dst_bytes" features used, in terms of 'buffer size or packet size ', to identify the number of data bytes from source to destination and destination to source. The number of connections to the same host is the key features of DDoS attack. The 'Count' feature is used to monitor the number of connections to the same host during specified time window.

In our data selection method, the NSL-KDD dataset combined with CADIA dataset in order to create a new set of attack dataset that reveals the characteristics of DDoS attack. By processing the continuous flow of the packets which propagates from mixed dataset, key characteristics of network activity can be

achieved between hosts. From the dataset attributes, five features, as shown in Table 1, have been selected for accurate detection of DDoS attacks. These attributes mostly consist of spoofed source address and contain half-open connections.

Table 1
List of features of the DDoS attacks

Feature name	Feature Description
Time response	Variance of time difference between two connections during specific time window
Protocol_type	Type of the protocol, e.g., TCP, UDP, etc.
Src_bytes	number of data bytes from source to destination
Dst_bytes	number of data bytes from destination to source
Count	number of connections to the same host during specified time window

2.3. Fuzzy Q-learning Detection - Motivation

To detect the type of attack a node may face in the future; this research optimized the Fuzzy Logic Controller (FLC) by Q-learning algorithm to enhance the self-learning ability of the detector agent. The fuzzification process converts the variables $x \in X$, where X is the set of possible input variables to fuzzy linguistic variables by applying the corresponding membership functions. The Inference Engine (IE) maps input and output fuzzy sets to Q-value. The Q-value and its eligibility updates by fuzzy rules. Defuzzification computes a crisp value to adopt an action in terms of the action policy. Such an adaptation of Q-Learning allows to process continuous state and action spaces by a simple discretization of the action-value policy. Figure 1 demonstrates the proposed Fuzzy Q-learning (FQL) architecture.

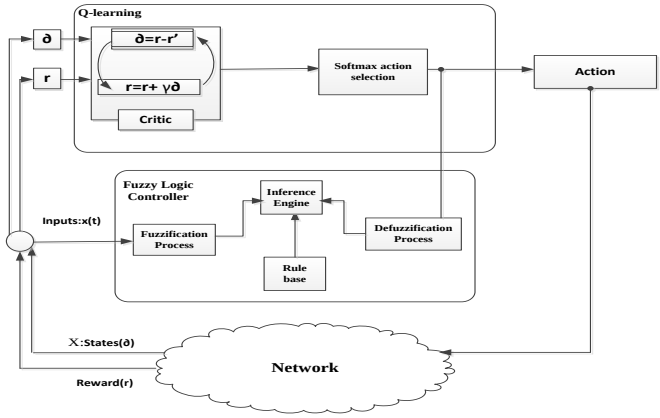


Figure 1
Block diagram of the Fuzzy Q-learning architecture

FQL-based detection instead of adding a new input to the FLC, the learning algorithm enables to control this performance indicator by the 'trial-and-error' methodology to avoid complex rules. The disadvantage of the FQL is that of the operator as fuzzy rules is fixed. For instance, IDS in a very congested network should be conservative as detection receives much more anomaly. Another special situation is when there is a small overlap between normal and abnormal states; the FQL produces an extreme change in the IDS margin that significantly increases the detection rate. In all cases the risk of resource consumption is higher when modifying FLC margins.

In our scheme, we modified the FQL algorithm by applying min-max action selection instead of ϵ-greedy action-selection and softmax action selection rule. The main drawback of ϵ-greedy action-selection is that when is explores it selects equally among all actions. The worst performing actions may be happened. To solve this problem, softmax method uses a Boltzmann distribution. The greedy action is given the highest selection probability according to their value estimates. The adjusting parameters of action selection methods must be set manually that decreases the speed of algorithm in training. To solve the problem of manually adjusting the action selection parameters, decrease the false alarm rate and increase the accuracy of attack detection, we used dynamic fuzzy min-max action selection method to improve the performance of algorithm.

2.4. Utility Function

To appraise the efficacy of the associations determined by the FQL and to determine the applicability of the rule at every point in time, Eq. 1 was utilized in this work, as suggested by Huang et al. [23]. In Table 2 the parameters of the utility function are described:

$$U = \rho * SP - \beta * FN - \theta * FP \tag{1}$$

Table 2
Utility function parameters

Parameters	Explanation
U	Is a utility
ρ	Symbolizes the weight of effective prediction, $q = 0.75$
SP	Characterizes the true confidence rate of attack patterns.
β	Signifies the weight of failed estimates (attack but no defense), b = 1
FN	Represents false negative of attack patterns - there are attacks but no defense
θ	Denotes the weight of failed predictions (defense but no attack), $h = 1$
FP	Represents false positive of attack patterns - there is defense but no attack

The fuzzy Q-learning principle approach entails detection accuracy with low time complexity, which only afterward begins to formulate a shield policy. The major drawback of the FQL theory is that if attacks are recurring over a short period, a

considerable amount of time is consumed in the detection phase, something that weakens the defense. It can be said that the detection precision may be low while the false alert rate is high. This problem is a worst-case scenario but can be addressed using the modified FQL proposed by [14]. Its principal contribution is identifying the probability of future attacks aimed at a wireless sensor node. For frequent attacks occurring over a short time, multi agent-based FQL was adopted to deal with the excessive time spent on detection. The aim of the proposed FQL is to obtain high detection accuracy with a low false alarm rate.

3 Proposed Model

3.1. WSN Model

In the present research study, Figure 2 illustrates the network model with hierarchical routing, which consists of clusters (C), their coordinators, or Cluster Heads (CHs), as well as the member sensor nodes (S). In the current scheme, the Cluster Head (CH) is assumed to be a Sink Node (SN) in a cluster. The SN monitors the behavior of sensor nodes by collecting data from the member sensor nodes and transmitting the critical status - the attack information of the sensor nodes, to a Base Station (BS).

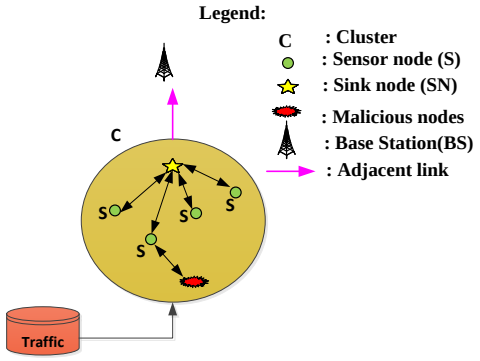


Figure 2
A network system perspective of a WSN

The route from a sensor node to a BS is deemed a hierarchical path that creates a hierarchical system with numerous routes, which is the main feature of cluster-based WSNs [24]. Figure 2 illustrates how sensor nodes send collected data from a sink node to a BS via other adjacent sink nodes, and the BS receives data only if SN within the routing formation are actively functioning. Attacks in this scenario can target the WSN in multiple ways, with DDoS attacks potentially originating either from the Internet or neighboring wireless sensor sources.

3.2. Methodologies and Techniques Used

The fuzzy Q-learning-based detection and defense mechanism operate to detect DDoS attacks, where the sink node and base station adapt to select the best strategy of detecting an immediate attack and respond to it. Regardless of whether the attacks are carried out on a regular or irregular basis, the IDS can adjust its learning parameters through fuzzy Q-learning to identify future attacks. Figure 3 depicts the proposed architecture of Fuzzy Q-learning Detection System (FQL).

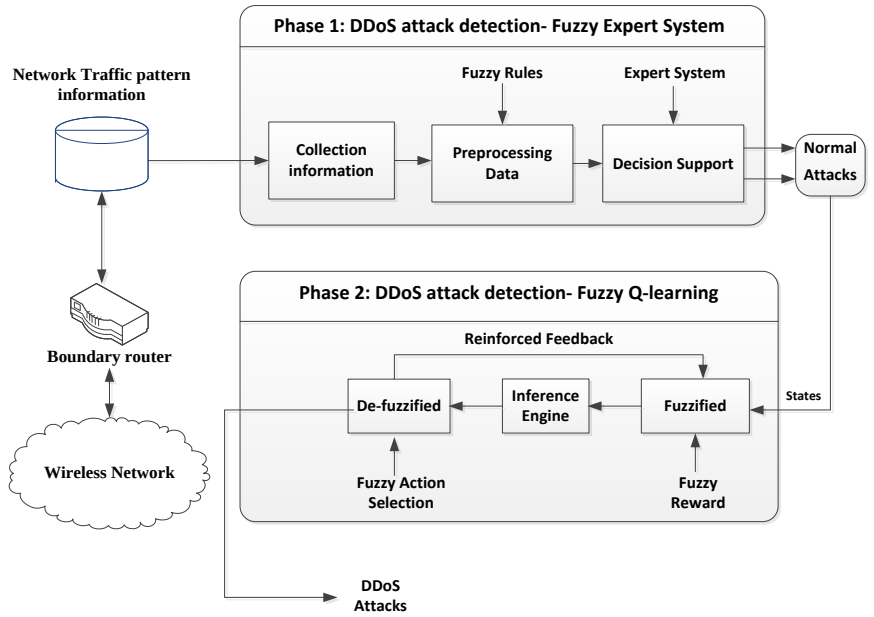


Figure 3

Architecture of Fuzzy Q-learning based Wireless Intrusion Detection System(FQL-WIDS)

In the first phase of proposed FQL architecture, Fuzzy Expert System (FES) concentrates to audit the attack records received from the traffic. When FES detects the possible attacks then send the new set of traffic dataset to the next layer. In the second phase, the FLC optimized by Q-learning to discover and detect the security treats captured by FES. The architecture of the proposed FQL-IDS is dual, that is, it has two phases (Figure 3).

- **Expert Policy:** It uses Expert System (ES) to decrease the state space for sink node due to increase in look-up table or Q-table. This policy broadcast the gain of ES engine (i.e. Abnormal, normal) through Base Station (BS).
- **Fuzzy Q-learning (FQL) policy:** It adopts to mitigate the possible faults escaped from Expert System policy. This learning policy identifies the anomalous data by optimizing Fuzzy Logic Controller based Q-learning.

4 Self-Tuning Scheme

In this section, we describe the FQL-based WIDS setup; define a primary detection based on an fuzzy expert system, express the FLC utilized Q-learning. We demonstrate how to optimize FLC based on a Q-learning algorithm. The process discovers six attributes: 1) Protocol type: ES chooses only TCP connection; 2) Source and destination IP: ES selects the IP range of acceptable by default; 3) Source and destination port: IDS pick out the corresponding port. 4) Time response: It deals with the time duration of response between sensor nodes. 5) Buffer size: It relies on the size of the buffer on processor storage. 6) Count: the number of connections to the same host at current connection in a past two second.

4.1. Fuzzy Expert System for DDoS Attack Detection

To fully exploit the suspicious level at the first phase, Expert System (ES) utilized Fuzzy Rules Base (FRB) to identify the anomaly conditions received from the traffic. The Fuzzy Expert System (FES) employed to decrease the record of anomalous data through fuzzy logic controller. We designed FES consists of the following components: the traffic capture, the feature extractor, the fuzzification, the fuzzy inference engine, the knowledge base, the defuzzification, and the expert analyzer. Figure 4 shows the details of component of the proposed Fuzzy Expert detection system architecture.

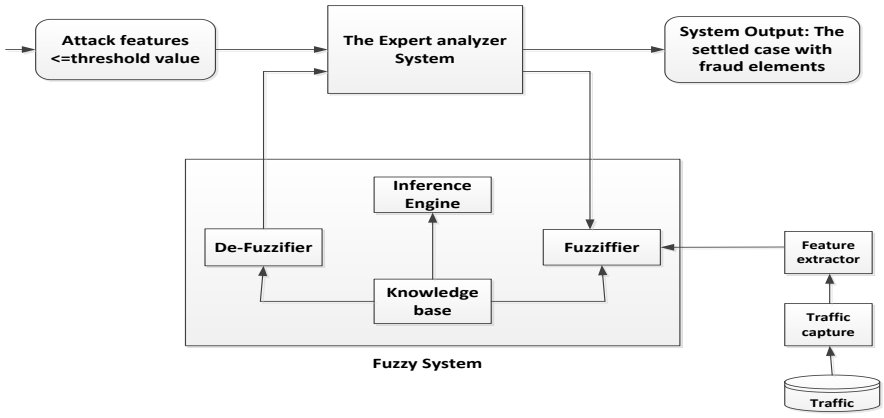


Figure 4
The architecture of the proposed fuzzy expert system

4.1.1. The Traffic Capture

The traffic capture component collects the traffic records and prepared the base information for traffic analysis. Currently the traffic capture is based on the popular network and hosts' intrusion detection tools and other scanning tools:

Snort, Sniffer, and wireshark. These different forensic tools real-time collect network attack traffics and intrusion host's information. In this research work, we utilized wireshark packet analyzer tools to pre-processed DDoS attacks data and their features.

4.1.2. The Feature Extractor

The feature extractor performs extracting features on the "network traffic" captured by the traffic capture component. Under the network and system environment, there are many traffic features that can be used for attack detection and analysis.

Definition 1:

Attack data source: The attack data source can be defined as a 5-tuple ADS={*Pt*,*Dp*,*Tr*,*Bs*,*Co*} according to the vulnerability scanning information, where *Pt* denotes as the type of protocol (TCP=1, UDP=2); *Dp* denotes as the destination port; *Tr* denotes as the variance of time difference between two connections during specific time window, Bs denotes as the length of packet from source to destination, *Co* denotes as the number of connections to the same host as the current connection in the past two seconds. Table 2 denotes as the major forensic parameters of DDoS attack data source.

4.1.3. The Fuzzification

Each input variable's sharp (crisp) value needs to be first fuzzified into linguistic values before the fuzzy decision processes with the rule base. The characteristic function of a fuzzy set is assigned to values between 0 and 1, which denotes the degree of membership of an element in a given set. Table 3 displays the linguistic terms and their fuzzy numbers used for evaluating the attack data source for time response, buffer size, and Count. Figure 5 indicates the membership functions for time response.

Table 3
Fuzzy rating for occurrence of attack traffic in ADS

Linguistic variables	**Fuzzy number**		
	Tr	**Bs**	**Co**
Low (L)	(-inf,-inf,0,40)	(-inf,0,2,3)	(-inf,0,1,1.5)
Medium (M)	(20,40,80,100)	(2,3,5,6)	(1,1.5,2,2.5)
High (H)	(80,120,inf,inf)	(5,6,8,inf)	(2,2.5,3,inf)

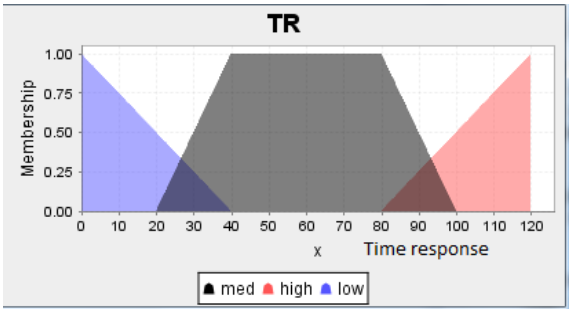


Figure 5

The membership functions of linguistic variables for attack data source Tr

4.1.4. The Fuzzy Inference Engine and Knowledge Base

Knowledge base stores the fuzzy rules which are used by the fuzzy inference engine to get a new fact from. The pseudo code of proposed FES is shown in Table 4 in parallel of Figure 6.

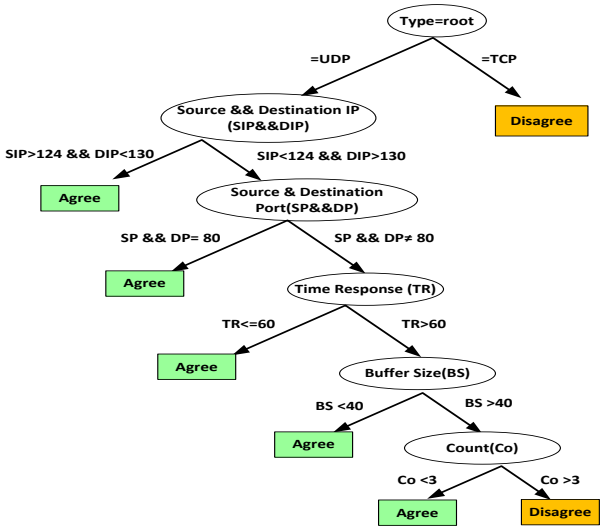


Figure 6

The state of decision fuzzy expert system to reach the goal

4.1.5. The Expert Analyzer

The expert analyzer decides the influence result from defuzzification whether inspected packets are attacks or not. If the crisp value is disparate than threshold value of the detection attack rule, then it adopts that an attack has occurred. The process of manually extracting rules may be time consuming and the rules may be approximate. Because these methods are off-line in nature, if a very large set of

data is involved, it can become expensive and impractical, and can not real-time detect the novel attacks. In order to overcome this problem, we propose a hybrid soft computing methods to identify DDoS attacks.

4.2. Fuzzy Q-Learning Algorithm for Anomaly Detection

To mitigate the learning time process FLC is optimized by Q-learning algorithm developed in. In this section, we optimized the FLC for anomaly detection by using the Q-Learning algorithm. Three fuzzy sets have been defined for the input of FQL to represent three different situations as a state space of Q-learning: These inputs are named as TBC. Time response deals with the time duration of response between sensor nodes, the Buffer Size relies on the size of the buffer for processor storage in sensor node by sending a huge number of fake messages. Count is the number of connections to the same host at current connection in a past two second. Figure 7 demonstrates block diagram of the optimization scheme for anomaly based–FQL.

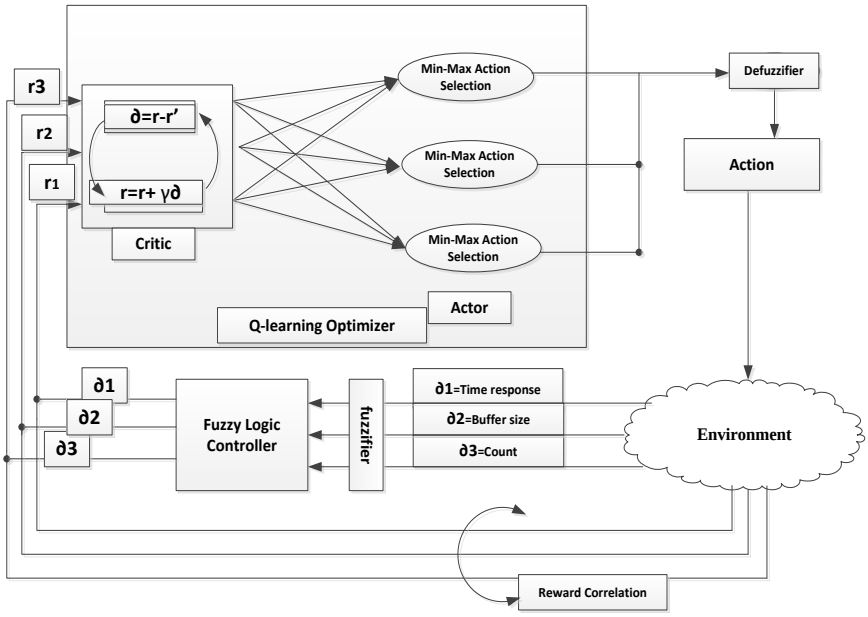
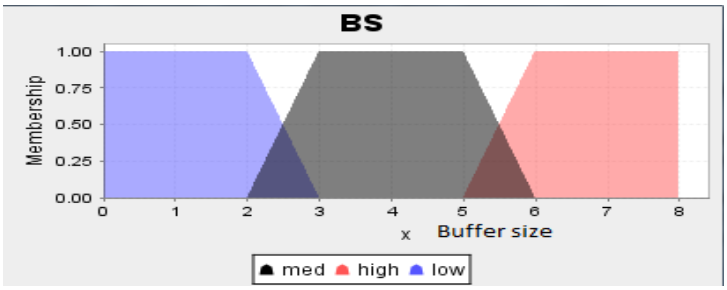


Figure 7
Block diagram of the optimization scheme for FQL

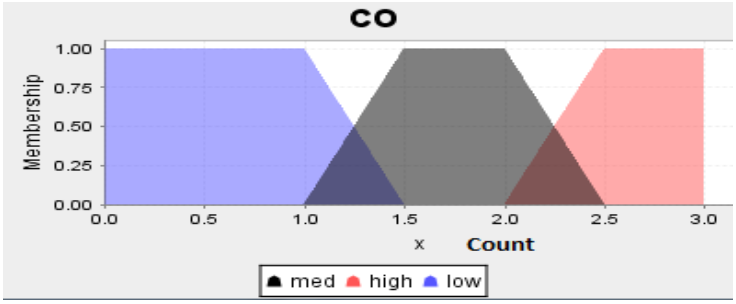
The FLC output, given by the Time response (Tr), Buffer size (Bs) and Count (Co), correspond to the fuzzy state of the network S (t),

- $S(t) = [Time\ response, Buffer\ size, Count] = [Tr, Bs, Co]$ (2)

The FLC output, given by the increment in the states, represents the action of the sink node, A(t). The reward signal, R (t), is built from the FLC, is measured in both modes of the adjacency in order to test if the sensors are experiencing attacks. The linguistic variables of Time response (Tr), Buffer size (Bs), and Count (Co) act as inputs and the Detect Confidence (DC) acts as an output are used in the experiments. Figure. 8 (a, and b) indicates the membership function for the input and Figure. 9 indicates the output variable of fuzzy systems.



a) Input membership function



b) Input membership function

Figure 8

(a), and (b), Input Membership function design in Java-fuzzy toolbox [25]

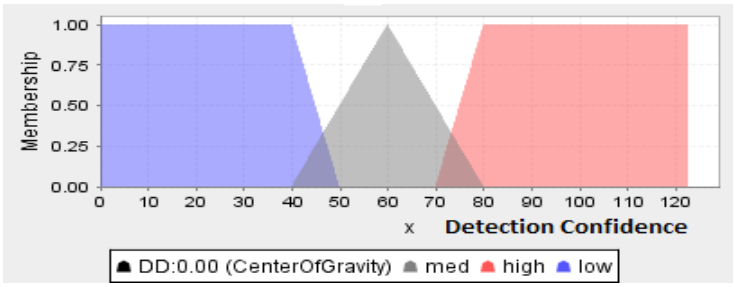


Figure 9

Output Membership function

Three fuzzy sets are identified in the current Buffer size (Bs), whose linguistic terms are 'Low' (L), 'Medium' (M), and 'High' (H). These three fuzzy sets discriminate the cases when Bs is less than (3k), which has been defined the length of packet received from source during specified time window. The output linguistic variable represents the system's Detect Confidence (DC) in the presence of abnormal behavior. To illustrate, if the confidence value is higher than 80, then the system is more than 80% certain that there is an abnormal entity, if the detection confidence is smaller than 40, it is more likely that there is no abnormality. However, input and output variables give us a notion of how traffic connection is changing. Figure. 8 (a, and b) indicates the membership function for the input variables of buffer size and count and Figure. 9 indicates the output variable of fuzzy systems. The membership functions are triangular or trapezoidal.

The number of selected rules in this section is smaller, resulting in a lower number of fuzzy rules. A small number of rules speed up the convergence of the Q-Learning algorithm since fewer states have to be visited during the exploration phase. The interpretation of each rule defined in this work is described as follows:

- **Rule 1**: It is activated when there is a high value in Tr and the Bs margin has a 'high' value and the number of connections to the same host has a high percentage, which is opposite to the desired value. A large increment in the linguistic variables should be necessary in this case to increase the grade of detection of anomalies. Thus, the consequent of rule 1 is set to 'high'.
- **Rule 2**: It is similar to rule 1 but with the difference that the Tr has a low value. The consequences of that rule should be a moderate change such as 'High'.
- **Rule 3**: The activation of rule 3 occurs when there is a medium Tr difference in the adjacency from the source to the destination but the Bs and Count margins have an appropriate ('low') value to monitor the traffic. For those reasons, the consequent for rule 3 is set to 'medium'.
- **Rule 4**: It is activated when the Tr has medium and the Bs margin has a 'high' value and Count is high. The selected consequent for rule 4 has been set to 'high'.

Each state defines by Time response, Buffer size and Count (TBC). The valuable range of TBC adopts the fuzzy membership function to represent the function of Q-learning.

In order to find the optimal action, the reinforcement signal r (t) used Eq.(2). FQL agent assigns a weight to all possible next states based on FLC. Associated to the threshold value, the optimal cost may be achieved. Thus, those FLC actions that lead to a Detect Confidence (DC) less than DCth should be rewarded with a positive value, while those actions producing a DC higher than DCth should be punished with a negative value. Formally, the reinforcement signal used in this work is defined by:

- $r(t+1) = \begin{cases} 100 \text{ , if } DC^{k}_{meatured}(t) < DC_{th} \\ -100 \text{ , otherwise} \end{cases}$ (2)

Where $r(t+1)$is the reinforcement signal for the K^{th} sink node in iteration t+1. The value of $DC^{k}_{meatured}(t)$ is calculated as the min-max weighted average:

$$\text{Detect Confidence} = \text{output}(C_j) = \left(\sum_{j=1}^{N} \alpha_j \; c_j\right) / \left(\sum_{j=1}^{N} \alpha_j\right) \quad (3)$$

$$\alpha_j = \left[\mu_j(x_0) * \mu_j(y_0)\right] \quad (4),$$

Where N is the number of rules, α_j is the degree of truth for the rule j and c_j is the selected output constant value for the same rule. The sufficient rules generate by look up table and shows in the Table 5.

Table 5
Fuzzy rules provide by a lookup table

Rule1: IF Tr = high AND Bs = high AND Co =High THEN output = Abnormal
Rule2: IF Tr = high AND Bs = low AND Co =med THEN output = Abnormal
Rule3: IF Tr = low AND Bs = low AND Co =low THEN output = Normal
Rule4: IF Tr = low AND Bs = high AND Co =low THEN output = Normal

These rules are typical of control applications in that the antecedents consist of the logical combination of the time response, buffer size and count signals, while the consequent is a control pattern output. The rule outputs can be defuzzified using a discrete centroid computation based on Eq. (3). Table 6 demonstrates the results of applying one of possible action selection for FQ- Learning algorithm.

Table 6
Possible action selection by FQL

Input variables state i			Input variables state j			Output desirable	Action(Min-Max)
Tr	Bs	Count	Tr	Bs	Count	Pattern	
Low (0. 2)	High (0. 8)	Low (0. 2)	High (0. 8)	High (0. 8)	High (0. 8)	Abnormal (0. 8)	Min Sj (0.8,0.8, 0.8) =0.8
			High (0. 8)	Low (0. 2)	High (0. 8)	Abnormal (0. 8)	Min Sj (0.8, 0.2, 0.8) =0.2
			Low (0. 2)	Low (0. 2)	Low (0. 4)	Normal (0. 2)	Min Sj (0.2, 0.2, 0.4) =0.2
			Low (0. 2)	High (0. 8)	Low (0. 4)	Normal (0. 2)	Min Sj (0.2, 0.8, 0.4) =0.2

Abnormal: Max (0.8,0.2) =0.8
Normal: Max (0.2, 0.2) =0.2
Threshold
$0.8 > 0.2 \rightarrow$ Abnormal > normal

Example

Consider the FQL is at start state ($\partial 1$) with the degree of membership function for parameter of TR to Low is 0.2, the degree of membership of BS to high 0.8 and the value of count is 0.2, so it is going to move to the goal state Sj (i. e $\partial 9$ in state diagram) with Tr=high, Bs=high and Count=high. It allocates a weighed fuzzy label for all next states by using fuzzy max min. Finally the simple threshold which compares the consequent is used to choose the best action.

5 Experimental Results

Three sets of experiments were conducted to examine the effects of attack detection accuracy based on Fuzzy Logic Controller (D1), Q-learning algorithm (D2), and Fuzzy Q-learning (D3). D3 is derived by taking D1+D2 functionality to produce a sophisticated attack detection algorithm. Table 7 shows the comparison of the proposed ensemble FQL detection algorithm versus the other existing standalone algorithms.

Table 7
Comparison of existing ensemble algorithms with proposed algorithm

Algorithm / Features	Fuzzy Logic Controller (D1)	Q-learning (D2)	Fuzzy Q-learning (D3)
Prior knowledge of data distribution	Required	Not Required	Not Required
Method used to combine classifiers	Fuzzy Classifiers	Markov Decision Process	Fuzzy rule base and Q-learning
Drawbacks	Work for small subset	Sensitive to noise and outliers, High cost consumption,	Limited by one classifier, The low speed of detection, Fail to high volume of traffic
Advantages	Simple to implement with good performance	Capable of handling multi-class attack detection	Prior knowledge of data distribution no needed,

We used FLC, which utilized min-max fuzzy method for improving classification scheme. If the new sets of fuzzy rules agree on the same class, that class is the final classification decision. If the fuzzy classifiers disagree, then class chosen by the second sets of fuzzy rules classifier is the final decision. The min-max fuzzy

classifiers show the good performance in reduced dataset, but inaccurate by increasing the high volume of traffics that fuzzy IDS may be crashed. In addition, prior knowledge of data distribution is required for fuzzy IDS algorithm. We also, modified the Q-learning algorithm to identify the DDoS attacks. The Q-learning based DDoS attack detection is capable of handling the minor class of DDoS attacks detection, but the multi objective procedure or major features of DDoS attack consumes maximum resources, especially in real time environment. In addition, the convergence of Q-learning takes much time. In Q-learning algorithm the observation is limited by one single classifier. Therefore, this algorithm fails due to high volume of real time traffic. To overcome the problem of accuracy of detection, false alarm rate and time complexity, we combine Q-learning algorithm with fuzzy logic controller to reach high accuracy of detection and low false alarm rate, especially in real time traffic.

Three investigates were carried out on publicly available datasets such as NSL-KDD dataset and CAIDA DDoS dataset, and mixed dataset using the Castalia and the results are discussed in Section 5.1, 5.2, and 5.3.

5.1. Performance Verification

Our proposed classification algorithm with cost per sample function is compared with existing soft computing methods D1, D2, in terms of accuracy of detection per sample on three dataset NSL-KDD, CAIDA, and mixed dataset of attacks. Comparing the false positive rate of FQL with cost minimization, it can be seen that FQL algorithm with cost minimization yields an improvement of 20% $\left(\frac{3.50-2.80}{3.50} * 100\right)$ over Q-Learning algorithm as shown in Table 8. Moreover, it can be inferred from Figure.10 that cost per percentage of samples or anomalous is less for FQL algorithm than the other methods.

The proposed Fuzzy Q-learning (FQL) algorithm with the cost function $U = \rho * SP - \beta * FN - \theta * FP$ was compared with existing soft computing methods (Fuzzy Logic Controller, and Q-learning) with respect to the attack detection precision of modeled Denial-of-Service attacks on three dataset NSL-KDD, CAIDA, and mixed dataset. A comparison between the average utility function and FQL with cost maximization indicates that the latter yielded an improvement of 20% $\left(\frac{3.50-2.80}{3.50} * 100\right)$ over Q-Learning algorithm as shown in Table 8. Moreover, it can be inferred from Figure 10 that cost per percentage of samples or anomalous is less for FQL algorithm than the other methods.

Table 8
Simulation result of detection algorithm for NSL-KDD dataset

Percentage of anomalous	FLC			Q-learning			FQL		
	True Positive (%)	False positive (%)	**UF**	True Positive (%)	False positive (%)	**UF**	True Positive (%)	False positive (%)	**UF**
1	70.20	1.80	50.85	75.20	1.40	55.00	80.10	1.20	58.88
5	71.50	2.20	51.43	76.70	1.60	55.93	81.20	1.40	59.50
10	73.20	2.80	52.10	76.90	1.90	55.78	82.50	1.90	59.98
15	75.40	3.20	53.35	77.60	2.10	56.10	83.70	2.10	60.68
20	75.90	3.70	53.23	78.50	2.40	56.48	83.90	2.40	60.53
25	76.10	4.10	52.98	79.80	3.10	56.75	84.20	2.60	60.55
30	77.10	4.60	53.23	80.10	3.40	56.68	85.80	2.80	61.55
35	80.10	4.90	55.18	82.30	3.90	57.83	86.40	2.90	61.90
40	81.90	5.10	56.33	83.60	4.20	58.50	87.70	3.20	62.58
45	78.20	5.30	53.35	79.80	4.90	54.95	88.50	3.40	62.98
50	76.80	5.90	51.70	78.90	5.20	53.98	89.60	3.90	63.30
55	75.60	6.10	50.60	79.30	5.60	53.88	90.40	4.10	63.70
60	74.80	6.20	49.90	80.00	5.80	54.20	92.40	4.50	64.80
Average	**75.91**	**5.77**	**52.63**	**79.13**	**3.5**	**55.85**	**85.88**	**2.8**	**61.61**

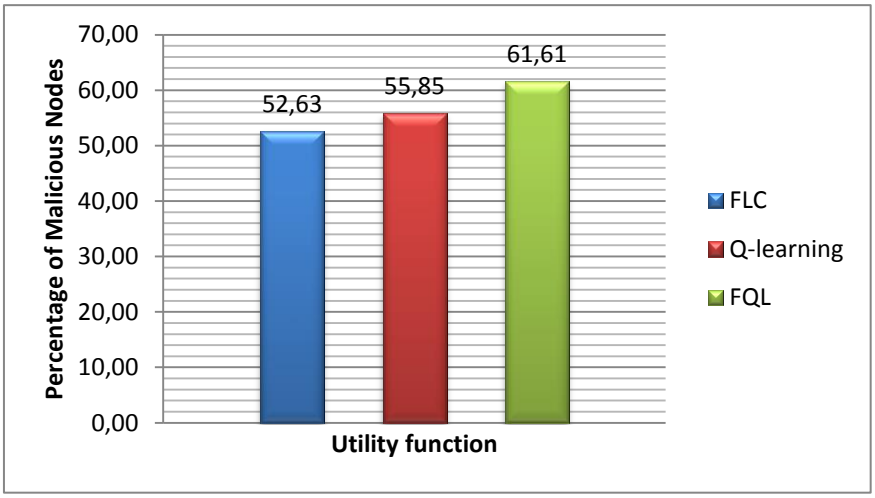


Figure 10
Cost per sample for existing DDoS detection and FQL from NSL-KDD attack source

The next tryout was conducted for CAIDA traffic. From Table 9, it can be seen that the detection accuracy is 78.59% with 5.79% false positive rate. Comparing the false positive rate, FQL algorithm with minimum cost function, it can be seen that FQL yields an improvement over Q-learning, and FLC. It can be inferred from Figure 11 that cost per samples is less for FQL algorithm than other methods.

Table 9
Simulation result of detection algorithm for CAIDA dataset

Percentage of	FLC			Q-learning			FQL		
	True Positive (%)	False positive (%)	UF	True Positive (%)	False positive (%)	UF	True Positive (%)	False positive (%)	UF
1	73.50	1.80	53.33	74.20	1.40	54.25	82.70	1.20	61.73
5	73.90	2.20	53.23	74.70	1.60	54.43	84.70	1.34	63.19
10	74.20	2.80	52.85	75.40	1.90	54.65	85.40	1.50	63.68
15	74.80	3.20	52.90	75.90	2.10	54.83	85.90	2.10	63.90
20	75.10	3.70	52.63	78.60	2.40	56.55	86.60	2.30	64.38
25	75.40	4.10	52.45	78.80	3.10	56.00	87.70	2.80	65.08
30	75.60	4.60	52.10	79.40	3.40	56.15	88.80	3.60	65.70
35	76.10	4.90	52.18	79.90	3.90	56.03	89.40	3.90	66.08
40	76.80	5.10	52.50	80.40	4.20	56.10	90.70	4.50	66.90
45	79.20	5.30	54.10	81.80	4.90	56.45	91.40	4.60	67.40
50	79.50	5.90	53.73	82.80	5.20	56.90	92.50	4.70	68.20
55	80.30	6.10	54.13	83.70	5.60	57.18	93.80	4.90	69.13
60	80.60	6.20	54.25	84.50	5.80	57.58	94.40	5.00	69.55
Average	76.54	6.43	53.10	79.24	5.85	55.93	78.59	5.79	65.76

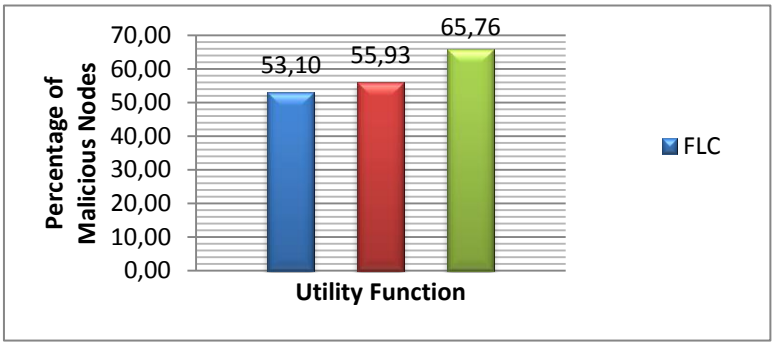


Figure 11
Cost per sample for existing DDoS detection and FQL from CAIDA attack source

Experiment 3 was performed in mixed dataset. From Table 10, it is evidence that the trained FQL algorithm was able to detect DDoS attack with high accuracy of detection, low false positive rate and minimum cost function. Figure 12 indicated the cost per sample of FQL.

Table 10
Simulation result of detection algorithm for Mixed dataset

Percentage of	FLC			Q-learning			FQL		
	True Positive (%)	False positive (%)	UF	True Positive (%)	False positive (%)	UF	True Positive (%)	False positive (%)	UF
1	78.20	1.80	56.85	78.70	1.34	57.69	82.70	1.00	61.03
5	78.60	2.40	56.55	79.20	1.47	57.93	84.70	1.10	62.43
10	79.00	2.80	56.45	79.40	2.10	57.45	85.40	1.30	62.75
15	79.30	3.50	55.98	79.80	2.80	57.05	85.90	1.80	62.63
20	80.10	3.90	56.18	80.10	2.40	57.68	86.60	2.10	62.85
25	80.60	4.30	56.15	80.50	2.92	57.46	87.70	2.40	63.38
30	81.30	4.60	56.38	81.30	3.40	57.58	88.80	2.70	63.90
35	82.10	4.90	56.68	81.80	3.90	57.45	89.40	2.90	64.15
40	82.50	5.10	56.78	82.70	4.20	57.83	90.70	3.00	65.03
45	83.10	5.30	57.03	83.90	4.90	58.03	91.40	3.30	65.25
50	83.40	5.90	56.65	84.60	5.20	58.25	92.50	3.60	65.78
55	84.10	6.10	56.98	85.90	5.60	58.83	93.80	3.70	66.65
60	84.20	6.20	56.95	86.60	5.80	59.15	94.40	3.90	66.90
Average			56.58			57.87			64.05

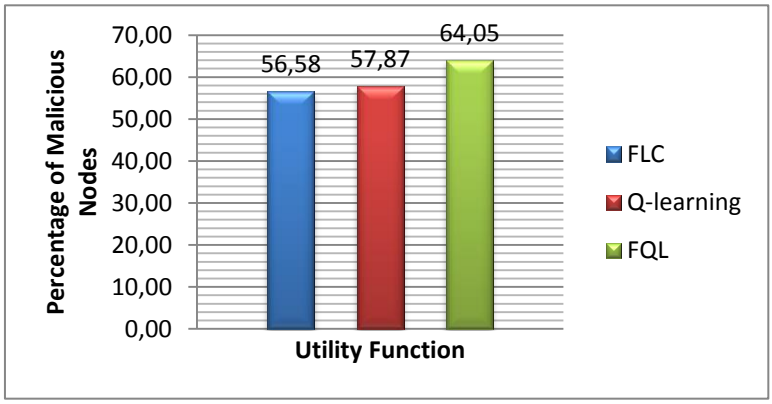


Figure 13
Cost per sample for existing DDoS detection and FQL from Mixed attack source

5.2. Computational Time of FQL Algorithm

Preprocessing time includes the time spent in feature extraction and normalization. The training time depends on the number of times the classifier needs training which in turn depends on the mean square error between iterations reaching goal minimum. Testing time includes the time spent in testing the unlabeled instances by weighted mean. Table 11 shows the performance comparison of the FQL in terms of consuming time obtained during the experiments. From Table 11, it can be realized that the training time of FQL is higher to QL, but it consumes more testing time than the FLC, Q-learning. Also, the computational time was calculated on Intel 3.10 GHz, Core i-5 Processor, 4 GB RAM computer.

Table 11
Performance comparison of Co-FQL with existing machine learning methods in terms of consuming time

Dataset	Algorithms	Training time (seconds)	Testing time (seconds)
Mixed Dataset	Fuzzy Logic Controller (D1)	3.10	1.30
	Q-learning (D2)	3.14	1.36
	Fuzzy Q-learning (D3)	3.22	1.40

Testing time of the proposed FQL method is a little high due to the ensemble output combination methods such as fuzzy logic controller with Q-learning algorithm, but more detection accuracy was achieved in FQL. The speedup of FQL can be improved when a hybrid classifier is executed in parallel processors. Thus, all the modules can be processed in parallel by different engines in order to reduce the overall processing time considerably.

Conclusions and Future Research

Development of the machine learning algorithmic technique for online IDS by modifying Fuzzy Q-learning mechanism detects DDoS attack with 85.88% accuracy, which is far superior to Fuzzy Logic Controller, and Q-learning algorithm by themselves. Reducing complexity and dimensionality of the selected feature set is learnt to reach to the goal state. In our research work discretization, feature selection and accuracy calculation are handled simultaneously, which reduces computational cost and build the detection in a comprehensive way. It has been observed that for detection of continuous attack attribute by fuzzy Q-learning, if different parameters are applied to all attributes, classification accuracy yields best result. The proposed method is tested with differently correlated data sets such as NSL-KDD, CAIDA, and Mixed datasets, showing effectiveness of the system in real time intrusion detection environment. It has been observed that the proposed method achieves higher classification by 88.77% accuracy and minimum cost function by 65.76% in CAIDA dataset compared to other existing detection methods (i.e., fuzzy logic controller, and Q-learning,) applied in the wireless networks.

Given the huge types and amounts of DDoS attacks, their optimum classification is very important for rapid detection, in which other performance indicators such as processing rate, energy consumption rate and accuracy of response would be needed to estimate the quality of the IDS. Novel detection of attacks is an important research area in security domain and has immense importance for IDPS. The characteristics of attacks changing with time and space and so handling of such attacks by using existing knowledge opens new avenue of research. Designing of classifiers using different approaches and then fusing those classifiers surely improve classification accuracy in IDPS. However, its deployment in real life operational environment is a huge challenge that still needs to be further researched.

Acknowledgement

This work is supported by the University of Malaya, Malaysia, under Research Grant RG108-12ICT.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless Sensor Networks: a Survey, J. Comput. Netw, 38 (2002) 393-422

[2] Y. E. Aslan, I. Korpeoglu, Ö. Ulusoy, A Framework for Use of Wireless Sensor Networks in Forest Fire Detection and Monitoring, Computers, Environment and Urban Systems (2012)

[3] J. M. L. P. Caldeira, J. J. P. C. Rodrigues, P. Lorenz, Toward Ubiquitous Mobility Solutions for Body Sensor Networks on Healthcare, Communications Magazine, IEEE, 50 (2012) 108-115

[4] A. Bonastre, J.V. Capella, R. Ors, M. Peris, In-line Monitoring of Chemical-Analysis Processes Using Wireless Sensor Networks, TrAC Trends in Analytical Chemistry, 34 (2012) 111-125

[5] N. Li, N. Zhang, S. K. Das, B. Thuraisingham, Privacy Preservation in Wireless Sensor Networks: A State-of-The-Art Survey, Ad Hoc Networks, 7 (2009) 1501-1514

[6] P. Schaffer, K. Farkas, Á. Horváth, T. Holczer, L. Buttyán, Secure and Reliable Clustering in Wireless Sensor Networks: A Critical Survey, Computer Networks, 56 (2012) 2726-2741

[7] DDOS Attacks against U.S. Banks Continue – LINKAGES Explored, Available from [www.gartner.com/], (2012)

[8] A. D. Wood, J. A. Stankovic, Denial of Service in Sensor Networks, Computer, 35 (2002) 54-62

[9] C. V. Zhou, C. Leckie, S. Karunasekera, A Survey of Coordinated Attacks and Collaborative Intrusion Detection, Computers & Security, 29 (2010) 124-140

[10] S. Shamshirband, S. Kalantari, Z. Bakhshandeh, Designing a Smart Multi-Agent System Based on Fuzzy Logic to Improve the Gas Consumption Pattern, Sci Res Essays, 5 (2010) 592-605

[11] A. Feizollah, S. Shamshirband, N. Anuar, R. Salleh, M. Mat Kiah, Anomaly Detection Using Cooperative Fuzzy Logic Controller, in: K. Omar, M. Nordin, P. Vadakkepat, A. Prabuwono, S. Abdullah, J. Baltes, S. Amin, W. Hassan, M. Nasrudin (Eds.) Intelligent Robotics Systems: Inspiring the NEXT, Springer Berlin Heidelberg, 2013, pp. 220-231

[12] D. Petković, N. T. Pavlović, S. Shamshirband, M. L. Mat Kiah, N. Badrul Anuar, M. Y. Idna Idris, Adaptive Neuro-Fuzzy Estimation of Optimal Lens System Parameters, Optics and Lasers in Engineering, 55 (2014) 84-93

[13] D. Petković, Ž. Ćojbašić, V. Nikolić, S. Shamshirband, M. L. Mat Kiah, N. B. Anuar, A. W. Abdul Wahab, Adaptive Neuro-Fuzzy Maximal Power Extraction of Wind Turbine with Continuously Variable Transmission, Energy, 64 (2014) 868-874

[14] S. Shamshirband, N. B. Anuar, M. L. M. Kiah, A. Patel, An Appraisal and Design of a Multi-Agent System-based Cooperative Wireless Intrusion Detection Computational Intelligence Technique, Engineering Applications of Artificial Intelligence, 26 (2013) 2105-2127

[15] X. Xu, T. Xie, A Reinforcement Learning Approach for Host-based Intrusion Detection Using Sequences of System Calls, Advances in Intelligent Computing, (2005) 995-1003

[16] S. S. Shamshirband, H. Shirgahi, S. Setayeshi, Designing of Rescue Multi Agent System Based on Soft Computing Techniques, Advances in Electrical and Computer Engineering, 10 (2010) 79-83

[17] S. A. Razak, S. M. Furnell, N. L. Clarke, P. J. Brooke, Friend-assisted Intrusion Detection and Response Mechanisms for Mobile Ad Hoc Networks, Ad Hoc Networks, 6 (2008) 1151-1167

[18] P. Muñoz, R. Barco, I. de la Bandera, Optimization of Load Balancing Using Fuzzy Q-Learning for Next Generation Wireless Networks, Expert Systems with Applications, 40 (2013) 984-994

[19] Kdd cup 1999 data, UCI KDD Archive [http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html] (1999)

[20] E. Çayirci, C. Rong, Front Matter, in: Security in Wireless Ad Hoc and Sensor Networks, John Wiley & Sons, Ltd, 2009, pp. i-xxiii

[21] The CAIDA DDoS Attack 2007 Dataset, Available from [http://www.caida.org/data/passive/ddos-20070804_dataset.xml] (2007)

[22] P. A. Raj Kumar, S. Selvakumar, Distributed Denial of Service Attack Detection Using an Ensemble of Neural Classifier, Computer Communications, 34 (2011) 1328-1341

[23] J.-Y. Huang, I. E. Liao, Y.-F. Chung, K.-T. Chen, Shielding Wireless Sensor Network Using Markovian Intrusion Detection System with Attack Pattern Mining, Information Sciences, 231 (2013) 32-44

[24] M. Eslaminejad, S. A. Razak, Fundamental Lifetime Mechanisms in Routing Protocols for Wireless Sensor Networks: A Survey and Open Issues, Sensors, 12 (2012) 13508-13544

[25] P. Cingolani, J. Alcala-Fdez, jFuzzyLogic: a Robust and Flexible Fuzzy-Logic Inference System Language Implementation, in: Fuzzy Systems (FUZZ-IEEE) 2012 IEEE International Conference on, 2012, pp. 1-8