Journal of Intelligent & Fuzzy Systems 28 (2015) 1345–1357
DOI:10.3233/IFS-141419
IOS Press

Co-FQL: Anomaly detection using cooperative fuzzy Q-learning in network

Shahaboddin Shamshirband[a,*], Babak Daghighi[a], Nor Badrul Anuar[a], Miss Laiha Mat Kiah[a], Ahmed Patel[b] and Ajith Abraham[c]

[a]*Department of Computer System & Technology, Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia*

[b]*School of Computer Science, Centre of Software Technology and Management (SOFTAM), Faculty of Information Science and Technology (FTSM), University Kebangsaan Malaysia, UKM Bangi, Selangor Darul Ehsan, Malaysia*

[c]*Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, WA, USA*

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. We propose a distributed intrusion detection system called Cooperative IDS to protect wireless nodes within the network and target nodes from DDoS attacks by using a Cooperative Fuzzy Q-learning (Co-FQL) optimization algorithmic technique to identify the attack patterns and take appropriate countermeasures. The Co-FQL algorithm was trained and tested to establish its performance by generating attacks from the NSL-KDD and "CAIDA DDoS Attack 2007" datasets during the simulation experiments. Experimental results show that the proposed Co-FQL IDS has a 90.58% higher accuracy of detection rate than Fuzzy Logic Controller or Q-learning algorithm or Fuzzy Q-learning alone.

Keywords: Intrusion detection, fuzzy system, reinforcement learning, multi agent system, cooperative IDS

1. Introduction

Recent advances in wireless communications particularly Wireless Sensor Networks (WSNs) have enabled the development of low cost and powerful multifunctional nodes in small size communicating over radio frequencies [3]. These type of networks are utilized with a wide range of applications from natural disaster relief [4], health monitoring [6], to hazardous events [5]. The existing application designs for wireless sensors afford greater flexibility in establishing communications and increasing system automation, but are deficient in security and privacy [10]. 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, adversaries may possess passive and active access to secret information, such as keys stored in a compromised node by eavesdropping [14] or Denial of Services (DoS) attacks. As a result, security is still a major design goal in WSNs.

To mitigate malicious attacks particularly DDoS attack, the Intrusion Detection Systems (IDSs) are employed to detect abnormal traffic patterns that diverge from the modeled expected normal traffic behavior [16, 17, 21]. Tsunoda et al. proposed a simple inspection packet mechanism to avoid stateful inspection against Distributed Reflective Denial of Service (DRDoS) attack [18]. However, the mechanism is vulnerable to DDoS attacks as the state tables were overwhelmed by the moderate size and complexity of DDoS attack features.

*Corresponding author. Shahaboddin Shamshirband, Department of Computer System & Technology, Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia. E-mail: shamshirband@um.edu.my.

The Q-learning algorithm has been modified by rough set theory (RST) to learn optimum value for different attack attributes to classify network traffic data [15]. Munoz et al. [11] utilized fuzzy Q-learning for congestion detection to drop packets that differs from normal features. Although, fuzzy Q-learning algorithms proposed by Munoz improved the accuracy of detection and consume minimum resources, due to increasing the high volume of traffic comes from DDoS attack the fuzzy Q-learning agent could not response well while, the accuracy of detection decreased.

Cooperative Intrusion Detection and Prevention System (CIDPS) based on a comprehensive set of requirements for wireless sensor networks; as proposed by Patel et al. [13] scrutinizes the special characteristics of a distributed framework structure within Smart Grid Networks (SGN) with Collaborative IDPS. Although, the proposed CIDPS influences on robustness, flexibility, adaptability and low resource consumption, but the accuracy of detection was not satisfied by using fuzzy threshold.

We aims to design a hybrid intrusion detection system called a Cooperative Fuzzy Q-Learning (called herein CoFQL) in this study to enhance the learning ability of attack detection and improve the speed of decision process. 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. The cooperative FQL algorithm aims to maximize the cost function (accuracy of detection) during attack detection.

The remainder of the paper is organized as follows. The related studies are explored in Section 2. Network model is described in Section 3. In Section 4, system's models including the fuzzy expert system, the fuzzy Q-learning for anomaly detection, and the design of the Cooperative-FQL algorithm are explained. The conducted experiments are discussed in Section 5. Section 6 concludes the paper.

2. Related works

2.1. DDoS attack

Presently, DDoS attack type reported to the CERT/CC [7], comprises sending flooding packets to a destination causing the consumption of resources (CPU and memory), network bandwidth, router processing capacity, or network stack resources, and disrupt the network connectivity to the victims. 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 [20].

The problem of DDoS attacks has already been addressed in many studies. Shiaeles et al. [2], proposed a fuzzy estimator approach for identifying a DDoS attack and classifying the malicious IPs. This method is very accurate in detecting the DDoS attack and fairly accurate for identifying the offending IP addresses within strict time limits that allow the system to respond in real time situation. Khatoun et al. [9] proposed a decentralized alert correlation technique to detect DDoS attacks based on P2P architecture, which correlates alerts produced by various intrusion detection systems and provides a high-level view of attempted intrusions.

2.2. DDoS attack dataset

The KDD Cup dataset was produced by processing the tcpdump portions of the 1998 DARPA Intrusion Detection System (IDS) evaluation dataset [9]. However, it is not synthetic and does not reflect contemporary attacks. NSL-KDD datasets [2] mitigates the weaknesses incurred by KDD'99 datasets. NSL-KDD contains fewer redundant and duplicate records in the training and test phases of learning-based detection, which makes the evaluation process of the learning system more efficient. CAIDA dataset [1] consists of DDoS attack dataset 2007, which can be availed by user's request. Tdataset consist of an hour of anonymized traffic traces from a DDoS attack.

In our research work, the similar key features of DDoS attacks collects from NSL-KDD and CADIA dataset and then combined into the new dataset that is called mixed dataset. The aim is to validate our security mechanism against DDoS attacks with different attack datasets.

2.3. System measurements

The most important outcomes derived from Wireless IDS are that the accuracy of detection is increased, and the percentage of false alarm rate is decreased. These two metrics are used to appraise the Collaborative-FQL performance, namely, the Detection Rate (DR), and the False Alarm Rate (FAR). Table 1 demonstrates the possible status for an IDPS reaction and Equations (1–5)

Table 1
Possible status for an IDPS reaction

Confusion matrix		Predicted class	
		Normal	Abnormal
Actual Class	Normal	True Negative (TN)	False Negative (FN)
	Abnormal	False Positive (FP)	True Positive (TP)

are applied as numerical evaluations to quantify the performance of IDSs.

True Negative Rate

$$(TNR) = \frac{TN}{TN + FP} = \frac{no.\ true\ alerts}{no.\ alerts} \quad (1)$$

True Positive Rate (TPR) or Sensitivity or Recall

$$(R) = \frac{TP}{TP + FN} = \frac{no.\ detected\ attacks}{no.\ observables\ attacks} \quad (2)$$

False Positive Rate

$$(FPR) = \frac{FP}{TN + FP} = 1 - \frac{TN}{TN + FP} \quad (3)$$

False Negative Rate

$$(FNR) = \frac{FN}{TP + FN} \quad (4)$$

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \quad (5)$$

2.4. Real time feature extraction

The main features of DDoS attacks was indicated in the study [13]. From the dataset attributes, five features, as shown in Table 2, have been selected for accurate detection of DDoS attacks. These attributes mostly consist of spoofed source address and contain half-open connections.

Our objective is to differentiate the DDoS attack and normal traffic. 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.

Table 2
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

2.5. Fuzzy Q-learning - motivation

FQL based congestion 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 addition, the single FQL is unable to store the huge amount of data received due to an increase in the lookup table. In all cases the risk of resource consumption is higher when modifying FLC margins. This algorithm is deemed to be expanded in order to propose our Cooperative Fuzzy Q-Learning algorithm.

In our scheme, we modified the Munoz FQL algorithm by applying min-max action selection instead of $\in$-greedy action-selection and softmax action selection rule. The main drawback of $\in$-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. In addition, we optimized FQL algorithm through cooperative algorithm that utilizes weight strategy policy.

3. Network model

According to the form of network organization, WSNs routing protocols can generally be divided into two categories: flat network routing and hierarchical network routing. In a flat topology, each sensor node, acting in the same role, communicates with other sensor nodes in WSNs by attempting to find a route to the sink node in the way of flooding. This method is very effective in relatively small networks; however, it is hard to apply to enormously dense networks due to flooded messages. On the other hand, a hierarchical routing protocol, each cluster has a coordinator, called Cluster Head (CH), and a number of member sensor nodes. In order to balance the energy consumption on inter-cluster sensor nodes, the CH must be re-clustered periodically. When member sensor nodes send their data to the responsible CH, the CH aggregates the data and sends them to the Base Station (BS) through other CHs. This clustering leads to be more scalable and energy-efficient for the fact that most of the data sensing and processing can be finished within the same cluster.

4. System model

We designed an architecture so-called Cooperative Fuzzy Q-leaning Detection System (Co-FQL) to identify potential DDoS attack on the wireless network. In the first layer of proposed Co-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 layer, the FLC optimized by Q-learning to discover and detect the security treats captured by FES. Finally, FQL uses cooperative Q-learning to provide a negotiation mechanism to improve the accuracy of attack detection and decrease the learning processing time process of detection. Figure 1 depicts the architecture of Cooperative Fuzzy Q-learning Detection System (Co-FQL).

In the paradigm of Cooperative Fuzzy Q-learning (Co-FQL) DDoS Detection System, sensor nodes capture packets received from traffic. Sink nodes utilized detection policies in three levels.

- **Fuzzy Expert System (FES) Policy**: reduces the state space for the sink node due to an increase in look-up table or Q-table. This policy broadcasts the gain of ES engine (i.e. Abnormal, normal) through a Base Station (BS).

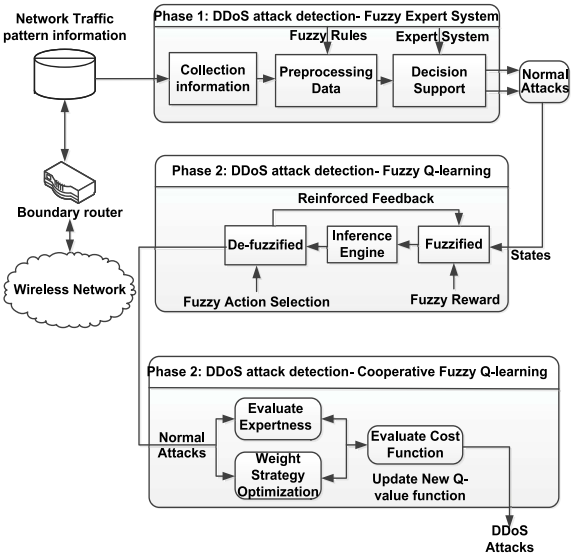

Fig. 1. Architecture of Cooperative Fuzzy Q-learning (Co-FQL).

- **Fuzzy Q-learning (FQL) Policy:** mitigates the possible faults escaped from the FES policies. The anomalous data is identified by optimizing Fuzzy Logic Controller based Q-learning.
- **Cooperative-FQL (Co-FQL) Policy:** synchronizes the negotiation policy in communication sensors by utilizing the measurement over multiple FQL. BS applies appropriate cooperative mechanism to identify anomalous usage.

4.1. Fuzzy expert system for DDoS attack detection

To fully exploit the suspicious level at the first layer, 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 2 shows the details of component of the proposed Fuzzy Expert detection system architecture.

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.

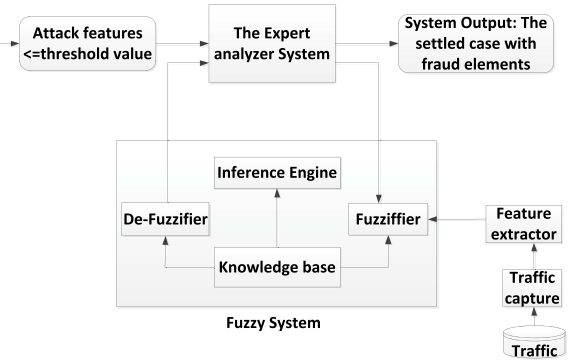

Fig. 2. The architecture of the proposed fuzzy expert system.

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.

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.

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

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)

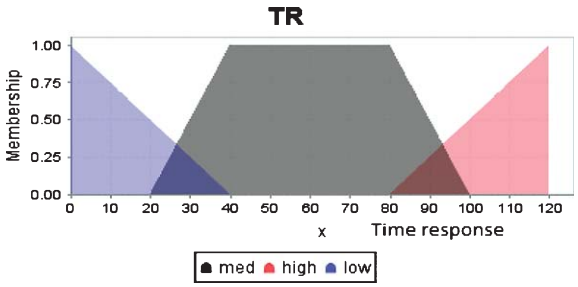

Fig. 3. The membership functions of linguistic variables for attack data source Tr.

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 3 indicates the membership functions for time response.

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.

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

Table 4
Pseudo code proposed for FES

```
REPEAT
READ current records
SET TRUST to True
IF Protocol Type = TCP THEN
  IF Source IP < 124 and Destination IP > 130 THEN
   IF Source Port && Destination Port ≠ 80 THEN
    IF Time Response > 60 MS THEN
     IF Buffer Size > 40 KB THEN
      IF Count > 3 times THEN
        SET TRUST to False
      END IF
     END IF
     END IF
   END IF
  END IF
END IF
Print TRUST
```

expensive and impractical, and cannot 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. FQL scheme for anomaly detection

An optimized FQL strategy is proposed in this section for detecting DDoS attack. The design of the anomaly based FQL utilized the FLC, which converts the continuous inputs into fuzzy sets. Three fuzzy sets have been defined for the input of FQL to represent three different situations as a state space of Q-learning: Fig. 4 demonstrates a block diagram of the optimization system for anomaly based–FQL.

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

$$S(t) = [Time\ response,\ Buffer\ size,\ Count] = [Tr,\ Bs,\ Co] \quad (6)$$

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.

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 5 (a, and b) indicates the membership function for the input variables of buffer size and count and Fig. 6 indicates the output variable of fuzzy systems. The membership functions are triangular or trapezoidal.

The number of selected rules in this section is smaller, that results 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.

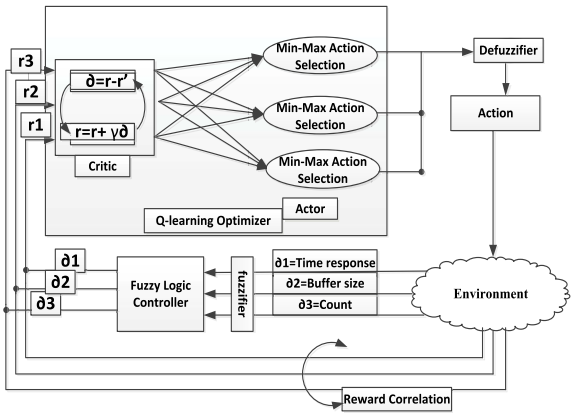

Fig. 4. Block diagram of the optimization system for FQL.

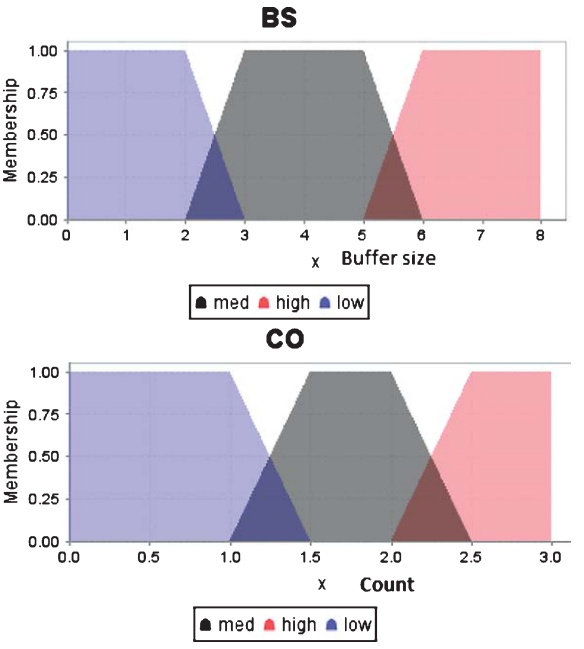

Fig. 5. (a), and (b), Input Membership function design in Java-fuzzy toolbox [8].

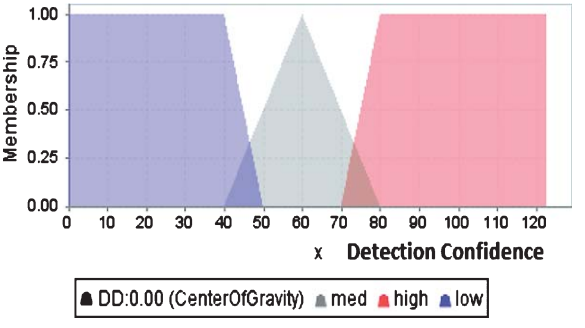

Fig. 6. Output Membership function.

- **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'.

In order to find the optimal action, the reinforcement signal r (t) used Equation (7). 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:

$$r(t+1) = \begin{cases} 100, \ if \ DC^{k}_{meatured}(t) < DC_{th} \\ -100, \ otherwise \end{cases} \quad (7)$$

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 fuzzy min-max weighted average:

$$Detect\ Confidence = output(C_j) \quad (8)$$

$$= \left(\sum\nolimits_{j=1}^{N} \alpha_j \, c_j\right) / \left(\sum\nolimits_{j=1}^{N} \alpha_j\right)$$

$$\alpha_j = [\mu_j(x_0) * \mu_j(y_0)] \quad (9)$$

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. Table 5 demonstrates the results of applying one of the possible min-max action selections for FQ-Learning algorithm.

Fuzzy min-max action selection mechanisms, which can reason with imprecise information, are good at explaining their decisions for a single FQL agent but the action performed to make decisions are not acquired when the complexity of anomaly variables is high. To improve the accuracy of action decision policy, cooperative mechanism strategy adopts to FQL through weight strategy policy.

4.3. Cooperative FQL (Co-FQL) algorithm

Previous researches on the impacts of Multi-Criteria Expertness (MCE) based cooperative Q-learning were presented [12] to improve the cooperative learning process in a hunter-prey problem. This shared information conducts episodes (state, action, and reward), sensation (state), and policies. The aim of this part is to contribute the following question: "How can an FQL agent enjoy exchanging information during the learning process, in order to improve the correct rate of the detection system?". In this research work, the cooperative policy evaluates the proficiency of an agent to optimize the cost function based on weight strategy sharing incorporating the expertness and weight assignment mechanisms for real time DDoS attack detection. These two mechanisms result as modules that are used in Co-FQL architecture and system implementation to speed up the learning process.

4.3.1. Expertness criteria

The agent's expertness is evaluated based on the Average of Positive of Reward Signals (APRS) as shown in Equation 10.

$$APRS(expertness) = ei = \frac{\sum |Reward|}{N} = \frac{sum\ of\ positive\ reward\ signals}{Number\ of\ total\ reward} = \frac{\sum reward\,|TP + TN|}{N} \quad (10)$$

where, the sum of positive rewards divided by the number of total rewards (i.e, negative and positive rewards) per epoch. In other word, the criterion of expertness of agent for DDoS attack detection calculates based on the following assumption: sum of positive reward signal received when no attack has taken place and no alarm is raised (True Negative), and the number of correct intrusion instances detected by the system (True Positive). The objective is to evaluate the proficiency measures of

Table 5
Min-Max action selection from ∂i to the next possible states

Input variables state I (∂i)				Input variables state I (∂j)			Output desirable	Min Max Fuzzy
Tr	Bs	Count	Rules	Tr	Bs	Count	Pattern	
$\mu_{Tr(low)} = 0.2$, $S_{Tr} = 40ms$	$\mu_{Bs(high)} = 0.9$, $S_{Bs} = 7k$	$\mu_{Co(low)} = 0.2$, $S_{Co} = 1.5\%$	$Rule^1$	$\mu_{Tr(high)} = 0.8$, $S_{Tr} = 110ms$	$\mu_{Bs(high)} = 0.8$, $S_{Bs} = 8k$	$\mu_{Co(high)} = 0.8$, $S_{Co} = 3\%$	Abnormal (0.8)	Min Sj (0.8,0.8.0.8) = 0.8
			$Rule^2$	$\mu_{Tr(high)} = 0.9$, $S_{Tr} = 120ms$	$\mu_{Bs(low)} = 0.2$, $S_{Bs} = 2k$	$\mu_{Co(med)} = 0.5$, $S_{Co} = 1\%$	Abnormal (0.9)	Min (0.9,0.2,0.5) = 0.2
			$Rule^3$	$\mu_{Tr(low)} = 0.2$, $S_{Tr} = 40ms$	$\mu_{Bs(low)} = 0.2$, $S_{Bs} = 2k$	$\mu_{Co(low)} = 0.2$, $S_{Co} = 1.5\%$	Normal (0.2)	Min (0.2,0.2, 0.2) = 0.2
			$Rule^4$	$\mu_{Tr(low)} = 0.2$, $S_{Tr} = 40ms$	$\mu_{Bs(high)} = 0.8$, $S_{Bs} = 3k$	$\mu_{Co(low)} = 0.2$, $S_{Co} = 1.5\%$	Normal (0.2)	Min (0.2,0.8,0.2) = 0.2
Detect confidence (DC)							Abnormal: Max (0.8,0.2) = 0.8	
Detect confidence Abnormal = 0.8							Normal: Max (0.2,0.2,0.2) = 0.2	
Detect confidence Normal = 0.2							Threshold: If $DC^k_{meatured}(t) < DC_{th}$	
							$\mathbf{0.8 > 0.2 \rightarrow Abnormal > normal}$	

agent during attack detection. The value of APRS can be used to the weight assignment mechanism.

4.3.2. Weights assignment mechanism

This mechanism is utilized the APRS to assign the suitable weight to correspond agents as shown in Equation 11:

$$W_{ij} = \begin{cases} 1-\alpha_i, & e_i = e_j \\ \alpha_i \frac{e_j - e_i}{\sum_{j=1}^{n}(e_j - e_i)}, & e_j > e_i \\ 0, & otherwise \end{cases} \quad (11)$$

where $0 < \alpha_i < 1$ is the impressibility factor that depicts how a fuzzy Q-learning agent depends on other's knowledge. For instance, when the proficiency measure of agent (i) is less than agent (j), then its weight is relative to the amount of expertness difference between the agent j and agent i divided by the sum of the other expert's differences. Figure 7 depicts the weight assignment strategy for the sink node which called FQL. In this case, FQL (i) evaluates the expertness of FQL and then assigns the weight based on Equation.12.

The procedure of negotiation between agents repeat until the weighted strategy is finished. After termination of the weighted strategy the Q-value is modified as shown in Equation 12 to update their knowledge to detect a DDoS attack.

$$Q_i^{new} = (1-\alpha_i) * Q_i^{new} + \alpha_i * \sum_{j \in Expert(i)} (W_{ij} * Q_i^{old}) \quad (12)$$

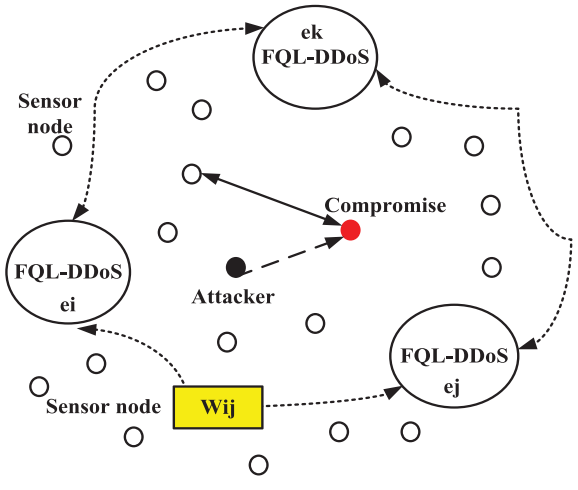

Fig. 7. The weight assignment for homogeneous agents.

Table 6
Comparison of existing ensemble algorithms with proposed algorithm

Algorithm / Features	Fuzzy Logic Controller (D1)	Q-learning (D2)	Fuzzy Q-learning (D3)	Cooperative Q-learning (D4)
Prior knowledge	Required	Not Required	Not Required	Not required
Method used to combine classifiers	Fuzzy Classifiers	Markov Decision Process	Fuzzy rule base and Q-learning	Weighted strategy optimization combine with fuzzy Q-learning
Drawbacks	Work for small subset	Sensitive to noise and outliers, High cost consumption,	One classifier, Low speed of detection, Fail to high volume of traffic	Large amount of traffic data is required.
Advantages	Simple implementation.	Capable of handling multi-class attack detection.	No require prior knowledge of data distribution.	Adaptive and Incremental learning

where, Q_i^{old} is the previous knowledge of each agent, $0 < \alpha_i < 1$ is the impressibility factor and shows how much agent i relies on the others knowledge.

The instances can be classified correctly or incorrectly in normalization layer (Fuzzy Expert). If the instances are classified correctly, the weight of these instances is reduced aggressively so that these instances will not be considered during next iteration of training. If the instances are incorrectly classified, there are two possibilities: (i) normal traffic classified as attacks (false positive) and (ii) attack classified as normal traffic (false negative). A closer look at the fuzzy min-max action selection in FQL reveals that the assignment of weight to correct classification is satisfied by threshold value. Consequently, if the weight exceeds less than our default threshold value, it drops the threshold, indicating the detection is bad.

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), Fuzzy Q-learning (D3) and Cooperative Fuzzy Q-learning (D4) algorithms. The cost function for various Fuzzy classifier, Q-learning, fuzzy Q-learning, and cooperative Q-learning algorithms are calculated. Table 6 shows the comparison of the proposed Co-FQL detection algorithm with existing ensemble algorithms.

An FLC is usually integrated with a fuzzyfier and a defuzzyfier that translates real-valued inputs to fuzzy terms and vice versa. The Q-Learning algorithm is adopted when both the states and the actions to be selected belong to finite, small sets in order to find a control rule that maximizes at each control cycle the expected discounted sum of future reinforcement.

FQL seems to be effective models to exploit the features of Q-Learning also with real-valued inputs and outputs. The main problems of this approach come from the fact that more than one fuzzy rule usually trigger on a single real-valued state, and it should be possible to reinforce each of them according to its contribution. Cooperative FQL, where the agents can share their performance statistics with their neighbor agents, so that agents try to learn the optimal action policy based on the overall reward of the neighborhood instead of just local selfish rewards.

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 better performance in the 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 and Fuzzy Q-learning algorithm the observation is limited by one single classifier. Therefore, these 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 the weighted strategy optimization with fuzzy Q-learning 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. We proposed the cost function to facilitate performance comparison of existing ensemble methods with our proposed Co-FQL algorithm. It was calculated using Equation (13).

$$Cost = (1 - TP) + (TP + FP) \tag{13}$$

Where TP, the number of instances is correctly predicted as attack class, FP is the number of samples incorrectly predicted as attack class. The algorithm with least cost function emerges out as the best detection system.

5.1. Performance verification

Our proposed classification algorithm with cost per sample function $Cost = (1 - TP) + (TP + FP)$ is compared with existing soft computing methods D1, D2, and D3 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 Co-FQL with cost minimization, it can be seen that Co-FQL algorithm with cost minimization yields an improvement of 13.57% $\left(\frac{2.80-2.42}{2.80} * 100\right)$ over FQL algorithm as shown in Table 7. Further, it is evident that FQL with cooperative mechanism achieves the maximum gain of accuracy of detection. Moreover, it can be inferred from Fig. 8 that cost per percentage of samples or anomalous is less for Co-FQL algorithm than the other methods.

The next tryout was conducted for CAIDA traffic. Table 7 depicts that the detection accuracy is 84.16% with 4.22% false positive rate. Comparing the false positive rate, Co-FQL algorithm with minimum cost function, it can be seen that Co-FQL yields an improvement of 27.11% over FLQ, 27.86% over Q-learning, and 34.37% over FLC. It can be inferred from Figs. 9

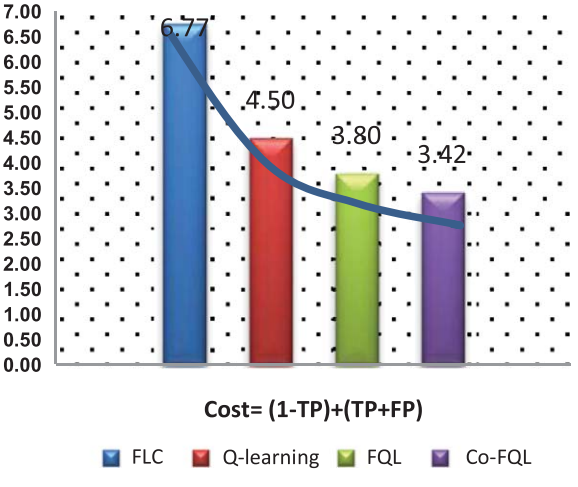

Fig. 8. Cost per sample for existing DDoS detection and Co-FQL from NSL-KDD attack source.

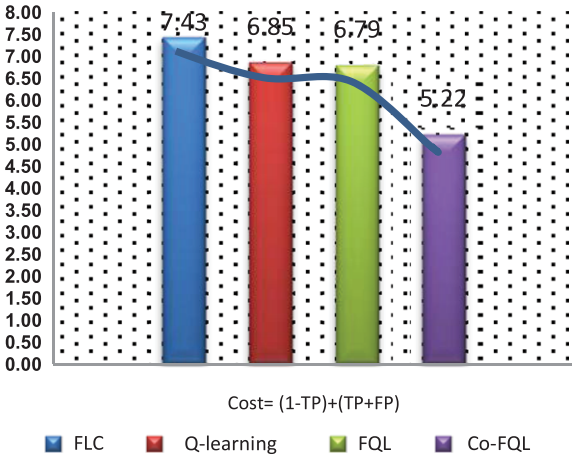

Fig. 9. Cost per sample for existing DDoS detection and Co-FQL from CAIDA attack source.

and 10 that cost per samples is less for Co-FQL algorithm than other methods.

Experiment 3 was performed in mixed dataset. The trained Co-FQL algorithm was able to detect DDoS attack with high accuracy of detection, low false positive rate and minimum cost function as shown in Table 7.

5.2. Computational time of Co-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 8 shows the performance comparison of the Co-FQL in terms of consuming time obtained during the experiments. It can be realized that the training time of Co-FQL is similar to FQL, but it consumes more testing time than the FLC, Q-learning, and FQL. Also, the computational time was calculated on Intel 3.10 GHz, Core i-5 Processor, 4 GB RAM computer.

Testing time of the proposed Co-FQL method is a little high due to the ensemble output combination methods such as fuzzy Q-learning and weight strategy sharing algorithm, but more detection accuracy was

Table 7
Simulation result of detection algorithm for NS-KDD dataset

Percentage of anomalous (%)	FLC						Q-Learning						FQL						Co-FQL					
	NS-KDD		CAIDA		Mixed		NS-KDD		CAIDA		Mixed		NS-KDD		CAIDA		Mixed		NS-KDD		CAIDA		Mixed	
	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)	TP (%)	FP (%)
1	70.2	4.7	73.5	4.9	78.2	3.9	75.2	1.4	74.2	4.7	78.7	3.8	80.1	1.2	74.9	4.2	8	3.8	83.2	1.2	75.7	2.1	84.3	3.2
5	71.5	4.8	73.9	5.1	78.6	4.1	76.7	1.6	74.7	4.9	79.2	4.1	81.2	1.4	75.1	4.3	84.7	3.9	83.4	1.3	76.2	2.6	85.8	3.1
10	73.2	4.9	74.2	5.3	79	4.5	76.9	1.9	75.4	5.2	79.4	4.5	82.5	1.9	75.6	4.8	85.4	4.1	84.3	1.5	76.8	2.9	86.9	4.4
15	75.4	5.0	74.8	5.7	79.3	4.7	77.6	2.1	75.9	5.3	79.8	4.6	83.7	2.1	75.9	5.1	85.9	4.4	85.6	1.7	77.5	3.1	87.3	4.6
20	75.9	5.1	75.1	6.2	80.1	4.9	78.5	2.4	78.6	5.4	80.1	4.8	83.9	2.4	76.4	5.2	86.6	4.7	87.9	1.9	77.9	3.4	88.9	4.8
25	76.1	5.8	75.4	6.5	80.6	5.1	79.8	3.1	78.8	5.6	80.5	5.0	84.2	2.6	76.8	5.6	87.7	4.8	88.3	2.1	80.4	3.7	89.6	4.9
30	77.1	5.9	75.6	6.7	81.3	5.4	80.1	3.4	79.4	5.8	81.3	5.1	85.8	2.8	77.6	5.9	88.8	5.1	89.7	2.4	80.9	3.9	90.4	5.0
35	80.1	6.0	76.1	6.9	82.1	5.9	82.3	3.9	79.9	6.0	81.8	5.3	86.4	2.9	78.9	6.2	89.4	5.3	90.5	2.6	90.3	4.4	91.6	5.2
40	81.9	6.2	76.8	7.0	82.5	6.1	83.6	4.2	80.4	6.2	82.7	5.6	87.7	3.2	79.6	6.4	90.7	5.6	91.7	3.1	90.6	4.8	92.2	5.4
45	78.2	6.4	79.2	7.1	83.1	6.2	79.8	4.9	81.8	6.4	83.9	5.8	88.5	3.4	80.5	6.7	91.4	5.8	92.4	3.2	91.2	5.2	93.8	5.6
50	76.8	6.5	79.5	7.2	83.4	6.5	78.9	5.2	82.8	6.6	84.6	6.1	89.6	3.9	81.2	6.8	92.5	5.9	94.2	3.3	91.7	5.7	94.5	5.8
55	75.6	6.8	80.3	7.4	84.1	6.8	79.3	5.6	83.7	6.9	85.9	6.2	90.4	4.1	83.4	7.0	93.8	6.0	96.5	3.5	92.4	6.2	95.9	6.0
60	74.8	6.9	80.6	7.6	84.2	6.9	80	5.8	84.5	7.1	86.6	6.5	92.4	4.5	85.8	7.1	94.4	6.1	98.2	3.7	92.5	6.9	96.4	6.1
Avg.	**75.91**	**5.77**	**76.54**	**6.43**	**81.27**	**5.46**	**79.13**	**3.5**	**79.24**	**5.85**	**81.88**	**5.18**	**85.88**	**2.80**	**78.59**	**5.79**	**83.02**	**5.04**	**89.68**	**2.42**	**84.16**	**4.22**	**90.58**	**4.93**

Table 8
Performance comparison of Co-FQL with existing machine learning methods in terms of consuming time

Dataset	Algorithms	Training time (Sec)	Testing time (Sec)
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
	Cooperative Q-learning (D4)	3.22	1.42

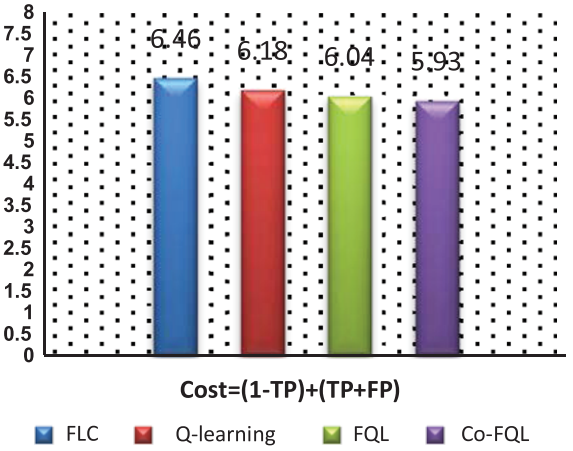

Fig. 10. Cost per sample for existing DDoS detection and Co-FQL from Mixed attack source.

achieved in Co-FQL. The speedup of Co-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.

5.3. *Convergences of Co-FQL in terms of expertness*

To evaluate the expertness of Co-FQL algorithm, we used the sum of positive rewards divided by the number of total rewards such as negative and positive rewards per epoch. For example, the percentage of expertness for our Co-FQL algorithm calculates based on the following assumption. In each epoch, the average number of positive rewards due to the correct attack detection will be calculated, and then the expertness value can be reached. Figure 11 shows the expertness of Co-FQL agent during 500 epochs.

Any DDoS method in order to be effective and offer added value to the infrastructure it protects should be able to perform in real time. The soft real-time processing is to detect anomalous activity as, or soon after, it

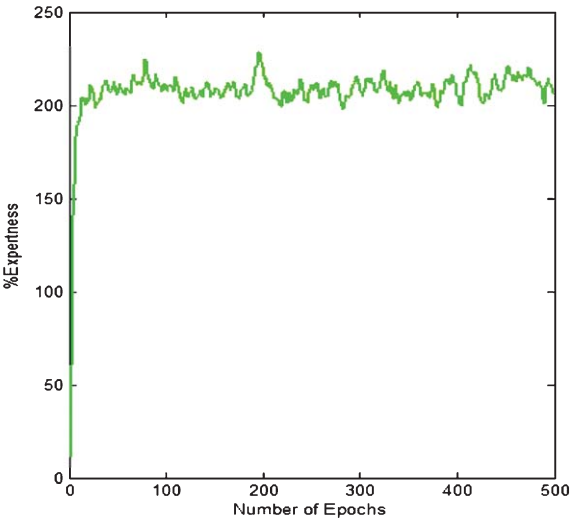

Fig. 11. Expertness of Co-FQL agent during 500 epochs.

occurs. We consider the upper limit for detection delay to be equal to the capacity of the server which is being protected. In a recent paper [19] a "real time" detection of DDoS was achieved by using fuzzy rules on the Hurst parameter. The time needed for the attack to be detected successfully was 13 s which can be classified as real-time in a certain context. According to the experiments, the Co-FQL converges to higher detection expertness with time detection less than 2 seconds. Therefore, it is deduced that the proposed framework is suitable for real-time scenario for detecting DDoS attacks.

6. Conclusion and future research

Development of the Co-FQL algorithmic technique for online IDS by modifying simple Fuzzy Q-learning algorithm and combining it with Cooperative mechanism detects DDoS attack with 90.58% accuracy, which is far superior to Fuzzy Logic Controller, or Markov decision processes or Fuzzy Q-learning 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 same parameters are applied to all attributes, classification accuracy varies widely. But combination of different expertness with weighted strategy policy for different attributes

in different cluster yields best result of classification accuracy. 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 90.59% accuracy and minimum cost function by 5.93% in mixed dataset compared to other existing detection methods (i.e., fuzzy logic controller, Q-learning, and fuzzy 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. A future initiative is to extend the proposed Co-FQL mechanism by incorporating data from various attack types and sources to further enhance its decision making capabilities in order to thwart existing or new attacks. Also, as part of future research work on complementing Co-FQL, studying a network evolutionary algorithm, such as the imperialist competitive algorithm, is considered of utmost importance.

Acknowledgments

This work is supported by the Ministry of Science, Technology and Innovation, Malaysia under Grant eScienceFund 01-01-03-SF0914

References

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

[2] The NSL-KDD Data Set, in: Available from [http://iscx.ca/NSL-KDD], 2009.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, Wireless sensor networks: A survey, *Computer Networks* **38** (2002), 393–422.

[4] Y.E. Aslan, I. Korpeoglu and Ö. Ulusoy, A framework for use of wireless sensor networks in forest fire detection and monitoring, *Computers, Environment and Urban Systems* (2012).

[5] A. Bonastre, J.V. Capella, R. Ors and M. Peris, In-line monitoring of chemical-analysis processes using Wireless Sensor Networks, *TrAC Trends in Analytical Chemistry* **34** (2012), 111–125.

[6] J.M.L.P. Caldeira, J.J.P.C. Rodrigues and P. Lorenz, Toward ubiquitous mobility solutions for body sensor networks on healthcare, *Communications Magazine, IEEE* **50** (2012), 108–115.

[7] C.C. Center, CERT/CC Statistics 1988-2003, in, 2012.

[8] P. Cingolani and 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.

[9] S. Hettich and S.D. Bay, Kdd cup 1999 data, in: U.K. Archive, ed., 1999.

[10] N. Li, N. Zhang, S.K. Das and B. Thuraisingham, Privacy preservation in wireless sensor networks: A state-of-the-art survey, *Ad Hoc Networks* **7** (2009), 1501–1514.

[11] P. Muñoz, R. Barco and 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.

[12] E. Pakizeh, M. Palhang and M. Pedram, Multi-criteria expertness based cooperative Q-learning, *Applied Intelligence* (2012), 1–13.

[13] P.A. Raj Kumar and S. Selvakumar, Distributed denial of service attack detection using an ensemble of neural classifier, *Computer Communications* **34** (2011), 1328–1341.

[14] P. Schaffer, K. Farkas, Á. Horváth, T. Holczer and L. Buttyán, Secure and reliable clustering in wireless sensor networks: A critical survey, *Computer Networks* **56** (2012), 2726–2741.

[15] N. Sengupta, J. Sen, J. Sil and M. Saha, Designing of on line intrusion detection system using rough set theory and Q-learning algorithm, *Neurocomputing* **111** (2013), 161–168.

[16] K.-A. Shim, Y.-R. Lee and C.-M. Park, An efficient identity-based broadcast authentication scheme in wireless sensor networks, *Ad Hoc Networks* **13** (2012), 1741–1749.

[17] B. Tian, S. Han, J. Hu and T. Dillon, A mutual-healing key distribution scheme in wireless sensor networks, *Journal of Network and Computer Applications* **34** (2011), 80–88.

[18] H. Tsunoda, K. Ohta, A. Yamamoto, N. Ansari, Y. Waizumi and Y. Nemoto, Detecting DRDoS attacks by a simple response packet confirmation mechanism, *Computer Communications* **31** (2008), 3299–3306.

[19] J. Wang and G. Yang, An intelligent method for real-time detection of DDoS attack based on fuzzy logic, *Journal of Electronics (China)* **25** (2008), 511–518.

[20] A.D. Wood and J.A. Stankovic, Denial of service in sensor networks, *Computer* **35** (2002), 54–62.

[21] A.A. Yavuz and P. Ning, Self-sustaining, efficient and forward-secure cryptographic constructions for Unattended Wireless Sensor Networks, *Ad Hoc Networks* **10** (2012), 1204–1220.