International Journal of Electrical, Electronics and Computer Engineering **7**(1): 42-48(2018)

ISSN No. (Online): 2277-2626

NIDSRL: Network Based Intrusion Detection System Using Reinforcement Learning

Darshana Kamavisdar[1] and Prof. Ram Ratan Ahirwal[2]

[1]M. Tech. Scholar, Department of Computer Science & Engineering, Samrat Ashok Technology Institute, Vidisha (Madhya Pradesh), India

[2] Professor, Department of Computer Science & Engineering, Samrat Ashok Technology Institute, Vidisha (Madhya Pradesh), India

(Corresponding author: Darshana Kamavisdar)

(Received 02 March, 2018 Accepted 23 April, 2018)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Intrusion detection is systems which continuously monitor the activity over the network perform by the intruders. In there any such misleading activity is found then that system report to the network manager. The intrusion detection systems are of two types network based and anomaly based. In this work, we focus for the detection and prevention of network intruders using reinforcement learning (Q-learning) techniques of artificial intelligence. The experimental analysis of the proposed methodology is performing using the KDDCUP'99 dataset and the performance measurement is done using parameters sensitivity, specificity and accuracy. After comparative analysis it is found that our proposed scheme is much better than the existing scheme.

Keywords: Intrusion detection System, Reinforcement learning, KDDCUP'99, Intruders.

I. INTRODUCTION

Intrusion Detection System (IDS) are software or hardware systems that automate the process of monitoring and analyzing the events that occur in a computer network, to detect malicious activity. Since the severity of attacks occurring in the network has increased drastically, Intrusion detection system have become a necessary addition to security infrastructure of most organizations. Intrusion detection allows organization to protect their systems from the threats that come with increasing network connectivity and reliance on information systems [1]. Given the level and nature of modern network security threats the question for security professionals should not be whether to use intrusion detection but instead which intrusion detection features and capabilities can be used. Intrusions are caused by: Attackers accessing the systems, Authorized users of the systems who attempt to gain additional privileges for which they are not authorized, Authorized users who misuse the privileges given to them. Intrusion detection systems (IDS) take either network or host based approach for recognizing and deflecting attacks. In either case, these products look for attack signatures (specific patterns) that usually indicate malicious or suspicious intent. When an IDS looks for these patterns in network traffic then it is network based (Fig. 1). When an IDS looks for attack signatures in log files, then it is host based. Various algorithms have been developed to identify different types of network intrusions; however there is no heuristic to confirm the accuracy of their results.

The exact effectiveness of a network intrusion detection system's ability to identify malicious sources cannot be reported unless a concise measurement of performance is available.

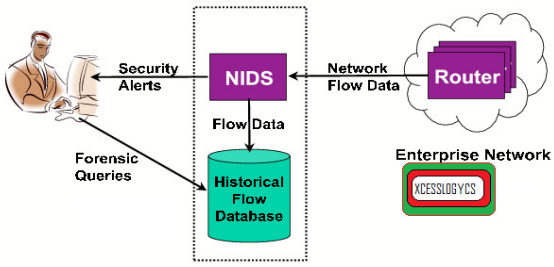

Fig. 1. Architecture of Network Intrusion detection System.

In this paper, we propose reinforcement learning (RL) [2] which is types of Machine Learning and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behaviour within a precise context in order to maximize its performance. Simple reward feedback is required for the agent to learn its behaviour this is known as the reinforcement signal. There are numerous dissimilar algorithms that undertake this issue. As a matter of fact, Reinforcement Learning is defined by an explicit type of problem and all its solutions are classed as Reinforcement Learning algorithms. In the predicament, an agent is supposed choose the best action to choose based on his current state.

While this step is repeated the difficulty is known as a Markov Decision Process. This method determines the network intruders efficiently and it helps to improve the performance of it.

II. RELATED WORK

Arunraj and Umarani [3] mentioned some common study techniques of Reinforcement learning and Q-learning algorithm with respect to the proposed IDPS which helps to see the best automated service on relate to subject topic. By using Q-learning algorithm we can identify the packet state as well as what it tries to do in the network. The user agent will take the decision according to the situation prevailing in the environment. Reddy [4] presented the Radial Basis Function (RBF) Networks are used to analyze the intrusion detection. The radial basis function network with the rough set theory combined addresses the intrusion problem. The RST approach has reduced the dimension for decreasing the training time of the RBF network. The trained RBF network is used to detect the new signs of intrusions. The model is tested for the three datasets and obtains the good results. The RBF network model adjusted the center based on the radial basis function is the crucial for accurately evaluation of the model. As the future research the model has to test directly on the real network data with unturned parameters. The RBF network has to design for the detection of the raw network data. Das *et al.* [1] used the Rough Set Theory (RST) and Support Vector Machine (SVM) to detect network intrusions. First, packets are captured from the network, RST is used to pre-process the data and reduce the dimensions. The features selected by RST will be sent to SVM model to learn and test respectively. The method is effective to decrease the space density of data. The experiments compare the results with Principal Component Analysis (PCA) and show RST and SVM schema could reduce the false positive rate and increase the accuracy. Kudenko and Servin [5] proposed a distributed Reinforcement Learning (RL) approach in a hierarchical architecture of network sensor agents. Each network sensor agent learns to interpret local state observations, and communicates them to a central agent higher up in the agent hierarchy. These central agents, in turn, learn to send signals up the hierarchy, based on the signals that they receive. Finally, the agent at the top of the hierarchy learns when to signal an intrusion alarm. They evaluated their approach in an abstract network domain. They presented solutions that enable the agents to learn an accurate signal policy and we have shown that the approach scales up to a large number of agents. Miller and Inoue [6] presented an intrusion detection system consisting of multiple intelligent agents. Each agent uses a self-organizing map (SOM) in order to detect intrusive activities on a computer network A blackboard mechanism is used for the aggregation of results generated from such agents (i.e. a group decision). In addition, this system is capable of reinforcement learning with the reinforcement signal generated within the blackboard and then distributed over all agents which are involved in the group decision making. Systems with various configurations of agents are evaluated for criteria such as speed, accuracy, and consistency. The results indicate an increase in classification accuracy as well as in its constancy as more sensors are incorporated currently this system is primarily tested on the data set for KDD Cup '99.

III. PROPOSED WORK

For detection of intruders we propose reinforcement learning and Q-learning techniques which effectively detects the intruders over the network. The brief overview about the proposed methodology is discussed below:

A. Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning inspired by behaviorist psychology, concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. The problem, due to its generality, is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In the operations research and control literature, the field where reinforcement learning methods are studied is called approximate dynamic programming. The problem has been studied in the theory of optimal control, though most studies are concerned with the existence of optimal solutions and their characterization, and not with the learning or approximation aspects. In economics and game theory, reinforcement learning may be used to explain how equilibrium may arise under bounded rationality. In machine learning, the environment is typically formulated as a Markov decision process (MDP), as many reinforcement learning algorithms for this context utilize dynamic programming techniques [7]. The main difference between the classical techniques and reinforcement learning algorithms is that the latter do not need knowledge about the MDP and they target large MDPs where exact methods become infeasible. Reinforcement learning differs from standard supervised learning in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected. Instead the focus is on on-line performance, which involves finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge) [8]. The exploration vs. exploitation trade-off in reinforcement learning has been most thoroughly studied through the multi-armed bandit problem and in finite MDPs. In other words, the problem that we are trying to solve with RL should be an MDP or its variant. The theory of RL relies on artificial intelligence (AI) and dynamic programming (DP).

With the help of this technique we are going to successfully implement this in IDPS. It's important to understand the players who make this as possible.

(1) Agent - who interact with the packet
(2) State - Absorb the current state
(3) Action - Based on the state, apply action
(4) Reward- Give some value for the action

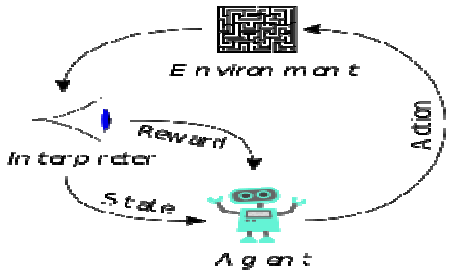

Fig. 2. Reinforcement Learning.

As shown in figure 2, first our IDPS will analyze the state of the packet and then send to the interpreter. Once it is passed to the interpreted it analyze each and every nook of the packet by separating each information of the packet and assign state for each information and then giving reward value for each state. At last our interpreter will decide that what action it should take based on cumulative reward and state of the packet. Here comes the next issue that how this interpreter analyze the state and perform the action. To answer that we choosing Q-Learning algorithm as a solution.

B. Q-Learning

Q-Learning is an Off-Policy algorithm for Temporal Difference learning. It can be proven that given sufficient training under any ε-soft policy, the algorithm converges with probability 1 to a close approximation of the action-value function for an arbitrary target policy. Q-Learning learns the optimal policy even when actions are selected according to a more exploratory or even random policy. A history of an agent is a sequence of state-action-rewards:

$$(s_0,a_0,r_1,s_1,a_1,r_2,s_2,a_2,r_3,s_3,a_3,r_4,s_4\ldots)$$

which means that the agent was in state s0 and did action a_0, which resulted in it receiving reward r_1 and being in state s_1; then it did action a_1, received reward r_2, and ended up in state s_2; then it did action a_2, received reward r_3, and ended up in state s_3; and so on. We treat this history of interaction as a sequence of experiences, where an experience is a tuple (s,a,r,s'). This new data point is called a return. The agent can use the temporal difference equation to update its estimate for Q(s, a):

$$Q[s, a] \leftarrow Q[s, a] + \alpha(r+ \gamma max_{a'} Q[s', a'] - Q[s, a])$$

or, equivalently,

$$Q[s, a] \leftarrow (1-\alpha) Q[s, a] + \alpha(r+ \gamma max_{a'} Q[s', a']).$$

The agent absorbs the current state s' and reward r' and returns an action based on the Q-table that stores action values indexed by state and action. The Q-value updating is done according to the above mentioned update equation. This updating process happens continuously until we reach an equilibrium i.e. we should not see change in the Q-values for two consecutive iterations. With the help of above processed Q-values, we select the action and therefore making user agent to reach its final stage. Through this way Q value tells the agent that what action it should take on time. According to our scenario we limiting our action in three categories such as to (1) Allow, (2) Report and (3) Deny as shown in figure 3.

As shown in figure 2, we can clearly see that our interpreter effectively inspect each and every corner of the packet as like state full inspection. With the help of Q-learning algorithm our interpreter analyzes the state of the packet as well as with whom it needs to communicate in the network. By doing so we can able to handle the packet effectively.

1. controller *Q-learning* (*S,A,γ,α*)
2: **Inputs**
3: *S* is a set of states
4: *A* is a set of actions
5: *γ* the discount
6: *α* is the step size
7: **Local**
8: real array *Q[S,A]*
9: previous state *s*
10 previous action *a*
11initialize *Q[S,A]* arbitrarily
12 observe current state *s*
13:**repeat**
14:select and carry out an action *a*
15: observe reward *r* and state *s'*
16: $Q[s,a] \leftarrow Q[s,a] + \alpha(r+ \gamma max_{a'} Q[s',a'] - Q[s,a])$
17: $s \leftarrow s'$
18: **until** termination

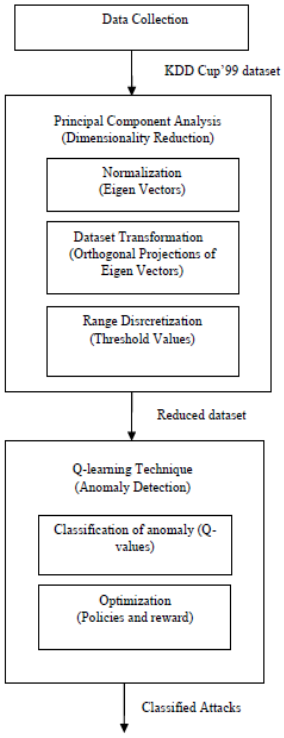

Fig. 3. Block diagram of proposed work.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of our proposed hardware and software requirement are as follows:
We use windows 8 operating system with dual core processor 2.0 GHz processor and 4 GB RAM. For simulation analysis of our proposed method for intrusion detection is executed in MATLAB2012a using KDDCUP99 Dataset.

A. Data Preparation
The KDD99cup data set used for the purpose of experimental research analysis, since we know that KDD 99 dataset has been extensively used for the assessment of network based intrusion detection. In the novel approach they have used KDDCUP99 intrusion detection dataset, which surrounds 26167 records with.50:50 training ratio.
Attack types are four categories:
1. Denial of Service (DoS)
2. Remote to Local (R2l)
3. User to Root (U2R)
4. Probe
In this we use reinforcement learning (RL) and Q-learning (QL) technique to detect the intrusions over network. Where RL and QL are putted extra efforts to optimize best classified of the category until they are not accurately classified. Subsequently, method has been tested on full (41 attributes) dataset as well as in reduced dataset (18 attributes), and used measurement parameters are [9]:

B. Performance Metrics
Sensitivity, specificity and accuracy, and method is compared with SVM, KNN and found that proposed method produced most accurate result into maximum cases. These parameters are defined as follows:

Sensitivity
It is also called the true positive rate, or the recall rate in some fields measures the proportion of actual positives which are correctly identified.

$$Sensitivity = TPR/PR = TPR/(TPR + FNR)$$

Specificity
It is also called the true negative rate, which measures the proportion of negatives which are correctly identified.

$$Specificity = TNR/NR = TNR/(FPR + TNR)$$

Accuracy

$$Accuracy = (TPR + TNR)/(FPR + TNR + TPR + FNR)$$

C. Result Analysis
The GUI environment for the proposed work is shown in figure 4 with the reduced feature of dataset.

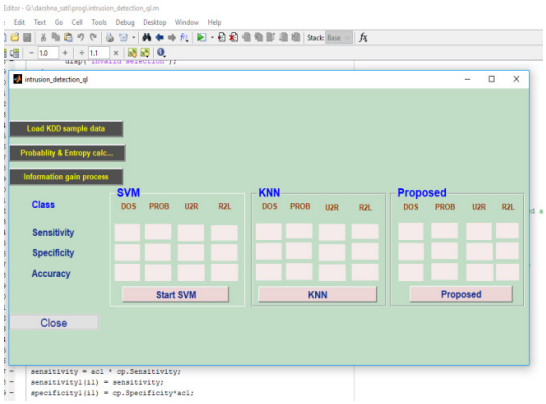

Fig. 4. GUI environment of proposed method.

Here table 1 & table 2 shows the result analysis of sensitivity of 41 attribute & reduced feature set of 14 attribute. In which we found that sensitivity rate of our method is much more efficient than the SVM and KNN method. This comparative analysis of the methods is depicted through chart in fig 5 & fig. 6.

Table 1: Sensitivity (41attribute).

Sensitivity(41attribute)			
	SVM	KNN	Proposed
DOS	82.3951	81.96	85.4312
PROB	82.4357	81.9769	85.4194
U2R	82.1211	81.9681	85.4239
R2L	82.4554	81.9869	85.4435

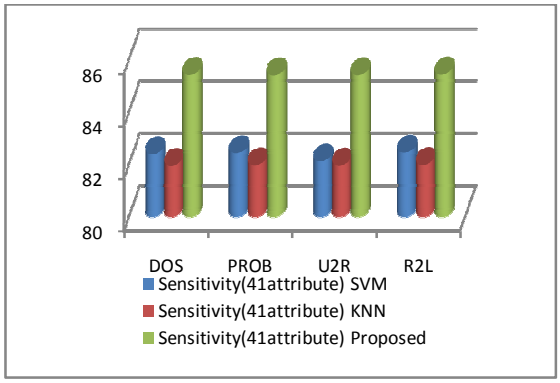

Fig. 5. Sensitivity Comparison of SVM, KNN and proposed work with 41 attribute.

Table 2: Sensitivity of reduced dataset of 14 attributes.

Sensitivity (Reduced 14attribute)			
	SVM	KNN	Proposed
DOS	99.2976	72.0065	98.5327
PROB	99.8726	94.6747	94.6285
U2R	99.0054	98.5136	98.476
R2L	96.9152	98.4609	98.4496

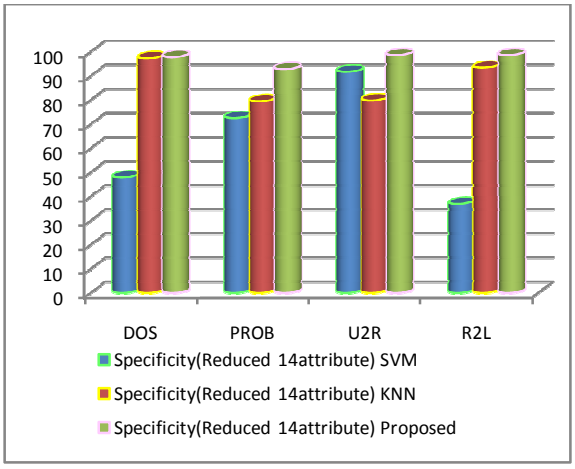

Fig. 6. Sensitivity Comparison of SVM, KNN and proposed work with 14 attribute.

Here table 3 & table 4 shows the result analysis of sensitivity of 41 attribute & reduced feature set of 14 attribute. In which we found that specificity rate of our method is much more efficient than the SVM and KNN method. This comparative analysis of the methods is depicted through chart in fig 5.4 & fig. 5.5. The specificity rate for detection of R2L with 41 attribute is less than the SVM & KNN but our method is more effective to detect DOS, Probe & U2R. With reduced feature set of 14 attribute is effectively identify all categories of attacks.

Table 3: Specificity (41attribute).

Specificity(41attribute)			
	SVM	KNN	Proposed
DOS	82.3762	81.9703	85.4363
PROB	81.3602	81.6014	85.3683
U2R	75.5841	77.435	85.45
R2L	82.4554	69.236	67.2691

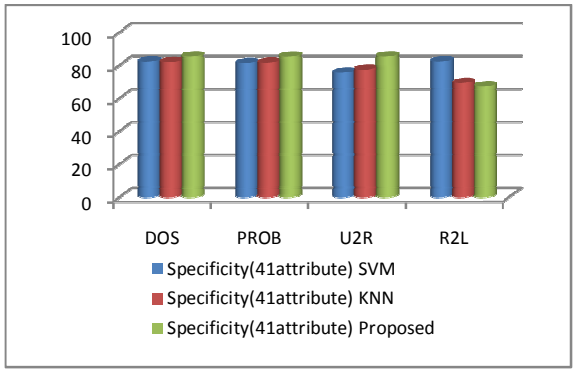

Fig. 7. Specificity Comparison of SVM, KNN and proposed work with 41 attribute.

Table 4 Specificity of reduced dataset of 14 attributes.

Specificity(Reduced 14attribute)			
	SVM	KNN	Proposed
DOS	47.8317	97.256	97.719
PROB	72.4858	79.3794	92.7535
U2R	91.6667	79.7705	98.54
R2L	36.8421	93.3537	98.54

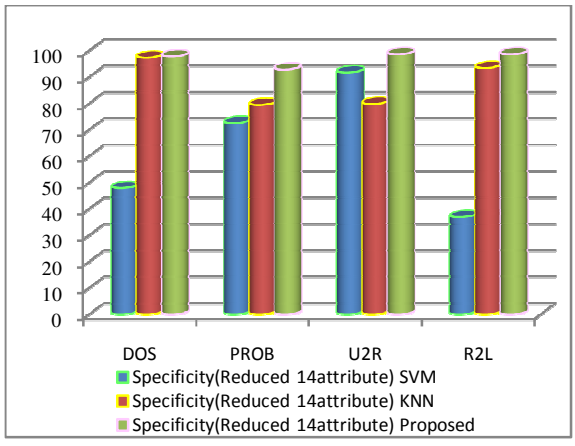

Fig. 8. Specificity Comparison of SVM, KNN and proposed work with 14 attribute.

Here table 5 & table 6 shows the result analysis of sensitivity of 41 attribute & reduced feature set of 14 attribute. In which we found that sensitivity rate of our method is much more efficient than the SVM and KNN method. This comparative analysis of the methods is depicted through chart in fig 9 & fig. 10.

Table 5: Accuracy of reduced dataset of 41 attributes.

Accuracy(41attribute)			
	SVM	KNN	Proposed
DOS	82.3861	81.9649	86.3937
PROB	82.3924	81.9618	85.8673
U2R	82.1151	81.9649	85.9639
R2L	82.4554	81.9649	86.9508

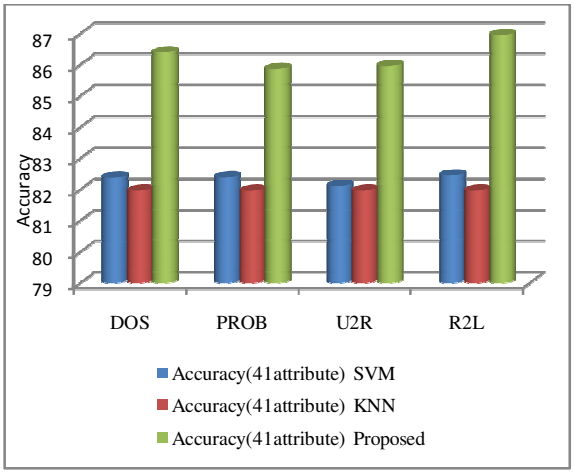

Fig. 9. Accuracy Comparison of SVM, KNN and proposed work with 41 attribute.

Table 6: Accuracy of reduced dataset of 14 attributes.

Accuracy(reduced 14attribute)			
	SVM	KNN	Proposed
DOS	74.7153	79.3381	99.1008
PROB	98.7694	94.6537	95.0773
U2R	98.9987	98.4986	98.5016
R2L	96.8279	98.4572	99.0996

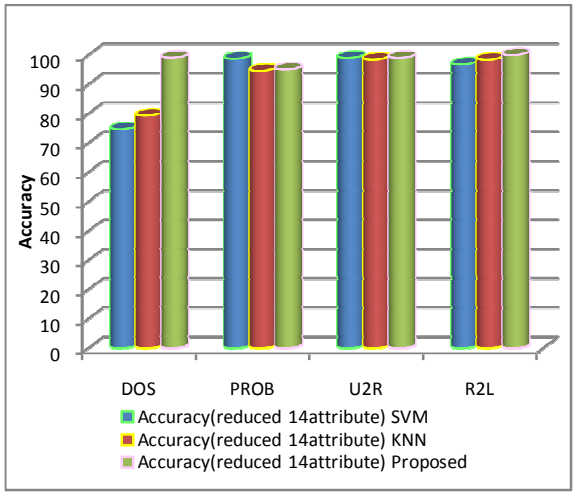

Fig. 10. Accuracy Comparison of SVM, KNN and proposed work with 14 attribute.

CONCLUSION & FUTURE WORK

These days security defiance becomes critical issues and to preserve is so many approaches has been developed. A system is designed to identify the intruders over the network which is IDS; it helps in improving the detection accuracy as well as reduces the manpower required in monitoring. In this paper, we propose reinforcement learning (RL) and Q-learning (QL) method. Firstly select the dataset from the KDD sample data on the basis of information gain, probability and entropy etc. Then the reduced attribute is chosen for training or testing set which required. If trained set of data sample is detected normal then put into normal class otherwise apply Q-learning to efficiently categorize different set of intrusions. The effectiveness of our method is depicted using KDDCUP99 dataset and the performance evaluation of the propose method is perform using performance metrics sensitivity, specificity and accuracy. The simulation result of our method is much more efficient than the other existing techniques such as SVM and KNN. In future work, develop such techniques which is also able to detect R2L attack together with DOS, U2R and Probe.

REFERENCES

[1]. Vipin Das, Vijaya Pathak, Sattvik Sharma, Sreevathsan, MVVNS. Srikanth, Gireesh Kumar T "Network Intrusion Detection System Based On Machine Learning Algorithms", *International Journal of Computer Science & Information Technology (IJCSIT),* Vol. **2**, No 6, December 2010.

[2]. Richard Sutton and Andrew Barto "Reinforcement Learning: An Introduction", MIT Press, 1998.

[3]. R. Arunraj, C. Umarani, "A Study on Applying Reinforcement Learning in Intrusion Detection and Prevention System", *Recent Advances in Technology and Engineering (RATE-2017) 6th National Conference by TJIT, Bangalore.*

[4]. Dr. R Ravinder Reddy " Network Intrusion Anomaly Detection Using Radial Basis Function Networks", *International Journal of Advanced Research in Computer Science*, Volume **8**, No. 3, March – April 2017, ISSN No. 0976-5697.

[5]. Arturo Servin and Daniel Kudenko "Multi-Agent Reinforcement Learning for Intrusion Detection", In proceeding of Springer, 2008.

[6]. Patrick Miller and Atsushi Inoue "Collaborative Intrusion Detection System", *In proceeding of IEEE, 2003.*

[7]. Van Otterlo, M.; Wiering, M. "Reinforcement learning and Markov decision processes", Reinforcement Learning. Springer Berlin Heidelberg (2012): 3–42.

[8]. Kaelbling, Leslie P.; Littman, Michael L.; Moore, Andrew W. "Reinforcement Learning: A Survey", *Journal of Artificial Intelligence Research.* **4** (1996): 237–285.

[9]. Dokas P., Ertoz I., Lazarevic, A. Shrivastava, "Data mining for network intrusion detection", *Proceeding of NGDM, pp-21-30, 2002.*