Immunity inspired Cooperative Agent based Security System

Praneet Saurabh and Bhupendra Verma
Department of Computer Science and Engineering, Technocrats Institute of Technology, India

Abstract: *Artificial Immune System (AIS) has evolved substantially from its inception and is utilized to solve complex problems in different domains out of which computer security is one of them. Computer Security has emerged as a key research area because of the ever-growing attacks and its methodology. Various security concepts and products were developed to overcome this alarming situation but these systems by some means fall short to provide the desired protection against new and ever-increasing threats. AIS enthused from Human Immune System (HIS) is considered as an excellent source of inspiration to develop computer security solution since the previous protect the body from various external and internal threats very effectively. This paper presents Immunity Inspired Cooperative Agent based Security System (IICASS) that uses Enhanced Negative Selection Algorithm (E-RNS) which incorporate fine tuning of detectors and detector power in negative selection algorithm. These features make IICASS evolve and facilitate better and correct coverage of self or non-self. Collaboration and communication between different agents make the system dynamic and adaptive that helps it to discover correct anomalies with degree of severity. Experimental results demonstrate that IICASS show remarkable resilience in detecting novel unseen attacks with lower false positive.*

Keywords: *Anomaly, human immune system, artificial immune system, agent.*

Received June 3, 2014; accepted May 24, 2016

1. Introduction

Computers and applications running on them have advanced significantly with age to facilitate users. The pervasiveness of computer network is of great importance because of ease and convenience which it brings to almost all the verticals of life. This importance and attached value attracts security concern. Potential chances of misuse and abuse always exist because it is exposed to many unintended recipient, who can take advantage.

The essence of network and computer security to keep information, secure from non-intended recipients, to preserve its integrity and availability at the same time [6, 10]. CSI survey [11] reported big jumps in different suspicious activities over internet. These points indicate and highlight the challenges and the gap between the existing and the desired solution in highly dynamic, unorganized, imperfect, uncontrolled and open network environment.

Biological systems have served as the driving force behind various computational learning systems (e.g., artificial neural networks and genetic algorithms) in the recent past [2]. Artificial Immune System (AIS) models the principles and processes of the biological immune system which enables all organisms to survive from the various threats and challenges [4]. The way Human Immune System (HIS) reacts and give response against different new attacks encourage and form base for researchers to model computer security system after it [5]. AIS use these theories to develop algorithms that help to solve problems in the domain of computer security.

Agent technology presents a new computing paradigm as agents are computational entities that act by coordinating, contributing and delegating tasks to other entities [13]. A Multiagent system (MA) has a number of agents that can interact and coordinate their actions in order to complete any task [9].

This paper presents Immunity Inspired Cooperative Agent based Security System (IICASS) that evolves to demonstrate capability of self learning and adaptive to new challenges. Section 2 contains viewpoint of AIS and Agents in the canvas of computer security. Section 3 presents Enhanced Negative Selection Algorithm (E-RNS), section 4 details the proposed whole system overview. Experimental results and discussion is covered in section 5 and section 6 concludes the discussion.

2. AIS and Agent perspective in Computer Security

Biologically inspired computational model offers a wide range of techniques and methods that can be used to develop a computer security solution. Biological Immune System (BIS) has ability to detect pathogens that it has never encountered. In context of computer security it performs anomaly detection [2, 3]. AIS model traits of BIS and thereafter use these qualities to design and develop a self-adaptive, self regulatory and distributed security solution.

2.1. Artificial Immune System

AIS are the systems conceptualised from the principles and processes of BIS [4]. BIS is a successful classification system which differentiates between self and non-self [5]. Immune system is constituted by central lymphoid whose purpose is to generate and mature immune cells. Lymphocytes are constantly generated by bone marrow, matures in thymus. Thymus releases only matured and beneficial T-cells to the blood stream and discards the remaining ones. These matured T-cells behave in the manner of a detector and identifies the invading antigens and takes suitable and appropriate measures [6].

2.2. Negative Selection Paradigm

Forrest *et al*. [5] proposed negative selection which revolves around immune system's ability to identify unknown antigens or non-self while not reacting to self-cells. It is shown to be efficient towards anomaly detection problem. Later on different variations in negative selection algorithms have introduced [6], but the crux remains same which is to build self profile, by recognising normal network patterns as self and other patterns as non-self. With reference to this built profile the non self patterns are very easily identified and marked as non-self or anomalous.

Negative selection algorithm is represented through various methods, most commonly used representations are either binary or string representation [1]. The real-valued representation is another very interesting representation in which detectors and antigens are represented as real valued vectors.

2.3. Agent Technology

Spirit of agents lies in the fact that it can move from one host computer to another while suspending its execution from the earlier one and can resume from there [14]. A multiagent system consists of several agents that interacts and coordinate their actions inorder to complete assigned task [15]. Apart from introducing distributedness it also brings in the features of adaptability, communication, intelligence and learning ability.

3. Enhanced Real Valued Negative Selection Algorithm

The most critical issue in negative selection algorithm is detector coverage of self and non-self area. It is an optimization problem and many works have focused on it. However it still remains to be realized efficiently and correctly. Many of the previous works generated detectors of small and of the same size to cover self/ non-self space due to this some of self/ non-self space is not covered or covered with holes represented in Figure 1 also called problem of similarity.

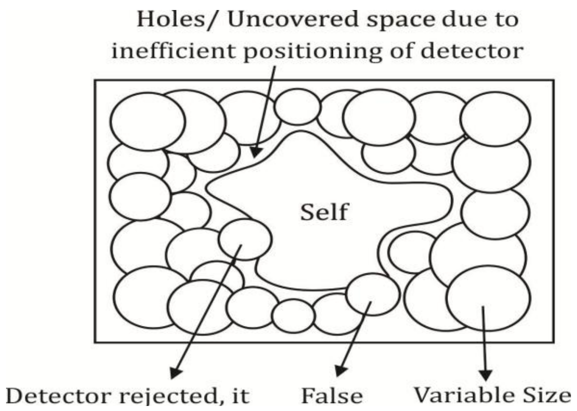
Holes/ Uncovered space due to inefficient positioning of detector
Self
Detector rejected, it
False
Variable Size

Figure 1. Problem of uncovered space.

This section presents Enhanced Real Valued Negative Selection Algorithm (E-RNS) with feature of fine tuning. Fine tuning enables the detectors to cover self and non-self space efficiently and correctly. During detector generation process E-RNS uses real value random number generator to generate unique dissimilar random numbers to generate detector (d). This unique value overcomes problem of similarity. In next step of detector selection, (d) is compared with the instances of Training Set (TS), and if the difference between detector (d) and TS instance is less than binding threshold (B_T) given by $d(dx_i, TSy_i) <$ B_T; then detector is not discarded but it is kept in the Weak Detector Set (WDS).

E-RNS incorporates fine tuning with tuning factor (T_V) in this stage. Detector (d) in WDS is altered by a factor of ‘V’ so that it becomes unlike and $d\ (dx_i, TSy_i) <$ B_T becomes false. Besides this feature E-RNS introduces detector power (d_P) as an additional attribute. It is a relative attribute which means strength of detector (d) to match with an antigen. Final value of detector (d) is determined after comparisons with all the instances of TS. Since detector (d) is not discarded after just one comparison, there are more comparisons with TS. Each negative comparison help detector (d) in increasing power of the detector (d_P). Furthermore if some detector covers self space then fine tuning maks it dissimilar, so that wrong and overlapping coverage is reduced. Self tuning makes a detector to cover as much as and correct space as possible and reduce chances of false positive. Finetuning achieves dissimilarity among the detectors, attribute d_P makes a detector more efficient to detect anomaly. Both these mechanism facilitate detectors to cover more self and non-self space correctly and minimizes computational cost. E-RNS forms base for IICASS for detector selection and anomaly detection.

4. Immunity Inspired Cooperative Agent based Security System

This section presents IICASS. IICASS combine E-RNS with collaborative agents. Binding of these techniques enable IICASS to generate efficient detectors and identify anomalies proficiently and correctly.

4.1. System Design and Components

IICASS has manger agent that has detector generator module to create random detectors and detector Selector module to select detectors. Detector Selector module works on the principle of E-RNS to select the best and fine tuned active detectors to form Detector Set (DS). Detection Agent have two modules, detector agent creator module to create agents and detector agent module to detect the anomalies. Agents have DSin quest to find anomalies when Detector Agent comes across Test dataset (TeS). Preprocessing Module (PPM) looks after dimension reduction in dataset used for training and testing.

4.2. Preprocessing Module

KDD Cup 99 dataset is most common, widely acceptable and recognised dataset [7]. Although it is released in year 2000, but still it is used extensively as a standard dataset for anomaly detection in computer security. Various researchers used it to train and verify their findings [12]. The complete dataset have about 5 million records and each record represents a TCP/IP connection that is composed of 41 features which are both qualitative and quantitative in nature plus a label of either “normal” or “attack.” This module engages two key concepts, Principal Component Analysis (PCA) for dimension reduction [8] in KDD Cup 99 dataset and Min Max Normalisation to convert the reduced features into computationally relevant form.

PCA is applied on Self Set (S) that contains all vectors of KDD cup dataset. Training Set (TS) include only normal records. It yields the most significant principal components called network features (f_1, f_2,…….f_5). These obtained network features (f_1, f_2,…….f_5) are spread in a wide range (e.g., -32322 to 54334), and requires large space for representation. This output contributes in computational complexities. Min-Max Normalization is used to overcome this limitation. It normalizes principal components network features (f_1, f_2,…….f_5) in the range (0, 1). Pre-processing configuration is saved to be used on Test Set (TeS) too.

4.3. Manager Agent

Manager Agent (MA) comprises of Detector Generation Module (DGM) and Detector Selection Module (DSM). The main goal of DGM is to generate detectors. DSM has the task of selecting fittest and powerful detectors to form detector set.

4.3.1. Detector Generation Module (DGM)

Detector generation is a search problem, as the purpose of any individual detector is to cover up the whole of the non-self space in order to identify an anomaly. A detector (d) in IICASS is of variable value and generated by using a random real number generator engine that describes any value in the shape space. Detector generation aims to accomplish two goals, first to maximize the coverage of the non-self subspace with an effort to lessen the uncovered spaces and second to minimize the coverage of the self samples. DGM uses Real value random number generator to generate unique dissimilar random numbers within a specified range [0.0-1.0] to achieve above said goals. Based on this unique number detector generator engine generates detector (d) with an additional attribute detector power (d_P), which is its strength to match with an antigen. d_P is a relative parameter and its final value is determined after comparisons with all the instances of Self S. Non similar values by real value random number generator help in achieving uniqueness of detector and also overcomes problem of similar detectors and uncovered space highlighted in Figure 1.

Algorithm 1: Detector Generation & Selection

```
Input: S = Preprocessed Self Set
Input: TS = Training Set
Input: $D_N$ = Number of Detector that needs to be generated
Input $d_P$ = Power of detector
Input: B_T= Binding Threshold
Input: P_T= Power Threshold
Input: $T_V$= Tuning constant
Output: Detector Set DS
Do until $D_N$ achieved
    Generate Detector d;
    for all S in self set do
        CALCULATE A= DISTANCE ($x_i$, d)
        if A > B_T
            INCREASE $d_P$ by 1
        else
            ADD (d, WDS)
            DECREASE $d_P$ by 1
            MANIPULATE detector '$T_V$'
        end else
        end if
    end for
    if ($d_P$ > P_T)
        ADD (d, DS)
    else
        discard d
    end else
    end if
end do
```

4.3.2. Detector Selector Module (DSM)

This is the second module of manger agent, it uses the E-RNS to select the best and efficient detectors to form active DS out of the generated detectors depicted in Figure 2.

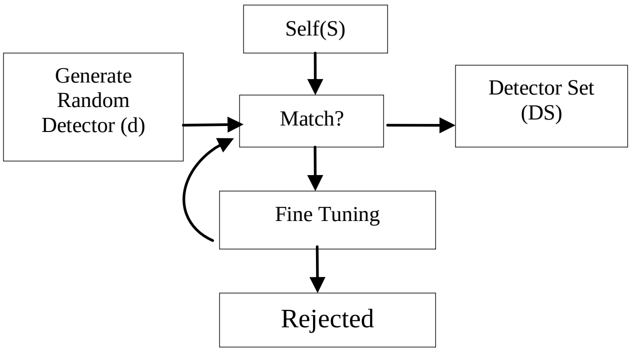

Figure 2. Representation of generation and selection of detectors in E-RNS.

Detector selection is an optimization problem as the task is to select the best detectors from the generated ones that DS to detect anomalies. Generated detectors (d) are compared with all the instances of TS. Binary representation of detectors employs rcb (r-contiguous bit), r-chunks, and Hamming distance as matching rules while Euclidean distance is used for matching in real-valued representation of detectors. It tells how much the d is close to instances of TS by using formula:

$$D(dx_i, TSy_i) = \sqrt{\sum_i (dx_i, TSy_i)^2} = \| x_i, y_i \| \quad (1)$$

d(dx$_i$, TSy$_i$) in Equation (1) represents the closeness between detectors (d) and any pattern of TS. In immunological terms it is called binding of an antibody, which is the power it boasts to bind with an antigen. Small difference between an antibody and an antigen characterize strong binding between them. If distance D (dx$_i$, TSy$_i$) is less than binding threshold; then detector is not discarded but it is kept in a WDS. Fine tuning introduced in IICASS at detector selection stage fine-tunes detector (d) through a variation of (T_V) and makes it dissimilar so that it is selected to DS if it has gathered enough detector power (d_P). Each negative comparison with Self (S) increases detector power (d_P), and if $d_P > P\_T$ is true then only detector (d) is selected to DS. These selected and fine tuned detectors are kept into DS which are later associated with the Detector Agent (DA) to identify anomalies. The tuning factor 'T_V' self configures d and helps it in to adapt dynamically with respect to the detector's closeness towards the self-sample while (d_P) makes detector efficient. These new features enable detectors to efficiently cover uncovered space.

4.4. Detection Agent

Detection Agent has two agent modules; one is Detector Agent Creator Module (DACM) and second is Detector Agent Module (DAM). The first one looks after creating detector agents which is assigned the DS and the second one has the goal to detect the anomalies when the detector agents comes across the test dataset TeS.

4.4.1. Detector Agent Creator Module (DACM)

This module creates Detector Agents (DA$_1$, DA$_2$,…DA$_N$) which are equipped with Detectors set (DS$_1$, DS$_2$…..) having fittest detectors. These agents are later deployed to detect the anomalies.

DACM create Detector Agents (DA$_1$, DA$_2$) and associates it with the latest and fittest detector sets (DS$_1$, DS2...., DS$_N$) containing detectors (d$_1$, d$_2$, d$_3$, …., d$_n$). DACM clones agents to create new ones whenever required. It also holds the feature to destroy the older agents to control the population. The new cloned agents are associated with a new Detector Set (DS) having more recent and fine tuned detectors.

Algorithm 2: Anomaly Detection.

```
Input: RS = Rule Set
Input: $A_n$ = Number of Agents that needs to be assigned
Input: DS= Detector Set
Input: TeS= Test Set,
Input: A_T= Affinity Threshold=0
Input: Al_T= Alert Threshold=0
ASSIGN DS and RS to DA
For all DA(DS, RS)$_{1 \text{ to } k}$ for all packets in TeS$_{1 \text{ to } k}$
    if DA$_K$ detects TS$_K$ as Anomaly
        increase Vote by 1
    end if
    if (Vote >= Al_T)
        ANOMALY
    else
        NORMAL
    end if
end for
```

4.4.2. Detector Agent Module (DAM)

It is the second module of the detection agent. It detect anomalies through the DA efficiently in TeS and report detection of anomalies with an Alert level (Al_T) to the manager agent. Detector agents (DA$_1$, DA$_2$…) having latest detector sets (DS$_1$, DS$_2$…) are compared with the samples of TeS for anomaly detection. TeS is preprocessed with the same configuration setting that are used for detector generation and selection. Concept of voting is also incorporated in this module.

Each DA uses its own DS compromising of detectors (d$_1$,d$_2$,d$_3$,….d$_n$) to detect the anomalies in TeS Euclidean similarity measure is calculated between D of DS and 1 to kth element of TeS. If similarity measure is greater than affinity threshold (A_T) predefined threshold, then test sample instance is labeled as an anomaly and DA increases vote count 'V' by 1. Information about this test packet with a vote count 'V' is sent to the other detector agents. Other DA also compares the test sample with its own DS and if the new detector agent also finds it an anomaly, it contributes to its vote count 'V'. This process is repeated till all DA test all TeS instances. If final vote-count 'V' for a packet increases over the Alert Threshold (Al_T), then the packet is finally classified as an anomaly and an alert signal with different priorities is sent to the manager agent with comprehensive information.

5. Experimental Results and Discussion

This section contains the experimental results and discusses the performance and the potential advantages of proposed IICASS over RNS with constant detectors. All the experiments for both the methods use KDD Cup dataset for training and testing purposes. TS is composed of 972,781 normal records while TeS is composed of 5000 randomly selected unseen data, which includes both normal and attack data. All the results are the average of 40 runs on the same configuration. Detection Rate and False Alarm are the two parameters that define the efficiency of detection system, detection rate notify the anomaly and false alarm rate of the detection system generates an alarm in normal conditions. High detection rate and low false alarm rate are prerequisites for any detection system.

5.1. Calculations Involved:

Following measures are used to compute the performance of the IICASS.

1. Detection Rate $(DR):\frac{TP}{TP+FN}*100$ (2)

2. False alarm rate $(FAR):\frac{FP}{TN+FP}*100$ (3)

The results in Table 1 and Figure 3 illustrate effect of training samples in detection when it is varied from 25% to 100%. Against this variation in training sample the relative average detection rate and false alarm of RNS and IICASS is calculated and their respective curves for Detection Rate and False Alarm Rate are generated for the same test set. d_P is 0.7 while the T_V remains 1. Detectors of both the methods are trained with same percentage of training sample and then tested with same test set with variation in the affinity threshold for detection. From the results it is observed that affinity threshold is an important parameter to determine detection rate and false alarm rate.

Table 1. Affect of training data and affinity threshold.

Training Data	Affinity Threshold	Method	Detection Rate	False Positive
25	0.3	RNS	100	100
	0.3	IICASS	82.72	0.71
	0.4	RNS	100	100
	0.4	IICASS	100	0.8
50	0.3	RNS	91.33	0.75
	0.3	IICASS	91.97	0.72
	0.4	RNS	100	1.12
	0.4	IICASS	100	0.8
100	0.3	RNS	90.5	0.71
	0.3	IICASS	92.9	0.67
	0.4	RNS	91.7	0.72
	0.4	IICASS	100	0.69

Lower value of self samples for training indicates that limited information is available to generate the self profile but IICASS successfully overcomes this limitation and creates the full profile. Due to this reason IICASS achieves high detection rate with lower false alarm rate. It also maintains stability in the results as compared to RNS. RNS in some cases of lower affinity threshold have very high detection rate but false alarm rate is also on the higher side and at occasions it becomes unstable, whereas IICASS gets high detection rate and low false alarm rate, which are the main objectives of any detection system. Results also represent the fact that that as affinity threshold increases false alarm comes down in IICASS but at many instances RNS become unstable and detects normal as anomaly by covering false non-self region which contributes in higher false alarm rate.

Figure 3 reflects the detection rate and false alarm rate of RNS and IICASS when trained by 100% of the samples and measured under different detection thresholds. Detection Rate of RNS remains lower to the curve of IICASS against all the variation in the detection threshold.

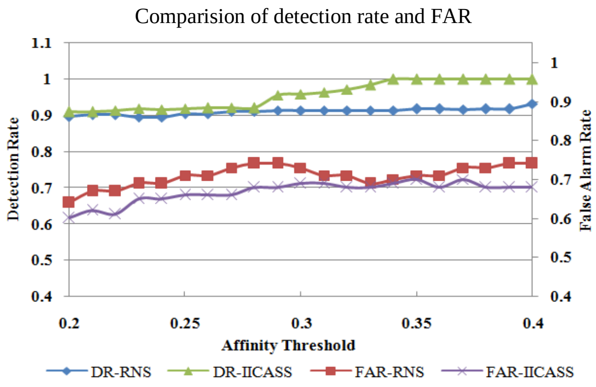

Figure 3. Affinity threshold vs detection rate vs FAR

Also FAR of RNS is higher when compared FAR of IICASS. It clearly indicates that under all the values and test condition IICASS attain better results and achieves a Detection Rate of 100% in some cases, with only 0.69% false alarm rate in the best case. It reflects in high detection rate and low false alarm rate. This ascertain the fact that IICASS adapts extremely well and non similar detectors accompanied with theory of fine tuning and voting achieve both the goals of high detection rate and lower false alarm rate.

Table 2. Affect of alert threshold

Alert Threshold	Scenario	Agents with detection threshold	False Alarm Rate
1	I	0.2, 0.3, 0.4	0.62
	II	0.25, 0.35, 0.45	0.72
2	I	0.2, 0.3, 0.4	0.58
	II	0.25, 0.35, 0.45	0.66
3	I	0.2, 0.3, 0.4	0.53
	II	0.25, 0.35, 0.45	0.59

The results in Table 2 are in continuation with the first results. IICASS introduces Al_T to lower false alarm rate. It achieves this task through collaborative agents aided different DS assigned to different DA for detection. Detectors are trained by 100 % of the self samples. The fine tuning constant of the detectors remains 1. Three different DS are assigned to DA, and then alert threshold is changed from 1 to 3. Furthermore value of alert threshold represent that how many votes are required to term any record as an anomaly, by different DA using dissimilar DS. This method of voting between agents helps in identifying any record in the TeS as an anomaly more precisely and results into lowering false alarm rate. All experiments are performed on two different scenarios; Scenario 1 has detectors sets with detection threshold 0.2, 0.3, 0.4 assigned to Agents 1, 2, 3 and then detection rate and false alarm rate is calculated. As Al_T increases false alarm rate comes down this indicates that voting between agents enable them to cover more self space correctly. In scenario 2 agents are equipped with detector sets with detection threshold 0.25, 0.35, 0.45, and again as the number of votes to term any record as an anomaly increases then false alarm falls down quite considerably, this lowering of false alarm signify that agents are covering the self space as self and not identifying it as an anomaly. Results in Table 2 very clearly points out that the proposed IICASS performs well in lowering the false alarm rate by adapting well and creating inclusive self profile. Low FAR contributes to effective detection of anomalies.

6. Conclusions

This paper presents IICASS which incorporated characteristics of agents and AIS to build correct and appropriate profile of self and non-self by using limited information. Fine tuning and voting powers IICASS to have self learning and adaptive capabilities that reflect in high detection rate with low false alarm rate for new and unseen TeS. Furthermore under different conditions and no matter what would be the training and TeS IICASS remains stable. Collaboration and communication between agents make IICASS distributed, robust, autonomous, adaptive and self configuring with self learning capabilities. Experimental results firmly illustrate that IICASS adapts well and reconfigures its profile to recognize self and non-self space effectively and efficiently with high detection rate and low false alarm rate for both existing and new unseen anomalies.

References

[1] Ayara M., Timmis J., Lemos R., Castro D., and Duncan R., "Negative Selection:How to Generate Detector," *in proccedings of 1st International Conference on Artificial Immune System*, UK, pp. 182-196, 2002.

[2] Dasgupta D., "Immunity-based Intrusion Detection System: A General Framework," *in procceding of 22nd Information Systems Security Conference*, pp. 147- 160, 1999.

[3] Denning E., "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222-232, 1987.

[4] Forrest S., Hofmeyr S., and Somayaji A., "Computer Immunology," *ACM Communications*, vol. 40, no. 10, pp. 88-96, 1997.

[5] Forrest S., Perelson A., Allen L., and Cherukuri R., "Self-Nonself Discrimination in a Computer," *in proceedings of IEEE Symposium on Research in Security and Privacy*, Oakland, pp. 202-212, 1994.

[6] Kim J. and Bentley P., "The Human Immune System and Network Intrusion Detection," *in proceedings of 7th European Congress on Intellegent Techniques and Soft Computing (EUFIT'99)*, pp. 1244-1252, 1999.

[7] KDD Cup. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Last Visited 1999.

[8] Lin S., Ying K., Lee C., and Lee Z., "An Intelligent Algorithm with Feature Selection and Decision Rules Applied to Anomaly Intrusion Detection," *Applied Soft Computing*, vol. 12, no. 10, pp. 3285-3290, 2012.

[9] Ou C., "Host-based Intrusion Detection Systems Adapted from Agent-based Artificial Immune Systems," *Neurocomputing*, vol. 88, pp. 78-86, 2012.

[10] Overill E., "Computational Immunology and Anomaly Detection," Information Security Technical Report, 2007.

[11] Richardson R., *CSI Computer Crime and Security Survey*, Computer Security Institute, 2011.

[12] Sereshtn N. and Reza A., "MAIS-IDS: A Distributed Intrusion Detection System using Multi-agent AIS Approach," *Engineering Applications of Artificial Intelligence*, vol. 35, pp. 286-298, 2014.

[13] Sobh T. and Mostafa M., "A Cooperative Immunological Approach for Detecting Network Anomaly," *Applied Soft Computing*, vol. 11, no. 1, pp. 1275-1283, 2011.

[14] Yang J., Liu X., Li T., Liang G., and Liu S., "Distributed Agents Model for Intrusion Detection Based on AIS," *Knowledge-Based Systems*, vol. 22, no. 2, pp. 115-119, 2009.

[15] Zhang P. and Tan Y., "Immune Cooperation Mechanism Based Learning Framework," *Neurocomputing*, vol. 148, pp. 158-166, 2015.

Praneet Saurabh has obtained his B.Tech (CSE) from IIIT-Kolkata, Viswa Bharti in 2004 and M.Tech (CTA) from SOIT, Bhopal in 2007. He is a Ph.d student at RGPV, Bhopal and working as Assistant Professor in Department of Computer Science and Engineering at TIT, Bhopal. He has published more than 10 research papers in different journals and conferences. His area of research includes Computer Security, Evolutionary Computation and Mobile Adhoc Networks.

Bhupendra Verma has done B.E and M.Tech in Computer Science and Engineering from SATI, Vidisha, M.P., India. He has completed his Ph.D. in Computer Science and Engineering from RGPV Bhopal in 2008. He is working as Director TIT (Excellence), Bhopal. He has published 52 research papers in journals and conferences. His area of research includes but not limited to Artificial Intelligence, Soft Computing, Computer Security, Evolutionary Computation, Human computer Interaction.