IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 25, 2020, accepted February 6, 2020, date of publication February 17, 2020, date of current version February 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973608

Immune System Based Intrusion Detection System (IS-IDS): A Proposed Model

INADYUTI DUTT[1], SAMARJEET BORAH[1], AND INDRA KANTA MAITRA[2]
[1]Department of Computer Applications, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majhitar 737136, India
[2]Controller of Examinations Department, St. Xavier's University, Kolkata 700160, India

Corresponding author: Inadyuti Dutt (inadyuti@gmail.com)

ABSTRACT This paper explores the immunological model and implements it in the domain of intrusion detection on computer networks. The main objective of the paper is to monitor, log the network traffic and apply detection algorithms for detecting intrusions within the network. The proposed model mimics the natural Immune System (IS) by considering both of its layers, innate immune system and adaptive immune system respectively. The current work proposes Statistical Modeling based Anomaly Detection (SMAD) as the first layer of Intrusion Detection System (IDS). It works as the Innate Immune System (IIS) interface and captures the initial traffic of a network to find out the first-hand vulnerability. The second layer, Adaptive Immune-based Anomaly Detection (AIAD) has been considered for determining the features of the suspicious network packets for detection of anomaly. It imitates the adaptive immune system by taking into consideration the activation of the T-cells and the B-cells. It captures relevant features from header and payload portions for effective detection of intrusion. Experiments have been conducted on both the real-time network traffic and the standard datasets KDD99 and UNSW-NB15 for intrusion detection. The SMAD model yields as high as 96.04% true positive rate and around 97% true positive rate using real-time traffic and standard data sets. Highly suspicious traffic detected in the SMAD model is further tested for vulnerability in the AIAD model. Results show significant true positive rate, closer to almost 99% of accurately detecting the file-based and user-based anomalies for both the real-time traffic and standard data sets.

INDEX TERMS Computer networks, computer security, intrusion detection, immune system, anomaly detection, network, T-cell, B-cell, innate immune system, adaptive immune system.

I. INTRODUCTION

Security of internet and intranet facilities are at continual risk due to the over reliance of government, military and commercial bodies on them for their day to day activities. Both the internet and intranet facilities face attacks from the network. An intrusion can be defined as an attempt to gain unauthorized access to network resources [1]. Intrusion detection system has become phenomenal in detecting attacks from both outside and inside network. An Intrusion Detection System (IDS) is a software or tool which can address attacks borne from either internet or intranet facilities. It can monitor, log the network traffic and apply detection algorithms for detecting intrusions within the network. An IDS can be categorized into two categories based on the type of model being used: signature-based IDS and anomaly-based IDS. In signature-based IDS, the pre-defined signatures of the attacks are stored in a database and the network is monitored against these existing signatures. In anomaly-based IDS, the network traffic is monitored and compared against the normal usage patterns of the network. Any deviation from the normal usage patterns are considered as an intrusion attempt. Anomaly-based detection can detect new attacks whilst the signature-based detection cannot detect new attacks that are not pre-defined.

Anomaly-based detection was originally proposed by Denning [2] and since then it has gained immense popularity in detecting new attacks using various methods including bio-inspired approaches. Nature and natural organisms have always inspired researchers in the field of network security. Bio-inspired computing mimics the behavior of nature and natural organisms. Immune-inspired network security has become popular due to its marked resemblance with the natural Immune System (IS).

A. IMMUNE SYSTEM (IS)

The natural Immune System (IS) protects our body from harmful invasion of pathogens like virus, bacteria or parasites. The natural immune system is a multi-layered system with the innate immune system as the external and adaptive immune system as the internal layers respectively.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

1) INNATE IMMUNE SYSTEM (SKIN)

Skin acts as the most important part of the innate immune system. It has natural Keratinocytes (KCs) that are involved in sensing pathogens and danger signals [3]. KCs form the central skin sentinels that recognize foreign bodies or pathogens. Pathogens form a microbial structure known as Pathogen Associated Molecular Patterns (PAMPs). KCs recognize these PAMPs using their own Toll-Like Receptors (TLRs) and produce inflammatory responses. These inflammatory responses generate threat signals to the immature dendritic cells (DC).

When an immature dendritic cell takes up a pathogen in infected tissue, it becomes mature activated dendritic cell. This pathogen specific mature dendritic cell has specific protein called MHC (Major Histocompatibilty Complex) on its surface. These secrete cytokines and chemokines that influence the activation of the naïve T-cells of adaptive immune responses. Naïve T-cells further produce armed effector T-cells on encountering the activated dendritic cell with pathogen specific protein called Protein: MHC complex.

2) ADAPTIVE IMMUNE SYSTEM

Naive T cells that recognize their antigen on the surface of a dendritic cell cease to migrate and embark on the steps that generate armed-effector cells. Armed-effector T cell recognizes the pathogen–specific MHC complex; it releases more effector molecules CD8 and CD4 that bind more strongly to the target pathogen cells. CD8 T cells are of two types (CD α and CD β) T cells that are responsible for killing the target cells most notably viruses, bound to MHC class I molecules at the cell surface. CD4 T cells express TH1 cells and TH2 cells, on the cell surface bound to MHC class II molecules. TH1 cells from CD4 activate macrophages, enabling them to destroy intracellular microorganisms more efficiently. TH2 cells, on the other hand, initiate B-cell responses by activating naive B cells to proliferate and secrete important antibodies. Therefore, TH2 cells are responsible for activating the B cells. Signals from the pathogen bound cells induce the B cell to proliferate and differentiate into a plasma cell secreting specific antibody. Table 1 shows these similarities with the proposed system.

In this research work, an unsupervised anomaly detection model has been devised where relevant features from header and payload potions are considered for effective detection of intrusion. The entire work has been segregated into five sections, where section I is introduction, section II contains related works from literature, section III illustrates the proposed model and section IV highlights experimental set-up and results. Finally, section V concludes the works which is followed by the references.

II. LITERATURE REVIEW

IDS form part of any complete security system and their goal is to detect any action that violates the security policy of a computer system. There are mainly two detection approaches

TABLE 1. Similarity between natural immune system and intrusion detection system.

Natural Immune System	Intrusion Detection System
Skin (Innate Immune System) A. Keratinocytes (KCs) B. Dendritic cells (DCs)	*First Layer of defense* A. Pre-processing–I module (ANOVA) B. Pre-processing–II module (3-Sigma)
Adaptive Immune System	***Second Layer of Defense***
T- cells i) CD8 T cells a. CD α T cells b. CD β T cells ii) CD4 T cells a. TH1 type T cell b. TH2 type T cell	T- cells i) T-cell_User_Activation_Module a. Off Hours Access Detection b. Remote Access Detection ii) T-cell_File_Activation_Module a. File Anomaly Type Detection Module b. File Parsing Detection Module
B-cells i) Plasma cells	B-cells i) B-cell_Activation_Module a. Alert Generation Module

that IDS can acquire namely, the misuse detection approach and anomaly detection approach. The misuse detection approach which is used in systems such as MADAM/ID [4] had utilized machine learning techniques to label the data. The classifier learns from a set of labeled connections, where there is normal traffic and attacks, and in subsequent use it recognizes known attacks. This detection approach has two problems. The first problem is to identify and obtain complete labeled network traffic and the second problem is to detect new attacks. Therefore, this method cannot solve the ''zero-day'' problem and as a result newer attacks would succeed in compromising the security of the system.

The second approach of handling intrusion is by anomaly detection, primarily proposed by Denning [2]. The main idea is to profile normal network traffic behavior and observe the deviation from the normal for the real-time traffic. The problem with this approach is to capture all the kinds of normal traffic. In order to model the normal traffic, this approach needs purely normal data. In a real-time system, it is very difficult to have either purely normal data or labeled data. If any attack is left as normal, then this attack will never be considered by the IDS and alert will not be generated.

Considering the problems of the two previous approaches, a third one is becoming more popular: the hybrid approach that is effective in taking care of the deficiencies of both the above approaches [5]. These kinds of systems do not need purely normal data and therefore unlabeled data can be used, which can be easily obtained.

Flood-based attacks can be detected by scanning the TCP/IP headers of network packets whereas the non-flood attacks cannot be detected using these headers [6]. Such attacks that are usually ''User to Root'' (U2R) and ''Remote to Local'' (R2L), have the intruders sending very few number of packets in order to malign the root directly or remotely in the network. For detecting such attacks, packet payloads can be used to detect intrusions.

Different approaches have been implemented to detect intrusions based on the payload of a packet. As the payload of different network depends upon the kind of service

it provides therefore service specific approaches are developed. In paper [7], the authors Krügel et al. have taken into consideration R2L attacks based on service-specific knowledge to increase the detection rate of intrusions. In work [8], the authors have used byte frequency distribution of the application-level payload. In paper [9], the authors Z. Morley et al. have analyzed the DDoS attacks and suggested that the attack duration, packet count, packet rate and prototype are vital during packet feature selection. In [6], the authors have considered both the header and payload information to improve the detection. In papers [10], [11], appearance frequency of 255 ASCII codes have been considered to characterize the attack. Keisuke Kato et al. in their work [12] analyzed the DDoS attacks. According to their work, the bytes per second and packet size of normal and attack packets are found to be same (approximately) sizes. F. Iglesias et al. in their work [13] selected 16 features from 41 features of the DARPA set using multi-stage feature selection which includes both the header and data features from the dataset. Irshad M. Iqbal et al. in their paper [14] have also used data related features from the network traffic.

From the above literatures point of view, it can be understood that intrusion detection approach has evolved with time and become more requirement specific. Since then, several methods have been developed for detecting anomaly. Machine-learning, data mining, genetic algorithms, neural networks, and statistical methodology are some of them. Some methods namely genetic algorithms and neural networks have addressed the anomaly-based detection problem by mimicking the nature and natural organisms. Nature and natural processes have always led to new dimensions in problem-solving algorithms and devices and therefore have become important in the fields of engineering and applied sciences in the last few decades [15]. Nature and natural organisms have always inspired researchers in exploring and combating environmental issues and humans are no exceptions. Underlying this concept, bio-inspired computing has gained key role in the field of evolutionary computing, artificial immune systems, swarm systems and membrane computing systems. In recent times, deep learning approach for detecting intrusion has been applied. In [16] and [17], approaches based on recurrent neural network and deep convolutional neural network (CNNs) are used to capture the essential features. Also, nature-inspired swarm optimization technique has been used in paper [18].

Artificial Immune System (AIS) is one such domain where intrusion and its detection have been widely used for about two decades. AIS is believed to work similarly like its biological adaptation, Immune System (IS) by taking into consideration both the Innate Immune layer and the Adaptive Immune layer respectively. Bio-inspired intrusion detection model have been widely used by various researchers in the last decade. Forrest et al. have first and foremost the conceptualized the self-discrimination and non-self-discrimination in their paper [19]. This common approach for discrimination of IS was used for further to discriminate self (legitimate users, protected data) from non-self (unauthorized users, viruses, etc.). Hofmeyr in [20] has contributed towards the distributed approach that can be used for intrusion detection using the IS adaptation. Stibor *et al.* [21]–[23] have shown both the positive and negative selection methods of IS for appropriation of anomaly detection.

In recent times, Uwe Aicklein et al. in their paper [24] have used Dendritic Cell Algorithm (DCA) of Artificial Immune System (AIS) for intrusion detection. DCA is a population-based algorithm, inspired by functions of natural DCs of the innate immune system. In [25] and [26], the authors have used immune system and employed agent-based systems to detect the anomalies and intrusions based on network profiles. Walid Mohamed Alsharafi et al. in their paper [27] have generated large pool of detectors to find the correct match with any change probably intrusion, virus or misusing etc. Clonal Selection is another approach inspired by IS, that has been used in [28]–[31], where the emphasis is given on population of immune cells that are able to recognize the foreign bodies and thus are proliferated accordingly in the body. This approach works well as because it is dynamic and clone copies that have greater affinity towards recognition are only proliferated.

III. THE PROPOSED MODEL (IS-IDS)

IS-IDS mimics the immune system by imitating the functionalities of the KCs, DCs of innate immune system and T-cells, B -cells of the adaptive immune system. The current work gives emphasis on developing an IDS model inspired by the immune system. It consists of two layers. The first layer of the IDS imitates the innate immune system and termed as *Statistical Modeling based Anomaly Detection* (SMAD). Subsequently, the second layer, *Adaptive Immune-based Anomaly Detection* (AIAD) imitates the adaptive immune system.

The first layer SMAD has two modules. The *Pre-processing-I Module* works in analogous to KCs as it is responsible for sensing the external intrusions. It tries to capture the external traffic in order to find out the first-hand vulnerability of the traffic. If the traffic through the *Pre-processing-I Module* appears to be vulnerable, then it is considered for the pre-processing-II module or otherwise the traffic is a normal one and passed through the internal network of the organization.

The *Pre-processing-II Module* works more like the DC cells, which can initiate the immune responses by activating the naive T-cells for further adaptive responses. The *Pre-processing–II Module* identifies the traffic in order of vulnerabilities to the system, which can be considered as a) Normal b) Least Suspicious c) Moderately Suspicious and d) Highly Suspicious and forwards the most suspicious traffic to the adaptive immune system. This system is very similar to the skin, as it identifies the traffic according to the maliciousness and decreases the amount of traffic by a significant amount for the adaptive immune system.

The second layer AIAD takes into consideration the activation of the T-cell and the B-cell modules. The traffic that

is found to be highly suspicious in the first layer is captured and the features from the header and payload portions relevant for the anomaly detection are disseminated to the *T-cell Activation Module*. This module further identifies the features from the traffic that are relevant for the *User-based anomaly and File-Based anomaly Detection Modules*. *T-cell Activation Module* acts like the armed-effector T-cell that binds strongly with the pathogen-specific MHC complex and releases proteins namely, CD8 and CD4 respectively. CD8 + T cells are proteins that bind further with MHC Class I molecules and release CDα and CDβ-cells. On a similar note, the features that are recognized by *User-based anomaly Detection Module* are further disseminated for the *Off-Hours Access Detection and Remote Access Detection Modules*.

The CD α-cell binds with the MHC I pathogen molecule in order to kill the pathogen further whereas the CD β-cells are specifically used for intravascular communication. In analogous to this, in the proposed system, the *Off-Hours Access Detection Module* would try to detect whether a user accesses the host-computer during off-hours of the system. And, the *Remote-Access Detection Module* tries to detect whether a user accesses the system in remote location or not. If the file finds any kind of remote access, then the user anomaly is recognized by the T-helper cell and sent to the B-cell Activation module.

On the other hand, CD4+ T cells are proteins that bind further with MHC Class II molecules to release TH1 and TH2 cells. Similarly, features that are recognized by the *File-based anomaly Detection Module* are further disseminated to the *File Anomaly Type Detection and File Parsing and Detection Modules.*

TH1 cell's main function is to activate macrophages to kill the intravascular pathogens presented by MHC Class II molecules. *In analogous to this, File Anomaly Type Detection Module* finds any anomaly related to file i.e. virus, worms etc. and deletes them immediately from the system.

TH2 cells, or helper T cells, are responsible for activating naive B cells to make antibody. *File Parsing and Detection Module* further activates the File Identity Database for detecting any anomaly that was previously archived in the database.

A. STATISTICAL MODELLING BASED ANOMALY DETECTION (SMAD): THE INNATE IMMUNE SYSTEM

The proposed model mimics the innate immune system and is the first layer of the proposed IDS. It captures the characteristics features of a system after fixed interval of time and compares the observed value of F in ANOVA against the F table to identify intra-group and inter-group differences.

It captures the characteristics features say X_1, X_2, X_3 of a system with factor A at a levels, factor B at b levels and factor N at n levels. Factor A can be system-based resource usages with 1, 2, ..., a levels and factor B can be user-based resources with $1, 2, \ldots, b$ levels. The factors are independent treatment variables like system or user-based usages whose settings are controlled by the levels say the time interval. Table 2 represents the feature capture over time.

TABLE 2. System or user based feature capture over time.

Levels →

Factors ↓

System/ User Feature	T_1	T_2	T_3	T_4	Total (T_i)
X_1	$X_1(T_1)$	$X_1(T_2)$	$X_1(T_3)$	$X_1(T_4)$	$X_1(T_i)$
Factor A X_2	$X_2(T_1)$	$X_2(T_2)$	$X_2(T_3)$	$X_2(T_4)$	$X_2(T_i)$
Factor B X_3	$X_3(T_1)$	$X_3(T_2)$	$X_3(T_3)$	$X_3(T_4)$	$X_3(T_i)$
Factor N T'^2	$T_1'^2$	$T_2'^2$	$T_3'^2$	$T_4'^2$	$\sum T(i)$

Let A_i be the sum of all observations of level i of factor A, for $i = 1,..., a$. The A_i are the row sums.
Let B_j be the sum of all observations of level i of factor B, for $j = 1,..., b$. The B_j are the column sums.
Let $(AB)_{ij}$ be the sum of all observations of level i of A and level j of B. These are cell sums.

The ANOVA model can be represented mathematically as:

$$Y_{ij} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \in_{ij} \quad (1)$$

where,

- μ is the overall mean response
- α_i is the effect due to the i^{th} level of factor A
- β_j is the effect due to the j^{th} level of factor B
- γ_{ij} is the effect due to any interaction between the i^{th} level of A and the j^{th} level of B
- $\in_{ij}$ is the random error in the j^{th} observation of the i^{th} treatment

ANOVA calculates Sum of Squares (SS) in order to determine the variability within the group or between the groups. The above can be mathematically expressed as ''equation (2),'' shown below:

$$SS(Total) = SS(A) + SS(B) + SS(AB) + SSE \quad (2)$$

where,

- $SS(Total)$ is the sum of total of all observations
- $SS(A)$ is the sum of all observations due to factor A
- $SS(B)$ is the sum of all observations due to factor B
- $SS(AB)$ is the sum of all observations due to factor

A and its interaction with B and

- SSE is the random error

ANOVA considers each characteristic feature i.e. X_1, X_2, X_3 as a group to evaluate the system behavior. If the mean of all the characteristics features are equal then the variation within a feature say, $X_1(T_1), X_1(T_2)$ *and* $X_1(T_3)$ is taken into consideration.

If the means of all the features are not equal then the variation between the system features $X_1(T_i)$, $X_2(T_i)$ *and* $X_3(T_i)$ is taken under consideration.

The main intention is to test the null hypothesis (H_0) that considers whether the means taken from different system behavior are equal or accept the alternative hypothesis (H_1) that considers at least one of the system behavior means is different from others.

Thus, the main objective is to evaluate the variability either within the specific feature or between two or more features. Any variability would denote change in characteristic feature and would represent anomalous behavior. ANOVA uses F-tests to statistically test the equality of means using F-table. The F-table (F_t) tests the test-statistics (F_r) against the tabular value based on the chosen confidence level say 95% and the Degrees of Freedom (DF).

Let us assume time in seconds as time intervals, T_1, T_2, T_3, T_4 seconds for n system or user-based resource features X_1, X_2, X_3 *and* X_n. Following is the stepwise pseudo-algorithm for the SMAD model.

Algorithm 1 Algorithm for SMAD Model

Input: Accept the system-based or user-based resource features X_1, X_2, X_3 and X_n for time intervals T_1, T_2, T_3, T_n in seconds

Output: X_1, X_2, X_3 and X_n vary significantly or non-significantly

1: Initialization: $h \leftarrow$ number of rows
$k \leftarrow$-number of columns
$N \leftarrow$ number of observations

2: Compute Total of the squares of all observations, (*SS)* as in "equation (3),":

$$SS = X_1(T_1)^2 + X_1(T_2)^2 + \ldots + X_n(T_n)^2 \quad (3)$$

3: Compute *Correction Factor*, (*CF)* as in "equation (4),":

$$CF = \sum T_i^2/N \quad (4)$$

4: Compute Total Sum of Squares *(TotalSS)*, as in "equation (5),":

$$TotalSS = SS - CF \quad (5)$$

5: Compute Sum of Squares Total *(SST),* as in "equation (6),":

$$SST = ((T_1'^2 + T_2'^2 + T_3'^2 + T_4'^2)/h - CF) \quad (6)$$

6: Compute Square Sum Between *(SSB)*, as in "equation (7),":

$$SSB = (X_1(T_i)^2 + X_2(T_i)^2 + \cdots + X_n(T_i)^2)/k - CF \quad (7)$$

7 : Compute Sum of Square Error *(SSE)*, as in "equation (8),":

$$SSE = TotalSS - (SST - SSB) \quad (8)$$

8 : Compute Degrees of Freedom *(DF) as below:*

i) DF for Total SS,

$$dfT = (h * k - 1) \quad (9)$$

ii) DF for Sum of Squares Total,

$$dft = (h - 1) \quad (10)$$

iii) DF for Square Sum between,

$$dfb = (k - 1) \quad (11)$$

iv) DF for Sum of Square Error,

$$dfe = dfT - (dft + dfb) \quad (12)$$

9: Compute Mean Square *(MS)* and F_r against F_t as shown below in "equation (13), (14) and (15)":

$$\text{i)} \quad MSA = SST/dft \quad (13)$$
$$\text{ii)} \quad MSB = SSB/dfb \quad (14)$$
$$\text{iii)} \quad MSE = SSE/dfe \quad (15)$$

10: Compute F ratio (F_r) as in "equation (16) and (17)

$$F_1 = MSA/MSE \text{ and} \quad (16)$$
$$F_2 = MSB/MSE \quad (17)$$

11: Compute F_t as $F_{t(\alpha,dft,dfe)}$ and $F_{t(\alpha,dfb,dfe)}$

12: Compare F_r with F_t

13: *If* $F_r > F_t$, *then*
13.1: *Message "Variability between the features"*
Call *Function_Critical_Difference ( )*
13.2: *Else, Message "Variability within the features"*
Call *Function_Three_Sigma ( )*
13.3: *End if*

14: ***Function_Critical_Difference ( )***

1: Compute

$$s = \sqrt{MSE} \quad (18)$$

where, s is the standard error

2: Compute

$$CD = s\sqrt{2n} * t_{0.025} \quad (19)$$

where, *CD* is the critical difference
n is the sample size and
t is based on *df* of *dfe*

3: Compute *absolute as* $|X_i(T_i) - X_n(T_n)|$

4: *If* $|X_i(T_i) - X_n(T_n)| > CD$ *then*

4.1: *Message "Usage of one feature is dependent on the usage of another feature"*

4.2: *Else*
Message "Usage of one feature is not dependent on the usage of another feature

4.3: *End if*
End _Function ( )
15 : ***Function_Three_Sigma ( )***
1: Compute

$$Avg = (X_1(T_1) + X_1(T_2) + \cdots + X_1(T_i))/no.ofT_n$$

2. Compute $S = \sum_{i=1}^{n} X_i(T_i) - Avg$
3. Compute $Square = S^2$
4. Compute $Avgnew+ = Square$
5. Compute $Variance = Avgnew/N$
6. Compute $One\_Sigma = \sqrt{Variance}$
7. Compute $Two\_Sigma = 2 * One\_sigma$
9. Compute $Three\_Sigma = 3 * One\_sigma$
End _Function ( )
16: *If* $X_n > Three\_Sigma$ *then*
16.1: *Capture the {TIME, SRC, SRCPORT, DST, DSTP ORT, PKTS, DATA}* of the network packet
16.2: *End If*
17: End

Once the highly suspicious network traffic is captured, it becomes necessary to determine the features of the packet that can be significant for detecting anomaly. The main issue is whether to capture the whole network packet or consider the TCP/IP header part only as considering whole packet including the data part would increase the efficiency of detection but at the same time increase the time and cost for capturing the whole network packet. However, considering only the header portion would reduce the detection rate and at the same reduce the time and cost for capturing the header.

Therefore, there are two options: either to capture the header information and discard the payload information or to capture header and, the payload of network packet and, subsequently devise a hybrid detection approach. This hybrid detection approach has some features from both the portions could be used for anomaly detection.

B. ADAPTIVE IMMUNE-BASED ANOMALY DETECTION (AIAD): THE ADAPTIVE IMMUNE SYSTEM

This proposed model (AIAD) acts like the second layer of the IDS. It imitates the adaptive immune system by taking into consideration the activation of the T-cells and the B-cells. The proposed model tries to learn and recognize specific kinds of pathogens through T-cells and retain a memory of them for speeding up future responses through B-cells. Highly suspicious traffic data from SMAD is considered as input for capturing the features of the header. It considers the traffic that is found to be highly suspicious from the first layer i.e. SMAD and captures the features of the header and payload portions of the network packet relevant for the anomaly detection. The features are disseminated to the *T-cell Activation Module*.

The *T-cell Activation Module* further activates *T-cell_File_Activation Module* for file-based anomaly and *T-cell_User_Activation Module* for user-based anomaly detections. The features that are relevant to file-based anomaly detection are sent to the *T-cell_File_Activation Module* whereas the features relevant to the user-based anomaly detection are sent to the *T-cell_User_Activation Module*. The primary objective of these modules is to identify file and user-based anomalies.

T-cell_File_Activation Module has two sub-modules namely: *File Anomaly Type Detection and File Parsing Detection Modules*. Sub-modules are described as follows:

1) FILE ANOMALY TYPE DETECTION MODULE

This module is responsible to determine the integrity of the file once it is saved into the host-computer. This module determines the unique hash value of the file as it arrives into the host-computer and if the value matches with the already existing value in the File Identification Database then the file is stored otherwise it is deleted. Any new file or modification of an existing file can only be completed with permission from the system administrator.

For this file anomaly detection positive selection algorithm of T-cell has been taken into consideration. Using this positive selection algorithm, the hash values of the different files present in the system is generated and stored as set of antibodies, $Ab = \{ab_1, ab_2, \ldots, ab_n\}$ of the set of files, $F = \{f_1, f_2, \ldots, f_n\}$ *in the File Identity Database*. When an incoming file, f_i arrives its hash value is generated as antigen, Ag_i. Then the value of the antigen, Ag_i is compared with each of the antibodies, $ab_n \epsilon Ab$. If it matches to any of the antibodies, ab_n, then the file, f_i is read to the system. However, if it does not match any of the antibodies then the file, f_i is not read to the system. Following shows the pseudo-algorithm for file anomaly type detection.

Algorithm 2 Algorithm for File Anomaly Type Detection Using Positive Selection Algorithm

Input: Create an initial set of hash values as *Ab, where* $Ab = \{ab_1, ab_2, \ldots, ab_n\}$ *of the set of files, F,*
Output:*Self file or Non- Safe file*
1: Accept hash value of the incoming file, f_i as antigen , Ag_i
2 : For each $ab_n \epsilon Ab$,
2.1: Compare (ab_n, Ag_i)
2.2:*If* $(ab_n == Ag_i)$ *then*
2.3: *Message "Self file" Go to Step 4*
2.4:*Else*
Message "Non- Safe file"
2:5: *End if*
2.6:*Loop until* $Ab = \{\emptyset\}$
3 : Read the file, f_i
4 : End

2) FILE PARSING DETECTION MODULE

It tries to detect whether the known virus codes prevail in the file or not. This module is activated when a file whose unique hash value does not matches with the already existing values of the *File Identification Database*. The file is checked against the virus code signatures that are present in

the database. The file is accepted if it does not contain any of the virus codes.

For implementing the proposed model, negative selection algorithm is used. Negative selection algorithm is like negative selection (clonal deletion) of the thymus. This algorithm consists of two phases namely a) detector generator phase and b) monitor phase.

a) Detector generator phase is responsible for generating set of detectors. Here, the detectors are the virus signatures that do not match any of the protected data of the files already existing in the system. The virus signatures or the detectors are the strings that are compared with each of the protected data of the files. The set of protected data of a file can be considered as a set of strings over a finite alphabet and any change in the data of the file would exhibit a change in the string not in the original set. To generate detectors, strings which do not match any of the strings of the protected data of the files are considered. A candidate detector is deleted if it matches any of the strings of the protected data of the files.

The negative selection algorithm was used for protecting files in the DOS operating system from corruption by viruses [16]. The self-set is obtained by taking into consideration the contents of the file, f that are already present in the *File Identity Database*. The contents of the file have strings of different length. Each such string can be considered asfeatures that are to be matched with the virus signatures or codes. In the detector phase, each of the virus signatures is the antigens, a_{gi} that belongs to Ag where Ag is the file containing the virus signatures that are matched against each of the contents or the strings of the file. The strings are considered to be the features or the antibodies, $A\{ab_1, ab_2, a_3, \ldots, ab_n\}$ that belongs to f.

Each such string is compared with the set of virus signatures, Ag. Each feature or the antibody in A are compared with each of the virus signatures in Ag. The similarity between them is referred to affinity (Dis) calculation in negative selection algorithm. If the affinity does not match then the antigen, ag_n is accepted as an antibody and stored in the *File* Identity *Database*. Otherwise the antibody is rejected as "Self". The antibody is accepted as the detector. This is repeated until there are no antibodies left. Following shows the pseudo-algorithm for Detector Generator.

Algorithm 3 Algorithm for Detector Generator

Input: Accept the set of virus signatures as $Ag = \{ag_1, ag_2, ag_3, \ldots, ag_n\}$ as antigens from File Identity Database

Output: Store antibody, ab_n or reject as *"Self"*

1: Create an initial random set of features of antibodies as $A\{ab_1, ab_2, ab_3, \ldots, ab_n\} \epsilon f$

2: For all features in Ag do

2.1: Calculate affinity with each antibody in A *as*

$$Dis(ab_n, ag_n) = \sqrt{\sum_{i=1}^{n} (ab_n - ag_n)^2} \tag{20}$$

2.2: *If (Dis == 0) then*

2.3: Reject the antibody, ab_n as *"Self"*

2.4: *Else*

Store the antibody , ab_n in the File Identity Database

2.5: *End if*

2.6: Loop until $A = \{\emptyset\}$

3: End

b) Monitor phase is responsible for monitoring the protected data of the all the incoming files by comparing them with the detectors generated from the above phase. Each copy of the detector is run against the protected incoming file data. If any of the detectors matches the protected data of the files, then an alert is generated to the server machine. Following shows the algorithm for Monitor Phase.

Algorithm 4 Algorithm for Monitor Phase

Input: Accept the set of features to be recognized for the incoming file, $f_i as\ S = \{s_1, s_2, s_3, \ldots, s_n\}$ as antigens and random set of features of antibodies as $A\{a_1, a_2, a_3, \ldots, a_n\}$ in the File Identity Database

Output: Alert message and remove the file, f_i

1: For all features in S do

1.1: For all features in A do

1.2:*Calculate affinity with each antibody in A*

$$\text{as } Dis(a_i, s_i) = \sqrt{\sum_{i=1}^{n} (a_i - s_i)^2} \tag{21}$$

1.3:*If (Dis == 0) then*

1.4: Remove the file, f_i and Message *"Alert to Server Machine"*

1.5:*Else* accept the next feature

1.6:*End if*

1.7:*Loop until* $A = \{\emptyset\}$

1.8:*Loop until* $S = \{\emptyset\}$

2: End

T-cell_User_Activation Module also has two sub-modules namely: Off-Hours Access Detection module and Remote-Access Detection module. Sub-modules are described as follows:

3) OFF-HOURS ACCESS DETECTION MODULE

This module tries to detect whether an intruder accesses the host-computer during off-hours of the system. According to Denning [2], individual profiles of the legitimate users can be useful in detecting such attacks. These profiles would provide the individual login frequencies of the legitimate user. This would help in detecting the masqueraders who try to log into unauthorized account during off-hours when the legitimate user is not expected to use the account. Off- hours or non-working hours may be assumed when the user is not expected to access system or his account. Individual profiles of legitimate users can provide the details

of the user logged in during his/her office hours from the office location or aggregate of all locations the user accesses his/her account.

4) REMOTE-ACCESS DETECTION MODULE

This module aims to detect whether any external user accesses the system from remote location or not. If it finds any kind of remote access, then the user anomaly is recognized by the T-helper cell and sent to the B-cell Activation module. This module uses the basic idea that a user sends packets to a machine over the network in which he or she does not have the privilege to access like the local user would have the access. The external user tries to access the system in order to control the remote machine through the local user.

Both these modules attempt to find out any unauthorized access during off-hours or from remote locations. For the implementation of these modules, immune-inspired clonal selection classification algorithm (CSCA) is used. This is based on the idea that the cells (antibodies) that are capable of recognizing the foreign bodies (antigens) will only be selected for further proliferation. The cells undergo affinity maturation (mutation) which further improves their affinity towards the foreign bodies or antigens.

The interactions between antigens and antibodies *(Ag-Ab)* can be considered as a generalized shape (S) comprising of the set of features $(F = \{f_1, f_2, f_3 \ldots, f_n\})$ that characterizes them. This generalized shape (S) can represent both real and binary-valued features. A distance measure is used to calculate the degree of similarity (affinity) between them. Each of the antigen or antibody is considered to have same length, l. The length and cell representation depend upon the problem.

Following is the CSCA algorithm used for detecting anomalies in both these modules:

Algorithm 5 CSCA Algorithm for Off-Hours Access Detection and Remote Access Detection

Input: *S*et of features, *F where,* $F = \{f_1, f_2, f_3 \ldots, f_n\}$ as antigens *(Ag)* as candidate solutions
Set of features, *S where,* $S = \{s_1, s_2, s_3 \ldots, s_n\}$ as antibodies *(Ab)* from the User Identity database

Output: Remove the antibody *(Ab)* or store

1:*For all features in F do*
1.1: *For all features in S do*
1.2: Compute

$$D = \sum_{i=1}^{l} \alpha \tag{22}$$

where, $\alpha = \{^{1 iff_i = s_i}_{0 iff_i \neq s_i}$
for all binary-valued f_i *and* s_i

or,
Compute

$$D(f_i, s_i) = \sqrt{\sum_{i=1}^{n} (f_i - s_i)^2} \tag{23}$$

for all real-valued f_i *and* s_i
1.3:*If* $(D == 1$ *or* $D >= \varepsilon)$ *then*
1.4: *Se* lect best features of the F with best D *(affinity)*
1.5: *End if*
1.6: *Loop until* $S = \{\phi\}$
1.7: Loop until $F = \{\phi\}$
2: Generate clones for n best features of the F using

$$num\_of\_clones = (D_i / \sum_{i=1}^{l} D_i) * (n) \tag{24}$$

where, D_i*is the fitness score for each of the antibodies in the current set, and* l *is the total number of antibodies and n is the total number of antibodies in the current set*

3: Mutate the clones using

$$\alpha = e^{-\rho * D} \tag{25}$$

where, D is the candidates affinity and

ρ *is the mutation rate*

4. Add clones and antigens selected to the population
5 : *If* $(D_i < \varepsilon)$ *then*
5.1: Remove antibodies
5.2:*Else Store in the* User Identity database
5.3:*End if*
6 : Modify the threshold (ε)
7 : End

B-cell Activation Module: This module takes the results of the above modules and store in two different files User-Identity Database and File-Identity Database. The main function of B-cell Activation Module is as follows:

a) Storage of new user and updating an existing user identity in User Identity database
b) Storage of new file and updating an existing file identity in File Identity database
c) Alert generation to Server Machine and other Host machines
d) File or User access grant or denial

IV. EXPERIMENTAL SETUP & RESULTS

A. RAW NETWORK DATA PREPARATION

The proposed model has considered both the real-time network traffic and the standard dataset for intrusion detection. For capturing real-time traffic, the experiment was conducted for three weeks on all incoming files. The system performance log is generated for each day, depicting the Network bandwidth usage, CPU usage, Memory usage respectively which is considered as the normal data set. The proposed model considers the system resource usages for generating

the system performance log. The IDS monitors the usages of network bandwidth, CPU and RAM and captures them after every 5 minutes of time interval for all incoming files. The mean sample size for real-time traffic of all incoming files was 30,377. The maximum number of files that were considered on a particular day was 30,390. The mean population size was 2,20000 approximately. In order to verify the performance of the proposed system, unknown HTTP traffic files were also considered for the vulnerability test purposes. The mean sample size of such HTTP files was 3,000 per day approximately. An intruded data set has been created by disabling the graphics driver, audio driver and USB driver.

Consider the following sample set of observations after every 5 minutes of time interval, $T_n(T_1, T_2, T_3 \ to \ T_4)$ for system resource features like network bandwidth, CPU and RAM usage respectively as in Table 3:

TABLE 3. Set of observations based on system resource usage.

System Resource Usage	T_1	T_2	T_3	T_4
Network Bandwidth usage	90	100	90	100
CPU usage	120	110	90	110
RAM usage	110	120	100	120

The above usages depict system resource usages in percentage. The following Table 4 represents observations reduced by dividing a constant (say, here 10) from each observation.

TABLE 4. Set of observations after reduction.

System Resource Usage	T_1	T_2	T_3	T_4
Network Bandwidth usage	0.9	0.10	0.9	0.10
CPU usage	0.12	0.11	0.9	0.11
RAM usage	0.11	0.12	0.10	0.12

The network packet has been partitioned into header and the payload portion. The header portion of the packet considers the packet sequence_number, source_port, destination_port, source_address, destination_address, length_of_the packet and protocol_type respectively whereas the data portion considers the raw data in binary bits representation.

The proposed model also has been tested on 10% KDD 99 standard dataset. The non-numerical attributes are converted to numeric values using discretization. Z-score normalization is also performed at the onset of the experiments to transform all attributes into the normalized format. The features that are selected for intrusion detection from the KDD 99 dataset are based on the type of attacks. Attacks in the dataset are categorized into Denial of Service (DoS), Probe, User to Root (U2R) and Remote to User (R2L) respectively.

Denial of Service attack takes place when an attacker makes the computing resources too busy to handle the legitimate users request and ultimately denies the user access to the system. The attacks that are categorized under this type in the KDD dataset are back, land, Neptune, smurf, pod and teardrop. Probe attacks are those where the attacker tries to gain information or sometimes abuse the host machine's features to look for exploits. Under this category, satan, nmap, portsweep, ipsweep are the four types of attacks.

For conducting the tests, DoS and Probe attacks have been considered for the SMAD model that has been proposed. These attacks are based on the assumptions that large number of packets is send to the host system over a short span of time [6]. These flood attacks are detected by scanning the TCP/IP headers of the network packets from the real-time traffic [6]. Features *(f1-f9)* have been observed from the 10% KDD dataset. The packets that are found to be highly suspicious are considered with their payload data for the second layer of the IDS. The payload-based features that are considered for these packets are content-based features *(f10-f22)* for file-based anomaly detection in *File Anomaly Type Detection and File Parsing Detection Modules*.

User to Root (U2R) attack occurs when an attacker tries to access legitimate user account or gain the root access. Loadmodule, buffer overflow, rootkit and perl are the four types of such attacks that an attacker tries to intrude through regular programming mistakes and environmental assumptions in U2R. For conducting tests on the standard dataset, these U2R attacks are used for the *Off-Hours Access Detection module.*

Remote to Location (R2L) attacks are those types of attacks where the attacker tries to gain access of the user machine remotely. The types of attacks in this category that are considered are phf, guess_passwd, warezmaster, imap, multihop, ftp_write, spy and warezclient respectively. For the purpose of experimentation, we have considered the R2L attacks for conducting tests of the *Remote- Access Detection module.*

The proposed model has also considered UNSW-NB15 dataset. It has 49 features. It is categorized in three groups: Basic, Content and Time. Some additional general-purpose and connection features are also present in the dataset. The features that are considered for the SMAD model are the basic features of the UNSW-NB15 dataset. The content-based features are considered for the file-based anomaly detection in *File Anomaly Type Detection and File Parsing Detection Modules.* The time-based and additional features are used in *T-cell_User_Activation Module* for user-based anomaly detections.

B. RESULTS AND DISCUSSION

The proposed model has been implemented in C# and the experiment was conducted for three weeks on all incoming files which were as high as 30,390 files on a particular day with mean sample size of 30,377 files each day. The system performance log is generated for each day, depicting the Network bandwidth usage, CPU usage, Memory usage respectively which is considered as the normal data set. An intruded data set has been created by disabling the graphics driver, audio driver and USB driver. Performance of the experiment has been measured by taking into consideration of the true positive rate against the false positive rate. The proposed system yields as high as 96.04% true positive rate that increased incrementally each day for real-time traffic using the first-layer, SMAD model. Similarly, the proposed SMAD model shows 7.8% as the false positive rate. The

graph for True Positive Rate vs. True Negative Rate and ROC curve are given below in Figure 1 and Figure 2.

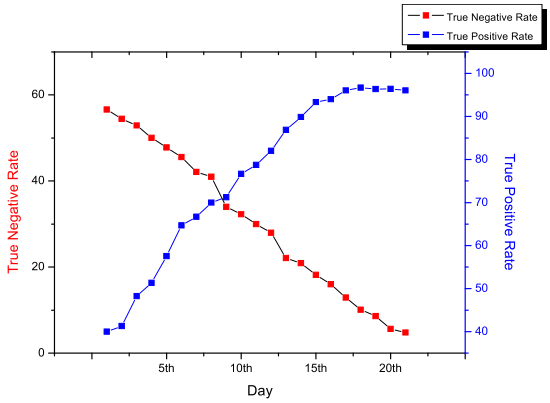

FIGURE 1. TPRs. vs. TNR in real-time traffic for *SMAD Model*.

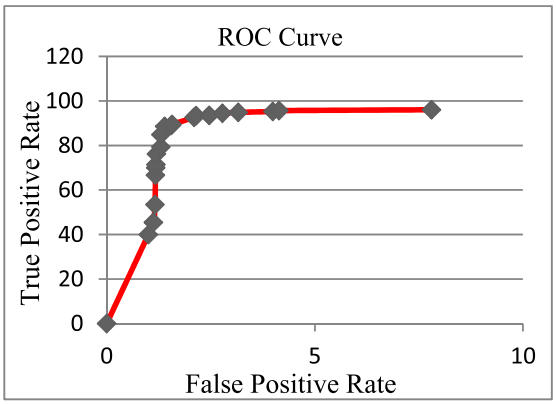

FIGURE 2. ROC Curve in real-time traffic for SMAD Model.

In the second layer of the proposed IDS, the payload portion of the highly suspicious traffic has been extracted and tested with the help of *T-cell Activation Module*. The *T-cell Activation Module* further activates *T-cell_File_Activation Module* for file-based anomaly and *T-cell_User_Activation Module* for user-based anomaly detections. In *T-cell Activation Module*, the file-based anomalies are determined by self and non-self tests using their hash values and determining the repeated known virus codes in the file with the help of *File Anomaly Type Detection and File Parsing Detection Modules*. Similarly, in *T-cell_User_Activation_Module*, user–based anomalies are identified on the basis of determination of the user in remote location or working in abnormal, off-hours. The data portion of the packet is considered for file-based anomaly detection. The Figures 3 and 4 show the results of true positive rates vs. false positive rates and true negative rates vs. false negative rates of the *T-cell_File_Activation Module* for file-based anomaly and *T-cell_User_Activation Module* for user-based anomaly detections using the real-time traffic.

For standard KDD99 dataset, features *(f1-f9)* are considered for the SMAD model and the payload-based features that are considered are content-based features

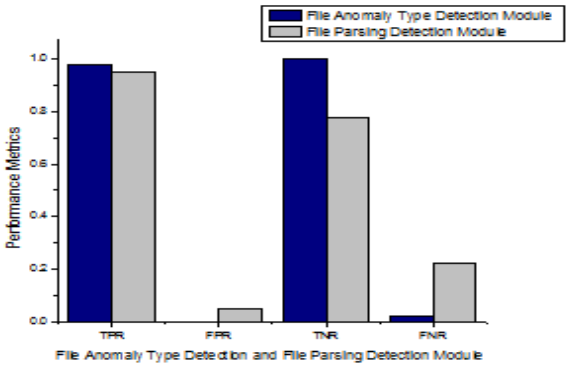

FIGURE 3. TPRs. vs. FPR and TNR vs. FNR in real-time traffic for *T-cell_File_Activation Module*.

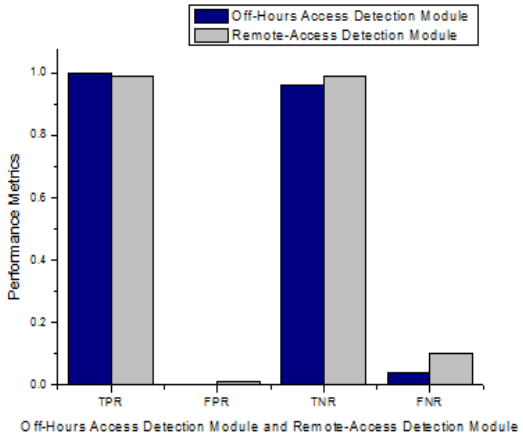

FIGURE 4. TPRs. vs. FPR and TNR vs. FNR in real-time traffic for *T-cell_User_Activation Module*.

(f10-f22) for file-based anomaly detection in *File Anomaly Type Detection and File Parsing Detection Modules*. Features that are found to be relevant are *src_bytes, dst_bytes, flag, urgent* for SMAD model and *is_guest_login, is_host_login, num_outbound_cmds, num_access_files, logged_in, su_attempted, num_failed_logins* respectively for file-based anomaly detection.

The results of SMAD model shows 97.1of true positive rate and 2.79 of false positive rate. Figure 5 shows the results of SMAD model with increasing ROC.

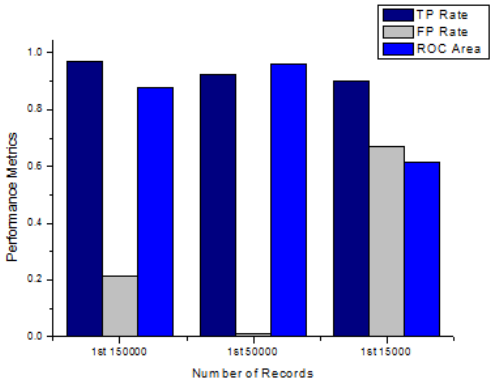

FIGURE 5. Results of SMAD Model with ROC Area for KDD Dataset.

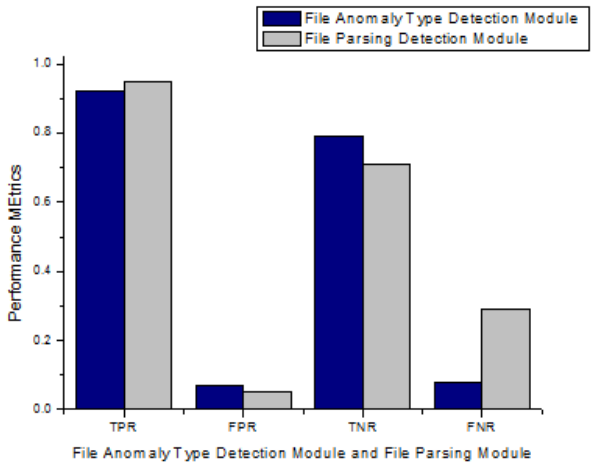

FIGURE 6. TPRs. vs. FPR and TNR vs. FNR using KDD Dataset for *T-cell_File_Activation Module*.

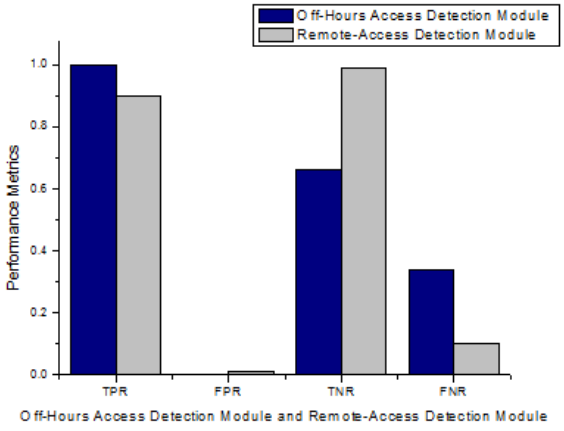

FIGURE 7. TPRs. vs. FPR and TNR vs. FNR using KDD Dataset for *T-cell_User_Activation Module*.

For detection of remote location and off-hours office anomaly, we have considered both the content-based *(f10-f22)* and traffic-based features *(f23-f41)* of the standard dataset. The features that are found to be useful for R2L and U2R attacks are *hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempetd, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_guest_login, is_host_login, count, srv_count, dst_host_count, dst_host_srv_count, dst_host_diff_srv_rate, dst_host_srv_serror_rate, dst_host_srv_rerror_rate.* Figures 6 and 7 show the results for both these modules.

UNSW-NB15 dataset has also been considered in order to meet the current demands of network security. The UNSW-NB15 dataset has 49 features in total which are categorized in three groups: Basic, Content and Time. Some additional general-purpose and connection features are also present in the dataset. The features that are considered for the SMAD model are the basic features of the UNSW-NB15 dataset.

The features that are found to be relevant are for the SMAD model are *spkts, dpkts, dloss, dbytes, sloss, sttl, state, service, rate, sbytes, sloss, dload, dttl, dur.* The result of the SMAD model is represented in Figure 8.

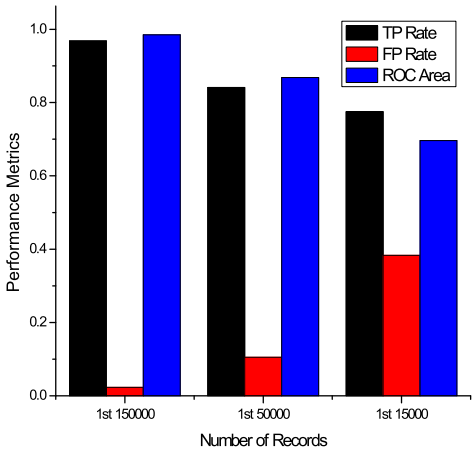

FIGURE 8. Results of SMAD Model with ROC Area for UNSW-NB15 Dataset.

For the AIAD model, content, time and rest of the additional features are considered. The content-based features are considered for the file-based anomaly detection in *File Anomaly Type Detection and File Parsing Detection Modules.* The features that are found to be relevant for this module are: *stcpb, dtcpb, dmeansz, swin, dwin, trans_depth*. The result of the *T-Cell_File_Activation Module* is shown in Figure 9.

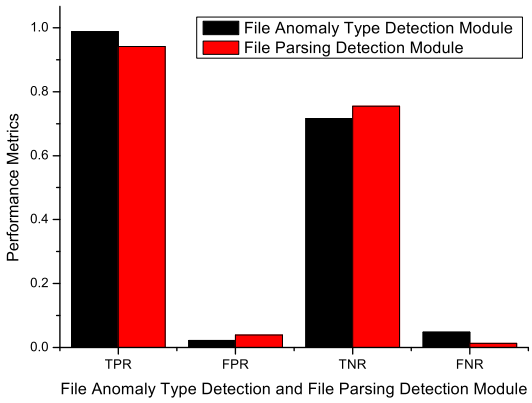

FIGURE 9. TPRs. vs. FPR and TNR vs. FNR using UNSW-NB15 Dataset for *T-cell_File_Activation Module*.

The time-based and additional features are used in *T-cell_User_Activation Module* for user-based anomaly detections. Features that are found to be relevant are: *ct_dst_src_ltm, ct_srv_dst, ct_srv_src, ct_src_dport_ltm, ct_dst_ltm, tcprtt, synack, ackdat, ct_state_ttl, is_ftp_login, ct_ftp_cmd, ct_flw_http_mthd, djit, sinpkt, is_sm_ips_ports, ct_ftp_cmd, is_ftp_login, dinpkt.* The result of the *T-Cell_User_Activation Module* is shown in Figure 10.

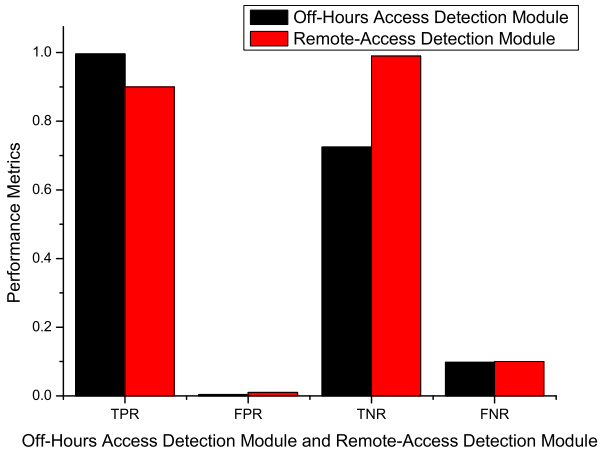

FIGURE 10. TPRs. vs. FPR and TNR vs. FNR using UNSW-NB 15 dataset for *T-cell_User_Activation Module*.

V. CONCLUSION

The current work is a proposal on intrusion detection system based on artificial immune system. It has two layers as on a natural immune system. Statistical Modeling based Anomaly Detection (SMAD) is the first layer of IDS. The SMAD works as the Innate Immune System (IIS) interface. It can be considered analogous to the skin of innate immune system. Its primary intention is to capture the initial network traffic in order to find out the first-hand vulnerability of the traffic. It captures the characteristics features of a network after fixed interval of time and compares the observed value against the pre-defined value of F in ANOVA. The SMAD captures the network traffic that is found to be highly suspicious. The second layer is based on the adaptive immune system and has considered analyzing features of the network packets of the highly suspicious traffic. It is termed as Adaptive Immune-based Anomaly Detection (AIAD) and acts like the second layer of the IDS. It imitates the adaptive immune system by taking into consideration the activation of the T-cells and the B-cells. It captures relevant features from header and payload potions for effective detection of intrusion.

The proposed model has considered both the real-time network traffic and the standard dataset for intrusion detection. For capturing real-time traffic, the experiment was conducted for three weeks on all incoming files. In real-time traffic, the source port, destination port, source IP address, destination IP address, packet length, payload data have been considered from the network packet. The proposed system yields as high as 96.04% true positive rate that increased incrementally each day for real-time traffic using the first layer, SMAD model. Similarly, the proposed SMAD model shows 7.8% as the false positive rate. The proposed system yields 97.1% true positive rate and 2.79% false positive rate for KDD 99 dataset. Highly suspicious traffic from the SMAD model are further tested for vulnerability in AIAD model. Results show significant true positive rate which is closer to almost 99% of accurately detecting the file-based and user-based anomalies. The proposed model also has been tested on 10% KDD 99 standard dataset. For standard KDD99 dataset, features *(f1-f9)* are considered for the SMAD model and the payload-based features that are considered are content-based features *(f10-f22)* for file-based anomaly detection. For detection of remote location and off-hours office detection, we have considered both the content-based *(f10-f22)* and traffic-based features *(f23-f41)* of the standard dataset. Results show high true positive rates and lower false positive rates for this standard dataset.The proposed system has been tested on UNSW-NB15 dataset that considers recent anomalous behaviors of current network security scenarios. For this dataset, the basic features were considered for the SMAD model whereas the content, time and additional features were considered for the AIAD model. Results exhibit very high true positive rates and ROC area for the SMAD model. The AIAD model also scores high true positive and lower false positive rates respectively.

REFERENCES

[1] S.E. Smaha, "An intrusion detection system," in *Proc. IEEE 4th Aerosp. Comput., Secur. Appl. Conf.*, Orlando, FL, USA, Dec. 1988, pp. 37–44.
[2] D. E. Denning, "An intrusion-detection model," *IEEE Trans. Softw. Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
[3] P. Di Meglio, G. K. Perera, and F. O. Nestle, "The multitasking organ: Recent insights into skin immune function," *Immunity*, vol. 35, no. 6, pp. 857–869, Dec. 2011.
[4] W. Lee, S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," in *Proc. IEEE Symp. Secur. Privacy*, Jan. 2003, pp. 120–132.
[5] L. Portnoy, E. Eskin, and S. J. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 386–395.
[6] I. Perona, I. Gurrutxaga, O. Arbelaitz, J. I. Martín, J. Muguerza, and J. M. Pérez, "Service-independent payload analysis to improve intrusion detection in network traffic," in *Proc. 7th Australas. Data Mining Conf.*, vol. 87, 2008, pp. 171–178.
[7] C. Krügel, T. Toth, and E. Kirda, "Service specific anomaly detection for network intrusion detection," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2002, pp. 201–208.
[8] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection* (Lecture Notes in Computer Science), E. Jonsson, A.Valdes, and M. Almgren, Eds. Berlin, Germany: Springer, 2004, pp. 203–222.
[9] Z. M. Mao, V. Sekar, O. Spatscheck, J. Van Der Merwe, and R. Vasudevan, "Analyzing large DDoS attacks using multiple data sources," in *Proc. SIGCOMM Workshop Large-Scale Attack Defense (LSAD)*, 2006, pp. 161–168.
[10] Y. Otsuki, M. Ichino, S. Kimura, M. Hatada, and H. Yoshiura, "Evaluating payload features for malware infection detection," *J. Inf. Process.*, vol. 22, no. 2, pp. 376–387, 2014.
[11] A. Jamdagni, Z. Tan, X. He, P. Nanda, and R. P. Liu, "RePIDS: A multi tier real-time payload-based intrusion detection system," *Comput. Netw.*, vol. 57, no. 3, pp. 811–824, Feb. 2013.
[12] K. Kato and V. Klyuev, "Large-scale network packet analysis for intelligent DDoS attack detection development," in *Proc. 9th Int. Conf. for Internet Technol. Secured Trans. (ICITST)*, Dec. 2014, pp. 360–365.
[13] F. Iglesias and T. Zseby, "Analysis of network traffic features for anomaly detection," in *Machine Learning*, vol. 101, no. 1. Cham, Switzerland: Springer, 2015, pp. 59–84.
[14] I. M. Iqbal and R. A. Calix, "Analysis of a payload-based network intrusion detection system using pattern recognition processors," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, Oct. 2016, pp. 398–403.
[15] H. Richter, "Artificial immune systems, dynamic fitness landscapes, and the change detection problem," in *Bio-Inspired Computational Algorithms and Their Applications*, S. Gao, Ed. London, U.K.: IntechOpen. [Online]. Available: https://www.intechopen.com/books/bio-inspired-computational-algorithms-and-their-applications/artificial-immune-systems-dynamic-fitness-landscapes-and-the-change-detection-problem, doi: 10.5772/37083.

[16] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[17] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.
[18] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.
[19] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, Dec. 2002, pp. 202–212.
[20] S. A. Hofmeyr, "An immunological model of distributed detection and its application to computer security," Ph.D. dissertation, Dept. Comput. Sci., Univ. New Mexico, Albuquerque, NM, USA, 1999.
[21] T. Stibor, J. Timmis, and C. Eckert, "On the appropriateness of negative selection defined over hamming shape-space as a network intrusion detection system," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Dec. 2005, pp. 100–995.
[22] T. Stibor, "On the appropriateness of negative selection for anomaly detection and network intrusion detection," Ph.D. dissertation, Darmstadt Univ. Technol., Darmstadt, Germany, 2006.
[23] T. Stibor, "Phase transition and the computational complexity of generating r-contiguous detectors," in *Artificial Immune Systems* (Lecture Notes in Computer Science), L. N. de Castro, F. J. Von Zuben, H. Knidel, Eds. Berlin, Germany: Springer, 2007, pp. 142–155.
[24] F. Gu, J. Greensmith, and U. Aicklein, "The dendritic cell algorithm for intrusion detection," in *Biologically Inspired Networking and Sensing: Algorithms and Architectures*, G. Cirincione, D. Watson A. Swami, and D. Towsley, Eds. Pennsylvania, PA, USA: IGI Global, 2012, pp. 84–102.
[25] F. Hosseinpour, S. Ramadass, A. Meulenberg, P. Vahdani Amoli, and Z. Moghaddasi, "Distributed agent based model for intrusion detection system based on artificial immune system," *Int. J. Digit. Content Technol. Appl.*, vol. 7, no. 9, p. 206, 2013.
[26] S. J. Xu and Y. Li, "Multi-agent intrusion detection system based on immune principle," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 3, no. 4, pp. 2681–2689, 2015.
[27] W. M. Alsharafi and M. N. Omar, "A detector generating algorithm for intrusion detection inspired by artificial immune system," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 2, pp. 608–612, 2015.
[28] C.-M. Ou, "Host-based intrusion detection systems adapted from agent-based artificial immune systems," *Neurocomputing*, vol. 88, pp. 78–86, Jul. 2012.
[29] N. Afzali Seresht and R. Azmi, "MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach," *Eng. Appl. Artif. Intell.*, vol. 35, pp. 286–298, Oct. 2014.
[30] C. Yin, L. Ma, and L. Feng, "Towards accurate intrusion detection based on improved clonal selection algorithm," *Multimedia Tools Appl.*, vol. 76, no. 19, pp. 19397–19410, Oct. 2017.
[31] L. Ma, J. Qu, Y. Chen, and S. Wei, "An improved dynamic clonal selection algorithm using network intrusion detection," in *Proc. 14th Int. Conf. Comput. Intell. Secur. (CIS)*, Nov. 2018, pp. 250–253.

INADYUTI DUTT has been in the field of academics, industry, and research for more than 18 years. She is currently doing the research in network security with the Department of Computer Applications, Sikkim Manipal Institute of Technology, Sikkim Manipal University (SMU). She is currently working as an Assistant Professor with the Department of Computer Applications, B. P. Poddar Institute of Management and Technology, Kolkata, India. She has more than 30 publications, authored a book and few book chapters to her laurels and also has keen research interest in the field of data mining, neural networks, and machine learning. She has been associated with ACM (CSTA), IAENG, and IACSIT. She has also been a Technical Reviewer in various journal and conferences like IC3 2018, *Informatica*, IJECE, and IJBM.

SAMARJEET BORAH is currently working as a Professor with the Department of Computer Applications, Sikkim Manipal University (SMU), Sikkim, India. He handles various academics, research, and administrative activities. He is also involved in curriculum development activities, board of studies, doctoral research committee, and IT infrastructure management. along with various administrative activities under SMU. He is involved with various funded projects in the capacity of a Principal Investigator/Co-Principal Investigator. The projects are sponsored by agencies like AICTE, Government of India, DST-CSRI, Government of India, and Dr. TMA Pai Endowment Fund. He is associated with ACM (CSTA), IAENG, and IACSIT. He organized various national and international conferences in SMU. Some of these events include ISRO Sponsored Training Programme on Remote Sensing & GIS, NCWBCB 2014, NER-WNLP 2014, IC3-2016, and IC3-2018. He is also associated with various other conferences in the capacity of a TPC Member, an Editorial Board Member, a Volume Editor, and a Reviewer. He is involved with various book volumes and journals of repute in the capacity of an Editor/Guest Editor/Reviewer such as IJSE, IJHISI, IJGHPC, IJIM, IJVCSN, JISys, IJIPT, IJDS, IJBM, and IEEE Access.

INDRA KANTA MAITRA received the Ph.D. degree in computer science from the University of Calcutta with rich academic, research, and administrative experience of more than 19 years. He is currently spearheading as a Controller of Examinations (Acting) with St. Xavier's University, Kolkata, India. Accomplished the research project entitled Computer Aided Detection (CAD) System for Automatic Detection of Breast Cancer sponsored by the Department of Electronics & Information Technology (DeitY), Ministry of Communication and Information Technology, Government of India, along with CDAC-T and RCC-T as a Co-Investigator with the University of Calcutta. Apart from that he is associated with the Lincoln University College, Malaysia; the Vignan's Institute of Information Technology (VIIT), Visakhapatnam; the Institute of Engineers India (IEI); and the Computer Society of India (CSI) in different capacity. He has authored two books and contributor of chapter of two books and has more than 50 research publications in international journals and conference proceedings to his credit. His areas of specialization are medical image processing and CAD, data structure and analysis of algorithm, object oriented programming, programming language, bioinspired network security, and computer organization and architecture. His areas of research are biomedical image processing and CAD. He was a recipient of the Best Poster Award from the 96th Indian Science Congress, 2009.