Intelligent Intrusion Detection and Prevention System using Smart Multi-instance Multi-label Learning Protocol for Tactical Mobile Adhoc Networks

M. Roopa[1] and S. Selvakumar Raja[2]
[1]Research Scholar, Department of Electronics and Communication Engineering,
Sathyabama Institute of Science and Technology, India
[e-mail: roopamsriram@gmail.com]
[2]Department of Electronics and Communication Engineering, TKR College of Engineering and Technology
Meerpet, Hyderabad-500097
[e-mail: selvakumarraja@yahoo.com]
*Corresponding author: M. Roopa

Received August 4, 2017; revised December 1, 2017; accepted December 23, 2017; published June 30, 2018

Abstract

Security has become one of the major concerns in mobile adhoc networks (MANETs). Data and voice communication amongst roaming battlefield entities (such as platoon of soldiers, inter-battlefield tanks and military aircrafts) served by MANETs throw several challenges. It requires complex securing strategy to address threats such as unauthorized network access, man in the middle attacks, denial of service etc., to provide highly reliable communication amongst the nodes. Intrusion Detection and Prevention System (IDPS) undoubtedly is a crucial ingredient to address these threats. IDPS in MANET is managed by Command Control Communication and Intelligence (C3I) system. It consists of networked computers in the tactical battle area that facilitates comprehensive situation awareness by the commanders for timely and optimum decision-making. Key issue in such IDPS mechanism is lack of Smart Learning Engine. We propose a novel behavioral based "Smart Multi-Instance Multi-Label Intrusion Detection and Prevention System (MIML-IDPS)" that follows a distributed and centralized architecture to support a Robust C3I System. This protocol is deployed in a virtually clustered non-uniform network topology with dynamic election of several virtual head nodes acting as a client Intrusion Detection agent connected to a centralized server IDPS located at Command and Control Center. Distributed virtual client nodes serve as the intelligent decision processing unit and centralized IDPS server act as a Smart MIML decision making unit. Simulation and experimental analysis shows the proposed protocol exhibits computational intelligence with counter attacks, efficient memory utilization, classification accuracy and decision convergence in securing C3I System in a Tactical Battlefield environment.

Keywords: MANET, tactical networks, intrusion detection and prevention system, virtual clustering, smart MIML-IDPS protocol.

http://doi.org/10.3837/tiis.2018.06.025 **ISSN : 1976-7277**

1. Introduction

Mobile Adhoc Networks (MANET) are infrastructure-less, self-organizing, rapidly deployable wireless networks that are more suitable for communicating in regions where crisis such as natural disasters, military operations occur. Advantages like accessing information and services regardless of its geographical position, decentralized administration and self-configuring capability makes it appropriate for recovery operation during emergencies. For example, in military operation, it serves to provide voice and data communication among roaming entities like dismounted platoon of soldiers, inter-battlefield tanks etc,. Role of MANET is not limited to applications on ground, but can be located on land, on sea or in air. Moreover, as nodes function very much like a router , it is capable of executing computations and perform data exchanges among their peers. It also performs networking functions in a self-organizing manner, hence, securing such network becomes a challenge. Subsequently, MANET's mobility-induced dynamic topological changes, complex routing strategy, security threats makes it still more challengeable.

Though there have been significant improvements in the fields related to complex routing protocols, location strategy and mobility prediction, addressing challenges in security aspects [1] particularly to those employed in military applications have rarely been addressed. Considerable research and development efforts at Centre for Artificial Intelligence and Robotics (CAIR) is directed towards realization of feasible, reliable and secure MANETs. On the other hand, complex properties and its unique characteristic has lead to new security problems [2] that has recently attracted significant attention. Nevertheless, its open nature makes it still more vulnerable to internal and external attacks [3]. As most MANET routing protocols assume nodes to cooperate among each other to relay data; while, this assumption provides attackers with higher opportunities to achieve significant impact by compromizing nodes in the network. Comprehensive measures for detecting (malicious nodes, misbehavior links etc) and preventing attacks [4] should be added as a defense before an attacker can breach the system. By completely eliminating the attacker as soon as they enter the network can resolve potential damages caused to the network. Moreover, as MANET's characteristics make them susceptible to many new attacks on different layers of network protocol stack [5], intelligent and smart intrusion detection and prevention engine that can combat unknown attacks and facilitate situation awareness for timely decision-making is primarily required.

To address these limitations and act as a key line of defense against major security attacks, this paper presents a "Smart MIML-IDPS" methodology that exhibits computational intelligence using Multi-Instance Multi-Label [6] technique to protect the network against complex multistage attacks where fixed relationship is unattainable. MIML-IDPS is a behavioral intrusion detection and prevention technique that uses fast supervised machine learning algorithm to determine the attacks occurring in multiple instances. These instances are captured and classified to different labels on the basis of time stamp and events. It combines multi-instance learning and multi-labeling for classifying the events for decision support. Aim of the proposed method is to

- enhance the intelligence of the system through fast supervised learning mechanism
- self-organize and adapt to the environment dynamics
- enhance the potential to detect and combat unknown attacks

and make it inevitable for MANET environment. Nevertheless, in a virtually clustered non-uniform network topology, it's distributed and centralized self-adaptive intrusion detection

and prevention approach ensures improved classification accuracy and decision convergence for securing C3I system in a Tactical Battlefield (TBF) [7] environment.

Simulation for the proposed method was conducted using MATLAB Simulink. A prototype was developed to demonstrate the practicality and flexibility of using Smart MIML-IDPS in Tactical mobile ad-hoc network. Experimental analysis shows the proposed method is an attack resistant lightweight IDPS mechanism. When tested with high node density, it provides less computation complexity and low communication overhead and proves to be power efficient, faster, and highly secured on devices over existing methods. This scheme has made the security system stable by detecting ~90% - ~95% of malicious nodes under various attacks in a collision constrained environment. Performance analysis proves the proposed approach to be an attractive scheme with good potential to be included in TBF environment. The rest of the paper is organized as follows: in Section 2, relevant previous work is reviewed, Section 3 presents an overview and detailed architectural specification of the proposed scheme. Section 4 evaluates the performance of the system, Section 5 presents the system's experimental evaluation. In Section 6 the paper concludes along with a discussion on future directions.

2. Related Work

Several research works on intrusion detection [8, 9] such as anomaly based, knowledge based and specification based techniques have been proposed. In this section we review the previous work related to IDS scheme.

Marti et al proposed a scheme named Watchdog [10] based on standard Dynamic Source Routing (DSR) [11] protocol. This scheme detects malicious nodes misbehaviour by eavesdropping on the transmission of the next hop in network. The standalone watchdog module is deployed in each node to identify malicious activity. By overhearing, the Watchdog node increments its failure counter if the next node fails to forward the packet within certain period of time. When the failure counter exceeds a predefined threshold value, then the next node is considered to be misbehaving and reported as malicious. Though this scheme proved to be efficient, it lead to false accusations as it failed to detect misbehavior in the presence of limited transmission power and collision constrained environment. ie., accuracy in monitoring the neighbourhood degraded when each node has different transmission ranges.

The weakness of Watchdog scheme was considered by Liu et al in TWOACK [12] approach. This approach uses acknowledgment-based detection technique to detect misbehaviour in the network. In this approach, TWOACK detects misbehaving links by acknowledging every data packet transmitted over every three consecutive nodes along the path from the source to the destination. When a packet is received, each node along the route is required to send back an acknowledgment to the node that is two hops away from it down the route. Though the receiver collision and limited transmission power problems posed by Watchdog is successfully solved by this scheme, the acknowledgment required in every packet transmission added a significant amount of unwanted network overhead. Such redundant transmission degrades the life span of the network. Moreover, it was extremely important to ensure that all acknowledgment packets are authentic [13] and untainted. Otherwise this scheme will be vulnerable as the attackers are smart enough to even forge acknowledgment packets.

Enhanced Adaptive ACKnowledgment based Digital Signature Algorithm [EAACK (DSA)] [14] proposed by Shakshuki et al, authenticates if the destination node has received the reported missing packet through a different route. Digital signature is incorporated to ensure integrity of the IDS [15]. It requires all acknowledgment packets to be digitally signed before they are sent out and verified when they are accepted. However, this scheme requires extra resources with the introduction of digital signature in MANETs. Distributed algorithms for clustering and electing a head node to handle ID is recently proposed as a solution to this problem. Usually, conventional MANET use IDS such as:

- *Standalone IDS* – It requires IDS to be installed and executed on each node independently. Decision is made only on information collected from each node. This method lacks cooperation among nodes; therefore nodes in same network do not know anything about the situation on other nodes. Due to these limitations, this architecture is not suitable for MANETS.
- *Distributed and Cooperative IDS* – It requires every node to participate in ID by having an IDS agent running on them. These agents are responsible for collecting local events to identify possible intrusions and initiate response independently. Neighboring nodes cooperate in global ID when evidence is inconclusive. This mechanism suits well for standalone or flat network architecture but not for multilayered one.
- *Hierarchical IDS* – This method is proposed for multilayered architecture by extending distributed and cooperative IDS functionalities. Here network is divided into clusters and more functionalities are allocated to cluster head rather than to all the nodes. The cluster head is responsible for detecting intrusions locally for itself and globally for its neighbors. Lack of load sharing among nodes is its major problem. While, securing nodes that perform IDS activities are their challenges.

Our proposed method is a colloboration of distributed and centralized IDS mechanism for identifying and preventing vulnerability against major security attacks and sustaining the performance in MANET.

3. System Design and Architecture

Let us consider a dynamic network configuration, where the topological connectivity is subject to frequent unpredictable changes. The network consists of mobile nodes randomly distributed. Node in the network operates in one of the roles - either as a Virtual Cluster Node (VCN), or as a Virtual Cluster Head (VCH). Network is accompanied with a centralized Command Control Center (CCC) – the server. Virtual cluster heads are dynamically elected to acts a client ID agent that serve as the intelligent decision processing unit while the centralized CCC acts as server IDPS and serves as the smart MIML decision making unit. Network model comprising the proposed Smart MIML-IDPS scheme is depicted in **Fig 1**.

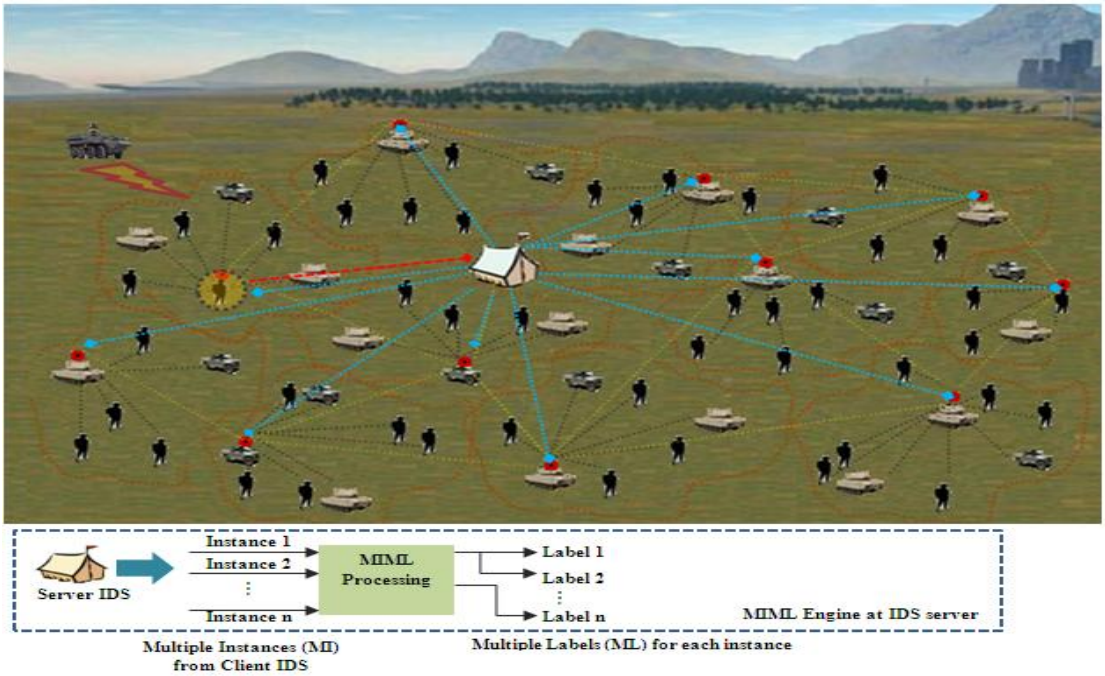

Fig. 1. Network Model of Smart MIML-IDPS System

Proposed Smart MIML-IDPS architecture includes the following components:

- **Virtual Cluster Node (VCN):** These nodes form the members of the virtual cluster and communicate with other nodes to exchange information. These nodes maintain its local data and VCH identifier information in its Local Aware Table (LAT).
- **Virtual Cluster head (VCH):** It is the head node of the cluster and its primary role is to monitor and analyze malicious activity within the cluster. It acts as client ID agent and empowered with ID capability. It collects information from its VCNs to determine misbehaving activities that violate the security rules within cluster. When an unusual activity is determined, it notifies security administrator for further investigation on such events.
- **Command Control Centre (CCC):** It is the most vital part of the architecture empowered with Smart MIML-IDPS engine support. It is responsible for detecting unusual network instances and initiating a global response for preventing such intrusion. It makes use of Smart MIML-IDPS service to categorize and classify instances either to normal or abnormal labels. The multi-layered Smart MIML-IDPS service engine exhibits computational intelligence and fast learning through MIML technique for efficient memory utilization, improved classification accuracy and decision convergence.

Table 1 displays the summary of nomenclatures used in the proposed scheme.

Table 1. Summary of nomenclature used in Smart MIML-IDPS mechanism

Notations	Description
$\mathbf{M_n^i}$	Mobility range of i^{th} node
$\mathbf{Tp_n^i}$	Transmission power range of i^{th} node
$\mathbf{Sc_n^i}$	Storage capacity of i^{th} node
$\mathbf{Mem_n^i}$	Memory value of i^{th} node

$\mathbf{V_{count}}$	Count value of virtual cluster head
$\mathbf{Mem_n^i N_n^i}$	Mem_n^i memory of i^{th} node &N_n^i i^{th} neighbor node
$\mathbf{M_{label}}$	Master label- list containing combination of good and bad labels list.
Mnode	Monitoring node
Mnode	Monitored node
#(∗, mnode)	Number of incoming packets on the monitored node mnode.
#(mnode, nhop)	Number of outgoing packets from mnode of which nhop is the next hop.

Our proposed Smart MIML-IDPS mechanism is categorized into the following phases:

- Phase I: Virtual Cluster Formation and Head Selection
- Phase II: Smart Multi-Instance Multi-Label Intrusion Detection and Prevention System (Smart MIML-IDPS) Mechanism

Flow model of the proposed scheme is referred in **Fig. 2.**

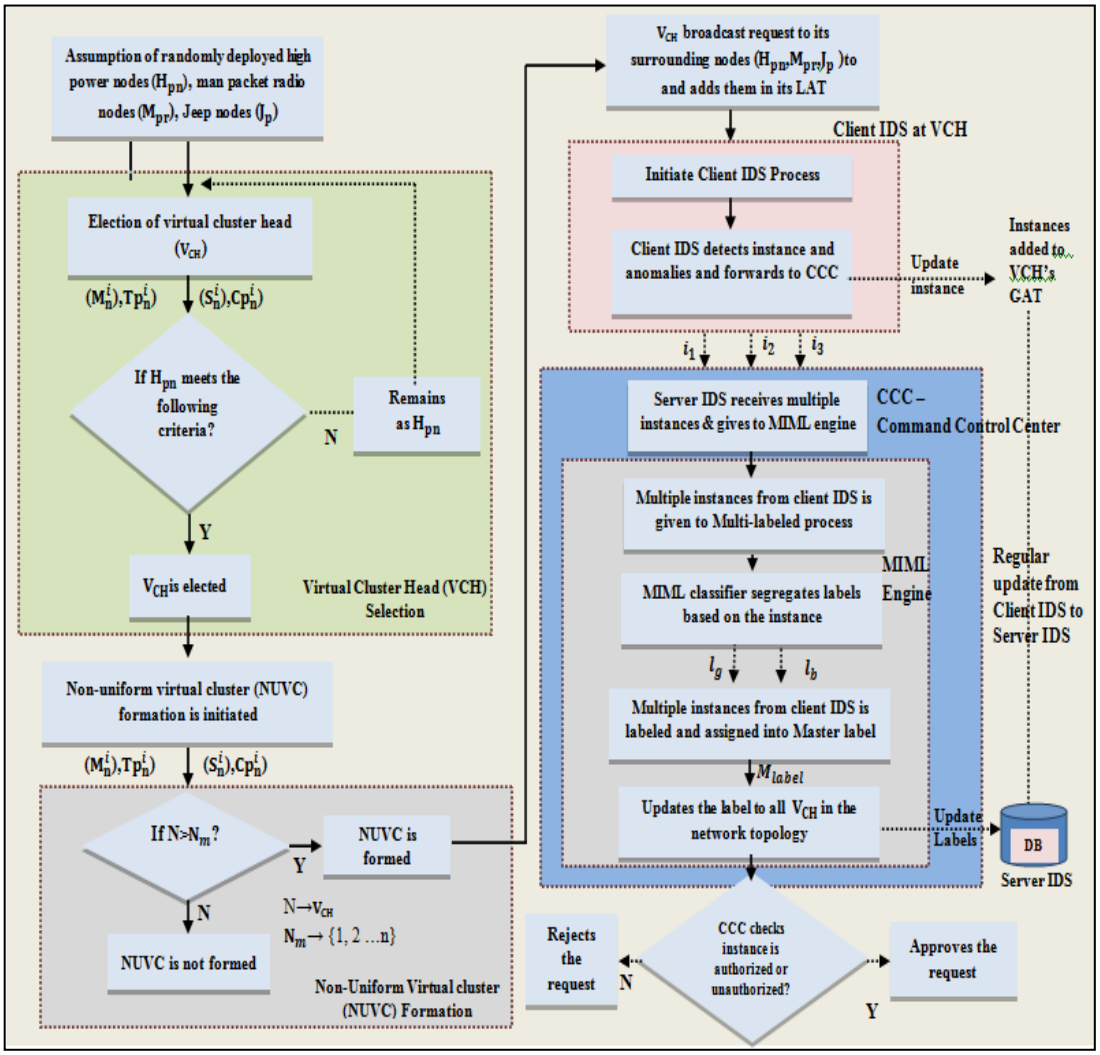

Fig. 2. Flow model of Smart MIML-IDPS scheme

3.1 Phase I - Virtual Cluster Formation and Head Selection

Topology of ad-hoc network often changes dynamically. The proposed system should be able to self-organize and re-configure by itself to these variations without manual interventions. Primary objective of virtual cluster formation and head selection is to help the system to self-adapt to dynamic changes. Initially, nodes in non-uniform distribution is randomly selected to form Virtual Clusters (VC). VCs are formed in such a way that the resulting network is virtually cluster connected. VC construction is based on the following conditions; i) each cluster should have optimal range of nodes ($n_{optimal}$), ie.,
$n_{min} \geq n_{optimal} \leq n_{max}$, where $n_{optimal}$ indicates the optimal number of nodes, n_{max} and n_{min} indicates maximum and minimum number of nodes within a cluster. The n_{max} and n_{min} increase or decrease depending on the coverage area for effective cluster formation. ii) overlapping between the inter-clusters should be avoided to prevent anonymous nodes that may fall out of the transmission range of VCH. Similarly, it is assumed that,

- VCH is aware of all its neighbors.
- prior to data communication, control packets are transmitted by nodes to VCH for authentication.
- all control messages are processed one at a time by VCH in the order in which they occur.
- all packets transmitted over a link are received correctly and in proper sequence within a finite time.

Fig. 3 illustrates VC formation and VC head selection in Smart MIML-IDPS.

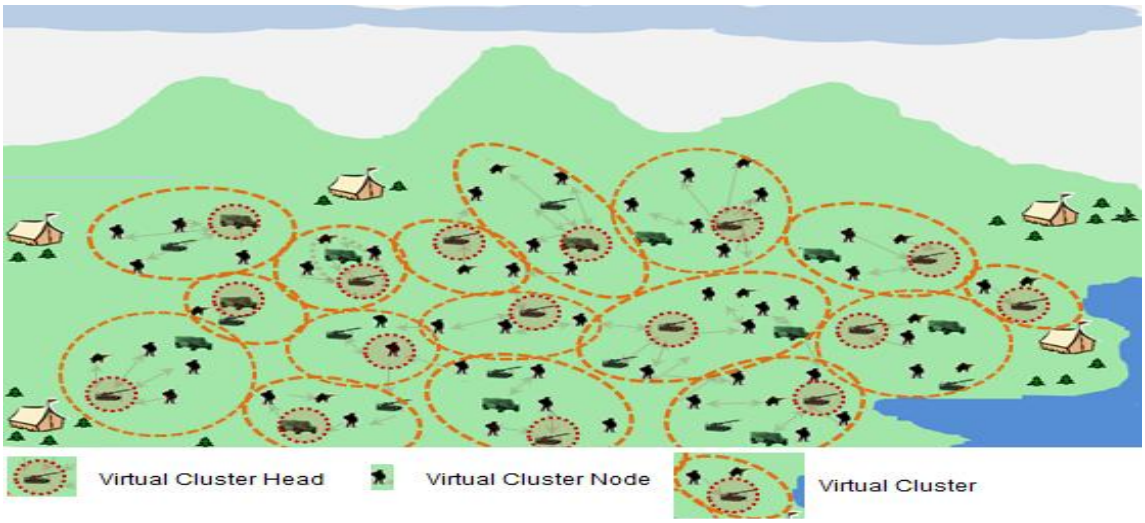

Fig. 3. Virtual Cluster Formation and Head Selection in Smart MIML-IDPS

VC formation among nodes is initiated through Special System Information Message (SSIM) broadcast. The steps involved in VC formation and head selection in Smart MIML-IDPS mechanism is as follows:

Step 1: Initially, let node 'A' broadcast "Hello" message (SSIM) for discovering its neighbors. The "Hello"message consists of the node's - identity(N_{Id}), mobility(N_{Mb}), transmission power(N_{Tp}), computation capability(N_{Cc}) and storage capacity (N_{Sc}). The message is broadcasted at a low power level. The aim of using low power is to send the message only to neighbors who are in close geographic vicinity and to save power.

Step 2: Upon receiving the "Hello" message, neighbor nodes registers sender's SSIM information in its Local Aware Table (LAT) and responds with "Ack_Hello" message back to the sender. Sender receives "Ack_Hello" message and records SSIM information about its neighbors.

Step 3: Using the information recorded in LAT, each node calculates its as well as its neighbor nodes score by considering factors such as its mobility(N_{Mb}), transmission power(N_{Tp}), computation capability(N_{Cc}) and storage capacity (N_{Sc}).

Step 4: Node with low mobility (N_{Mb}), high transmission power(N_{Tp}), optimal computation capability(N_{Cc}) and storage capacity (N_{Sc}) is given a higher score. Each node compares its score with its neighbors. Among the set of nodes, the node with highest score is considered for VCH selection candidate and is elected as VCH i.e., if a node 'n_i' has greatest score in its neighborhood, it declares itself a Virtual Cluster Head.

Step 5: Node 'n_i' declares itself as a VCH by propagating "VC Formation" message to its neighbors. Neighbor nodes which listen to this message become part of the virtual cluster by sending a "Virtual Confirmation Message". Once a neighbor node becomes a part of a virtual cluster, re-current messages for VCH election or formation are discarded by those nodes. After the election, the VCH updates its LAT – table that maintains SSIM information of neighbors. Up-to-date view of group members and their registered identity information are managed by VCH in its LAT. Finally, VCNs becomes the members of the virtual cluster.

VCH rotation is triggered after every 't_i' time slot (where 't_i' depends specific to application). ie., after 't_i' time slot if node 'n_i' is found inappropriate (by comparing updated scores of nodes) to act as a VCH, it sends a "VCH Message" to node 'n_r' (node with highest score) to become a VCH and waits for a fixed duration of time 'δ' to receive a response back from the node 'n_r' . If the node 'n_r' or any other node in the neighborhood fails to respond with "Virtual Cluster Formation" message, then the node 'n_i' itself remains to be a VCH. **Algorithm 1** depicts the steps involved in virtual cluster formation and **Algorithm 2** illustrates virtual cluster head election process in Smart MIML-IDPS mechanism.

Algorithm 1. Virtual Cluster Formation in Smart MIML-IDPS

$N_A \rightarrow$ SSIM_nodeA(N_{Id} ,N_{Mb}, N_{Tp}, N_{Cc} ,N_{Sc}); // Node A broadcasts SSIM
$N_B \leftarrow$ SSIM_nodeA(N_{Id} ,N_{Mb}, N_{Tp}, N_{Cc} ,N_{Sc}) // Neighbor Node B receives SSIM of N_A
If (N_B (Msg_nodeA) == 'T') then // Node B receives SSIM of Node A successfully
    $LAT_B \leftarrow$ fetch_data(N_{Id} ,N_{Mb}, N_{Tp}, N_{Cc} ,N_{Sc}); //fetch the data of Node A and store in LAT of Node B
    $N_B \rightarrow$ AckMsg; //Node B responds with "Ack_Hello" message
    If (N_A (AckMsg_nodeB) == 'T') then //Node A receives "Ack_Hello" message of Node B
        $LAT_A \leftarrow$ fetch_data(N_{Id} ,N_{Mb}, N_{Tp}, N_{Cc} ,N_{Sc}); // fetch the data of N_B and store in LAT of N_A
    end;
end;
/* Update SSIM for all neighbors nodes. Trigger Virtual Cluster Head identification among neighbors*/
    $Score_{data}$ = find_score(LAT_A); // score or weight of each node is calculated
    $Score_{high}$ = get_highest_score($Score_{data}$); // compare the scores of nodes to find the highest score
    $Node_i$ = get_node($score_{high}$); // node with highest score is identified.
    VCH_{node} = $Node_i$;
    $VCH_{node} \rightarrow$ VCH_fmsg; // VCH propagates cluster formation message to neighbors
    $Node_{neighbor} \leftarrow$ VCH_fmsg; // Neighbors receives VCH message and becomes part of cluster
    $Node_{neighbor} \rightarrow$ VCfrm_msg; // Neighbors send confirmation message to VCH_{node}

Algorithm 2. Virtual Cluster Head Selection Process in Smart MIML-IDPS

1. Initialize M_n^i , Tp_n^i, Sc_n^i , Mem_n^i, V_{count} = 0, j=1;
2. for set of 'n' nodes
   if ($N_{VCH}^i(M_{VCH}^n) \leq M_\delta$) &&(N_{VCH}^i (Tp_{VCH}^n) $\geq Tp_\delta$) ||($N_{VCH}^i(Sc_{VCH}^n) \geq Sc_\delta$)&&($N_{VCH}^i(Mem_{VCH}^n) \geq Mem_\delta$) then
   $Best_{VCH}$ = store_data (M_{VCH}^n, Tp_{VCH}^n, Sc_{VCH}^n,Mem_{VCH}^n);
   $V_{count} = V_{count} + 1$;
   end;
   end;
3. if $V_{count} > P_{VCH}^i$
   // $NBest_{VCH}$ is number of nodes satisfying the norms, δ_{adj} is adjusted threshold limit, O_{VCH} is Optimal VCH
   for i=1 to $NBest_{VCH}$
   $optimal_{count} = 0$;
   if ($Best_{mob}^i \leq \delta_{adj}$) &&($Best_{tp}^i \geq \delta_{adj}$)|| ($Best_{sc}^i \leq \delta_{adj}$) &&($Best_{mem}^i \geq \delta_{adj}$) then
   O_{VCH} = store_data ($Best_{mob}^i$, $Best_{tp}^i$, $Best_{sc}^i$,$Best_{mem}^i$);
   $optimal_{count} = optimal_{count} + 1$;
   else if $V_{count} < P_{VCH}^i$
   re-adjust threshold limit;
   repeat step 2;
   end
   end
   end;
4. Repeat step (2) and (3) for effective VCH selection.

The frequency for re-electing VCH depends on application.

Non-Uniform VC and head selection ensures cooperativesness among mobile nodes. Details of Smart MIML-IDPS architecture specification is illustrated in Phase II.

3.2 Phase II - Smart Multi-Instance Multi-Label Intrusion Detection and Prevention System (Smart MIML-IDPS) Mechanism:

Primary focus of IDPS is to identify vulnerable incidents and initiate preventive measures. Just analyzing a small part of the entire event information may not prove to be a sustainable approach for protecting the whole network. Rather, one must dig deep in-order to truly understand security threats caused within the network. The proposed model makes it possible to have a better understanding of an incident when it occurs rather than analyzing a single or a number of distributed unconnected logs. It brings together multiple layers of technology into a single monitoring using a multi-layered MIML-IDPS service engine that works on system logs (sys logs), IDS logs, firewall logs etc. Aim is to:

- conceptualize large amount of data (collect logs and events from different location) through aggregation and provide event linkage among them to make it easier and quicker for monitoring and analysis.
- create a single source of valuable information which is clear, comprehensive and concise for decision making.
- maintain major security logs at centralized location thereby permitting techno-logistical detection to happen within short duration and respond faster without having to scrounge across the network and nodes for related logs.
- reduce number of time consuming steps during the analysis.
- make the network more secure by bringing a whole new view and depth of operation.
- increase situational awareness.

Most prominent activities of Smart MIML-IDPS are categorized as follows:

1. Client ID Agent's Rule Registration and Activation
2. Security based MIML-IDPS Server Support Specification

3.2.1 Client ID Agent's Rule Registration and Activation: Within each cluster, security is enforced by empowering VCH with intrusion detection capability. VCH acts as client ID Agent. Role of VCH is to verify the authenticity and permit valid VCNs and reject invalid VCNs from further communication. The client ID module plays a vital role in determining the members who violate security rules. Such violated incidents (abnormal activities) are sent to CCC for further investigation. Steps involved during client ID agent's rule registration and activation in Smart MIML-IDPS mechanism is as follows:

Step 1: The selected VCH (client ID agent) sends client registration (C_{reg}) notification to CCC regarding its election.

Step 2: CCC receives the notification, registers the VCH in its client ID agent registration (CID-C_{reg}) list. Similarly, through client registration mechanism the client ID agent information received from various VCH across the network are received and registered into the list. The updated list provides complete information of newly elected VCH (along with its registered members) across the network.

Step 3: Using the latest updated member list, MIML-IDPS engine generates new set of IDPS rule ($IDPS_{rule}$) through the IDPS rule generation (IDPS-R_{gen}) mechanism. The generated IDPS rule consists of the latest white (valid IP and Port) and black (invalid IP and Port) list candidates. To avoid complexity, in this paper, we have considered two constraints such as IP address and Port information. Various other important and relevant features such as time, length, sequence number etc., can be included to enhance the accuracy is estimating host or suspicious activity during authentication.

Step 4: Next, CCC sends IDPS rule registration ($IDPS_{rule\_reg}$) response to VCH. This message consists of newly updated IDPS rules (list of white and black list candidate's IP and Port address).

Step 5: Upon receiving the message, VCH updates its LAT and GAT (Global Aware Table). The updated LAT and GAT provides consistent and up-to-date view of the network at a particular instant of time. In addition, GAT maintains the identity of other VCH and routing information of the network. VCH then initiates rule registration ($Rule_{reg}$) process in-order to update the IDPS rule with its members. It sends notification to all its members indicating,

i) Good IP and Good Port (GIP_GP) candidates: GIP_GP indicates valid IP and valid Port. These are the registered set of nodes that form the network. These nodes are authorized nodes to (or from) which data communication is permitted.
ii) Bad IP and Bad Port (BIP_BP) candidates: BIP_BP indicates invalid IP and invalid Port. These nodes are unauthorized malicious nodes to (or from) which communication is rejected.
iii) Grey List (GL) candidates: GL indicates unknown list of IP and Port details. Due to inadequate information, the incident may not be able to classify the node as either "good" or "bad" candidate. These types of nodes are categorized as "grey" candidates. If GL nodes are identified such incidents are sent to CCC for authentication.

Apart from IP address and Port data, other constraints such as time stamp, sequence number, length etc., are also used as parametric information to validate the authenticity of nodes prior to data communication. VCNs using the registered rule either approves or rejects the request from white or black list candidates and subsequently reports incidents that are abnormal to its VCH.

Step 6: Apart from rule registration mechanism, VCH activates the light weight client ID module for detecting suspicious activities. Aim of the module is to determine abnormal behavior and report them to CCC for analysis and authorization. Abnormal behavior plays a

vital role in deciding the malicious activity. For example, in Tactical battlefield setup, even registered nodes that do not communicate for longer period of time and suddenly sends request for data communication is considered to be abnormal. Communication request for such nodes are rejected and those request are forwarded to CCC. Here, even when the node is said to be registered member of the network, it's behavior has caused a suspicion and hence CCC blocks communication to such nodes and keeps the node in monitoring state. **Algorithm 3** depicts the details of Client ID agent's rule registration and activation mechanism.

Algorithm 3. Client ID Agent's Rule Registration and Activation Mechanism

```
VCH (C_reg) → CCC;
CCC ← VCH (C_reg) ;
initiate_registration(CID-C_reg);
rule_registration(IDPS-R_gen);
generate_new(IDPS_rule);
CCC → IDPS_rule_reg ;
VCH ← IDPS_rule_reg ;
initiate_rule_registration(Rule_reg);
enable_VCH(client ID);
set_VCH_GAT-LAT( IDPS_rule);
Let N_i ← N_A;
N_i = verify_LAT(N_A_ipp);
if (N_A_ipp == GIP_GP) then
        if (N_A_ipp(T_slot) == 'S') then        // S indicates suspicious indication
                Status = 'Block';
        else
                Status = 'Permit';
        end;
        if (Status == 'Block') then
                N_i = reject_req(N_A);
                N_i → send_node_info_to_VCH(N_A);
                VCH → fwd_info_CCC(N_A);
                CCC = set_node_status(N_A , 'B');
                CCC  = monitor(N_A);
        else
                N_i = approve_req(N_A);
                N_i → N_A;                      // N_i communicates with N_A
        end;
end;
if (N_A_ipp == BIP_GP) then
        N_i = reject_req(N_A);
end;
if (N_A_ipp == GL) then
        N_i → send_node_info_to_VCH(N_A);
        VCH → fwd_info_CCC(N_A);
        CCC = set_node_status(N_A , 'B');
        CCC = monitor(N_A);
end;
```

Primarily, VCN rejects invalid node's request by looking-up its LAT. While it approves the request for valid nodes provided the behavior of those nodes are said to be normal. In case, if any abnormal or uncertain behavior is noticed (for example: frequent variation in time stamp, abnormal change in mobility pattern, etc.,) then, communication for those nodes are blocked. Such abnormal incidents are forwarded to VCH for further analysis. VCH analyses such incidents and if the available evidence is inconclusive but requires a broader search to derive a conclusion, then it sends it to CCC for final decision-making. Client ID module in VCH

has the capability to analyze and detect local intrusions. While for events that are inconclusive, the MIML service engine at CCC tends to analyze and detect intrusions globally. This mechanism of distributing the Client ID module helps to share the load across the network and collaborate with centralized CCC for decision making.

3.2.2 Security based MIML-IDPS Server Support Specification: At CCC, the server IDPS engine implements a smart MIML-IDPS service mechanism, a promising approach that can significantly categorize unknown incidents (instances), classify them as white or black list labels using its clearly defined multi-layered specifications.
At CCC, the server IDPS agent performs the following steps:
Step 1: CCC receives aggregated message from client ID agent and forwards it to the server IDPS. Server IDPS constitutes a service engine that performs a "Smart MIML-IDPS" execution.
Step 2: Collective response from multiple client ID agents received by server IDPS is categorized based on incidents (similar request from various VCHs) and grouped together to form a similar set S = (x_i,y_i), $1 \leq i \leq n$, consisting 'n' instances. Instances ($x_i \in X$, $Y_i \in Y$) are then fed as input to the multi-layered smart MIML-IDPS service engine for processing.
Step 3: Smart MIML-IDPS service engine is a multi-layered process that optimizes each set using specific evaluation function to produce a multi-label classification. High computation capability of multi-layered engine is capable of mining instances faster and classify as per layered specification to appropriate labels and store them to the stationary secure database (SSD). The SSD has more storage capacity for storing all patterns of known and unknown signatures. These patterns are then used for rule mining and classification during MIML-IDPS service processing. The idea behind MIML-IDPS model is to reduce the computational complexity at each level and make the rule mining efficient. Detailed specification regarding multi-layered MIML-IDPS service engine is elaborated in 2.2.2.1 section.
Step 4: The results (or the multi-label classification of instances to good or bad candidates list) from the rule-based engine are collected and sent from the CCC to all the VCH across the network.
Step 5: VCH receives the response, updates it's LAT (and GAT) and forwards the new IDPS rule set to its VCNs for LAT updates. By viewing the LAT, VCNs either initiate or rejects communication for blocked nodes. **Algorithm 4** depicts MIML-IDPS mechanism.

Algorithm 4. The steps involved in the MIML-IDPS process.

Input: The instances i_1(attack 1), i_2 (attack 2), i_3(attack 3) received from VCHs is fed to MIML_IDPS engine.
$\bar{I}$ = transform_instances(i_1, i_2, i_3); // $\bar{I}$ denotes the multi-instance bags (X_i, Y_i, Z_i).
Initialize $i_{count} = 0$; j=1;
i_{count} = getCount(I); // get the number of instances (i_{count})
for j = 1 to i_{count}
  repeat // repeat for each instance and validate multi-instance bags X_i, Y_i, Z_ibag
    if ($i_j == X_i$)then // verify the presence of 'black hole attack' instance in the bag
      if ($Hc_n^i < D_n^i$)&& ($Sq_n^i > D_{sq}^i$)then //verify hop count and sequence number and label (l) the instance
        Assign $i_j = l_b$; //Assigned as bad label as hop count is low and sequence number is high
      else
        Assign $i_j = l_g$; //Assigned as good label
      end
    else if($i_j == Y_i$) then // verify the presence of 'Man-in the-middle attack' instance in the bag

```
        if(S_IP^i ∉ D_IP^i)&& (S_MAC^i ∉ D_MAC^i)then  // verify IP,source and destination address to send and receive
        the packet
          Assign i_j = l_b;                    //Assigned as bad label
        else
          Assign i_j = l_g;                    //Assigned as good label
        end
    else if( i_j == Z_i)then                   //verify the presence of 'Denial of service attack' instance in the
bag
          if (Sc_n^i < N_n^i)&& (Mem_n^i < N_n^i)then  // verify the storage space and memory of neighbor nodes
            Assign i_j = l_b; /*bad label*/
          else
            Assign i_j = l_g; /*good label*/
           end
        else
                attack_status = classify_unknown_attacks(i_j);
                 if (attack_status = 'T' then  // verify the storage space and memory of neighbor nodes
                   Assign i_j = l_b;           //Assigned as bad label
                else
                   Assign i_j = l_g;           //Assigned as good label
                 end
        end
        j=j+1;
        until (j < i_count);  /* continue for all instances .
end /* end of for loop
collect_label_list();
add_to_Master_label( M_label)          // M_label is master label which has the complete list of good, bad and
grey labels.
send_to_VCH( M_label) → CCC.
```

Almost all activities in the proposed scheme are logged on a centralized system. The inspection of these logs not only detects and prevents the intrusions, but also helps to analyze and audit the extent of damage caused, trace back the attack etc.,. It is claimed to detect most of the suspected threats with minimum overhead. Fast mining to detect suspected instance more quickly makes the proposed approach suitable for Tactical Battlefield environment.

3.2.2.1 Server MIML-IDPS Service Engine Functionalities: The multilayered MIML-IDPS service engine performs the following activities:

Data Representation and Encoding: In this stage, incidents or events collected as raw information in different format are processed and converted to standard format, encoded (encoded information here in after is referred as 'instances') and stored into the Stationary Secure Database (SSD). This layer is vital as the encoding of information is critical to the rest of the process. Primarily, the packet header portion is parsed and stored into SSD.

Discovering Relationship and Generating Rules: During this stage, the relationships among instances are identified by verifying the rules in the rule set archive. Rule set archive acts as a knowledge base for storing the rules. It plays a vital role in categorizing and aggregating instances into appropriate set through semantic information handling, inference reasoning and event correlation. Relationships and similarities found among instances forms the base (metrics) for rule generation. For example, classification similarity (based on the type of attack), time similarity (time when the attack happened, time when the attack was detected), source similarity (source that triggered the attack) etc., are used during

relationship discovery for grouping instances. During network initialization, rules are generated and added to rule set for known attacks (which has per-defined norms). While when the network is in operational stage, new rules are dynamically created and added to the rule set for unknown attacks (which do not have any pre-defined norms but are identified on the fly). Usually, rules are created using traffic and non-traffic related statistics. Various types of traffic related patterns (example, the number of packet received, the number of packet forwarded, the number of route reply messages, etc.,), statistics (example, route statistics such as route count, average route length, route updated etc.,), features and routing operations are used for detecting intrusions. While, trace logs maintained in each node is used for capturing non-traffic related statistics. The derived statistics is considered to be an attack if it deviates from pre-computed results using the existing rule set.

Several identification rules are pre-defined for known attacks by using relationships of the mentioned statistics. Once a deviation or variance (irregularity) is detected, investigation is performed by the Smart MIML-IDPS to determine the detailed information of the attack from a set of these identification rules. These rules enhance the system to identify the type of the attack and, in some cases, the attacking node. For well known attacks, the identification rules are defined as follows.

Rule for identifying black hole attack: This rule uses Global Forward Percentage (GFP) and it relies on information available on the M_{node} node. Let $N(M_{node})$ denote M_{node}'s 1-hop neighbors.

$$GFP^{\,s\_node}_{\,m\_node} = \frac{\#^{L}(*,Mnode) - \#^{L}(*,[Mnode])}{\sum_{i\epsilon N(Mnode)} \#^{L}(i,Mnode) \; - \sum_{i,j\epsilon N(Mnode)} \#^{L}(i,[j]) - \#^{L}(*,Mnode)} \quad (1)$$

If packets from N(MNode) destined to other nodes than itself or another N(Mnode) is not zero and

$GFP^{\,s\_node}_{\,m\_node}$ = 1, it means that the blackhole attack is detected and Mnode is the attacker.

Rule for identifying Unconditional Packet Dropping: This rule uses Forward Percentage (FP) over a period L to define the attack.

$$FPm_{node} = \frac{\#^{L}(mnode,Mnode) - \#^{L}([mnode],Mnode)}{\#^{L}(Mnode,mnode) - \#^{L}(Mnode,[mnode])} \quad (2)$$

Unconditional packet dropping attack is said to occur if there are packets to be forwarded and $FPm_{node} = 0$, and the attacker is monitored node m_{node}.

Motive behind maintaining the rule set archive is to instinctively search, verify and derive a solution to features concerning a problem. Nevertheless, the final solution is either a failure (mark as 'black' label) or a success (mark as 'white' label) derived using the MIML process.

Multi-Instance Multi-label Processing: The rules generated in earlier stage play a vital role in categorizing and aggregating instances into appropriate set of vectors. Multiple such sets are generated during this process, where each set consist of multiple instances. These instances are processed by the MIML engine layers to categorize into appropriate labels. In this section we investigate the multi-instance and multi-label learning considering the ambiguity in input and output spaces simultaneously. Assume that instances fed to the input units are $A^k = A^1, A^2, \ldots A^n$ where k =1, 2, 3... n, 'k' refers to the index of the instance and 'n'

is the number of input units. The value fed as input to the second-layer 'j' from the first-layer unit 'i' is:

$$BIn_{ij} = \exp - (\frac{A_i{}^K - \theta_{ij}}{\alpha_{ij}})^2 \quad (3)$$

Where θ_{ij} and α_{ij} are the responsive center and the responsive characteristic width of the Gaussian weight connecting unit 'i' with unit 'j'. The second-layer unit 'j' computes its activation value according to:

$$b_j = f - \left(\sum_{i=1}^{n} BIn_{ij} - \theta_j\right) \quad (4)$$

Where θ_j is the bias of unit 'j', 'f' is the sigmoid function, $f(u) = \frac{1}{1+e^{-u}}$. A leakage competition is carried out among all the second-layer units and the output is transferred to related third-layer units. The activation value of the third-layer unit 'h' is computed according to:

$$C_H = f\left(\sum_{i=1}^{n} b_j v_{jh} - \theta_h\right) \quad (5)$$

Where b_j is the activation value of second-layer unit 'j' connecting with unit 'h'. v_{jh} is the weight for second-third-layer and it is always 1. θ_h is the bias of unit 'h'. The output is transferred to fourth-layer units. The activation value of the output unit 'd' is computed according to:

$$d = C_H w_{hl} \quad (6)$$

where C_H is the activation value of third-layer unit 'h'. w_{hl} is the weight connecting third-layer to output layer. The error (E_{rr}) between real and expected output is computed. If the error (E_{rr}) is in the allowable range, it means that current instance is covered by an existing attracting basin. Then θ_{ij} and α_{ij} of the Gaussian weights connecting with the second-layer units are adjusted. If the E_{rr} is beyond the allowable range, it means that current instance is not covered by any existing attracting basins and need to find third-layer with minimum error. The unit 'u' whose characteristic error is the minimum among the entire third-layer unit is selected. If the error (E_{rc}) is in the allowable range, it means that the internal output classification represented by unit 'u' is applicable to the current instance. Also, it is the internal input classification represented by the second-layer units that should be adjusted. Thus, the unit whose activation value is the maximum among those connecting with unit 'u' is selected. If the error (E_{rc}) is beyond the allowable range, it means that both internal input classification and internal output classification is inadequate for current instance. Thus, two units are appended to the hidden layers, one in the second-layer and other in third-layer. The new second-layer 's' is connected with all the input units. The responsive centers of the Gaussian weights are respectively set to the input components of current instance, and the responsive characteristic widths are set to a default value. The new third-layer unit is connected with all the output units. **Algorithm 5** represents the steps involved in the internal layers of the MIMIL process.

Algorithm 5. The steps involved in the internal layers of the MIMIL process.

E_{rr} = Compute$_{error}$ (R_{output},E_{output}); // The error E_{rr} between real and expected output is computed
if (E_{rr} <=A_{range}) then // A_{range} is allowable range
    A_{value} = AdjustS$_{layer}$ (θ_{ij}, α_{ij}); //The $\theta_{ij}$and α_{ij} are adjusted.
    G_{label} = classifyLabel (E_{rr}, $C_{Instance,}$ A_{value}); //current instance is classified as good label
else if (E_{rr} >A_{range}) then
    T_{layer} = findMerrorTLUnit(); // find third-layer unit with minimum error for CI
    E_{rc} = ComputeCharacteristic$_{error}$ (sendtoT_{layer}($C_{Instance}$));
    If (E_{rc} <=A_{range}) then // A_{range} is allowable range

G_{label} = classifyLabel (E_{rc}, $C_{Instance}$);
else if (E_{rr} >A_{range}) then
$RC = IO_{Cinadequate}(I_{list})$;
If (RC < Th_{value}) then
G_{label} = classifyLabel (E_{rc}, $C_{Instance}$);
else
B_{label} = classifyLabel (E_{rc}, $C_{Instance}$);
end;
else
$I_{list} = add_{information}$ ($C_{Instance,}$, E_{rr}, E_{rc});
end;

Finally, the MIML engine initiates computation using the multi-instance vector in-order to classify them to appropriate labels. MIML engine considers the following criteria during classification:
Criteria 1: Instances satisfies conditions as per the pre-defined rule set (rules set for known attacks), then those are classified as "bad labels".
Criteria 2: Instances does not satisfy conditions as per the pre-defined rule set (rules set for known attacks), then those are processed to generate error ($\mathbf{E_{rr}}$)($\mathbf{E_{rc}}$) value, difference between the real and expected output, using which the classification is performed:

- If error value is within allowable range (The responsive centers and responsive characteristic widths of the Gaussian weights) then they are classified as "good" label.
- If error value is beyond allowable range then they are classified as "bad" label.

Criteria 3: Instances may not be classified either as "good" or "bad" label when the available information is inconclusive for decision making. Those are classified as "grey" labeled and categorized either to "good" or "bad" when adequate information satisfying the criteria is met based on monitoring.
MIML processed and labeled instances (classified either as valid/white/good or invalid/black/bad candidates) are stored in SSD i.e., the engine maintains the attack signatures and patterns of normal and abnormal behaviors in a SSD.

Decision Support and Reporting – Intrusion Detection and Prevention Process: During this stage, various patterns of known and unknown signatures are data mined and classified for decision making. High computation capability of multi-layered smart MIML-IDPS service engine makes it possible to generate and mine rules faster making the system more suitable for Tactical setup. It's capability facilitates storage, retrieval and visualization of tactical data thereby providing effective decision support to the commanders.

4. Performance Evaluation

In this section we concentrate to better investigate and evaluate the performance of various protocols and illustrate their scope in handling security vulnerabilities in Tactical MANET. We have chosen DSR, Watchdog, EAACK (DSA), and Smart MIML-IDPS as representative protocol for detailed evaluation and analysis.

4.1 Performance analysis for known attacks

MANET suffers from all-weather attacks, which can come from any node that is in the radio range of any other node in the network. The attacks mainly include passive eaves dropping, leakage of secret information, gray hole, black hole, worm hole, denial of service etc.,. To

better investigate the performance of DSR, Watchdog, EAACK (DSA) and Smart MIML-IDPS schemes various scenarios with different types of known misbehavior is considered for investigation:

Scenario 1: In this scenario, black hole attack insists the malicious node to simply absorb and drop the legitimate data packets it receives causing information to be lost.

To combat dropping packet attacks, nodes within listening range keep track of control packets that are sent by one node but not forwarded by the next. When the number of dropped packet reaches a threshold level, nodes that exists within the listening range sends Route Elimination Packet (REP) informing others about the blacklisted node. REP message consists of the malicious node's identifier (MN_{id}), the sending node's identifier (SN_{id}) and signature (S_{ig}). Nodes that receive a REP will break their routing links through that node, isolating it from the network. Though Watchdog scheme proved to be efficient in detecting malicious nodes in normal setup, it failed to detect such behaviors in the presence of receiver collisions and limited transmission power. While on the other hand, the digitally signed acknowledgement packets in EAACK (DSA) ensure authenticity and integrity of IDS making the scheme more reliable than Watchdog. Misbehavior detection in EAACK (DSA) was comparatively high, but significant amount of unwanted network overhead caused by acknowledgement packets in EAACK (DSA) led to degrade the life span of the entire network. The proposed Smart MIML-IDPS approach ensures fairness and secured communication through virtual cluster formation and head election. In this approach, instead of every node capturing all the features themselves and analyzing them for possible intrusion, the VCH makes itself solely responsible for capturing and analyzing traffic related statistics. As VCH is in-charge for authenticating registered members during communication, identifying misbehaving members becomes easy and such nodes are eliminated in early stages preventing multiple attacks. Apart from reducing individual nodes energy consumption, the detection accuracy is noticeably improved.

Scenario 2: In this scenario, the malicious node absorbs and drops the legitimate data packets they receive and sends back a false misbehavior report to the originator as a response whenever needed.

Malicious attackers generate false misbehavior reports and send it to originator to falsely report innocent nodes as malicious. This attack can be lethal to entire network when the attackers' breakdown adequate nodes to cause network division. Watchdog approach fails to detect misbehaving nodes in the presence of false misbehavior reports. To combat this attack, rather than just adopting acknowledgment-based schemes, it also becomes crucial to guarantee the authenticity of those packets. EAACK (DSA) approach adopts digitally signed acknowledgement packets to address this problem. These packets authenticates only if the destination node has received the reported missing packets through a different route. By adopting an alternative route to the destination, misbehaving reporter node is avoided. Destination node upon receiving the acknowledgement packet verifies if the reported packet was already received. If not, it is trusted and accepted, otherwise it safely concludes that this is a false misbehavior report and whoever generated it is marked as malicious node. Unlike Watchdog, EAACK (DSA) is thus capable of detecting malicious nodes despite the existence of false misbehavior report. Similarly, in the proposed approach, the VCH - client ID agent, monitors activities that violate the security rules. The lightweight client ID capability enabled in VCH determines false misbehaviors and notifies those incidents to CCC for further investigation. This mechanism makes the overall process simple in detecting false

misbehaving nodes and excludes them from further communication until authenticated by server. Managing misbehaviors within each clusters and preventing the adoption of authentic acknowledgement packets makes the proposed scheme comparatively better than EAACK (DSA).

4.2 Performance analysis for unknown attacks

Most existing IDPS mechanisms are signature based designed to detect and tackle a particular category of network attacks. These existing schemes are either implemented on top of existing protocols or have independent modules added to the mobile nodes to tackle known attacks. Most among them estimates and detects known attacks caused in network using supervised learning mechanism. Nevertheless, the number of new security threats is likely to increase quickly in MANET. Such attacks pose substantial threats to critical or military applications, and may be hard to distinguish from normal communications. These types of attacks should be detected and prevented before they harm the network, system or data. Smart MIML-IDPS scheme defense such unknown attacks using behavioral MIML learning mechanism. This approach learns, analyzes, categorizes and labels various instances instantaneously which make it unique from other existing methodologies.

Let us consider a tactical war scenario, where the vicinity of the nodes is available to the enemy or to the intruder. In a Tactical MANET, nodes (like man pack radio, tankers, other military vehicle etc.,) have different mobility patterns with varied velocity. There is high possibility for these nodes to detach from the network very often and thrown into multiple attacks. Let us consider a node 'x' (N_x) which is isolated due to different mobility pattern from its original pattern (due to war situation). This anonymous node is selected for hacking by the intruder. The intruder tries to flood or masquerade the isolated node by injecting malicious worms or software to damage the network (IPS signature or policy is modified). Then the same node (Nx) is introduced back into the network. The node tries to communicate and associate itself into the network.

In such scenarios, existing approaches find itself hard in detecting such malicious activity caused by such insider node. Whereas, in the proposed approach, when the node (Nx) tries to associate itself as a member within a cluster, VCH receives the request from the node and verifies the authenticity, if VCH finds the node to be suspicious (as the time slot variation results in suspicious state), it blocks the node from communicating with others and forwards the request to CCC for further investigation. CCC analyzes various other factors (such as loss of time interval, time duration the node was not communicating with VCH, etc.,) of the node, if found suspicious, it adds the node to block list and moves the node to monitoring mode (the state during which the node is blocked from communicating with other nodes in the network, while its activities are monitored and analyzed by VCH and reported to CCC). The VCH gets permission from server to vigilant this node in a suspicious and monitoring mode. If the behavior of the node matches to its past behavior(as per the adhered IPS - Intrusion Prevention System policy) then the node is considered as normal node and moved into a clean state otherwise it is considered abnormal (behavior like invalid port request, etc.,) and the node is moved into malicious category and permanently disassociated from the network. This information is updated at CCC and passed to all VCH. Here, our proposed algorithm tends to classify the node either as a malicious or normal node by assessing its behavior (like ports allocated, continuous request made) during monitoring state.

5. Simulation and Experimental Analysis:

Performance of the proposed Smart MIML-IDPS approach was evaluated using the self-written script developed using MATLAB Simulink. To analyze the behavior of the proposed protocol our experiment considers a simulation area of 1000 m X 1000m with a pack of MANET nodes. Mobile nodes are randomly deployed and set to move across the simulated area with varying speed. To diversify nodes mobility, high power with large transmission range (Tanker or Military Jeep) and low power with small transmission range (Man Pack Radio) is considered. Mobility speed is set in the range between 0m/s - 20m/s and limited to 20 m/s. Initially all nodes remain static at 0 m/s during simulation. Node density varies from 5 to 10 nodes per group. Group radius for virtual cluster formation is set to 100 meters to ensure that VCH is directly communicable to its registered members within the group. Reference Point Group Mobility Model (RPGM) with physical layer speed of 1Mbps-2Mbps is considered. **Fig. 4,** displays the Smart MIML-IDPS network model captured during simulation. Experiment is performed with background traffic generated by 10% to 20% of the nodes in each scenario. Virtual time slots concept was used to implement message sending and receiving. A node is randomly chosen in each time slot to generate a new message and let it send the message to the destination node.

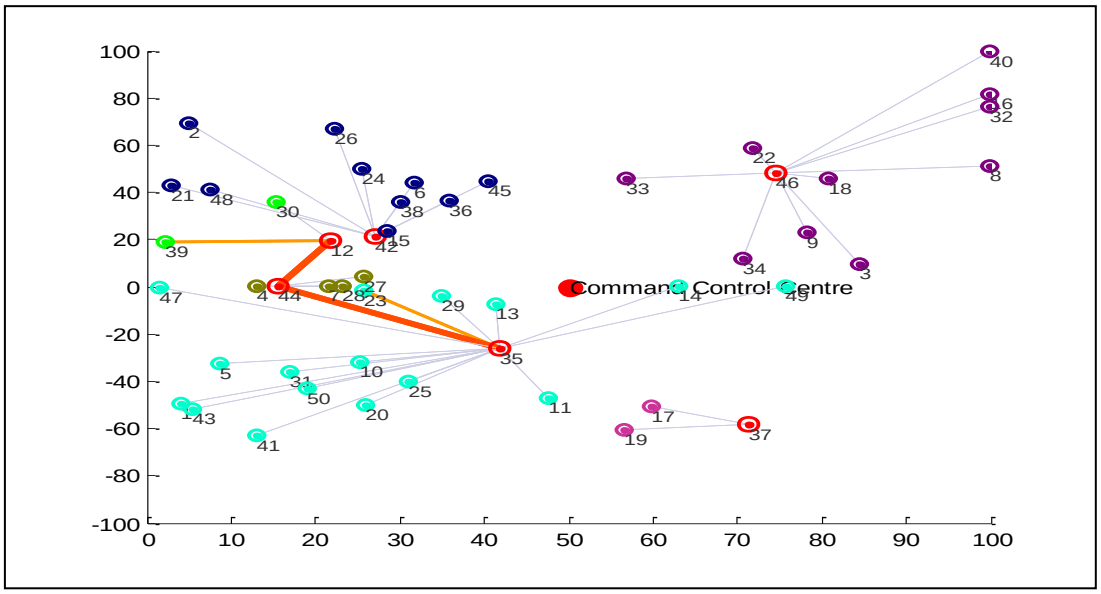

Fig. 4. MATLAB network model of Smart MIML-IDPS scheme

User Datagram Protocol (UDP) traffic with constant bit rate with a packet size of 512B is implemented. Nodes send packets of 512 B at a rate of ten packets per second. The simulation time was set to 500 time slots. Different deployments of mobile adhoc networks were generated with mobile nodes varying from 100 to 500 during the experiment. Average performance was evaluated by executing the network scenarios multiple times and by varying the mobile nodes. The desired delivery rate was set to be 99% (very high) and 85% (medium). Following metrics were used to evaluate the performance of the proposed and existing scheme.

Packet Delivery Ratio (PDR): Packet Delivery Ratio is the ratio of the number of packets successfully delivered to the destination against the total number of packets generated by the source. For better insight and comparative analysis among various approaches, simulation results for scenario1 and scenario2 are captured and presented in **Table 2** and **Table 3.**

Table 2. Scenario1 Packet Delivery Ratio

Scenario 1 : Packet Delivery Ratio					
	Malicious Nodes: 0%	**Malicious Nodes: 10%**	**Malicious Nodes: 20%**	**Malicious Nodes: 30%**	**Malicious Nodes: 40%**
DSR	1	0.82	0.73	0.68	0.66
Watchdog	1	0.83	0.77	0.7	0.67
EAACK(DSA)	1	0.96	0.97	0.93	0.91
Smart MIML-IDPS	1	0.99	0.99	0.98	0.97

Table 3. Scenario 2 Packet Delivery Ratio

Scenario 2 : Packet Delivery Ratio					
	Malicious Nodes: 0%	**Malicious Nodes: 10%**	**Malicious Nodes: 20%**	**Malicious Nodes: 30%**	**Malicious Nodes: 40%**
DSR	1	0.82	0.73	0.68	0.66
Watchdog	1	0.83	0.75	0.69	0.68
EAACK(DSA)	1	0.95	0.92	0.87	0.79
Smart MIML-IDPS	1	0.99	0.97	0.92	0.90

Scenario 1 Analysis: In scenario 1, the packets that pass through the malicious nodes are dropped. PDR data captured for various approaches are represented in **Fig. 5 (a)** and **Fig. 5 (b).** From the figure, **Fig. 5 (a)**, we observe that though EAACK (DSA) scheme performs better in delivering packets successfully to destination in the presence of malicious nodes when compared to DSR and Watchdog, the proposed Smart MIML-IDPS scheme demonstrates better PDR than EAACK (DSA). With 20% of malicious nodes, Smart MIML-IDPS surpassed Watchdog's and DSR's performance by ~25% - ~30% and EAACK (DSA) by ~3% - ~5%. Nevertheless, even when the number of malicious nodes is increased to 40%, proposed scheme was able to sustain malicious activity (by detecting and preventing misbehaviors) and remain stable in delivering packets successfully to destination when compared to other schemes. With the presence of 40% malicious nodes, Smart MIML-IDPS PDR was observed to be ~10%, ~30% and ~35% better compared to EAACK (DSA), Watchdog and DSR. Though EAACK(DSA) was able to detect malicious activity through authenticated acknowledgement, when it takes too long for the acknowledgement to reach the originator from the destination, the waiting time (predefined threshold) can exceed rejecting valid route for data delivery. This circumstance is avoided in the Smart MIML-IDPS approach, since the VCH itself is able to prevent suspicious behaviors among its registered members and avoids unregistered members to become part of the VC unless authenticated by the CCC. This enhances legitimate communication among nodes by consuming limited transmission power and minimizing receiver collision resulting in high PDR.

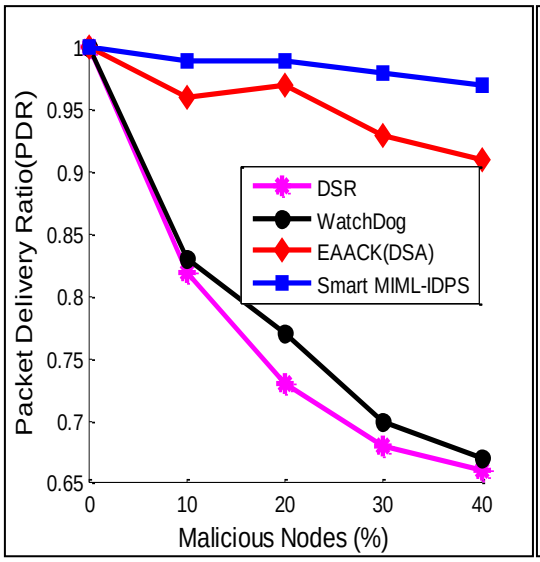

Fig. 5(a). Scenario 1 – Packet Delivery Ratio

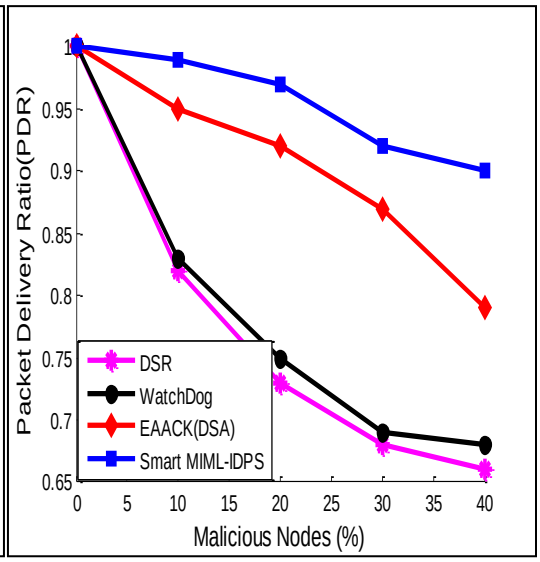

Fig. 5(b). Scenario 2 - Packet Delivery Ratio

Scenario 2 Analysis: In scenario 2, the malicious node sends false misbehavior report to the originator node. From the figure, **Fig. 5 (b)**, we can observe the PDR of the proposed scheme outperforms other schemes and maintains the PDR above 90% even when the malicious nodes density increases. Client ID module in VCH has the capability to determine false misbehavior activity among its members and such incidents are notified to CCC for further investigation. Additionally, the traffic load across the network is reduced in this scheme enhancing network performance. Though EAACK (DSA) has better PDR compared to Watchdog and DSR, its PDR starts degrading when the number of malicious nodes increases. It requires additional acknowledgement packet transmission thereby increasing traffic load and thus degrading the overall performance of network. From the result it is observed that when the malicious node is 40%, the Smart MIML-IDPS surpassed Watchdog's and DSR's performance by ~20% - ~23% and EAACK (DSA) by ~10% - ~12%. Average PDR for Smart MIML-IDPS is found to be ~5% - ~10% more compared to EAACK (DSA). Moreover, the PDR is sustained above 90% even when the number of malicious node increases.

Routing Overhead: Routing Overhead defines the ratio of number of routing related transmission. Simulation results of routing overhead for scenario1 and scenario 2 are captured and presented in **Table 4** and **Table 5** for comparative analysis among various approaches.

Table 4. Scenario 1 Routing Overhead

Scenario 1 : Routing Overhead					
	Malicious Nodes: 0%	**Malicious Nodes: 10%**	**Malicious Nodes: 20%**	**Malicious Nodes: 30%**	**Malicious Nodes: 40%**
DSR	0.02	0.023	0.023	0.022	0.02
Watchdog	0.02	0.025	0.025	0.023	0.023
EAACK(DSA)	0.15	0.28	0.35	0.44	0.58
Smart MIML-IDPS	0.03	0.034	0.036	0.033	0.025

Table 5. Scenario 2 Routing Overhead

Scenario 2 : Routing Overhead					
	Malicious Nodes: 0%	**Malicious Nodes: 10%**	**Malicious Nodes: 20%**	**Malicious Nodes: 30%**	**Malicious Nodes: 40%**
DSR	0.02	0.023	0.023	0.022	0.02
Watchdog	0.02	0.025	0.025	0.023	0.023
EAACK(DSA)	0.22	0.25	0.33	0.32	0.64
Smart MIML-IDPS	0.031	0.036	0.038	0.034	0.028

Scenario 1 Analysis: From the results as shown in **Table 4** and **Fig. 6(a)**, we can observe that the Watchdog and DSR outperforms EAACK (DSA) with low RO as they are not acknowledgement-based. This significantly reduces the RO transmission thereby reducing the network overhead. Whereas, acknowledgement is required at all levels for misbehavior detection in EAACK (DSA) resulting in high RO. Interestingly we can notice from the observation that the proposed scheme displays low RO similar to DSR and Watchdog. In Smart MIML-IDPS, as nodes do not exchange routing information but are used only for transporting IDPS messages which is handled by VCH and CCC. Thus, virtual clusters formation and head selection makes the system stable and efficient even when the malicious node increases making is more preferable for Tactical MANET.

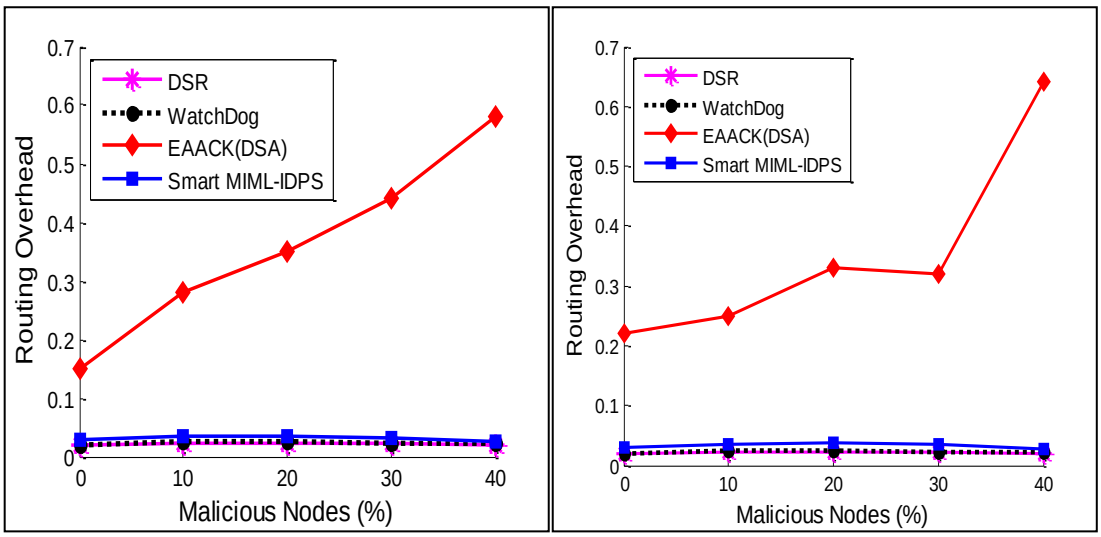

Fig. 6(a). Scenario 1 – Routing Overhead **Fig. 6(b).** Scenario 2 – Routing Overhead

Scenario 2 Analysis: In scenario 2, we observe from the results as shown in **Fig. 6(b),** that EAACK(DSA)'s RO rapidly increases when the number of malicious nodes increases to 40%. Fact being, more malicious nodes require a lot more acknowledgment packets and digital signatures. The best performance is achieved by DSR and Watchdog as they do not require acknowledgment for misbehavior detection. The RO difference between DSR and Watchdog against Smart MIML-IDPS is negligible, which indicates proposed scheme is able to resist RO when malicious nodes increases unlike EAACK (DSA), making it a more desirable scheme for MANET, as it mitigates the network collision and contention by reducing routing overheads so as to improve the Quality of Service (QoS).

CPU Consumption and Memory Utilization: By varying the network traffic and mobility every 5 to 10 minutes, the change the system's memory utilization and CPU consumption was analyzed and the summary of the results is presented in **Table 6.**

Table 6. Average CPU consumed, Memory utilized and Packet loss of Watchdog and Smart MIML-IDPS.

	Watchdog	**Smart MIML-IDPS**
CPU Consumed (%)	79	0.71
Initial Memory Utilization(KB)	456	108
Packet Loss (%)	7.8	1.15

From the results, we observe that an average of ~70% - ~80% of the CPU is consumed by Watchdog whereas, Smart MIML –IDPS consumes ~0.7% of CPU and under no conditions did it exceed 1%. Similarly, initial memory footprint was about 450KB for Watchdog and 110KB for Smart MIML-IDPS. The standalone IDS loaded and executed in each node resulted in high CPU consumption for Watchdog. While in Smart MIML-IDPS, only the VCH is loaded with a lightweight client ID module (it gets enabled when the node is selected as head and disabled whenever node is deselected) for intrusion detection locally. On an average, the rate at which data was "written to" and "read from" the hard disk was double in the machines Watchdog was running than in the machines VCH was enabled. In Watchdog, each node handles all control and data packets themselves, while in Smart MIML-IDPS, VCH handles only IDPS packets. From the experimental analysis, the packet loss varied from 7% - 10% for Watchdog and 1% - 2% for Smart MIML-IDPS. As per the observation, the packet loss was high for both schemes when sudden topological changes are caused with increase in traffic. Packet loss was low when topological changes occurred occasionally (low mobility and low traffic) and high when sudden topological changes (high mobility and high traffic) occurs. As expected, an additional packet loss is increased by an extra of 1% - 4% depending on the case scenario such as environmental interference, potential routing algorithm implementation problems, driver stability issues, etc. Final observation of our experimental analysis indicates that Smart MIML-IDPS is a light weight solution which offers a significant improvement with efficient use of limited resources.

End-to-End Delay: End-to-end delay of a packet is defined as the time elapsed between the time slot the packet is generated at its source and the time slot it is delivered to its destination. To validate end-to-end delay, customization was done to simulate packet generation, distribution and delivering processes. **Fig. 7,** displays results on packet end-to-end delay with varying node density (n), packet-broadcast probability q = {0.1, 0.3, 0.5} and system load $\rho = 0.6$ ($\rho = \lambda/\mu$, where λ maximum packet arrival rate and μ indicates per node throughput capacity).

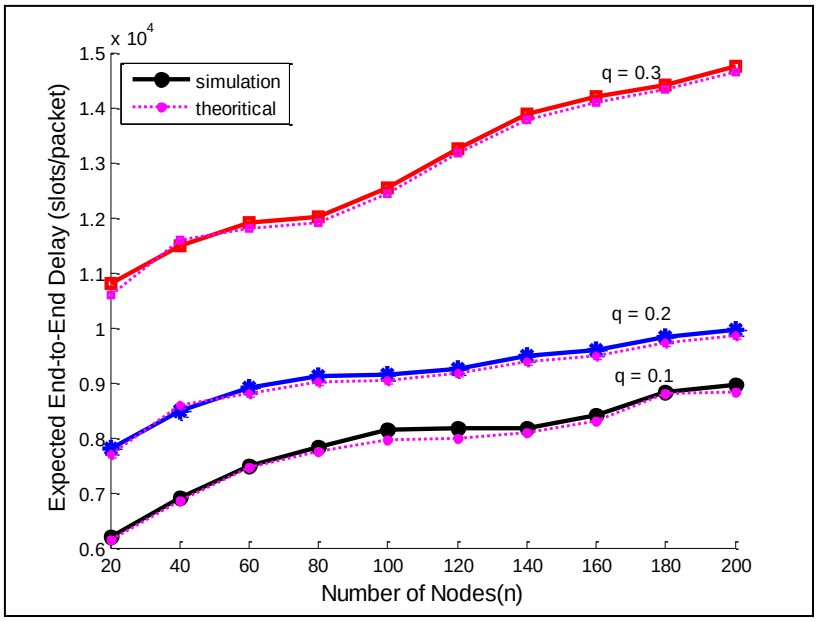

Fig. 7. Expected packet End-to-End delay Vs Number of nodes

The figure, **Fig. 7,** clearly shows that theoretical Quasi-Birth-and-Death (QBD) [16] results match very nicely with simulated ones, indicating the proposed approach is efficient in capturing the expected packet end-to-end delay. Also, as node density increases, packet end-to-end delay increases as well due to increase in contention of wireless channel access leading to longer packet end-to-end delay.

Receiver Operating Characteristic (ROC) and Detection Error Tradeoff (DET): ROC considers the True Positive Rate (TPR) and False Positive Rate (FPR) or False Acceptance Rate (FAR). A True positive in this case occurs when an instance from authorized node is correctly classified either as a good or bad labeled candidate by the MIML classifier resulting in a success. A False positive on the other hand occurs when an instance from an unauthorized node is incorrectly classified either as a good or bad labeled candidate by the MIML classifier resulting in a success. For better insight, the ROC of Smart MIML-IDPS and EAACK (DSA) scheme are analyzed and shown in **Fig. 8(a)**. From the figure we can observe that the attempt made by the authorized node's success rate increases (TPR is higher) while the attempt made by the malicious node's success rate is negligible. Smart MIML-IDPS performs better than EAACK (DAS) scheme due to the fact that the cooperativeness among the client IDS and server IDPS makes the overall system highly efficient in detecting suspicious behavior in early stages and eliminating such incidents from affecting the network.

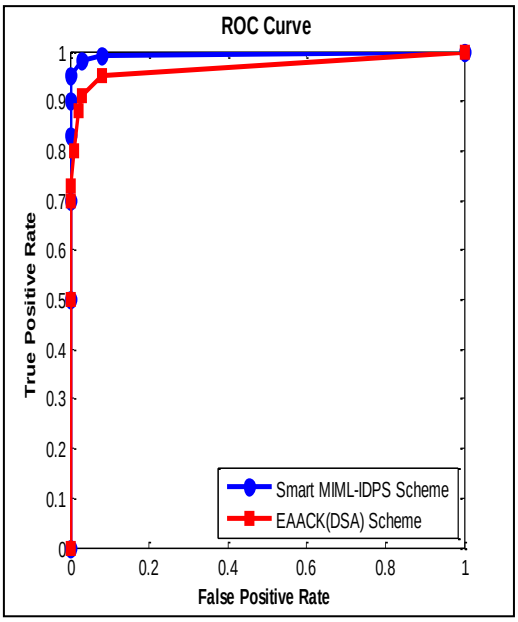

Fig. 8(a). ROC of Smart MIML-IDPS Vs EAACK(DSA)

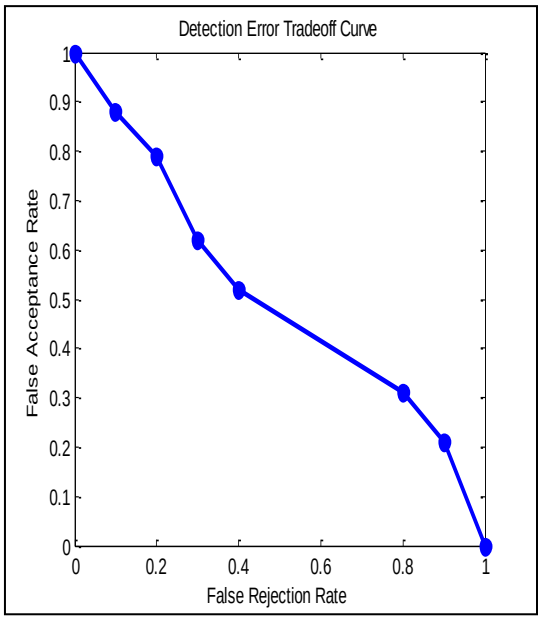

Fig. 8(b). DET of Smart MIML-IDPS

Behavioral supervised learning mechanism adapted at the server IDPS helps to learn, analyze various unknown instances, categorize and classify them appropriately. Ie., MIML internal layers helps to dynamically classify uncertain instances either as good(white) or bad(black) labels providing higher security at the receiver. Error rates in Smart MIML-IDPS scheme was derived through Detection Error Tradeoff (DET) graph by plotting False Rejection Rate (FRR) against False Acceptance Rate (FAR). Objective of this analysis is to verify if the proposed system could tolerate various unknown incidents caused between the sender and the receiver. Observation from the DET curve shown in **Fig. 8(b)** indicates that, when the attempt made by the authorized node's failure rate (FAR) increases, the attempt made by the malicious node's success rate (FRR) decreases. From the results, we can predict that when the FAR is challenged continuously with respect to the injection of FRR, it gradually decreases and becomes ineffective. Collaborating information or evidences through multilayered makes the situation significantly clearer during decision making. The light weight functional task is distributed to client agents and a heavy weight MIML-IDPS classification service is handled by the server facilitating reduced FAR and making Smart MIML-IDPS scheme more robust and secure against other existing schemes.

6. Conclusion

In this paper, we have proposed a novel network security mechanism for Tactical communication networks, predominantly deployed in a battlefield or extreme military operations. The deployment considered was typically a semi-mobile adhoc networks based on MANET architectures for military environment. As Tactical communication network provides an extreme challenge in deploying network architecture and network security, a typical IDPS product may not be suitable for such environment. Considering such an environment, we have proposed a novel Smart MIML-IDPS which is a centralized and distributed behavioral based architecture. Primarily, the proposed scheme makes use of a

virtual clustering technique and multi-layered MIML service processing to distribute functional task between client ID agents and server IDPS making it a hybridized Smart MIML-IDPS. Based on various theoretical and simulation analysis we found that our architectural re-modification and introduction of our IDPS technique proves to be more effective, secure and robust in a C3I based Tactical network environment. The results were compared at base level with existing methods, as these methods do not assume an extreme network architecture condition. Our future work relies on the challenges imposed in forming full MANET and semi-MANET conditions for successful implementation of our advanced Artificial Intelligence based IDPS system

References

[1] Y.Zhang, W.Liu, W.Lio and Y.Fang, "Securing Mobile Ad Hoc Networks with Certificateless public keys," *IEEE Transactions on Dependable and Secure Computing*, Vol.3, No.4, pp.386-399, Oct.- December, 2006. Article (CrossRef Link).

[2] KekeGai, Meikang Qiu, , Zhong Ming, Hui Zhao, Longfei Qiu, "Spoofing-Jamming Attack Strategy Using Optimal Power Distributions in Wireless Smart Grid Networks," *IEEE Transactions on Smart Grid*, Vol.8,Issue5,pp. 2431 – 2439, February 2017. Article (CrossRef Link).

[3] T. Anantvalee and J. Wu, "A Survey on Intrusion Detection in Mobile Ad Hoc Networks," *Wireless/Mobile Security*, New York: Springer-Verlag, 2008. Article (CrossRef Link) .

[4] KekeGai, Longfei Qiu, Min Chen, Hui Zhao, Meikang Qiu, "SA-EAST:Security-Aware Efficient Data Transmission for ITS in Mobile Heterogeneous Cloud Computing," *ACM Trans. Embedded Comp. Syst*., 2017. Article (CrossRef Link).

[5] Minho Jo, "A Survey: Energy Exhausting Attacks in MAC Protocols in WBANs," *Telecommunication Systems*, Vol.58, No.2, pp. 153-164, February 2015. Article (CrossRef Link).

[6] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, 89(1-2):31–71, 1997. Article (CrossRef Link).

[7] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, "A survey on mobility models for performance analysis in tactical mobile networks," *Journal of Telecommunications and Information Technology*, vol. 2, pp. 54–61, 2008. Article (CrossRef Link).

[8] KekeGai, Meikang Qiu, Lixin Tao and Yongxin Zhu, "Intrusion detection techniques for mobile cloud computing in heterogeneous 5G," *Security and communication networks*, February 2015. Article (CrossRef Link).

[9] Tran Hoang Hai, Eui-Nam Huhand and Minho Jo,"Lightweight intrusion detection framework for wireless sensor networks," *Wireless Communications and Mobile Computing*, Vol.10, No.4, pp.559-572, April 2010. Article (CrossRef Link).

[10] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *Mobile Computer Networks*., Boston, MA, pp. 255–265, 2000. Article (CrossRef Link).

[11] D. Johnson, Y. Hu, and D. Maltz, "Rfc 4728: The dynamic source routing protocol for mobile ad hoc networks for ipv4," 2007. Article (CrossRef Link).

[12] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in MANETs," *IEEE Trans. Mobile Comput.*, vol. 6, no. 5, pp. 536–550, May 2007. Article (CrossRef Link).

[13] Xiong Li, Maged Hamada Ibrahim, SaruKumari, Arun Kumar Sangaiah, Vidushi Gupta and Kim-Kwang Raymond Choo,"Anonymous Mutual Authentication and Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks," *Computer Networks*, 2017. Article (CrossRef Link).

[14] Elhadi M. Shakshuki, Nan Kang, and Tarek R. Sheltami, "EAACK—A Secure Intrusion-Detection, System for MANETs," *IEEE Transactions on Industrial*, VOL. 60, NO. 3, March 2013. Article (CrossRef Link)
[15] Xiong Li, JiangweiNiu, SaruKumari, Fan Wu, Arun Kumar Sangaiah and Kim-Kwang Raymond Choo, "A Three-factor Anonymous Authentication Scheme for Wireless Sensor Networks in Internet of Things Environments," *Journal of Network and Computer Applications*, 2017. Article (CrossRef Link).
[16] Juntao Gao, Xiaohong Jiang, Osamu Takahashi, Norio Shiratori, "End-to-End Delay Modeling for Mobile Ad Hoc Networks: A Quasi-Birth-and-Death Approach," *Ad Hoc & Sensor Wireless Networks*, 2015. Article (CrossRef Link).

Roopa.M received the Bachelor Degree from Institution of Electronics and Telecommunication Engineering (IETE), Delhi and Masters Degree in Applied Electronics from Sathyabama Institute of Science and Technology. She is currently pursuing the Ph.D. degree in Electronics Engineering at Sathyabama Institute of Science and Technology. Presently she is working with Dhanalakshmi College of engineering as assistant professor. Her main research areas of interest are: mobile adhoc networks, network security, and neural networks. She has published several papers in various national and international journals.

Dr.S.Selvakumar Raja received his B.E. degree in Electronics and Communication Engineering from A.C.College of Engineering and Technology, Madurai Kamaraj University during 1989, M.E. degree in Applied Electronics from Government College of Technology, Bharathiar University during 1991 and Ph.D degree in Information and Communication Engineering from Anna University, Chennai during 2011. He has got 24 years of teaching and research experiences in reputed engineering colleges in and around Chennai in Tamil Nadu. Presently he is associated with TKR College of Engineering & Technology, Hyderabad as Professor in the Department of Electronics and Communication engineering. His current research interests are in pattern recognition, image and signal processing. He is Life member of Indian Society of Technical Education (ISTE). He has published 9 research papers in leading International and National Journals and 15 research papers in leading International and National Conferences.