INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A HIDDEN MARKOVIAN METHOD TO SECURE INTRUSION DETECTION USING RANDOM CLUSTER HEAD ALGORITHM

T.Ramesh *, S.S.Meenatchi
Assistant Professor, Department of Information Technology,
Bharathiar University, India, Coimbatore-641046
Research Scholar, Department of Information Technology,
Bharathiar University, India, Coimbatore-641046

ABSTRACT

WSN network is vulnerable to various security breaches. Depends on the application and their environment, it may vary dynamically and it is very unpredictable. In the network, each layer is prone to security breach. Designing Security measures for WSN is a challenging task. As we design more and more preventive measures for the attacks, number of attacks increases exponentially day by day. WSN has several limitations such as low energy, limited battery power, etc. So we need to consider these aspects as well while designing the system. In this paper, a new approach called Hidden Markovian Intrusion detection system is proposed which uses a new algorithm called Random Cluster Head selection. Experimental results show that the proposed model has achieved higher defense rate than any other game theoretic approaches.

KEYWORDS: Intrusion detection, WSN, Game theory

INTRODUCTION

Wireless Sensor Network (WSN) is an emerging technology which is widely used in many applications in our day to day life. Initially it was used for Military purposes and now it is being used in many commercial applications. WSN is a tiny and most sensitive device. The devices in WSN are free to move independently and it can move in any direction. There can be several paths in the networks, but WSN can choose any path that is effectively given by a router. Unlike traditional networks, it does not need any base stations. The main challenge is to maintain proper information since it switches from one network to another as often several path breaks may occur. WSN are kinds of wireless ad hoc networks.

WSN faces several constraints such as Energy, Routing, Security, Power, Memory and many more. Among these, security is a major concern. WSN in an un-attended environment is subjected to several security risks. Intruders can cause a little or huge damage that cannot be recovered.

This paper introduces the problem of Intrusions in sensor networks and takes a step towards the development and implementation of a novel approach called Hidden Markov Model (HMM) for Intrusion detection. Aiding to HMM, a new algorithm called Random Cluster Head selection (RCH) is proposed. This selects the optimal Cluster Head (CH) based on the history of node states at different instance of time.

Using attack pattern mining algorithm, time of attacks at different instance of time is recorded, based on which future attacks are predicted. The prediction time is given to the CH which in turn informs the Base Station (BS). And the base station formulates the strategy for defending the attacks.

In case of fixed CH, the nominated cluster heads are exposed to attackers which increase the chances of an attacker to devise a strategy to attack the nodes. This is a two player game, where in even the attacker can mine the pattern and attack the CH. In order to mitigate this, RCH is being proposed. By applying the RCH algorithm, the possibility of attacker to devise a strategy is made impossible, as the cluster heads are nominated on the fly. Thereby improves the overall stability of the entire network. The detailed explanation of how HMM is designed will be discussed in subsequent sections.

The remainder of this paper is organized as follows: in Section 2, we discuss Proposed Model, integrating game theory with MDP into the IDS of a WSN. In
Section 3, we present the Simulation Result. Finally, the paper is concluded in Section 4.

All content should be written in English and should be in 2 column.
- Page type will be A4 with normal margin, word spacing should be 1.
- No space will be added before or after paragraph.
- This section should be typed in character size 10pt Times New Roman, Justified

**PROPOSED MODEL**

In this paper, we propose new IDS called “The Hidden Markovian IDS” where we have used game theory to select the best defense strategies and a Markovian Decision Process (MDP) to determine the weakest nodes based on the rewards assigned in the game. We have also discussed how cluster head is nominated randomly on the fly. Now, we will start discussing on the practical application of game theory with Hidden Markovian method section by section.

**SYSTEM ENVIRONMENT**

Assumptions:
Problems like Network congestions are not considered in this paper. And we have considered flooding attack by the intruders for our analysis.

The node deployment model considered for our discussion is as shown in Fig. 2.1. Here there two sets of sensor nodes, each containing 17 nodes reporting to a Base Station (BS). Among these two groups, any node can be nominated as Cluster head in the run time based on the Game played by the two players. For this analysis, we have considered two attacker nodes A1 & A2 which floods one of the nodes in Group1. And another attacker node A3, attacking one of the sensor nodes in Group2. Any attack to the node is reported to Cluster head which in turn is passed to Base station. So the details on how HMM and other algorithms are implemented to devise the strategy for defending the nodes under attack will be discussed going forward.

**Figure 2.1 Arrangements of the nodes**

**PROPOSED MODEL OF HIDDEN MARKOV METHOD**

In this section, the Game setup, reward function and how HMM defends and its defending strategy is defined. HMM is a statistical model, in which system is assumed to be Markov model but it has unobserved states. This is a simplest dynamic Bayesian network.

Here the attacker and the defender have their own strategies. The attacker strategies (AS) are defined as

\[ AS = \{ a_{s0}, a_{s1} \ldots a_{sn-1} \} \]

Similarly the defender strategy (DS) is defined as

\[ DS = \{ d_{s0}, d_{s1} \ldots d_{sn-1} \} \]

There are 4 possible decisions for the Hidden Markov Intrusion Detection System (HMIDS). That is listed in the following table:

<table>
<thead>
<tr>
<th>POSSIBLE STRATEGIES</th>
<th>HMIDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MISS</td>
<td>DEFEND</td>
</tr>
<tr>
<td>NO-ATTACK</td>
<td>LEAST DAMAGE</td>
</tr>
<tr>
<td>ATTACK</td>
<td>FALSE NEGATIVE</td>
</tr>
</tbody>
</table>

If the HMIDS defends the attacks properly, then we define a reward function \( g \) as the gain. If the HMIDS does not defend the attacks, then we give the reward as \( -g \). So that in the process of CH selection, if it has the reward value as \( -g \) then it will not be selected as CH.

When the process state “s” in a time interval t can be defined as “st”, then the defender strategy can be taken based on the rewards assigned to the nodes at any instance of time. At any instance of time, we define function \( f \) for defense (ds) and attacking strategy (as).

\[ f: \rightarrow d s X a s \]

Using this we calculate the reward function, \( r \) for the states at each instance of time.
This is the reward function for all the 4 possible strategies. And the last strategy is the best one as it defends most of the attacks. Here, i is the attacker variable and j is the defender variable.

\[
r(f(s_t)) = \begin{cases} 
0 & \text{if } i = 0, j = 0 \\
\delta_i(g - c_i) & \text{if } i \neq 0, j = 0 \\
-g & \text{if } i = 0, j \neq 0 \\
gk_{ij} - c_i & \text{if } i \neq 0, j \neq 0
\end{cases}
\]  

(1)

This is the reward function for all the 4 possible strategies. And the last strategy is the best one as it defends most of the attacks. Here, i is the attacker variable and j is the defender variable.

We assume that the state of node x is s0 at t = 0. If defense strategy d is taken against attack strategy a, the state of node x evolves from s0 to s1, and node x receives a reward \( r(f(s_0)) \) and so on, as shown by Eq. (2). In the Hidden Markovian Decision Process (HMDP), the state of node x transits from s0 to s1 and eventually to sp, where 1 ≤ p ≥ k - 1. Thus, the accumulated reward received by x is as follows:

\[
\begin{align*}
\tau^p_x &= \sum_{t=0}^{p} \gamma^t r(f(s_p)) \\
\end{align*}
\]  

(2)

Where \( \gamma \in (0,1) \) is the discount rate parameter.

At a discount rate of 0, the system considers only the current reward, but at a value of 1, it would regard the long-term high reward. The objective of the HMIDS is to choose an appropriate defense strategy against an attack strategy to accumulate rewards

\[
\begin{align*}
\tau^p_x &= \sum_{t=0}^{p} \gamma^t r(f(s_p)) \\
\end{align*}
\]  

(3)

Table 2.2 Definitions of the notations in Attack Pattern mining

| T | Time durations |
| S | State space of a node |
| Ds | Defense strategies |
| As | Attacker strategies |
| G | Reward |
| k_ij | Effectiveness of the reward |
| \( \delta_i \) | False positiveness of the reward |
| c_i | Denotes the cost |

A. PROPOSED ALGORITHM

In this paper, 2 algorithms are used and the detailed information of both are given below:

i. ATTACK PATTERN MINING

This algorithm is used to record the pattern of time of attacks at different instances. This information is being sent to the CH, which in turn forwards the information to the BS. The purpose of this algorithm is to record the pattern of attacks and based on the recorded data, the future attack is predicted. The advantage of implementing this framework improves the chance of detecting intrusion in an optimal way. In this framework, we have different types of nodes they are

I. Sensor nodes
II. CH
III. BS

I. SENSOR NODES

Sensor nodes are members of a cluster. These nodes sense the data and they can share their information to other sensor nodes. They have designated CH which is also a normal sensor node. Any information to the Base station is communicated via CH. It has no direct connection with the Base Station.

II. CLUSTER HEAD

Each node in the network is connected with one CH. CH is connected with the Base Station. The CHs and the BS forms a backbone. This connected backbone helps in detection of Intruder and appropriate defense mechanism is taken.

III. BASE STATION

This is the root node, it controls all other nodes. This takes appropriate decision and instructs the CH to execute. If the decision taken by the BS is not an appropriate protective strategy, BS revises the decision and instructs back to CH. It uses Bayesian approach for taking the decision.

In our system, HMIDS records the pattern of attacks using pattern mining algorithm and predicts the next time of attack. We have provided additional buffer time of 0.5 sec and starts defending the nodes before.
the predicted time of attack. The notations used in the algorithm are:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tba</td>
<td>Time Between Attack</td>
</tr>
<tr>
<td>AvgTba</td>
<td>Average of time between attack</td>
</tr>
<tr>
<td>AP</td>
<td>Attack Pattern</td>
</tr>
<tr>
<td>Toa</td>
<td>Time of attack</td>
</tr>
<tr>
<td>Ptoa</td>
<td>Predicted time of attack</td>
</tr>
<tr>
<td>N</td>
<td>Number of attacks</td>
</tr>
</tbody>
</table>

**Table 2.3 Notations used in Attack Pattern Mining**

### A. DESIGN OF THE ALGORITHM

**Step 1:** Record the time of attacks at regular intervals as

\[ \sigma = (toa_1, toa_2, \ldots, toa_n) \]  \hspace{1cm} (4)

Where \( \sigma \) denotes the sequences toa is the time of attack that are recorded in the regular time intervals.

**Step 2:** Calculate the Time between Attacks

\[ \delta = (toa_{n-1} - toa_n) \]  \hspace{1cm} (5)

Where \( \delta \) denotes Time Between the attack

\( \sigma \) denotes the time of attack.

\( n \) represents the number of attack

**Step 3:** From the previous step, the attack intervals are stored in the system. Now we can calculate the average time as

\[ \text{avgTba} = \frac{(\sum( (toa_1) + \cdots + (toa_{n-1}) + (toa_n)) ) }{n} \]  \hspace{1cm} (6)

Where, \( \text{avgTba} \) is the average time between attacks.

\( toa \) denotes the time of attack

\( n \) represents the number of attack

**Step 4:** From the average calculated, we predict the future attack.

\[ pToa = (toa_c + \text{avgTba}) - \text{buf} \]  \hspace{1cm} (7)

Where \( pToa \) is the Predicted time of attack.

\( toa_c \) denotes the current time of attack.

\( \text{buf} \) denotes the buffer time in seconds.

\( \text{avgTba} \) denotes the average time between attack

**Step 5:** Based on the possibilities defined in table 3.1, appropriate security measure is taken by the defender.

The above steps are described below:

Process begins whenever there is an attack in the cluster. Whenever there is an attack, then the time of attack is being recorded in the base station, so that we can calculate the parameters like time between attack and avgTba. Based on which the future attack is predicted.

To calculate the time between attacks the formula used is given below

\[ \delta = (toa_{n-1} - toa_n) \]

Here the time between attacks is stored in the base station. We take that in account and calculate the time between attacks.

For example if the time of the first attack has occurred in 5th minute and the next attack has occurred in 10th minute, the time between attacks is 10-5 = 5 minutes.

Next is to calculate average time between attacks. This is calculated as

\[ \text{avgTba} = \frac{(\sum( (toa_1) + \cdots + (toa_{n-1}) + (toa_n)) ) }{n} \]

Using this average, we have to predict the future attack using

\[ pToa = (toa_c + \text{avgTba}) - \text{buf} \]

Here we need to provide some buffer in seconds so that even before we expect an attack the system starts defending for more stability. As per our previous example the time of attacks happens at 5th and 10th minute, so that the next predicted attack will be at 15th minute. After the prediction using mining algorithm the next step is to devise an appropriate strategy to defend the attacked nodes. For that the states of the different nodes are considered. Like discussed earlier reward points are calculated based on the game results between the two players.

### ii. RANDOM CLUSTER HEAD SELECTION ALGORITHM

Static CH exposes the state of nodes to the intruder. So in order to hide the state and identity of the nodes to the intruder, a new algorithm called RCH is proposed. This algorithm nominates the CH based on the history and state of the nodes randomly. New CH is selected based on how well that particular node is performing. The following figure illustrates the Random CH selection.

![Random CH Selections](image-url)
Here \( x \) defines the possible states, \( y \) defines the possible observations, \( a \) defines the state transition probabilities and \( b \) defines output probabilities. The random variable \( x(t) \) is the hidden state at time \( t \) \( (x(t) \in \{ x_1, x_2, x_3 \}) \). The random variable \( y(t) \) is the observation at time \( t \) \( (with \ y(t) \in \{ y_1, y_2, y_3, y_4 \}) \).

The arrows in the diagram denote the conditional dependencies.

**A) DESIGN OF THE ALGORITHM**

**Step 1:** Identify the physical cluster.

**Step 2:** Input the data’s

**Step 3:** Split the nodes individually.

**Step 4:** Calculate the reward values for all the nodes at different instances to select the CH.

This is given as

\[
    r_x^p = r(f(s_0)) + \gamma r(f(s_1)) + \gamma^2 r(f(s_2)) + \cdots + \gamma^p r(f(s_p))
\]

Here \( r_x^p \) denotes the reward function for the predicted attack at a period of time

\( f \) denotes the function

\( s \) denotes the state of transmission

\( \gamma \) denotes the discount rate parameter

**Step 5:** Check the reward function.

**Step 6:** If the reward function is less than 5 then do not nominate the node for CH selection.

**Step 7:** If the reward function is greater than 5 nominate them for CH selection

**Step 8:** The CH nominated and rejected nodes are stored in the database.

**Step 9:** Based on the stored values the nominated CH is compared and the highest rewarded node is selected. If it contains more than one same reward value it selects randomly.

The above steps are described below:

Based on the input data’s we begin our process. We split all our nodes to calculate the reward value. The reward value is calculated by using the following formula

\[
    r_x^p = r(f(s_0)) + \gamma r(f(s_1)) + \gamma^2 r(f(s_2)) + \cdots + \gamma^p r(f(s_p))
\]

This reward value is the most important for the random cluster head selection. By default each node is assigned the reward value as 5. If any attack occurs the counter is reduced by 1 and if it defends properly we increment the counter by 1.

After calculating the reward value based on the attacks, the CH selection begins. Here we have a comparison between nodes if it is greater than 5 it is nominated for CH selection or if it is less than 5 the node is not sent for CH selection.

**SIMULATION RESULT**

The results from the simulation have shown below in the table and the related line graph with variable time. In the table 3.1 the time vs. packet drop ratio compared to existing work is recorded and portrayed in the subsequent figures. The defending strategy is done for flooding attack. In the table 3.2 a comparison chart between existing and proposed Time vs. Packet Delivery Ratio.

<table>
<thead>
<tr>
<th>Time</th>
<th>Existing (Fixed CH)</th>
<th>Proposed (Random CH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.9</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Existing (Fixed CH)</th>
<th>Proposed (Random CH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.80</td>
<td>0.80</td>
</tr>
</tbody>
</table>
The above figure shows the packet drop induced in the existing system due to the highlighted flaw (State of the node is visible) in this thesis. From the graph we understand that few attacks had happened at 2.0, 4.0 & 6.0 seconds respectively. Ideally after sixth second, intrusion detection system should predict the attack pattern and defend the node under attack accordingly. But in this case, since CH itself is attacked by the attacker, the prediction of the future attack is prevented resulting in failure in defending the node under attack. From the above observation, even after the second attack, the packets are getting dropped due the failure in communication to base station from cluster head. This situation will finally lead to a network congestion which would break the system.

The above figure highlights the benefits we have yielded after implementing the hidden markovian model to our system. Here as we notice, once HMM enabled security system comes into picture, the packet drop gets drastically reduced to zero. From then on, all the nodes are successfully defended and health of the entire network is maintained. There is not even a single packet drop which is a clear indication of the successful defensive strategies devised by the defender as a player.

The below figure finally compares both the output in a single graph thereby highlighting the benefits gained in proposed system.
In Intrusion detection system, the cluster head selection is the most important part which will improve the quality of tracking the data. The proposed Random Cluster Head selection (RCH) algorithm is implemented to select the best cluster heads to monitor the nodes in order to keep track the node and also to defend the attackers. Thus an enhanced version of intrusion detection using Hidden Markovian Approach is proposed.

This approach overcomes the problem in the existing markovian approach wherein if a CH is attacked, then the entire effort taken while defending becomes vain. In this approach a CH is not revealed to the attacker. As the CH is not static even if the attacker attacks the correct CH that is pattern mined and another CH is nominated which communicates the attack to Base station and defends the attack. Experimental results verify that the proposed algorithm can enhance the Intrusion detection and defend the intruders using Hidden Markov Method.

REFERENCES

“Sensors 2012, 12, 9055-9097; doi:10.3390/s120709055


[27] Yu Liu, Cristina Comaniciu, Hong Man “A Bayesian Game Approach for Intrusion Detection in Wireless Ad Hoc Networks” In: 2006 ACM 1-59593-507-X.

Author Bibliography

T.Ramesh
Received the Master’s Degree in Computer Application from Anna University, Tamil Nadu, India in 1996, M.Phil, degree in Computer Science from Bharathiar University, Coimbatore, Tamil Nadu, and India in 2008 and M.Tech(IT) from Anna University, Coimbatore. He is currently working as Assistant Professor and Assistant Technical officer in Department of Information Technology, School of Computer Science and Engineering, Bharathiar University, India, Coimbatore-641046. He is having more than 18 years of experience as teaching and research. He has published 20 articles in National, International Journals and International Conference and written Two books. His current research includes: Computer Networks, Mobile Computing, and Information Security.

S.S.Meenatchi
Received her Master’s Degree in Computer Science and Information Technology from G.T.N Arts College, Dindigul, Tamil Nadu, India in 2007. Worked as Lecturer for 1 year in G.T.N Arts College, Dindigul, Tamil Nadu, India. Completed M.Phil (Research Scholars), in Computer Science , School of Computer Science and Information Technology, Bharathiar University , India, Coimbatore -641046. Has attended 1 National conference and published 3 articles in National Journals. Her area of interest includes: Network Security, WSN, Game theory.