3
Bio-inspired Machine Learning Model for Wireless Sensor Networks

Intrusion comes into the picture as and when the attacker breaks into the system by exploiting a piece of network and gains access to its data. An Intrusion Detection System (IDS) in general, acts as a second line of defense just like a human immune system, intrusion prevention schemes (first line of defense) like authentication, encryption may fail. An IDS is a system that dynamically monitors the events taking place in the machine just like a human lymphatic system and checks network traffic and decides whether these are symptoms of an attack or not. It has two major components namely, misuse detection and anomaly detection.

(1) Misuse Detection: It matches new observations with the signatures stored in the database.
(2) Anomaly Detection: It detects abnormal activities from predefined normal profiles in order to identify possible attacks. It can be supervised (with prior knowledge of the classes), semi-supervised (with knowledge of one class) or unsupervised (with no knowledge).

In WSNs, sensor nodes can behave abnormally and send data outside the normal expected behavior. Figure 3.1 shows three possible scenarios of data being reported by a sensor node which is measuring temperature of a remote site. The plot *normal stable* reflects normal temperature behavior under stable conditions. The plot *normal rising* reflects normal behavior in a situation where the temperature of the remote site is increasing. The plot *fraudulent* reflects the possible fraudulent behavior. In the first two scenarios, the sensor node is behaving well, unlike the last scenario where the sensor node shows a fraudulent behavior. Detection of such anomaly (fraudulent behavior) needs to be rectified.

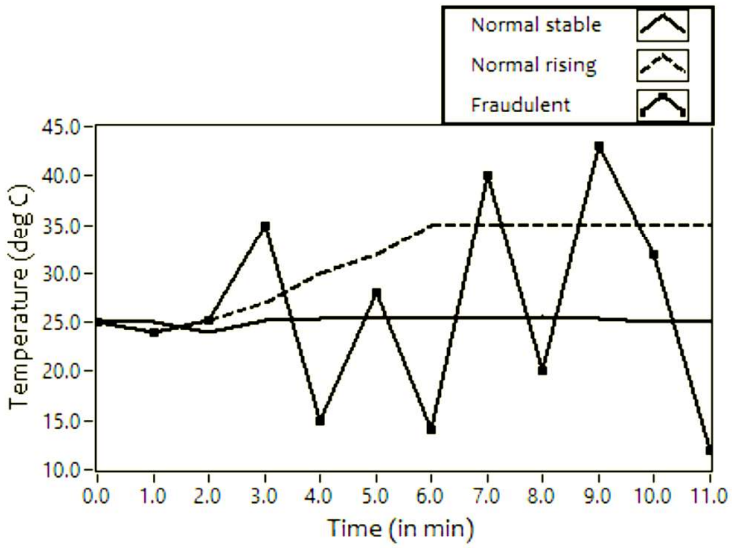


Figure 3.1: Fraudulent Data Behavior

The objective of this chapter is to present the design of a security system for WSN. It has two models. A Machine learning model is used for the detection of fraudulent nodes in WSN since machine learning algorithms are accurate and performance efficient. It is a type of intrusion detection system which aims to detect the anomalies. Techniques such as k-means, Support Vector Machine (SVM) and Anamoly Detection Engine (ADE) are used for the detection. An Immune model is used for reducing the effect of those fraudulent nodes which were flagged as fraudulent nodes from the machine learning model. The Immune model uses the human immune system as inspiration. The antigen and antibody concept is used for the removal of the fraudulent node. Section 3.1 gives the background information related to machine learning techniques and describes the human immune systems. Section 3.2 elucidates the concept of T-cells and B-cells in human body. Section 3.3 gives the outline of the proposed work and describes the proposed model having two blocks namely machine learning model and immune model. Section 3.4 presents the experimental results followed by Section 3.5 which summarizes the chapter.

3.1 MACHINE LEARNING

Machine learning models aim at developing an algorithm to make predictions from the data for accomplishing a particular task [Sedjelmaci and Feham 2011]. Tom Mitchell describes machine learning as a technique to formulate a machine for facilitating learning with respect to time i.e. a machine is said to learn from experience E_m with respect to some task T_m and some performance measure P, if its performance on T_m, as measured by P, improves with experience E_m [Mitchell, 2006]. Machine learning techniques help in learning the pattern of data by using a set of rules and predictors. There are two broad algorithms that are used in machine learning:

(1) *Unsupervised Learning*: Learning is carried out on an unlabelled data set which reflects the statistical pattern of the overall behavior of data.
(2) *Supervised Learning:* Learning is performed on a labelled data set to describe the pattern and create a decision boundary. It has two parts:
    (i) Regression
    (ii) Classification

3.1.1 Unsupervised Learning

The intent of unsupervised learning is to find a structure on unlabeled data set. K-means is a widely used technique under unsupervised learning. It is a type of algorithm which groups the data to produce clusters [Wagstaff, 2001]. The procedure follows a simple and effective way of creating clusters via two major tasks:

(1) *Cluster assignment*: Here random centroids (consider two centroids) are assigned and distance of each observation is measured with respect to the two centroids. Accordingly, assignment of each observation is made to the nearest centroid.
(2) *Centroid movement*: Calculate the new means to be the centroids of the observations in the new clusters i.e. calculate the average of all the observations assigned to each centroid. Later assign the average position of these observations to the centroid.

Finally, repeat the above two steps till convergence. Algorithm 3.1 describes the k-means algorithm.

3.1.2 Supervised Learning

Supervised learning is learning performed on labelled data set. It is a learning which is formulated to produce pattern and decision boundary on the data set. It can be performed via two techniques namely regression and classification.

Regression: Regression is a technique to fit the data measurements for predicting continuous valued output [Mitchell, 1977]. It gives real valued output to obtain a curve. Given the input values x_i and output values y_i, different curves/fits can be obtained as seen in Figure 3.2 (a).

ALGORITHM 3.1 K-MEANS ALGORITHM

Input:

k : Number of clusters

s: Number of measurements

Training set $(x^{(1)}, x^{(2)}..... x^{(s)})$, where $x^{(i)} \in R^{(n)}$

Output: k number of clusters generated out of random data measurements

1 Randomly initialize k cluster centroids $\mu_1, \mu_2.....\mu_K$,

2 **while** *till convergence* **do**

3 **for** $i \leftarrow 1$ *to* s **do**

4 c^i = index from 1 to k of cluster centroid closest to x^i

$c^i = \min_k || x^{(i)} - \mu_k ||$

5 **for** $i \leftarrow 1$ *to* s **do**

6 μ_k = average (mean) of measurements assigned to cluster k.

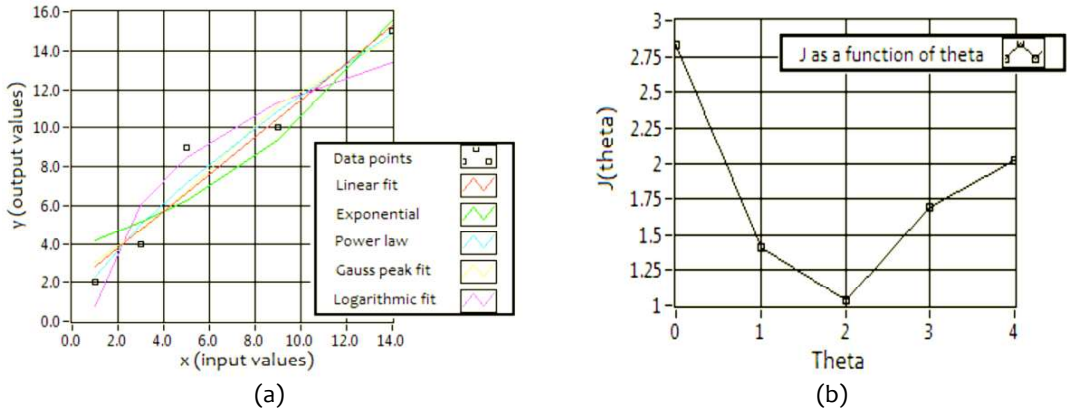


Figure 3.2: Various Regression Fits: (a) Various fit such as linear curve fit, power fit, logarithmic fit and exponential fit on data measurements; (b) Cost function of power fit

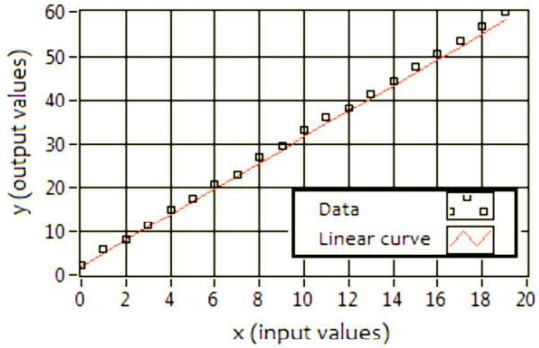


Figure 3.3: Gradient Descent Algorithm

Fitting can be implemented in various ways depending upon the type of data set. In the given example, the best model which fits the data measurements as learned from the cost function is a power fit. The following hypothesis has been used [Ng, 2012]:

$$h(x) = \theta_0 + \theta_1 x \quad (3.1)$$

where h is the hypothesis which maps the input to output, θ_i are the parameters and x is the input variable. To find the values of θ_0 and θ_1 the following cost function has been used to obtain the best value by regressively minimizing the function as shown in Figure 3.2 (b) [Ng, 2012]:

$$J(\theta) = \frac{1}{2s}\sum_{i=1}^{s}(h_\theta(x_i) - y_i)^2 \quad (3.2)$$

where s is the number of training samples and y_i are the output variable.

A typical example of regression technique is gradient descent model. This model helps in finding the best fit model by modifying the values of a parameters i.e. θ_j. Figure 3.3 depicts how the data fits by using the gradient descent algorithm as explained in Algorithm 3.2. For present study, the linear fit seems to fit data measurements shown as amplitude in Figure 3.3.

ALGORITHM 3.2 GRADIENT DESCENT ALGORITHM

Input:

α_r: Learning Rate

Output: Fitting function

1 **while** *till convergence* **do**

$$\theta_j := \theta_j - \alpha_r \frac{\delta}{\delta\theta_j} J(\theta_0, \theta_1) \;\; for\, j = 1 \; and \; j = 0$$

Classification: Support vector machine (SVM) is a popular tool used to classify the data and create a decision boundary to distinguish between fraudulent and benevolent users/data. It is a type of classification and regression prediction tool. Over-fitting of data is avoided to maximize predictive accuracy by utilizing machine learning theory. The aim of SVM classifier is to determine a set of vectors called support vectors to construct a hyper plane in the feature spaces [Tong and Koller, 2002]. Initially, the input vectors or features of the training data were mapped onto a very high dimensional feature space. The mapped feature space, SVM then constructs a linear decision function which separates the two classes of events, namely benevolent and fraudulent. SVM searches for a decision function with maximum margins [Joseph et al, 2011].

Given the training datasets, $(x_i, y_i, i = 1, 2 \ldots n, y_i \in \{-1, +1\}, x_i \in R^d)$, a hyper plane is determined having a maximum margin [Suykens and Vandewalle, 1999]:

$$w_v . x = b \quad (3.3)$$

where w_v is a normal vector and the parameter b is offset. In order to find the optimal hyper plane, the following convex optimization problem is solved:

minimize

$$\left(\frac{||w_v||^2}{2} + \theta \sum_{i=1}^{n} \epsilon_i\right) \quad (3.4)$$

subject to $y_i(< w_v, x_i > + b) \geq 1 - \epsilon_i, \; \epsilon_i \geq 0, 1 \leq i \leq n$

where $\sum_{i=1}^{n} \epsilon_i$ relax the constraints on the learning vectors, and θ is a constant that controls the tradeoff between number of mis-classifications and the margin maximization. Equation 3.4 can be dealt using Lagrange classifier [Scholkopf and Smola, 2001]:

maximize

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_j, x_i) \quad (3.5)$$

subject to $\sum_{j=1}^{n} y_i \alpha_i = 0 \;\; and \; 0 \leq \alpha_i \leq c \; for \; all \; 1 \leq i \leq n$

where $K(x_j, x_i)$ is the kernel function and α_i is the Lagrange multiplier. According to the condition of Kuhn-Tucker (KKT), the x_is that correspond to $\alpha_i > 0$ is called support vectors (SVs). Once the solution to Eq. (3.5) is found, following equation is obtained:

$$w_v = \sum_{i=1}^{n} \alpha_i y_i x_i \quad (3.6)$$

Thus, the decision function can be written as:

$$f(x, a, b) = \{\pm 1\} + sign(\sum_{i=1}^{n} y_i \alpha_i \, y_i K(x, x_i + b)) \quad (3.7)$$

The kernel functions can be of various types such as linear kernel, polynomial kernel, radial basis function kernel, sigmoid kernel, Gaussian kernel, convex optimization of kernels, normalization of kernels [Vapnik, 2000; Muller, 2001]. For the present study, linear kernel and polynomial kernel are used. They can be represented in following manner:

Linear Kernel $K(x_j, x_i) =< x_j, x_i >$ (3.8)

Polynomial Kernel $K(x_j, x_i) = (s < x_j, x_i > +c)^d$ (3.9)

where *s*, *c* and *d* are kernel specific parameters.

3.2 IMMUNE SYSTEM

Biological immune systems have intelligent capabilities of detecting antigens (foreign bodies in the system) in the body. As shown in Figure 3.4, immune system can be classified into two types, innate and adaptive.

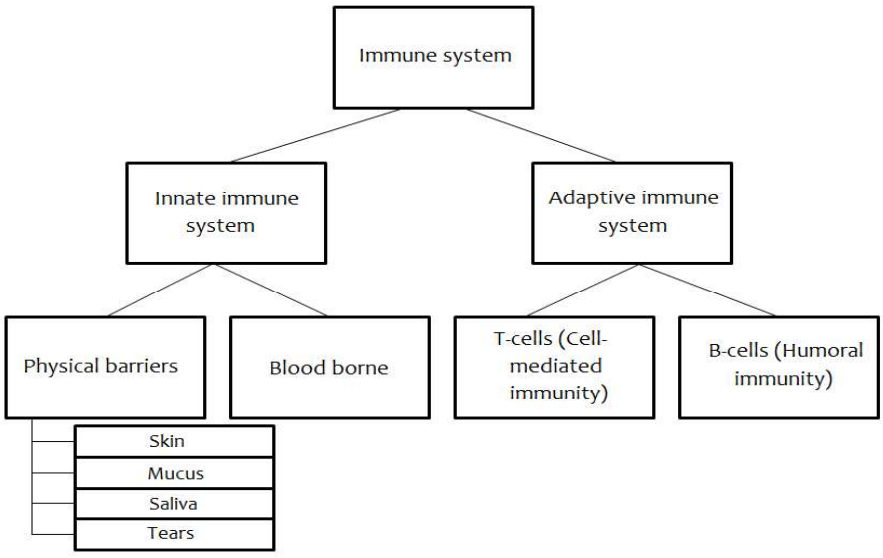


Figure 3.4: Classification of Immune System

Innate immunity is the first line of defense for pathogens. It is non-specific and is meant for rapid detection and elimination of pathogens. It generally refers to non-specific defense mechanisms that come into play within hours of an antigen's appearance in the body. It is referred to as non-specific defense mechanism since it is not designed for any specific pathogen. It can further be classified as physical barriers and blood borne. Physical barriers such as skin, tears, saliva and mucus stop the infection before entering the body [Greensmith, 2010]. If pathogens manage to get past the physical barriers, blood borne body cells come into picture. Their response will also be non-specific. This process, called as phagocytosis, is carried out by a number of different phagocytes, the most common types being the neutrophils and

macrophages. Neutrophil has molecules on their cell walls that help in identifying foreign particles. Once the foreign particles are identified, it will attach to the micro-organism wall, to engulf and enclose the micro-organism in the vacuole as shown in Figure 3.5. Enzymes present in Golgi body in the neutrophil are secreted, which then digest the micro-organisms. Macrophages perform the same task outside the blood vessels, so that the pathogens can be removed from tissue fluid.

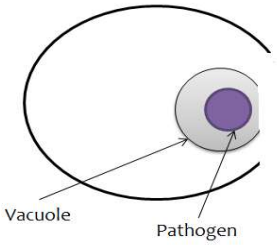


Figure 3.5: Engulfing of Pathogen in Neutrophil

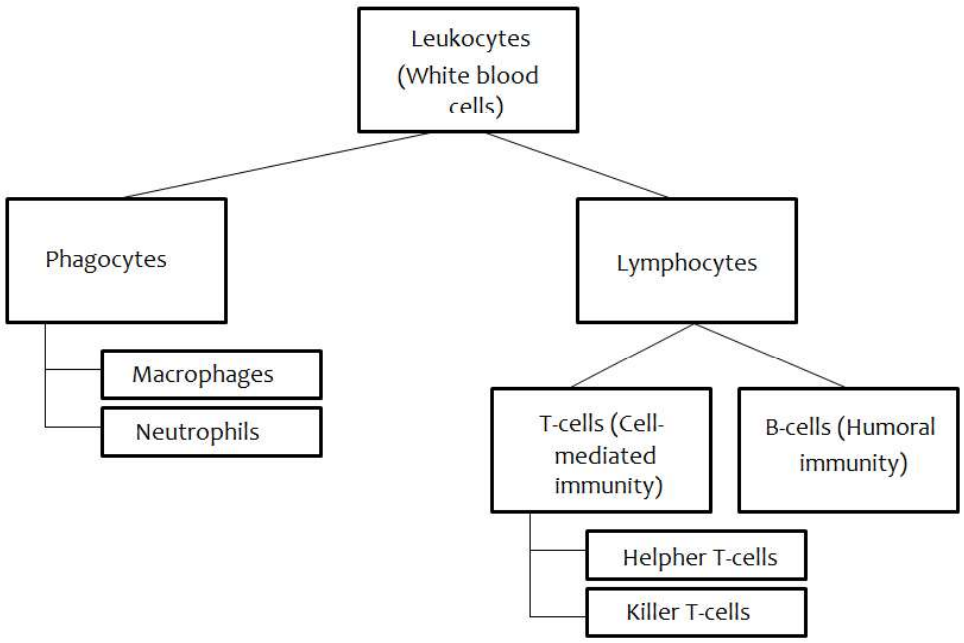


Figure 3.6: Classification of Leukocytes

If the innate immune system cannot remove the pathogen, then the adaptive immune system takes over. Adaptive immune system comprises of a network of cells, tissues, and organs that work in cohesion to protect the body. The cells involved are white blood cells, or leukocytes, which come in two basic types, phagocytes and lymphocytes [Durani, 2012]. Classification is depicted in Figure 3.6. Phagocytes have already been discussed in innate immune system. Lymphocytes are of two types, namely T-cells and B-cells. Leukocytes are developed from undifferentiated stem cells in the bone marrow. Lymphocytes start out in the bone marrow, stay there and mature into B-cells. Alternatively, they leave for the thymus gland,

where they mature into T-cells. They are called T-cells because the latter stages of their development occur in the thymus. Spleen, bone marrow and thymus are also called lymphoid tissues. Lymph nodes are specialized tissue harbouring cells of the innate immune system called lekocytes and macrphages, in addition to specialized cells of the adaptive immune system; T and B cells. These nodes are connected by the lymphatic circulation of the body and help the two arms of human immune system coordinately fight a pathogenic attack in coordination.

The adaptive immune system consists of two complementary systems, namely cellular immune system and humoral immune system. The humoral immune system is aimed at bacterial infections and extracellular viruses, but can also respond to individual foreign proteins. This system contains soluble proteins called antibodies which bind the bacteria, viruses, or large molecules identified as foreign and target them for destruction. Antibodies are produced by B-cells. Antigens are secreted by the pathogens which cause the immune system to respond. B-cells produce and secrete antibodies after they encounter antigens. The cellular immune system destroys host cells infected by viruses and also destroys some parasites. The agents at the heart of this system are a class of T-cells. B-cells act as the body's military intelligence system which alerts the body's defenses to seek and destroy the enemy targets. T-cells act as the soldiers, destroying the invaders that the intelligence system has identified. T-cells are broadly classified into two types, namely Helper T-cells and Killer T-cells. Killer T-cells interact with infected host cells through receptors on T-cell surface. Helper T-cells interact with macrophages and secrete cytokines that stimulate killer T-cells, helper T-cells, and B-cells to proliferate and produce antibodies specific to the pathogen.

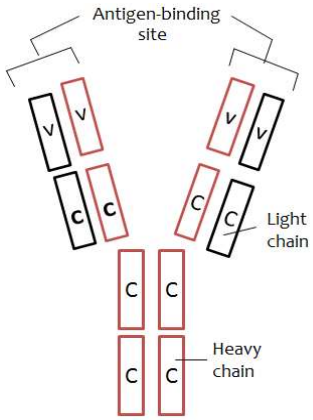


Figure 3.7: Antibody Structure

In the human immune system, B-cells produce an antibody on encountering with an antigen. As there can be any type of antigen that can excite human immune system, there is a requirement of a specific antibody that can bind to that antigen. In human bodies, B-cells produce n different types of antibodies that bind to the particular antigen to decrease its effect in the body. Further, B-cells are capable of maintaining a memory which keeps track of all the previous invaders. B-cells simulation can be modeled on three parameters:

(1) Affinity between B-cell and pathogen.
(2) Affinity among its neighbors.
(3) Separation (enmity) factor from the loosely connected neighbors of the B-cells.

The general structure of all antibodies is similar. However, a small region at the tip of the protein is extremely variable which allows millions of antibodies to have minute differences in the tip structures, or antigen binding sites, to exist. Typical structure of antibody is as shown in Figure 3.7. It is composed of 2 chains, one light chain (black color) and one heavy chain (red color). Both the chains consist of constant region and variable region as denoted by C and V respectively. Antibodies are diverse in nature because of the various processes such as domain variability, V(D)J recombination, somatic hyper mutation and affinity maturation, class switching or affinity designations [Wikipedia, 2012].

Once the antibodies are produced, they continue to exist in a person's body. It provides better immunity to the living beings in a way that if the same antigen is presented to the immune system again, the antibodies are already there to perform their task. Hence if someone gets sick with a certain disease, like chickenpox, that person typically does not get sick from the disease again. Although antibodies can recognize an antigen and lock onto it, they are not capable of destroying it without the help of other cells [Morphosys, 2012]. That is the job of the T-cells, which are a part of the system that destroys the antigens which have been tagged by antibodies or cells that have been infected or somehow changed. Furthermore, T-cells help signal other cells (like phagocytes) to do their tasks. All of these specialized cells and parts of the immune system offer body protection against a disease. This protection is called immunity.

To wrap up, antigens are secreted by the pathogens which cause the adaptive immune system to respond. B-cells produce and secrete antibodies after they encounter antigens. Once they produce a specific antibody for the antigen it forms a complex called antigen-antibody complex which in turn is engulfed by the T-cells. After the B-cells produce antibodies they give rise to plasma cells from which further antibodies are produced for that specific antigen.

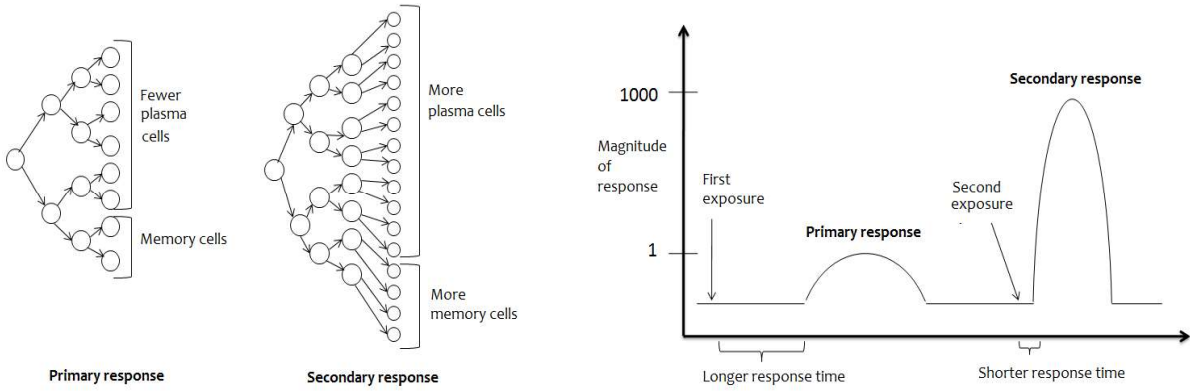
Thus, human immune system has following aims [Hao, 2005]:

(1) *Accountability*: Human body tries to identify, find and destroy the non self particles which have entered the human body.
(2) *Disposability*: Human body continues to work even if the non self particles are detected.
(3) *Correction*: The body prevents from attacking its own cells.
(4) *Integrity*: Genetic codes inside the cells are not modified.

3.2.1 Primary vs Secondary response

Primary response in the immune system is defined as the first exposure of the immune system to the antigen. Secondary response occurs when there is another exposure to the same antigen. When the immune system encounters the antigen for the first time, B-cells multiply to form two types of cells i.e. plasma cells and memory B-cells.

The plasma cells produce antibodies and memory B-cells just keep a track of the type of antigen. Secondary response occurs when another exposure to the same antigen causes the memory B-cells to rapidly form the plasma cells to produce antibodies out of it as shown in Figure 3.8(a). Secondary response is faster since it produces more antibodies than in the case of primary response as shown in Figure 3.8(b). It also has a high magnitude which rapidly decreases the count of antigen rapidly.



(a) (b)

Figure 3.8: Primary vs Secondary Response: (a) Clearly it is seen that number of plasma cells and memory cells are high in secondary response as compared to the primary response thereby increasing the number of antibodies; (b) The response time in primary exposure is less compared to the secondary response (Source: Sharp, 2011)

3.2.2 Mathematical Model

A mathematical model was devised to study the overall antigen-antibody concept [Dibrov et al, 1977; Fowler, 1981]. Dibrov's Model consists of three coupled equations for the antibody quantity a, the antigen quantity g, and the small B-cell population x. If the secondary and subsequent responses to an antigen injection are similar, it is reasonable to suppose that the small B-cell population is relatively unaffected by the later injections, so x is put as a constant. The second-order set of equations describing antigen-antibody interactions:

$$\frac{dg}{dt} = Kg - Qag \tag{3.10}$$

$$\frac{da}{dt} = AH(t-T) - Rag - Ea \tag{3.11}$$

where *K, Q, A, R, E* are rate constants. *K* is the overall growth rate of antigen. *T* is previous time constant. *H(t)* in Eq.(3.11) is the Heaviside step function whose value is zero for a negative argument and one for a positive argument.

$$H(T) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases} \tag{3.12}$$

The product '*ag*' is the complex formed as antibody-antigen complex. As the complex is formed, it results in a net loss of the antibody and antigen. The simplest assumption is that of the law of mass action, valid when the densities are below a certain saturation level i.e. the losses are proportional to the product of the antibody and antigen densities. The rate constants *Q* and *R* are necessarily not the same. The rate of antibody production at time *t* is supposed to be proportional to the rate of small B-cell stimulation at time *t* - *T*: i.e. there is a delay *T* between the stimulation of a small B-cell and the subsequent production of plasma cells from it.

3.3 PROPOSED WORK

Machine Learning Based Biologically Inspired Security model for WSNs can be divided into two essential blocks namely machine learning model and immune model as shown in Figure 3.9. The Machine Learning model has three essential blocks namely clustering, Support Vector Machine (SVM) and Anomaly Detection Engine (ADE). It is used for the detection of fraudulent nodes. The Machine learning model is followed by the immune model which is used

to remove the fraudulent nodes from the system just like the defensive mechanism is used to remove foreign particles in human body.

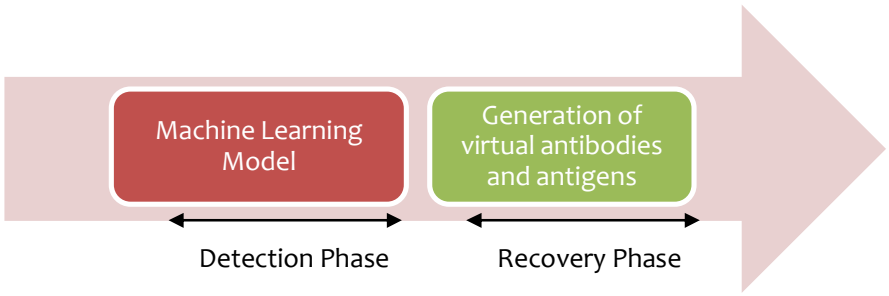


Figure 3.9: Two Phase Model Namely Machine Learning Model and Immune model

3.3.1 Machine Learning Model

Machine learning model is divided into two phases namely training phase and test phase as shown in Figure 3.10: where training phase is used to train the data and formulate the machine to learn the trend, test phase is used to check the hypothesis made by the training phase. The output of the training phase is the set of all fraudulent data measurements which is detected by the techniques of machine learning. Machine Model has three basic parts:

(1) Clustering
(2) Support Vector Machine
(3) Anomaly Detection Engine

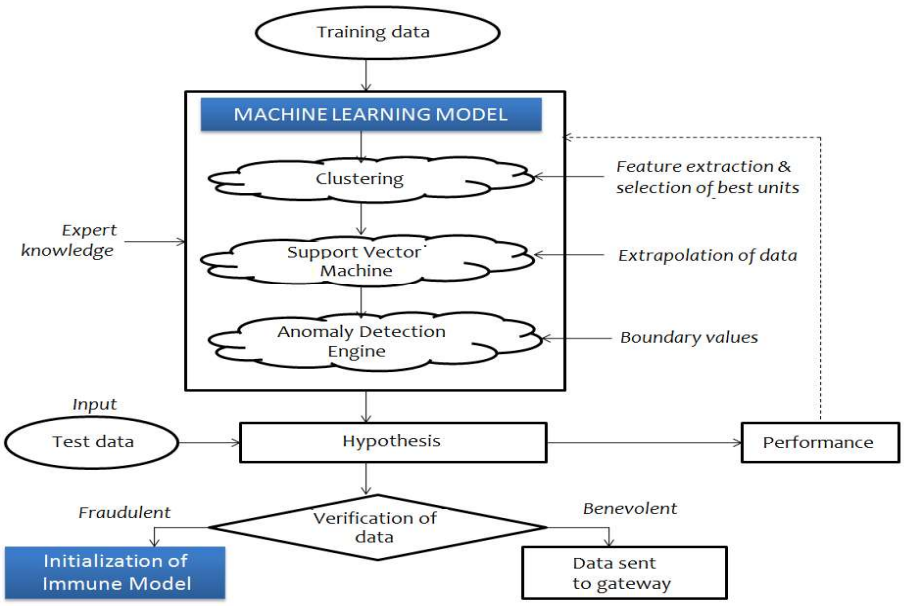


Figure 3.10: Flowchart of Machine Model

Clustering: It is an unsupervised machine learning method which works on the principle of finding a structure out of an unlabeled data set. K-means method can be used to take the mean of the training data set to prepare a lower and an upper bound defining the two clusters. The two centroids which are assigned in k-means are the higher and the lower bound mean values

of the data measurements. It groups the data to make clusters. Figure 3.11(a) shows the random measurements and 3.11(b) shows two random clusters generated out of it using k-means which is named as fraudulent cluster and benevolent cluster.

Support Vector Machine: The data which was classified into clusters from k-means algorithm is given to SVM to create the decision boundary as shown in Figure 3.12.

Anomaly Detection Engine: SVM creates the decision boundary; however the data which lies on the boundary needs to be further evaluated for better accuracy and precision. ADE helps in making more accurate decisions. The algorithm works in the following manner.

(1) After SVM implementation, the Gaussian distribution of benevolent features (x_j, y_j) is calculated that might be indicative of anomalous examples for clear cut demarcation between the fraudulent and the benevolent data.

(2) The Gaussian distribution is calculated by computing mean and standard deviation of the benevolent features.

$$\mu_j = \frac{1}{S}\sum_{i=1}^{S} x_j \tag{3.13}$$

$$\sigma_j^2 = \frac{1}{S}\sum_{i=1}^{S}(x_j - \mu_j)^2 \tag{3.14}$$

Where μ and σ are the mean and the standard deviation of the benevolent measurements

(3) Compute the probability distribution p(x) of benevolent features (measurements (x axis) and time (y axis)) with the help of following equation [Ng, 2012]:

$$p(x) = \prod_{j=1}^{n} p\left(x_j, \mu_j, \sigma_j^2\right) = \prod_{j=1}^{n} exp\frac{-(x_j-\mu_j)^2}{2\sigma_j^2} \tag{3.15}$$

(4) The combined probability distribution function of x and y is calculated using the following equation:

$$p(x) = p(x, \mu_x, \sigma_x^2)p(y, \mu_y, \sigma_y^2) \tag{3.16}$$

(5) Threshold ϵ is maintained and if the probability of the measurement is $< \epsilon$, it is considered as anomaly and vice versa.

The probability distribution function of x and y and combined probability is as shown in Figure 3.13. The outcome of the machine learning model is the set of fraudulent data measurements.

The algorithms for the training and the test phase of machine learning model are described in Algorithm 3.3 and Algorithm 3.4 respectively.

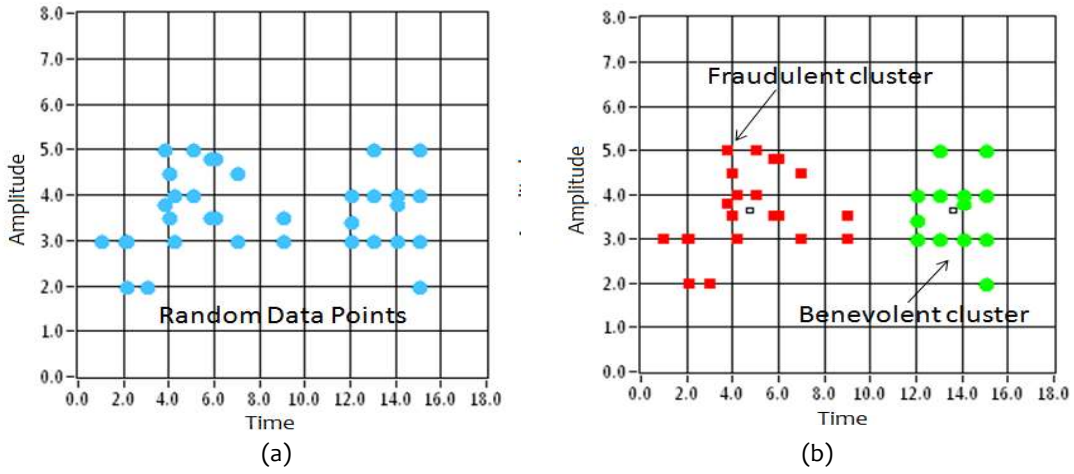


Figure 3.11: K-means Algorithm for Random Data: (a) random data measurements are fed into k-means algorithm; (b) Two clusters generated out of k-means algorithm

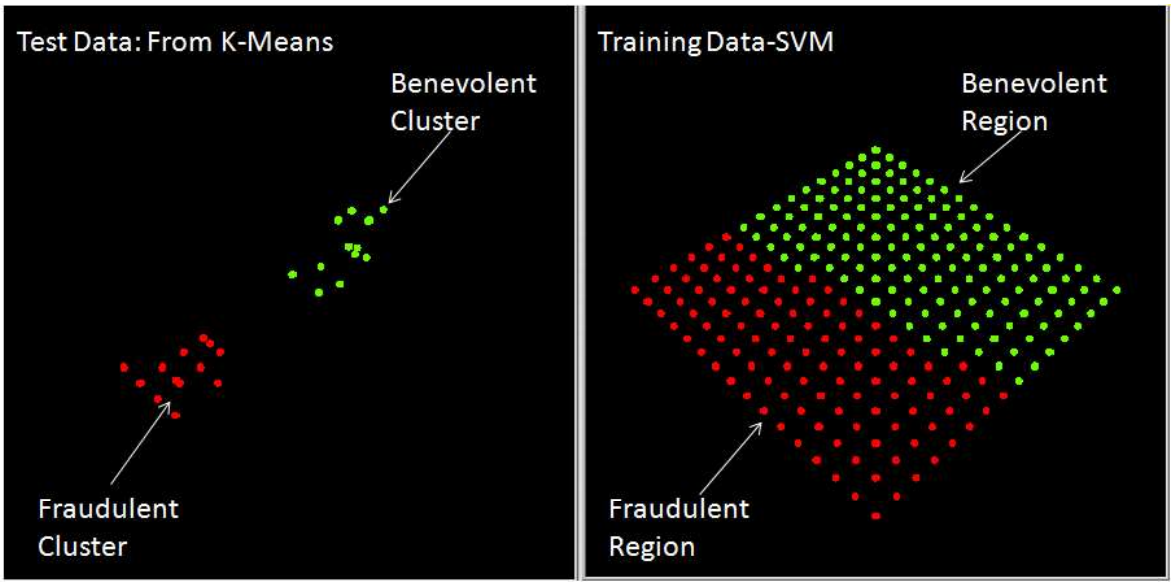
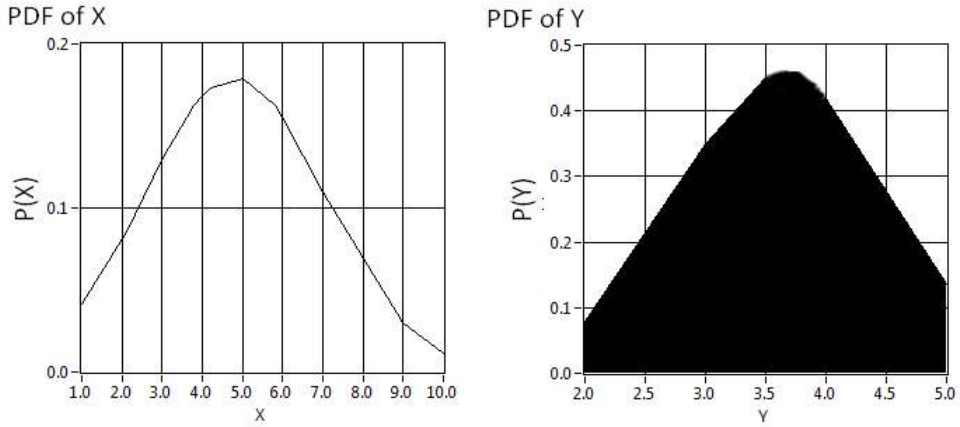
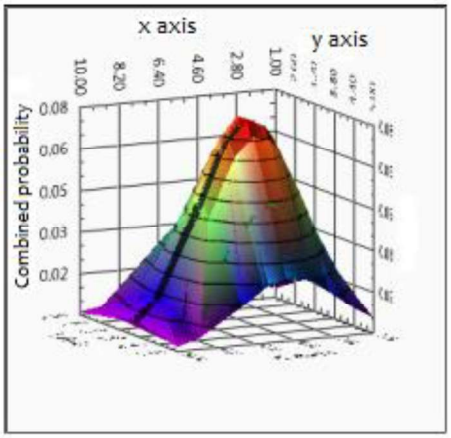


Figure 3.12: 3D view of Support Vector Machine: (a) The output of the k-means is fed into SVM; (b) Extrapolation of the measurements



(a) (b)



(c)

Figure 3.13: Probability Distribution Function of x (time), y (measurements) and Combined Probability (probability density) are Shown in (a), (b) and (c) respectively

ALGORITHM 3.3 TRAINING PHASE OF MACHINE LEARNING MODEL

Input:
m: Number of measurements
Training set $((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \ldots\ldots (x^{(m)}, y^{(m)})$ where $(x^{(i)}, y^{(j)}) \in R^{(n)}$. Here, $x^{(i)}$ denotes time and $y^{(j)}$ denotes measurements.
Output: Set of all fraudulent data measurements detected by the techniques of machine learning

```
1 Data acquisition
2 Clustering
3 Support vector machine
4 Anomaly detection engine
5     for each measurement in benevolent set do
6             if probability calculated using ADE < threshold (0.1) then
7                 Anomaly
8             Else
9                 not an anomaly
```

ALGORITHM 3.4 TEST PHASE OF MACHINE LEARNING MODEL

Input:
Run time data $((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \ldots\ldots (x^{(m)}, y^{(m)}))$ where $(x^{(i)}, y^{(j)}) \in R^{(n)}$.
g: Flag of antigen
Output: Detection of fraudulent node

```
1 Data acquired at run time
2 for each new measurement do
3     if acquired measurement is in benevolent set (Based on Algorithm 3.3) then
4       Set g = 1, (Initialization of immune model)
5     Else
6       send measurement to the training set for the update
```

3.3.2 Immune Model

Immune model guides a way to deal with fraudulent nodes which are detected as fraudulent nodes as an outcome of the machine learning model. The flowchart for the same is shown in Figure 3.14.

In the Immune model, Dibrov Model was taken as the foundation for the removal of fraudulent nodes. Simulations were carried out using the Runge-Kutta (variable) method for solving the Dibrov differential Eq. (3.10) and Eq. (3.11) and the following results were obtained as shown in Figure 3.15. It shows the graph of the rate of change of antigen and antibody as a function of time, for values of $K = 0.01$, $Q = 1$, $A = 1$, $R = 1$, $E = 1$ with initial conditions $a_0 = 0$ and $g_0 = 1$. This shows that initially, the antigen count is high which decreases when antibodies are produced in the body. In other words, when an antigen is detected, the B-cells start producing antibodies to form an antigen-antibody complex. The count of antigen decreases linearly with the increase in antibodies.

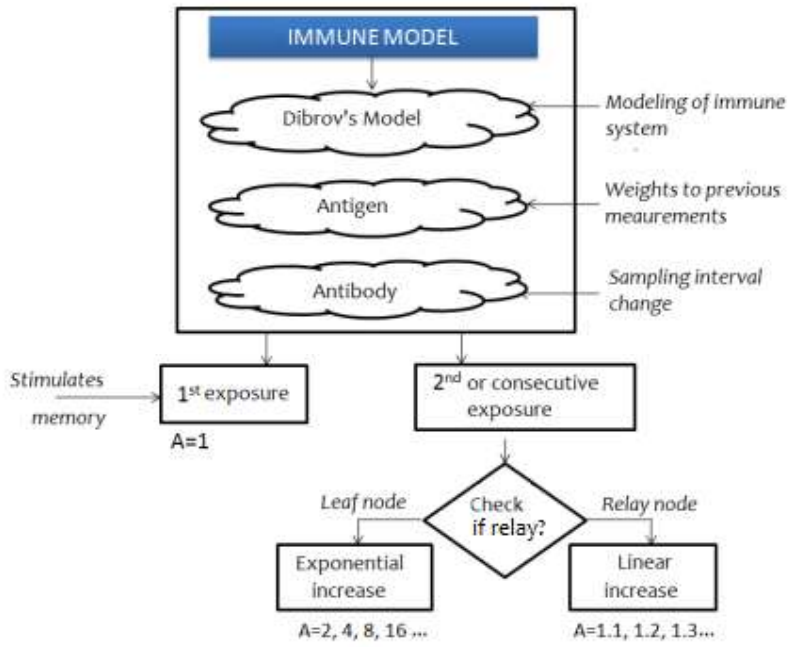


Figure 3.14: Flowchart of Immune Model

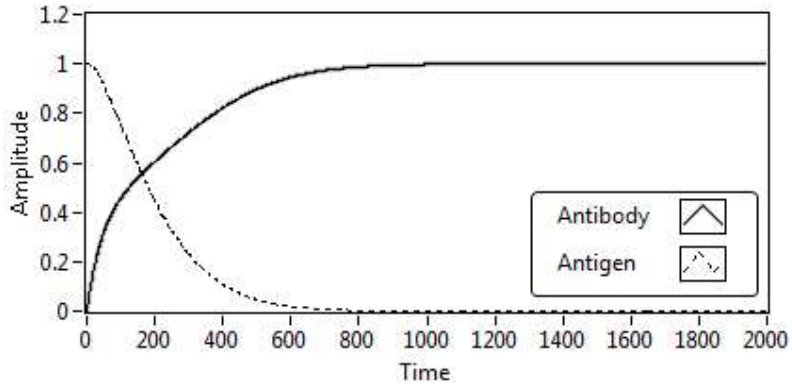


Figure 3.15: Rate of Change of Antigen and Antibody

Immune model in WSN focuses on giving correct measurements to the host controller even if the node is detected as fraudulent by the machine learning model. This is done so as to increase the lifetime of the node so that even if the node gets corrupted, it gives correct measurements. For this, weights are assigned to the measurement values of the fraudulent nodes. The weights assigned are proportional to the antigen values taken from the differential equations. Prior to the fraudulent node detection, the weight assigned is equal to 1. After the fraudulent node is detected, the expected measurement value relies upon the previous measurements and the weights would then be decreased proportionally to the antigen values taken from the differential equation as shown in Figure 3.16.

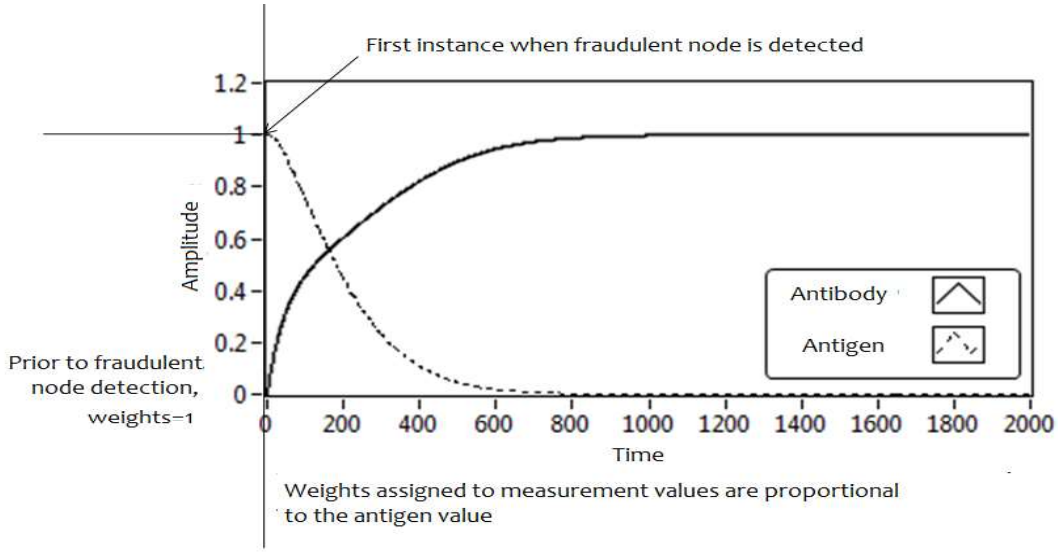


Figure 3.16: Relationship between Weights and Antigens

The measurements are calculated as per the following equation:

$$T_{new} = \frac{\sum_{i=1}^{k} W_i \times T_{prev}}{k} \quad (3.17)$$

where T_{new} is the new measurement obtained by applying weights, T_{prev} are the previous measurements, k is the history length fixed as 10, W_i are the weights which are kept proportional to the antigen values as per the following equation:

$$W_i = \theta \times g_i \quad (3.18)$$

where θ is a constant. However, weights are not normalized since they are kept analogous to the antigen values.

There are several ways to react to the fraudulent node. In order to minimize the effect of fraudulent nodes either the sampling interval can be increased or decreased. In WSN the measurements are acquired at a particular rate which is termed as a sampling interval. Prior to the fraudulent node detection the sampling interval is kept as constant. After the detection of fraudulent sensor node, it is required to reduce the effect of this node. However, turning off the fraudulent node immediately after detection is not a feasible solution since it would affect the stability of the system. So a better solution would be to either slowly decrease the sampling interval to zero or to increase the sampling interval depending upon the application. Increasing the sampling interval will not only help the host controller to check for on-off attacks but also help in increasing the lifetime of the fraudulent sensor node. The type of an attack, where the sensor node changes its behavior alternatively is known as an on-off attack [Lopez et al, 2012]. Increasing the sampling interval would allow the good measurements to last for a longer period since the rate at which the samples are collected gets increased.

The sampling interval is increased or decreased by taking into account the antibodies' value from the differential equation. The rate at which the sampling interval is increased is:

$$S_{after_attack} = 2^{\left\lfloor \frac{a_i}{\frac{a_{max}-a_{min}}{k}} \right\rfloor + 1} \times S_{prior_to_attack} \quad (3.19)$$

S_{after_attack} is the sampling interval after the fraudulent node is detected and $S_{prior_to_attack}$ is the sampling interval prior to the detection of the fraudulent node, a is the antibody value where a_{max} and a_{min} is fixed to 1 and 0 respectively. k is the number of steps desired to end the influence of fraudulent node (in this case it is fixed to 10).

The rate at which the sampling interval is decreased is:

$$S_{after_attack} = \frac{S_{prior_to_attack}}{2^{\left\lfloor \frac{a_i}{\frac{a_{max}-a_{min}}{k}} \right\rfloor + 1}} \tag{3.20}$$

Furthermore, the feature of the primary and the secondary response can be adapted in shutting down the fraudulent sensor nodes. In WSN, the shutting down process of the fraudulent node can depend upon the number of backup nodes:

(1) *Leaf Node*: Do not hamper the network stability. The response time to shut the leaf node should be rapid.
(2) *Relay Node*: Can hamper the network stability. It should be removed slowly.

Consider Figure 3.17 which shows node can be either a leaf node or a relay node. In Figure 3.17, *L1*, *L2* and *L3* are the leaf nodes and *R1* and *R2* are the relay nodes. Since the fraudulent node can either be a leaf node or a relay node, shutting off those nodes should be done proportionally. If the node is a leaf node it can be removed quickly, however if it is a relay node, the response time to shut off that node should be high since many leaf nodes would be dependent on it. This is done so as to improve the stability of the system. For instance, *R2* becomes fraudulent at some moment then *L3* node should route its packet to some other node other than *R2*. Therefore, shutting down process would depend on the number of backup nodes.

In order to adapt this feature for shutting down process, *A* constant used in Eq. 3.10 (rate of change of antibody) is increased either linearly or exponentially depending on the type of the node. The relay nodes which are detected as fraudulent should be removed slowly from the network. Linear response can be an effective solution in this regard. Linearity can be depicted by the following equation:

$$A(t) = \begin{cases} 1 & n = 1 \\ 0.1 + A(t-1) & n \geq 2 \end{cases} \tag{3.21}$$

where *A(t)* is the increase at time instant *t*, *n* is the number of exposure.

For shutting down the leaf nodes, exponential increase in *(A)* is an effective solution. Exponential increase of antibody (a) is mathematically modeled by the following equation:

$$A(t) = A(t-1)[1+r]^n \tag{3.22}$$

where *r* is the growth rate. Keeping *r = 1*, *(A)* is increased exponentially by following equation:

$$A(t) = \begin{cases} 1 & n = 1 \\ 2^n A(t_o) & n \geq 2 \end{cases} \tag{3.23}$$

where $A(t_0)$ is the initial antibody count i.e. equal to one. Algorithm 3.5 illustrates the overall algorithm for immune model.

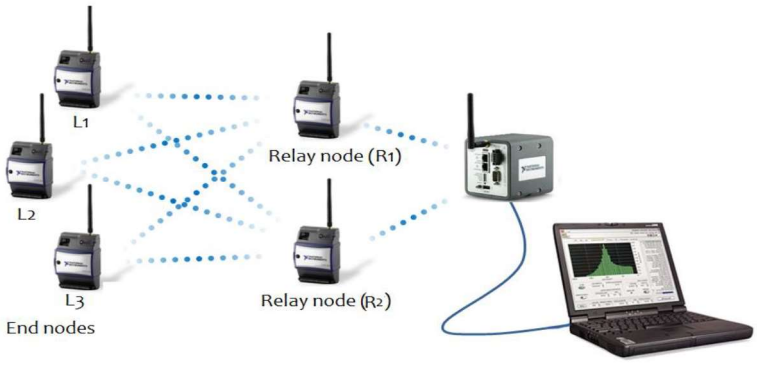


Figure 3.17: Leaf vs Relay node for Shutting Process

ALGORITHM 3.5 IMMUNE MODEL

Input:

$x^{(k)}$, $y^{(k)}$ as the measurement of a fraudulent sensor node

g: Flag for initialization.

Output: Turning off the fraudulent node slowly by ignoring its reading.

1 Set $g(t) = 1$ for that particular sensor node for others this parameter is zero
2 Solve Dibrov's differential equation.
3 Update the measurements of fraudulent node by assigning weights to its previous measurements using antigen values
4 **if** *first exposure* **then**
5 *Set A = 1*
6 **Else** *second exposure*
7 **if** *leaf node* **then**
8 *linearly increase A*
9 **Else** *relay node*
10 *exponentially increase A*
11 Change the sampling interval using values received from antibody production.

3.4 EXPERIMENTS AND RESULTS

Simulations were carried out in LabVIEW, on a network size of 10 sensor nodes out of which, one sensor node was introduced with high noise (See Figure 3.18). Noise faults are introduced where the temperature readings $T_{x,y}$ were replaced by $T_{x,y} + x_g$, where x_g was a Gaussian random variable, whose distribution was $N(0,2)$. Controlled environment setup is taken into account for training, where the temperature is consistent around 20^0-30^0C as shown in Figure 3.19(a). Any value outside this temperature range is considered as a fraudulent behavior.

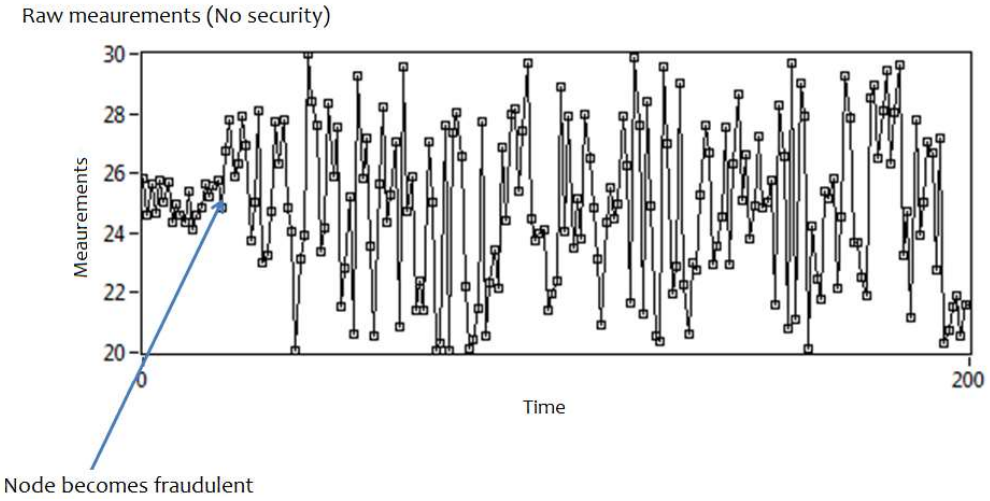


Figure 3.18: Fraudulent Sensor Node Measurements (Test Data)

3.4.1 Machine Learning Model

The sensor nodes were trained for 24 hours (a screenshot shown in Figure 3.19(a)), where machine learning tools like k-means, SVM and ADE were used. The k-means was used to

create two clusters namely, fraudulent cluster and benevolent cluster as shown in Figure 3.19(b) on account of training data. The data which was classified into clusters was given to SVM which created the decision boundary. Linear kernel and polynomial kernel are used for the simulations. However, polynomial kernel is preferred in this case where measurements till 36^0C were treated as benevolent measurements. This is because if the same type of data comes continuously it will be considered as a benevolent data as seen from Figure 3.20. The data which lies on the boundary needs to be further evaluated for better accuracy and precision as shown in Figure 3.20. ADE is used for the boundary values between the two regions as shown in Figure 3.21.

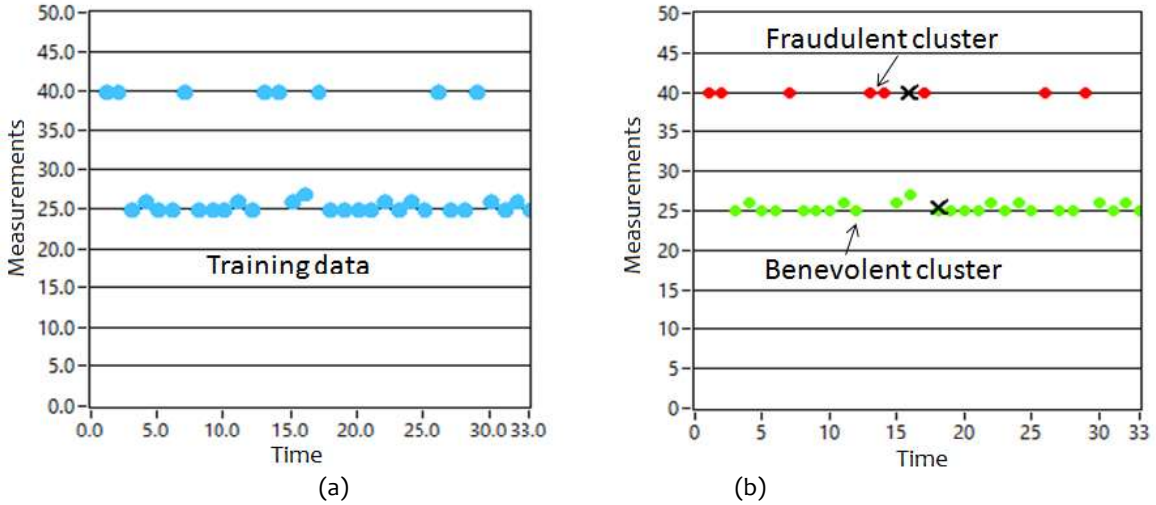


Figure 3.19: K-means Clustering: (a) Training set having temperature measurements; (b) Shows the learned data having two clusters generated out of k-mean algorithm having centroid assigned as mean value

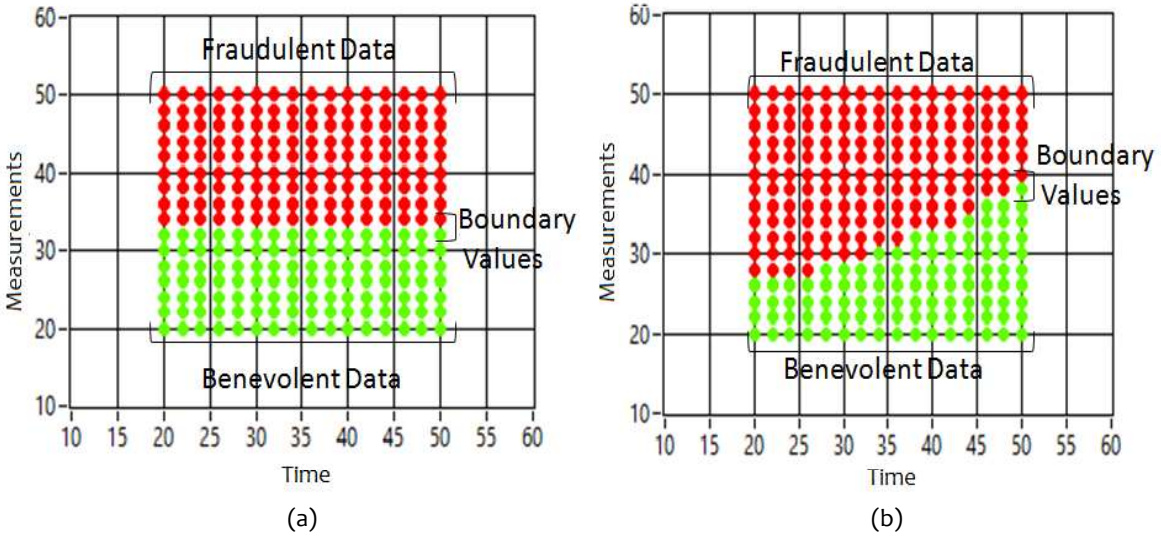


Figure 3.20: SVM Linear and polynomial output: (a) and (b) shows the extrapolation of data measurements using SVM linear and polynomial kernel respectively. Polynomial kernel is used for further processing because if the same type of acceptable data comes continuously it will be considered as a benevolent data as seen from the curve.

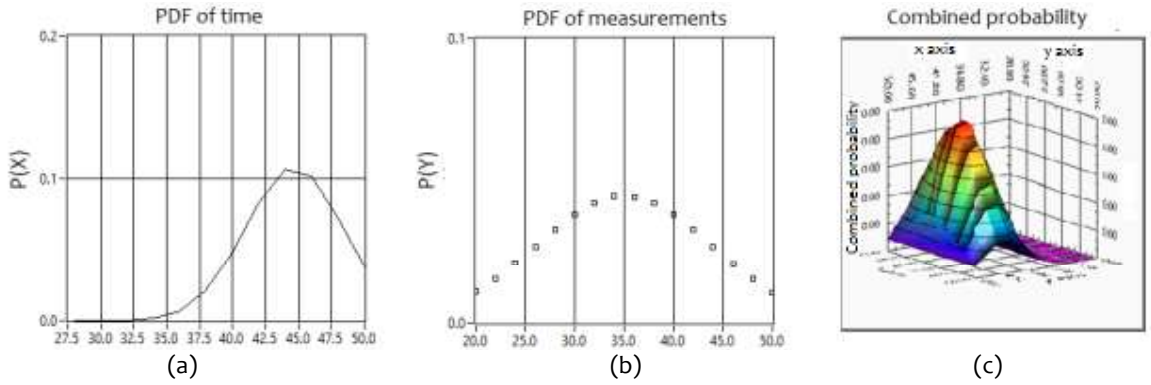


Figure 3.21: Probability Distribution Function of the Benevolent Measurements of SVM Polynomial Kernel: (a) PDF of x axis of the polynomial kernel i.e. the time; (b) PDF of y axis i.e. the temperature measurements. It explains the probability of having 25°C is 0.03; (c) Combined probability of the two

3.4.2 Immune Model

An Immune model is instantiated when the sensor node sends a fraudulent measurement which lies in the data set of fraudulent measurements as classified by the machine learning model. In an immune model, four different scenarios were considered for the purpose of comparison (See Figure 3.22):

(1) *Non-weighted averaging*: Simple averaging is performed, not assigning weights to the previous measurements. Here, the measurements would be varied according to the noise in the fraudulent node. Hence, the variance from the true measurement gets varied. Lifetime of the fraudulent node is infinity here, since there is no change in the sampling interval for capturing the future measurements.

(2) Weighted averaging: Weights are assigned to the previous measurements for the computation of new measurement readings. Here, the measurements would decrease since it is dependent on the weights given to the previous measurements as per Eq. (3.17) and Eq. (3.18). Weights are made proportional to the antigen values taken from the differential Eq. (3.10) and Eq. (3.11).

(3) *Weighted averaging and increase the sampling interval*: Besides weighted averaging, sampling interval is increased using Eq. (3.19). In this case, good measurements persist their state for longer duration making the lifetime of fraudulent node longer. The sampling interval is increased by taking into account the antibodies' value from differential Eq. (3.10) and Eq. (3.11).

(4) *Weighted averaging and decrease the sampling interval*: Weighted averaging is performed and sampling interval is decreased using Eq. (3.18). Here, variance from true measurements would be high which leads to a shorter lifetime of the fraudulent node. Also, when the sampling interval is changed, sampling interval of other channels would also get affected.

Figure 3.22 shows the results over the four scenarios. It can be seen that soon after 20 seconds, when the sensor node starts giving fraudulent measurements (Figure 3.18), host controller was able to get good measurements for a long time with the help of immune model. In case of weighted averaging and increase in sampling interval the lifetime of getting good measurements was high. However, with the increase in sampling interval, the measurements lasted for 200 sec. Furthermore, in case of weighted averaging and decrease in sampling interval, host controller was getting good measurements for 20 more seconds and then the node was getting turned off at the end of the simulation.

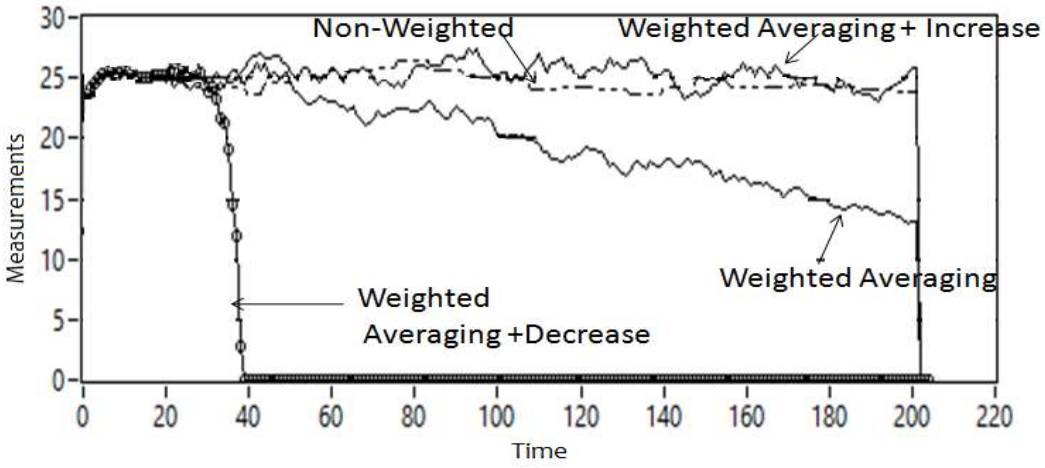


Figure 3.22: Measurements on Applying the Concept of Antigen and Antibody: Clearly it is seen that measurements would be good for some time even if the node becomes fraudulent

Table 3.1: Results Comparing Four Different Scenarios

Metric	Non-weighted Averaging	Weighted Averaging	Weighted Averaging + Increase sampling interval	Weighted Averaging + Decrease sampling interval
Fraudulent node lifetime (Response Time)	Infinity	Medium	Medium	Low
Variance from true measurement	Varied, depends on type of noise	High	Low	High
Impact on other channels in infected node	None	None	High	High

Table 3.1 gives the comparison to the four scenarios discussed above. As seen from Table 3.1, comparison is made on three metrics i.e. lifetime of fraudulent node, variance from true measurement and impact on other channels. Lifetime of fraudulent node should be less and in case of weighted averaging and decrease in sampling interval lifetime comes out to be the least. Variance from true measurement is less in case of weighted averaging and increase in sampling interval. The impact on other channels only happens in the last two cases. As seen from the results it is recommended to use weighted averaging. The sampling interval can either be increased or decreased depending upon the application. Evidently, the fraudulent node should be removed after its detection, for that reason weighted averaging and decrease in sampling interval is recommended. However, fraudulent node can be monitored for longer time to check for an on-off attack, where weighted averaging and increase in sampling interval would be preferred.

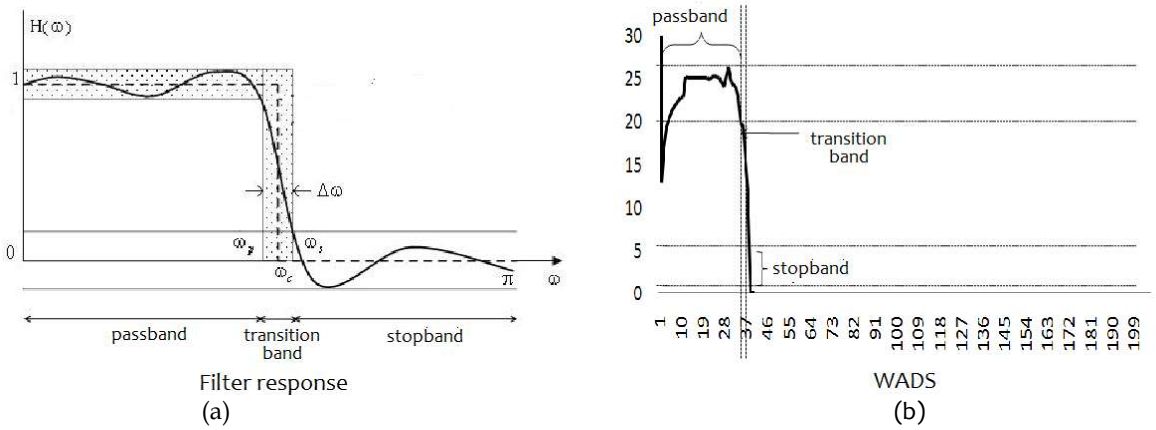


Figure 3.23: Filter Response and WADS Technique: (a) Filter response; (b) Weighted averaging and decreasing sampling interval can be compared beautifully with the filter response; there would be a passband, a transition band and a stop band where measurements would be neglected fully

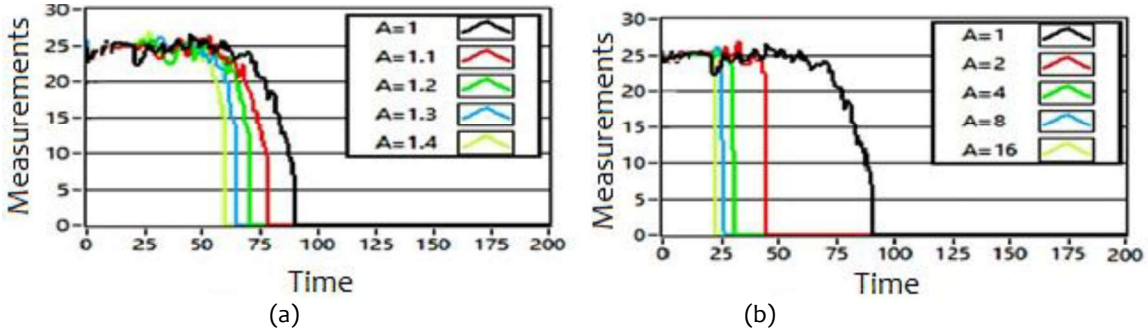


Figure 3.24: Secondary Response: (a) Linear increase of A on relay node. On second exposure the time required to turn off the fraudulent relay node is around 77 sec vs; (b) Exponential increase of A on leaf node. On second exposure the time required to turn off the leaf node is only 48 sec

Weighted Averaging and Decrease in Sampling interval (WADS) can be compared with the filter response. There are three bands in a filter namely passband, transition band and stop band. Passband response is the filter's effect on frequency components that are passed through. Frequencies in the middle which are not removed completely from the output signal, may receive some attenuation are represented by transition band. Frequencies in the stop band are removed completely from the output signal. Similar type of analogy can be established for the sensor nodes by assigning a pass band, a transition band and a stop band. These three phases can be compared very beautifully with the WADS approach as shown in Figure 3.23. There would be a passband where measurements would be taken from the fraudulent sensor node followed by the transition band giving the network some time to stabilize. Stop band is the phase where the measurements would be ignored totally.

Furthermore, primary and secondary responses of immune system can be adapted in WSN depending on the type of node. It was observed that while increasing the value of A as a linear function Eq. (3.19); the response time was low as shown in Figure 3.24(a) where time is in seconds. However, as A was increased exponentially Eq. (3.21), the response time was fast as shown in Figure 3.24(b). The figure explains that first exposure is same in both cases but it changes its behavior on second exposure depending on the type of the node.

3.5 SUMMARY

The chapter focuses on designing a security system for next generation WSN. In sensor network where fraudulent nodes can hamper the system it is necessary to remove these nodes without affecting the overall system. Various trust models have been developed for the detection of fraudulent nodes in the network. The proposed model is a mixture of machine learning model and immune model. Machine model is used for the detection of the fraudulent nodes and immune model is used for the removal of those nodes taking into account the antigen and the antibody concept. In the immune model, antigen values are used for the prediction of new values and antibody values are used to change the sampling interval. The immune model is a novel model for removal of the fraudulent nodes in the network. Furthermore, in the immune model different types of response have been applied depending upon whether the fraudulent node is leaf or relay node.

…