2
A Review of Security Challenges, Attacks, Trust and Reputation Models in Wireless Sensor Networks

Security issues have been studied on a variety of networks such as, wired networks, wireless networks, social networks etc. It is an important aspect which promises not only the reliability to the system, but also helps in sustaining the network. It is imperative to have a secure system where there is confidentiality and correctness to the data being sent from one system to another.

WSNs often monitors critical infrastructure and carries sensitive information which makes them desirable targets for attacks. Furthermore, since the network is openly accessible, anybody can access the deployment area which breaches the security of WSN. Some of the attacks in WSN may be facilitated by [Dargie and Poellabauer, 2011]:
(1) Remote and unattended operation
(2) Wireless communication
(3) Lack of advanced security features due to cost, form factor or energy
As a consequence, sensor networks require effective solutions to overcome from various types of attacks. Nevertheless, while designing efficient solutions for WSN security various challenges have been addressed.

2.1 CONSTRAINTS ON WSN

Obstacles are observed while setting up a WSN [Ringwald and Romer, 2007]. Resources of individual sensor nodes are intrinsically constrained in a WSN. They have limitations such as inadequate processing capability, communication bandwidth and storage capacity. The causes of having such limitations are due to limited energy and physical size. Additionally, these small sized nodes are highly expensive. Nevertheless, the processor and the radio used in sensor nodes consume a lot of energy while sending, processing and receiving data. Major energy consumers include sensor transducer, communication hardware and microprocessor computation. Owing to this, optimum communication with minimal processing is recommended in sensor nodes [Soderman, 2008].

Furthermore, as sensor nodes are deployed in hostile conditions where access to a permanent power supply is unlikely, nodes are configured with a battery source. Over and above that, if the battery becomes deficient, nodes turn superfluous. Also, since the nodes are remotely located, anybody can hamper the node not only at data level but also at infrastructure level, to breach the security of the network. As a result, it is critical to secure the overall system.

2.1.1 Threat Model

In WSNs, it is a common practice to assume that the attacker might be aware of the security measures that are deployed in a sensor network [Wang et al, 2006]. They may be able to compromise a node or even physically capture a node which in turn allows the attacker to steal the information within the node. Attacks in WSN can be classified into:

(1) *Outsider vs insider attacks*: The cause of an outsider attack is due to nodes which do not belong to the network. The insider attacks cause nodes of the same network to behave in an unintended manner.
(2) *Passive vs active attacks*: Passive attacks comprise eavesdropping or monitoring packets exchanged within a WSN; active attacks involve some modifications of the data steam or the creation of a false stream.

The metrics that can be used to evaluate security scheme in WSN should make the system resilient, energy efficient, flexible, and scalable. They should be fault tolerant, should have self-healing capability and high assurance.

2.1.2 Security Requirements for WSN

The goals of security services in WSN are aimed at protecting the information and resource from attacks and behavior [Sharma et al, 2010]. In general, the basic security goals are a set of properties which are listed below [Lee, 2007]:
(1) *Confidentiality*: The goal of confidentiality is to prevent any unauthorized usage or revelation of information in transit.
(2) *Integrity*: It aims to ensure that data being transferred has not been changed by the attacker. It safeguards the accuracy and completeness of the information.
(3) *Availability*: It is the quality of being at hand when needed which states that authorized users should have reliable and timely access to information.
(4) *Access control*: This property is required to permit or deny the use of an object by a subject.
(5) *Authentication*: It aims at validating a claimed identity of an end user or a device.
(6) *Authorization*: It is an act of granting access rights to the user.
(7) *Accountability*: It is the act of being explained and justifiable.
(8) Freshness: Data should be fresh and non redundant.
(9) *Robustness*: It ensures the network to be functional as well as capable of handling errors in catastrophic conditions and environment.

Consequently, while designing a security management system for the network, properties such as robustness, integrity, accountability should be attained for a concrete solution.

2.1.3 Focus and Contents

The present chapter gives the background and review for a secured WSN. The chapter is organized into three sections namely attacks, key management protocol and trust-reputation model as shown in Figure 2.1. Section 2.2 throws light on various types of attacks in WSN. Sections 2.3 and 2.4 illustrate ways to overcome from the attacks, where Section 2.3 gives a basic insight into key management protocol and Section 2.4 presents the details of trust and reputation models in WSN. Section 2.5 concludes the chapter.

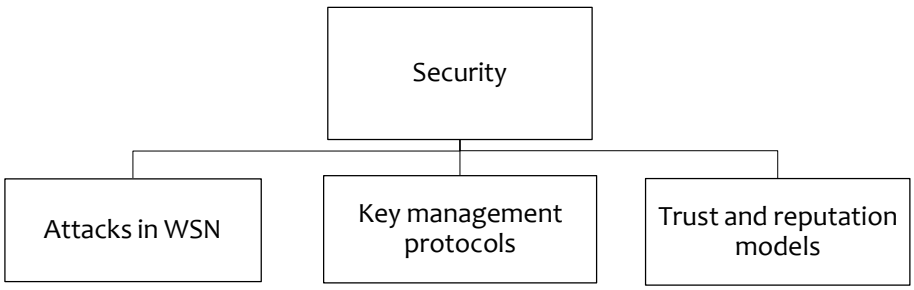

Figure 2.1: Organization of Chapter: The chapter is divided into three sections namely attacks, key management protocol and trust-reputation models.

2.2 ATTACKS IN WIRELESS SENSOR NETWORK

Attacks randomly come into picture which breaches the security of the system. It can aim on hindering the secrecy and authentication of the information by tampering the service integrity or it can modify and consume network availability. Attacks in WSN can be broadly classified into attack on data and attack on infrastructure (See Figure 2.2) [Baljak, 2012; Ni et al 2009; Sharma and Gose 2010].

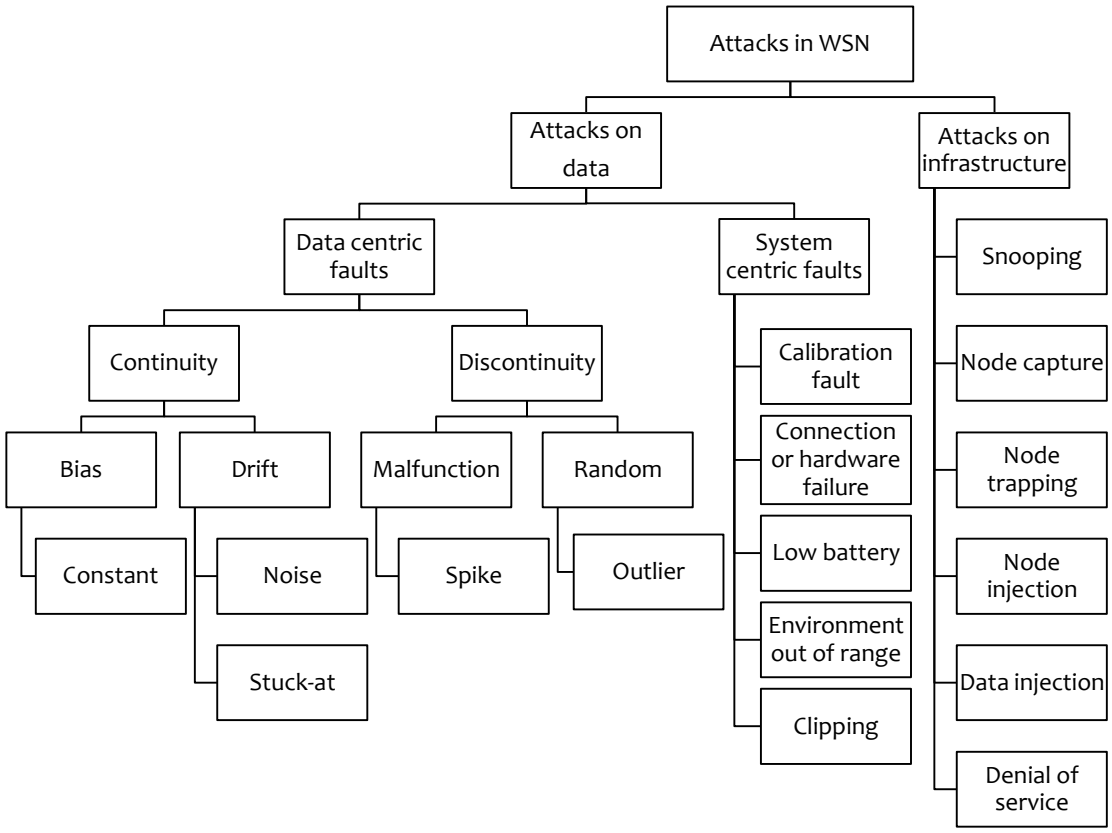

Figure 2.2: Attacks in WSN: Attacks on data modifies the data content whereas attacks on infrastructure claims the identity of the nodes and consume the resources of the nodes. Attacks on data can be further classified on data centric and system centric views. The two classification approaches are not completely disjoint and can overlap. The classification is on the basis of occurrence and pattern of fault. Various network layer attacks on infrastructure are shown in the figure.

2.2.1 Attacks on Data

Attacks on data can be simply defined as deviation from the expected normal behavior [Ni et al, 2009]. Attacks on data focus on stealing and modifying the data which includes injecting wrong packets, deleting packets and modifying the content. Data faults are classified on the basis of data centric and system centric views. Data centric faults change the data semantics and system centric faults are due to physical malfunction in the node. These two approaches are not disjoint and can overlap. For instance, spike fault can look like connection/hardware failure fault and stuck-at fault could look like clipping attack as shown in Figure 2.3.

(a) Data Centric Faults

Data centric faults can be classified on the basis of [Baljak, 2012]:

(1) Continuity of the occurrence
(2) Frequency of the occurrence
(3) Observable and learnable pattern

If a sensor reading is represented with $x_i + \varepsilon_i$, where x_i is the state information which are the physical parameters such as temperature, pressure etc. and ε_i is a fault, fault classification can be defined on the basis of continuity or discontinuity of occurrence as follows:

Continuity of the occurrence:

In continuous occurrence of faults, the sensor returns faulty data constantly, and a pattern can be observed in the form of a function (bias or drift). In bias, the function of the error is a constant or non zero but random mean i.e. $\varepsilon_i = constant$. The generated fault is termed as constant fault. It can have a positive or a negative offset.

In drift, the deviation of data follows a learnable function, such as a polynomial change. It can be observed in three forms:

(1) *Noise*: High noise and variance makes sensor values experience unexpectedly high variation which can be due to hardware failures, environment out of range or a weakening of battery supply. It is a very common and expected fault in sensor nodes.
(2) *Stuck-at fault*: Stuck-at is a series of data value that experiences zero variation and follows an unexpected jump in the behavior for a period of time. It is termed as stuck-at fault, since the data values get stuck-at one point and do not change their behavior for times greater than expected. The cause of this fault is due to sensor hardware malfunction.
(3) *Clipping*: A type of system centric fault which is caused when the environment exceeds the range of Analog to Digital Convertor. It exhibits either a ceiling or a floor function. It is defined under continuous faults because it follows a learned pattern where the frequency of occurrence is fixed.

Discontinuity of the occurrence:

Discontinuous means to occur from time to time on aperiodic basis i.e. occurrence of ε_i is discrete. It can be further divided into malfunction or random fault.

(1) *Malfunction fault*: The frequency of occurrence of faulty readings is $\varepsilon_i > \tau$, where τ is threshold frequency. Furthermore, there is no observable pattern in the fault occurrences. The fault which comes under this category is outlier faults. In case of outliers, isolated data points are observed on different time instances. The distance from the expected behavior is high and is often caused by unknown software inserted by the attacker.
(2) *Random fault*: The occurrence of the fault is infrequent and $\varepsilon_i < \tau$. Spike fault comes under this category where multiple isolated data points are observed outside the expected range of behavior. Random faults can occur due to the connection failures or other hardware problems like battery failure.

(b) System Centric Faults

System centric faults are the ones which are observed during sensor node deployment and while understanding real world parameters. They are of various types [Ni et al, 2009]:

(1) *Calibration fault*: Calibration of transducer response is required to increase the accuracy of the sensor nodes since their factory calibration might not be well suited to the conditions of the field. There should be consistent one to one mappings of output values to input values. Calibration error can also be of three types namely offset, gain and drift. In offset error, sensor faults are offset from the expected range by a constant amount. In gain error, measurements have offset with some gain value f which is a random multiplicative factor such as $x_i + x_i \times f$, where f lies within the interval for instance [0.1, 10.0]. In drift errors, measurements drift away from the original values.

(2) *Connection or hardware failures*: Poor connections or hardware malfunction of sensors can cause the sensor nodes to deviate from the actual measurements. Hardware malfunction occurs due to environment perturbations or sensor age. It is categorized by sending either very low or very high sensor measurements.
(3) *Low battery*: Battery is one of the limiting factors of sensor nodes. If the battery becomes deficient, sensor turns counter-productive. Battery losses occur when processor and the radio used in sensor nodes consume a lot of energy while sending, processing and receiving data. System performance is affected by battery supply by introducing noise and inaccurate measurements depending upon the type of sensor.
(4) *Environment out of range*: When the sensitivity range of transducer falls outside the acceptable range, errors due to environment occur. This will give way to much higher noise or flattening of the data.
(5) *Clipping*: A type of system centric fault (explained in data centric fault).

The observable pattern of all the attacks on data can be represented pictorially as shown in Figure 2.3. The results were generated in LabVIEW taking into account the fault pattern described above.

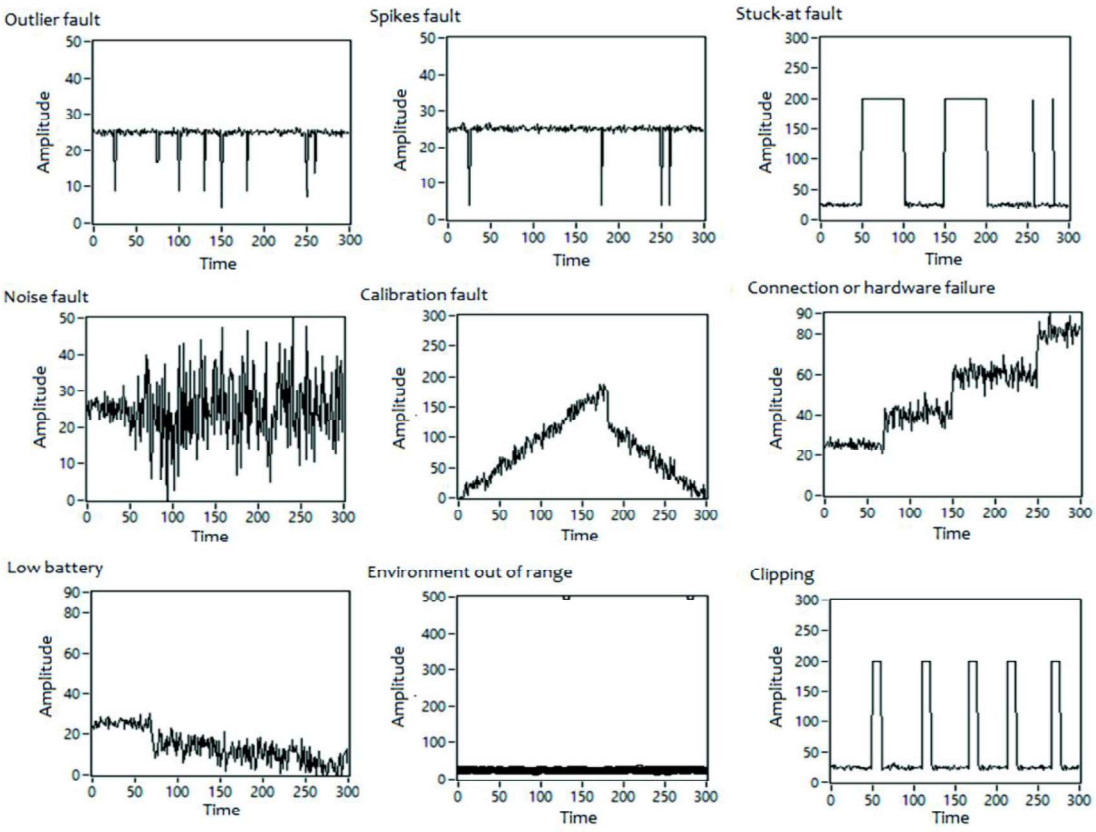

Figure 2.3: LabVIEW Simulations on Attacks on Data in WSN: The occurrence and pattern of fault is consistent in case of noise fault, calibration fault, connection or hardware failure, low battery and clipping fault. However, it is inconsistent in case of outlier, spikes, stuck-at and environment out of range faults.

2.2.2 Attacks on Infrastructure

Attacks on infrastructure claim the identity of the nodes and consume the resources of the compromised nodes. The attacks on infrastructure try to utilize the resources of the network

by sending unnecessary packets to the victim. These attacks can be seen at different layers of WSN protocol [Sharma and Ghose, 2010]. The protocol stack in sensor nodes contains physical, data link, network, transport and application layers.

(a) Physical Layer

Physical layer is responsible for frequency being selected, generation of carrier frequency, modulation, data encryption and signal deflection. At the physical layer, the attacker tries to jam the radio transmission by *jamming*. As the sensor nodes have limited power, this attack consumes a lot of energy. *Tampering* is another attack which can be seen at physical layer where the attacker compromises nodes and extracts sensitive information such as cryptographic keys or other data.

(b) Data Link Layer

Data frame detection, medium access, error control and multiplexing of data streams are managed by data link layer. At MAC layer, the attacker can send or violate guaranteed time slots forcing the other sensors to retransmit the data multiple times for consuming power. *Collision* is one of the major attacks where two sensor nodes transmit information on the same frequency. When the packets collide, changes occur in the data portion resulting in the deviation of checksum at the receiving end. *Exhaustion* is another attack where repeated collision occurs to cause resource exhaustion. *Unfairness* is another such attack, where the attacker causes other nodes to miss their transmission deadline in a real time MAC protocol.

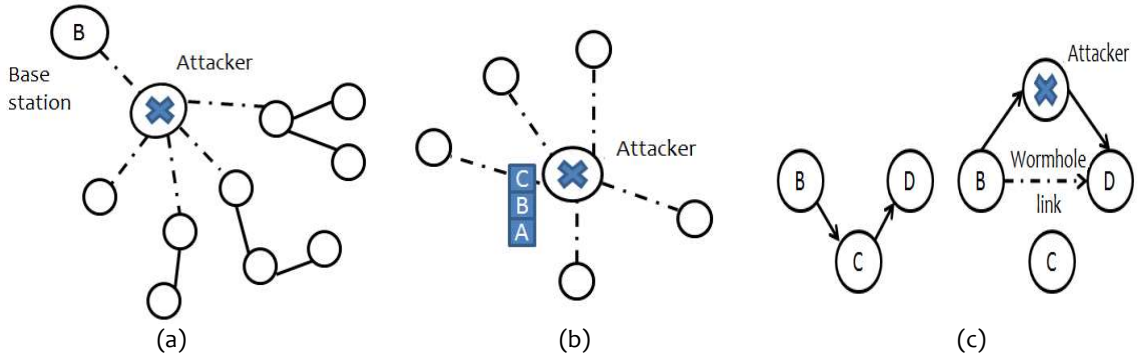

Figure 2.4: Network Layer Attacks: (a) Black-hole attack: Attacker captures the relay node and attracts all the traffic in the sensor network; (b) Wormhole attack: B base station broadcasts routing requests messages to all the nodes in the network where the attacker receives this packet being in the range of B; (c) Sybil attack: Attacker claims the identity of various genuine nodes (Source: Pathan et al, 2006)

(c) Network Layer

Forwarding of packets and the assignment of addresses is performed by network layer. At network layer, sending data long ways and disturbing the routing protocol can devastate global network performance. The common examples include spoofing, altering and replaying routing information. Black-hole, sink-hole, sybil, wormholes, hello flood and acknowledgement spoofing are the attacks observed at the network later. Some of the network layer attacks are:

Snooping

As WSN uses radio communication for transmission, any attacker which is in the range of the network can monitor the communication. It can listen to the data and record it.

Node capture
As nodes are not placed in a secure location, attacker can capture the node and can get access to all its information. Black-hole/sink-hole attack is a type of node capturing attack [Ahmed et al, 2005]. An attacker acts as a black-hole and attracts all the traffic in the sensor network. See Figure 2.4 (a).

Node trapping
In node trapping attack, the attacker can trap the node and can tunnel the data to some other place as shown in Figure 2.4 (b). Wormhole is a type of node trapping attack.

Node injection
To accomplish a task, the sensors may be required to work together in WSN. The sensors can accomplish the task by usage of redundant information and distribution of subtasks. If multiple identities are used by an attacker, then this type of attack is classified as Sybil attack (Figure 2.4 (c)). Sybil attack can be performed for attacking the routing mechanism, distributed storage, data aggregation, voting, resource allocation and misbehavior detection [Newsome et al, 2004].

Data injection
The attacker can inject fraudulent data without joining into the network. Replay attack is an example of data injection attack where the data is either delayed or repeated [Lupu, 2009].

Acknowledgement spoofing
Acknowledgements are sometimes required to be transmitted by routing algorithms used in sensor networks. Here, the acknowledgements of overheard packets, are spoofed by the attacker. These packets are intended for neighbouring nodes and provide them with false information. A typical situation is claiming the node as alive when in fact it is dead.

(d) Transport Layer

Transport layer is responsible for specifying the reliable transport of packets. *Flooding* is one such attack where an attacker may exhaust the resource which each connection requires. In such a scenario, the attacker may also reach a maximum limit of attempted connections.

(e) Application Layer

Application layer is responsible for specifying how the data is requested and provided for both individual sensor nodes and interaction with the end user. At application layer, applications can be forced to do extensive computation which uses the limited memory of sensor nodes.

The taxonomy of various types of faults and attacks elucidates that, there is a requirement of secure transmission of data or a mechanism where the identification of fraudulent nodes can be done which are behaving in an unpredictable manner.

2.3 KEY MANAGEMENT IN WSN

A key management protocol is used to encrypt and authenticate the data which can limit the attacks on WSN. The protocol must establish a key between all sensor nodes that wants to exchange data securely. The general idea behind this technique is to process, revoke and distribute keys. The hardest part in key management is the establishment and revocation of keys. The steps involved in key management protocol are:

(1) At the outset, nodes are initialized. It is the process of loading initial security settings.
(2) In the second place, keys are generated in a secure and random manner.
(3) To negotiate a key between the two sensor nodes without allowing the attacker to find out the key, a key establishment protocol is used. In this step, the key is transported with the help of key agreement steps. There are three types of general key agreement schemes:

trusted-server scheme, self-enforcing scheme, and *key pre-distribution scheme* [Zia et al, 2007]. Trusted server scheme like Kerbores depends on trusted server [Neuman and Ts'o, 1994]. This type of scheme is not preferable in WSN because of lack of trusted server. Self-enforcing scheme uses asymmetric cryptography such as public key certificates, Diffie-Hellman key agreement [Großschädl et al, 2007] or RSA [Kayalvizhi et al, 2010]. However, because of computation and energy constraints this scheme is seldom used. Key pre-distribution is the third type of key agreement scheme where the distribution of key information is performed prior to sensor node deployment. Multiple key pre-distribution schemes are applied in WSN. Allowing all the nodes to carry a master secret key can be one of the solutions. To achieve key agreement, new pairwise keys can be obtained by any pair of nodes utilizing the master secret key. Another solution is to let each sensor carry n– 1 secret pairwise keys, each of which is known only to this sensor and one of the other n - 1 sensors (assuming n is the total number of sensors) [Du et al, 2004]. The security of communications among nodes is not affected while compromising a single node. This provides an advantage to the scheme and makes it more resilient. However, as *n* could be large, this scheme is impractical for sensors which have limited amount of memory.

(4) A distributed system is required to control the keys for minimizing the communication. To avoid transporting large amount of usage information, the system is divided into key classes.

(5) Later, revocation of broken keys is made. Key revocation scheme was initially introduced by [Eschenauer and Gligor, 2002]. Here, the controller node broadcasts message to each node in the network informing about the compromised keys. A signature key is sent to each node prior to the broadcast message. Schemes were introduced later where revocation was made using a trusted third party or performing voting in the network or via time stamps [Soderman, 2008].

(6) Moreover, keys are to be stored. Keys should be minimal because of memory constraints of sensor nodes. Therefore, if the attacker gets physical access to the sensor node, its keys are broken to release memory.

The computation requirement for creating keys is very high. This is because; the embedded processors used in wired or ad-hoc networks are more powerful than those used in WSN. As such, complex cryptographic keys cannot be used in WSNs. Also, the memory used in sensor nodes usually includes flash memory and RAM. Flash memory is used for storing information and RAM is used for processing the computation. Therefore, there is not enough space to run complicated algorithms in sensor nodes. Also, a secure way for sending the private key is a challenge when you have large number of nodes. Key management protocols cannot alone provide security to WSNs for the attacks described in the above section. Therefore, WSN should have mechanisms to detect and react to various intrusions. An intrusion detection method could be a better solution in this respect where the network monitors for suspicious activity outside the normal and expected behavior. For this, cooperation among sensors is required to detect and nullify the effect of intruders in the network. Cooperation can be attained by building trust and reputation models for WSN.

2.4 TRUST AND REPUTATION MODEL

Trust and reputation are very important factors in the decision making processes of any network where uncertainty is a major factor for ruining the overall system. *Trust is a personal and subjective phenomenon that is based on various factors or evidence* [Jøsang, 2007] and *reputation is a collective measure of trustworthiness in a sense of reliability based on the ratings from the members of a community* [Boukerch et al, 2007].

In the context of wireless sensor network, while sensing data, nodes can produce ratings based on direct observation (known as first-hand information) and indirect observation (also known as second-hand information) as shown in Figure 2.5. Direct observation is the trust

ratings calculated selflessly, by taking into account many factors such as packet delivery ratio, amount of energy left, hardware error, deviation from sensor readings etc. It is more or less based on observations and experience, whereas indirect observation contains the observations from other nodes. Based on first-hand information and second-hand information, trust is calculated. Trust engine regularly calculates trust ratings based on risk engine decision [Momani and Challa, 2010]. Both trust engine and risk engine are dependent on each other in terms of user ratings and neighbour weights. Finally, trust ratings are generated which are regularly updated with time.

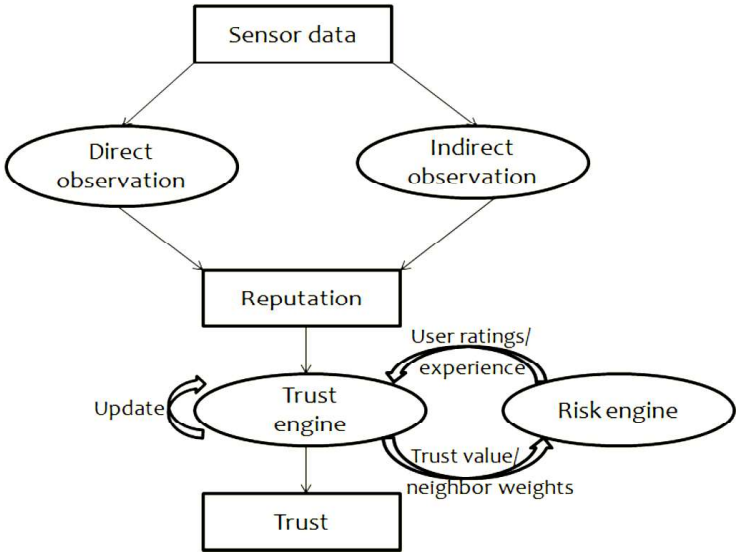

Figure 2.5: Trust Calculation Based on First-hand Information and Second-hand Information

2.4.1 Methods to Calculate Trust

There are four basic models to calculate trust as shown in Figure 2.6 [Reddy, 2012]:

(1) *Reputation Based Model*: Uses packet transfer rate to calculate trust.
(2) *Event Based Model*: Trust is calculated at particular or specific timed events or periodically.
(3) *Collaborative Model*: Models are developed to calculate trust based on first-hand and second-hand information.
(4) *Agent Based Model*: In this model, an agent node is introduced from a cluster of nodes within communication distance. This agent node is used to store the packet transfer information. Initially, all the nodes are given r-certificate which shows either they are highly trusted or not trusted at all [Mármol and Pérez, 2011]. Now consider two nodes n_i and n_j which want to communicate with each other. Here, n_i asks for r-certificate of n_j. Based on the certificate provided, n_i decides whether to send information or not. After the transaction is over, n_i makes trust evaluation on n_j based on quality of service it gets and submits its evaluation to its local mobile agent and sends t-instrument (local to n_j) for n_j. On receiving it, n_j updates its r-certificate.

2.4.2 Methodologies to Model Trust

In general, trust model takes into account both direct and indirect observations. There are various models which are developed to model trust. These are listed below:

(a) Ratings

This was the initial model which was developed that calculated trust on direct summation of first-hand information and second-hand information as seen from equation:

$T_{ij} = (T_{ij})_D + (T_{ij})_{ID}$ (2.1)

where T_{ij} is the trust of j node with respect to i node, D is the direct information and ID is the indirect information.

It was a very basic model which had lower detection rate of fraudulent nodes. The model was simple to implement and did not incorporate the detection of various attacks [Resnick et al, 2000; Jonker and Treur, 1999].

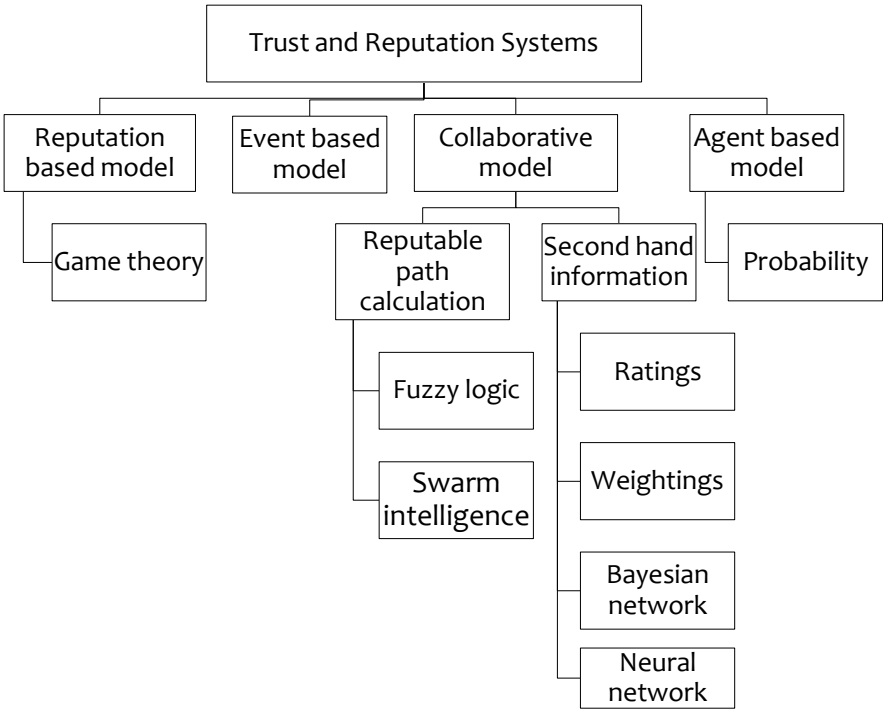

Figure 2.6: Trust and Reputation Models Classification: The classification is on the approaches such as reputation based, event based, collaborative based and agent based

(b) Weightings

In this model, weights were assigned to the indirect observation [Momani, 2008]. Nodes monitor the behavior of other nodes and compute the trust with the help of following equation:

$T_{ij} = (T_{ij})_D + w \times (T_{ij})_{ID}$ (2.2)

where w is the weight whose value vary between 0 to 1.

A weighted approach was proposed where initially every node was highly trusted with weights (= 1) [Atakli et al, 2008]. The network was adapted in the architecture between a group of sensor nodes and their forwarding node (FN) as shown in Figure 2.7.

The FN collects all information provided by sensor nodes and calculates an aggregation result A_r using the weights w_n assigned to each sensor node as given in the formula:

$$A_r = \sum_{i=1}^{n} w_n \times x_n \quad (2.3)$$

where x_n is the data acquired from sensor nodes. With the help of following equation weights w_n were updated for each node based upon the variation of A_r with respect to x:

$$w_n = \begin{cases} w_n - \theta \times r_n & if\ x_n \neq A_r \\ w_n & elsewise \end{cases} \quad (2.4)$$

where θ is a constant value and r is the ratio of sensor node sending fraudulent data to the total number of nodes under the same FN.

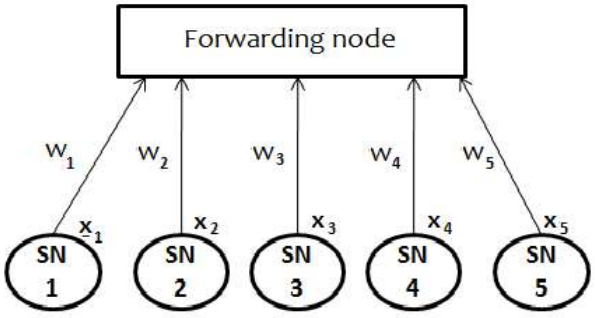

Figure 2.7: Topology of Weightings Method: Forwarding node collects all information provided by sensor nodes $SN_1 \ldots SN_5$ and calculates an aggregation result using $W_1 \ldots W_5$ and $x_1 \ldots x_5$ (Source: Atakli et al, 2008)

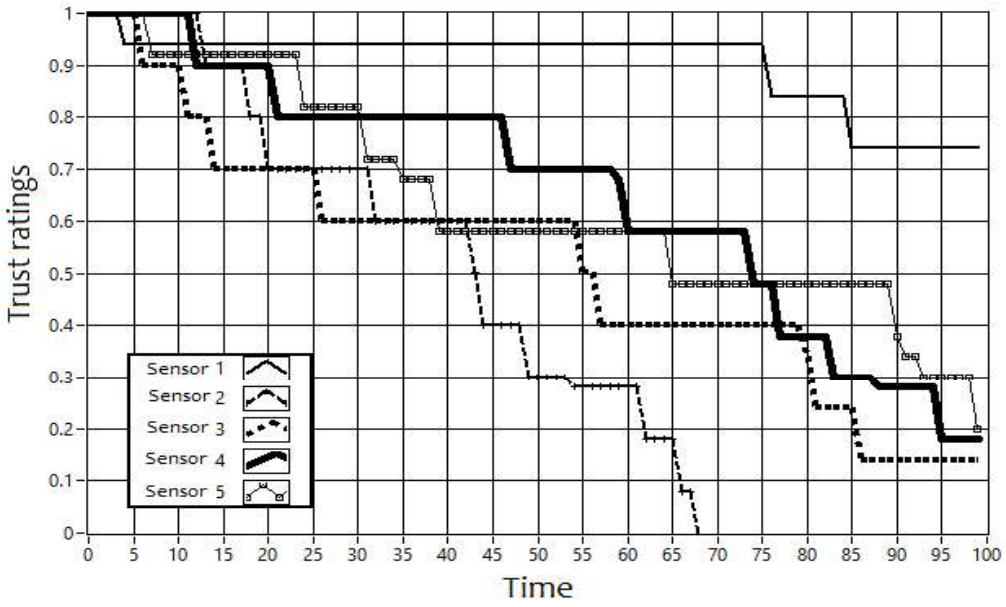

Figure 2.8: LabVIEW Implementation of Weightings Method: Showing how the weights are decreased on account of the information provided by the neighbouring nodes

Simulations are carried out for a network size of 5 nodes where initially every node has a weight equal to one and θ value equal to 0.1. Measurements of sensor nodes are in the range [10-20]. Deviation of each sensor node is checked with the mean value of past observations. If the variation was greater than 0.1 then weights are updated with the help of Eq. 2.4. Figure 2.8 shows the result depicting the rate of change of weights/trust ratings. The figure represents how the trust ratings are decreased with respect to time. The trust ratings decrease to zero since there is no rule for increase in trust ratings when the sensor nodes start behaving well. Once the trust ratings decrease, it never increases.

The work presented by Atakli et al, 2008 states that weighted approach is found effective only for small sized networks. For large sized networks, number of forwarding nodes was substantially increased. However, increasing the number of forwarding nodes subjects the network to another problem known as node-clustering problem. Furthermore, the approach is based on the assumption that base stations are trusted. Hence, if the attacker could gain control over the base stations, it can do any possible attack against the WSN.

(c) Probabilistic Method

An agent-based trust model for WSN was presented in [Chen et al, 2007] to observe the behavior of nodes and broadcast their trust ratings. The agent nodes monitor the sensor nodes and send the trust ratings to them. The sensor nodes on receiving them, compute and broadcast those trust ratings. According to the received information, sensor nodes make the decision about cooperation with their neighbours. This method of calculating the trust ratings maintains state count for every positive and negative outcome/decision [Chen et al, 2008]. Data collected is checked with respect to two parameters:

(1) *Routing*: looks at the data forwarding behavior.
(2) *Data processing*: monitors raw sensing data, performs data aggregation and checks delay in data transmission.

For every positive outcome on above parameters, positive count (p_c) was incremented and for every negative outcome negative count (n_c) was incremented. Given $\{p_c, n_c\}$ the probability x of obtaining a positive outcome was computed as follows [Chen et al, 2007]:

$$P(x) = P(x| < p_c, n_c >) = \frac{x^{p_c}(1-x)^{n_c}}{\int_0^1 x^{p_c}(1-x)^{n_c}dx} \quad (2.5)$$

Furthermore, certainty of each event was calculated using:

$$c(p_c, n_c) = \frac{1}{2}\int_0^1 \left| \frac{x^{p_c}(1-x)^{n_c}}{\int_0^1 x^{p_c}(1-x)^{n_c}dx} - 1 \right| dx \quad (2.6)$$

Trust was then calculated using the following equation:

$$Trust = \frac{p_c+1}{p_c+n_c+1} \quad (2.7)$$

Probabilistic method does not require second-hand information. It also guarantees better results than cryptographic results [Chen et al, 2007]. However, this scheme limits attacks such as bad-mouthing attack, on-off attack and conflicting behavior attack.

(d) Bayesian Network

Bayesian network uses Baye's rule as the principle rule. Baye's rule expresses how a subjective degree of belief should randomly change on account for evidence [Momani et al, 2008]. The advantage of this technique is that the framework is generic and allows more components to be added to and/or deleted from the framework very smoothly and transparently besides the direct and indirect observation as seen in Figure 2.9. Using Bayes' theorem, the probability of the total trust, given the direct and indirect observation, can be presented, as shown in equation [Momani et al, 2008]:

$$P(T|D, I) = \frac{P(D|T,I) * P(T|I)}{P(D|I)} \quad (2.8)$$

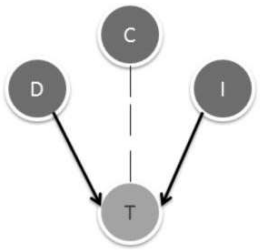

Figure 2.9: Bayesian Network: D and I are direct and indirect observation respectively. T stands for total trust computed on account of both D and I. Framework is generic since a third component C (communication observation) can be added.

A reputation based trust system was proposed to detect the selfish nodes using Bayesian formulation specifically beta reputation system [Ganeriwal et al, 2008]. In this method, r

eputation is calculated using a watch dog mechanism. After the packet gets passed to other nodes, the packet gets eavesdropped by each node in the network. This is performed to confirm that the packet has reached the destination. Based on the reputation scheme, the trust value is formulated.

The main disadvantage of Bayesian network technique is that the trust evaluation was based on node's QoS property which makes the false positive rates to range between 0.31 to 0.68 while the false negative rates range between 0.11 to 0.28 [Meng et al, 2013]. Furthermore, flat wireless structure was used which does not allow more nodes to be added in the network [M R Raj et al, 2013].

(e) Game Theory

Game theory is a method that deals with multi-person decision-making situations. It is designed to address situations in which the outcomes of a persons' decision depends not just on how they choose among the several available options, but also on the choices made by the people with whom they interact. It is used to determine a preferred strategy where such interactions are in play. A game comprises of a set of players, a set of strategies for each player, and a set of corresponding utility functions. Throughout the game, a strategy for a player is to perform complete plan of actions, in all possible situations. In any game, the players try to act greedily, according to their preferences, to maximize their consequences. A utility function expresses these preferences. These preferences map every consequence to a real number. Nash equilibrium provides a solution to describe a steady state condition of the game. A strategy is not changed by a player unless a better strategy is available that can result in more utility that is favorable for the current player. The normal form of a game is given by a tuple [Shigen et al, 2011]:

$$G = (I, S, U) \tag{2.9}$$

where G is the particular game, I is the set of players, S is the set of strategies used by the players and U is the set of utility functions that the player wishes to maximize.

Game theory can be used to capture the nature of conflict in WSNs security. In WSN, game theoretical model tries to mathematically capture the behavior of nodes in situations where the decisions depend upon the behavior of the other nodes. Transaction between the nodes demands some resources such as energy and storage for proper functionality [Chen et al, 2011]. Thus, the nodes may choose the selfish non-cooperative behavior to think about the limited resources for themselves. The node can behave cooperatively (CP) or non-cooperatively *(NCP)* giving it a game theoretical approach [Chen et al, 2011].

$$\begin{matrix} & CP & NCP \\ CP & \begin{pmatrix} R_p - C_e & -C_e \\ R_p & C_e \end{pmatrix} \\ NCP & \end{matrix} \tag{2.10}$$

Following scenario is observed:

(1) If both nodes choose cooperative strategy then they would gain R_p units of pay-off and consume C_e units of energy.
(2) If both of them use a non-cooperative strategy each of them gains a zero unit of pay-off.
(3) If one chooses a cooperative and other chooses a non-cooperative strategy, cooperative will gain zero and consume C_e units of energy and non-cooperative will gain R_p units of pay-off.

According to different applications, a taxonomy was proposed by [Shigen et al, 2011], which classifies the current existing typical game-theoretic approaches for WSNs security into four sets: preventing Denial of Services (DoS) attacks, intrusion detection, strengthening security, and coexistence with fraudulent sensor nodes as shown in Figure 2.10. To prevent DoS attacks, cooperative games [Agah et al, 2004], non-cooperative games [Agah et al, 2005] and repeated games [Agah and Das, 2007] have been proposed. Intrusion detection includes non-cooperative games [Reddy, 2009] and Markov games [Alpcan and Basar, 2006]. To further

strengthen security, auction theory [Agah et al, 2006] and coalitional games [Li and Lyu, 2008] have been suggested. For the detection of fraudulent sensor nodes, signalling games have been recommended [Wang et al, 2009]. Interested readers can get better insight by going through these papers.

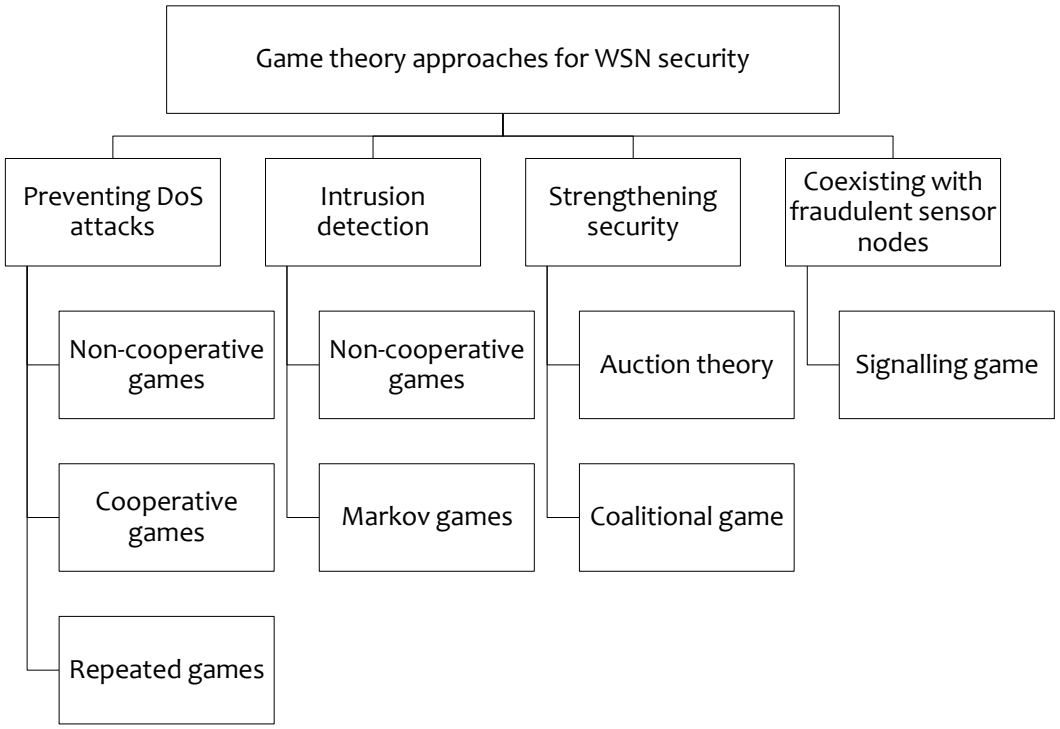

Figure 2.10: Taxonomy of Game Theoretical Approaches in WSN Security

The main disadvantage of game theoretical technique in WSN security is its computational complexity which is hard to implement. Also, the detection percentage is ranged between 30-60% [Shigen et al, 2011].

(f) Neural Network

Neural network approach employs analytical redundancy to estimate the values provided by a sensor node by considering the past/present values given by the adjacent neighboring sensor nodes. This estimate is compared with the actual value of the sensor node to increase/decrease its trust factor accordingly. In other words, sensor node compares the present value with the estimated value to increase/decrease the trust factor accordingly where the value provided by neighborhood nodes is cumulative estimation based on present values and past history as shown in Figure 2.11 [Curiac et al, 2007]. Consider the selected sensor node numbered as 5 in Figure 2.11 which sends the data x_A, whose neighbours 1, 2, 3, 4, 6, 7, 8, 9 send their data to the estimation and prediction block to compute x'_A. Error is calculated between the two values as:

$$Error = x_A - x'_A \tag{2.11}$$

Lesser the error between the two, more is the trust ratings and vice versa.

A typical ANN block has a block diagram as shown in Figure 2.12 (a). Here, 'input' is the sensor measurement and 'desired output' is the mean of past values. It is a typical 2-layer feed forward neural network (See Figure 2.12 (b)) [Auralius, 2009]. The activation function of the input layer is an identity function and output layer has a sigmoid function. It has two basic processes namely feed forward process and back propagation process [Jerome et al, 2005].

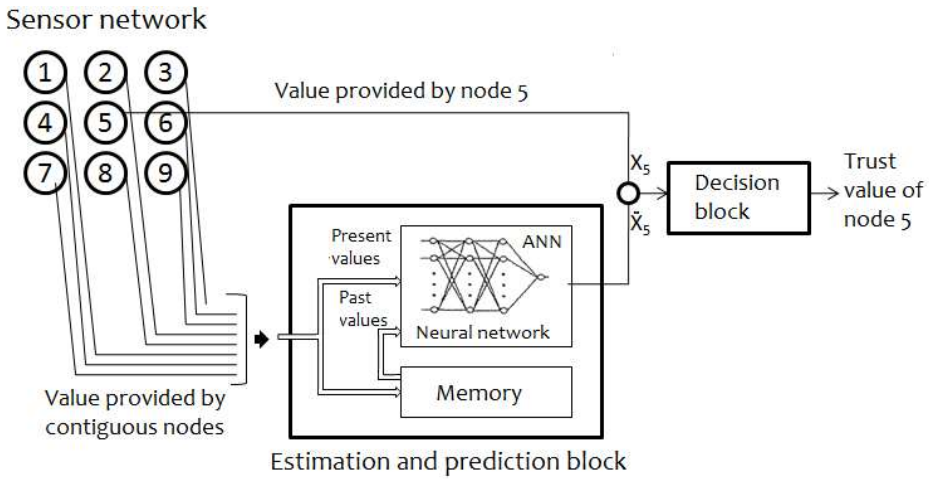

Figure 2.11: Neural network Approach: It has two blocks namely estimation & prediction block and decision block. The estimation and prediction block has a neural network block termed as Artificial Neural Network (ANN) which takes in the present and the past values of sensor nodes (Source: Curiac et al, 2007)

In the feed forward process, the network will receive single input and produce single output (See Figure 2.12 (c)). The input signal will be broadcasted to each neuron in the next layer with the help of following equations [Bhanot, 2008]:

$$h_j = \sum_k w_{jk} x_k \tag{2.12}$$

$$v_j = \frac{1}{1+\exp(-h_j)} \tag{2.13}$$

$$s_i = \sum_i w_{ij} v_j \tag{2.14}$$

$$y_i = \frac{1}{1+\exp(-s_i)} \tag{2.15}$$

In the back propagation process, weights are adjusted to minimize the errors with the following equations [Bhanot, 2008]:

$$\delta_i = y_i(1-y_i)({y_i}^d - y_i) \tag{2.16}$$

$$\delta_j = v_j(1-v_j)\sum_i \delta_i w_{ij} \tag{2.17}$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_i v_j \tag{2.18}$$

$$w_{jk}(t+1) = w_{jk}(t) + \eta\delta_j w_{ij} \tag{2.19}$$

Back propagation of errors was followed by training the neural network. Through this training, the sensor dynamics is identified by the neural network. The objective was to achieve identical behavior between the selected sensor network and the neural network. Accordingly,

for the selected sensor node, current value is given as the input and the mean of past values will be given as the desired output. While on training it was necessary to adjust the bias to reduce the error. Bias was adjusted using trial and error method where first bias was set close to zero. When the neural network converges to a certain value with an intolerable error, the bias value was increased gradually. After training, the best results were achieved with the parameters shown in Table 2.1. With these values, the neural network controller shows a good performance in term of input tracking capability as seen in Figure 2.13. The graph shows the black line as the current measurement and red line as the learned value from ANN. The lesser the gap between the two, higher are the trust ratings. The overall algorithm for the neural network approach in WSN can be explained in Algorithm 2.1.

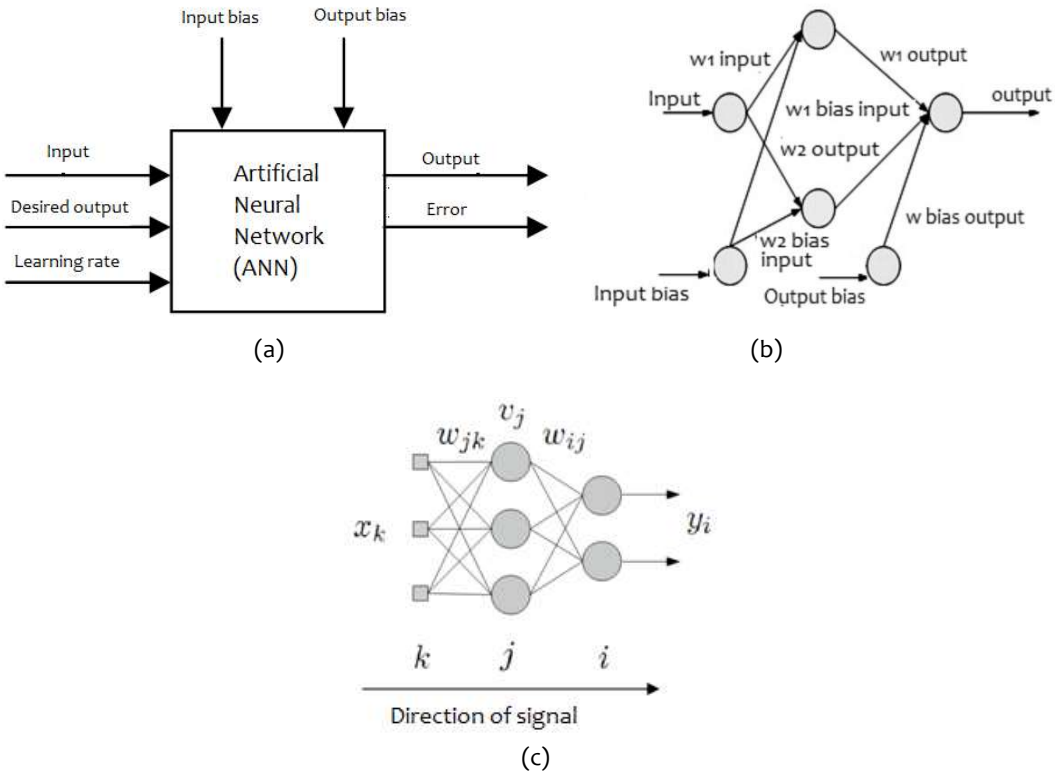

Figure 2.12: ANN Block: (a) It has 'input' as the sensor measurement and 'desired output' as the mean of past values; (b) Typical 2 layer feed forward network; (c) Feed forward process direction (Source: Auralius, 2009)

Table 2.1: Parameter Values for Neural Network Approach

Parameters	Values
w1 input	*-3.12317*
w2 input	*-7.0356*
w1 bias input	*-0.211902*
w2 bias input	*-0.211902*
w1 output	*-3.12317*
w2 output	*-7.0356*
w bias output	*11.846*
Input bias	*0*
Output bias	*0.2*

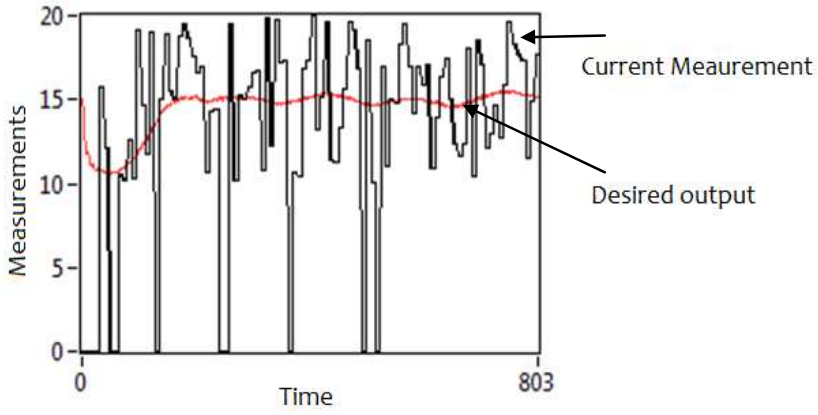

Figure 2.13: LabVIEW Implementation of ANN block: It shows the output of the selected sensor node. Lesser the gap between the two lines, higher is the trust rating

The main disadvantage of using neural network approach in WSN, is the complexity associated with it which makes it hard to implement. The space complexity of neural network in WSN is *O(RT)/cycle* where *R* is the total number of neurons and *T* is the maximum number of activation changes [Soliman et al, 2012].

ALGORITHM 2.1 NEURAL NETWORK ALGORITHM IN WSN

Input:
n: Number of sensor nodes
α_r: Learning rate = 0.5
Maintain history length
Sensor measurements ($x^{(1)}, x^{(2)}.....x^{(n)}$), where $x^{(i)} \epsilon R^{(n)}$
Output: Trust ratings for the nodes.

1 **Data** acquisition
2 Select a random node A and store the value as x
3 **for** $i \leftarrow$ 1 *to n* except A **do**
4 Pass the current value of *i* as the input and the mean of the past information of *i* as the desired output to the ANN block
5 Store output of the ANN block as = $x_{predicted}$
6 Calculate mean of $x_{predicted}$
7 Calculate errors (Error) as x-$x_{predicted}$
8 Compare error with the threshold (τ) set as = 0.05
9 **if** *Error* < τ **then**
10 *Node A has high trust rating*
11 **Else**
12 *Node A has low trust rating*

(g) Fuzzy Logic

IF-THEN rules are used to solve a problem in fuzzy logic. The logical steps followed in the fuzzy logic approach are fuzzy sets and the criterion of selection is predefined along with the input variables such that the fuzzy rules give output for the given input data. The results are regularly updated with the feedbacks.

Fuzzy logic security approach in WSN aims at finding the most trustworthy path using three principles namely [Kim and Seo, 2008]:
(1) *Fuzzy Matching*: calculates trust or mistrust degree.
(2) *Inference*: calculates rules and conclusion based on a matching degree.
(3) *Combination*: combines conclusion inferred by all fuzzy rules into the final one.

Initially, trust is calculated for each sensor node in the network followed by the summation of the trust values and combining them to select the best path as shown in Figure 2.14. Figure has a source (sensor node) having three paths to reach to the destination (base station). The most reputable path has nodes numbered 7, 8 and 9 which have trust equal to 1 as against 0.4 and 0.8 of the other paths.

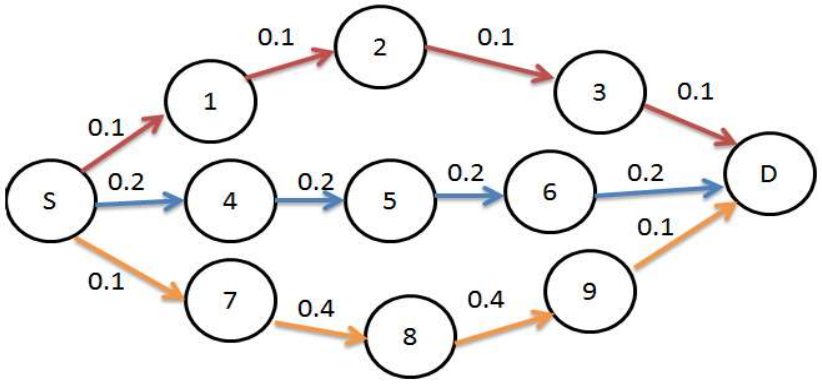

Figure 2.14: Fuzzy Logic Calculation: 7, 8 and 9 are the most trustworthy nodes

A TRIP mechanism was proposed which aimed at quickly and accurately distinguishing the fraudulent nodes spreading false information throughout the network [Gómez and Martínez, 2012]. RFSTrust based on fuzzy recommendation was proposed which comprises five types of fuzzy trust recommendation relationships according to the fuzzy relation theory and a mathematical description for MANETs. This approach not only provided a natural framework to deal with uncertainty but also the tolerance towards indistinct data inputs for the subjective tasks of trust evaluation, packet forwarding review and credibility adjustment [Lu et al, 2009].

The major problem in fuzzy logic approach is the memory overhead. Additionally, inefficiency occurs due to a lot of if-else rules.

(h) Swarm Intelligence

Swarm intelligence method also tries to achieve the most trustworthy path leading to the most reputable nodes in WSN [Mármol and Pérez, 2011]. Swarm intelligence approach uses ants as the inspiration for finding the most trustworthy nodes in the network. The main task of ants is to collectively search for food. They can adapt to the change in the environment, optimizing the path between the nest and the food source. This fact is due to stigmergy which

involves positive feedback, given by the continuous deposit of chemical substance, known as pheromone. Classic example of searching for food via shortest path can be seen in Figure 2.15.

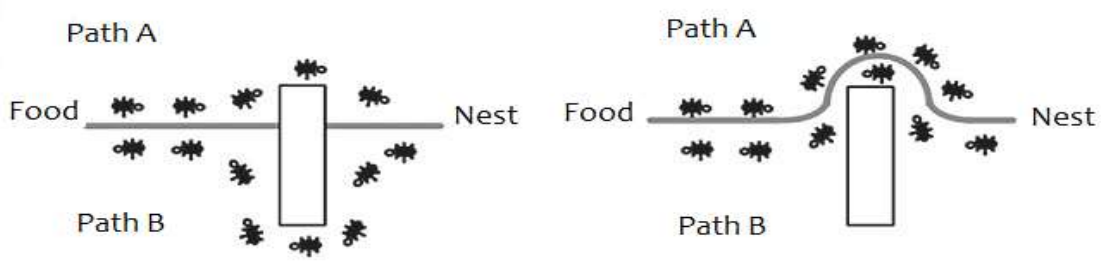

Figure 2.15: Ants Converging to Shortest Path (Source: Perretto and Lopes, 2005)

Ants converge to the shortest path by selecting the path which has a higher deposition of pheromone. Pheromone is the physical substance secreted by the ants. As seen from Figure 2.15, initially, half of the ants take one path and the other half take another path to reach to the food source. On their way, they deposit pheromone. The ants which first reach the food source are the first ones to come back again and give this information to other ants. Hence, the amount of deposition in the shorter path would be higher (See Eq. (2.20)) [Dorigo and Gambardella, 1997].

$$\Delta\tau_{ij}{}^{k} = \begin{cases} \frac{Q}{L_k} & if\ ant\ k\ used\ edge\ (i,j)\ in\ its\ tour \\ 0 & otherwise \end{cases} \quad (2.20)$$

where $\Delta\tau_{ij}$ is pheromone deposited from node i to node j and Q is a constant. The amount of pheromone deposition is inversely proportional to the length. Longer the path, pheromone deposition is less and shorter the path pheromone deposition is high. As time passes by, pheromone is updated using following equation) [Dorigo and Gambardella, 1997]:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k} \quad (2.21)$$

On a similar note, pheromone is evaporated with the evaporation rate of ρ as depicted in the equation [Dorigo and Gambardella, 1997]:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \quad (2.22)$$

By applying the above technique in WSN, computation of the most trustworthy path leads to the most reputable nodes in WSN. For implementation, 10 random active nodes were taken in a grid. Shortest path was computed having the most trustworthy nodes between the source and the destination as shown in Figure 2.16 where 1 is the source node and 10 is the destination node and the path containing nodes 4, 5, 6 and 9 is the most trustworthy within the network.

Overall the method was efficient with *90%* detection rate of fraudulent nodes. However, the worst case space complexity of maintaining pheromone table is $O(n^2)$ for a fully meshed network of n number of nodes [Bhaskaran et al, 2011].

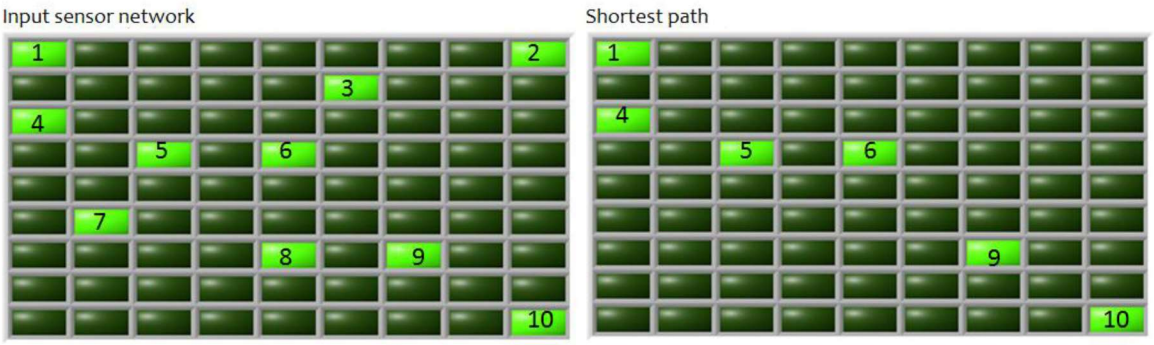

Figure 2.16: LabVIEW Implementation of Swarm intelligence Approach in WSN for Finding the Most Reputable Path

2.5 SUMMARY

A Wireless Sensor Network (WSN) is a network based on multiple low data rate and energy constrained sensor nodes. The purpose of the network is to monitor physical and environmental parameters such as temperature, pressure etc. The network consists of sensor nodes and gateway nodes, where the sensor nodes acquire the parameters and relay these cooperatively through other sensor nodes to the gateway nodes, which process and forward the data to the end-user. Security in a WSN is important since it not only provides reliability to the system but also helps in sustaining the network. Security threats can be introduced in a WSN through various means, where the ongoing data transmissions can be tampered or the nodes can be altered to behave in an unpredictable manner. The objective of the chapter was to present a review of various attacks and strategies used in overcoming the attacks. It discusses the security measures that can be implemented to detect the fraudulent nodes through trust and reputation models. Finally, comparative analysis is shown for the trust models discussed in the chapter.

Although research efforts have been made in regard to various approaches in trust and reputation model, there are still some challenges like high computational complexity pertaining to hardware implementation of WSN secure algorithms; memory; energy consumption and reliability of the overall WSN system, which need to be addressed. Since the sensors are characterized by constraints of energy, computation capability and memory, the design of the security services should be simple yet effective with regard to fraudulent node detection in WSN.

…