Intrusion Detection In Wireless Sensor Networks: A Survey

Alaa Eissa*, Hazem M. El Bakry, Mamoun H. Mamoun, and Samir Zied

Information Systems Dept., Faculty of Computer Science and Information Systems, Mansoura University
Mansoura, Egypt

Received: 03 December 2012; Accepted: 04 January 2013

Abstract

Since security threats to WSNs are increasingly being diversified and deliberate, prevention-based techniques alone can no longer provide WSNs with adequate security. However, detection-based techniques might be effective in collaboration with prevention-based techniques for securing WSNs. As a significant branch of detection-based techniques, the research of intrusion detection in wired networks and wireless ad hoc networks is already quite mature, but such solutions can be rarely applied to WSNs without any change, because WSNs are characterized by constrained resources, such as limited energy, weak computation capability, poor memory, short communication range, etc. The development of intrusion detection techniques suitable for WSNs is therefore regarded as an essential research area, which will enable WSNs to be much more secure and reliable. In this survey, a few of the key design principles relating to the development of intrusion detection techniques in WSNs are discussed and detection technique categories (statistical techniques, rule based, data mining, computational intelligence, game theory, graph based, and hybrid, etc.).

Keywords: Wireless Sensor Networks; Intrusion Detection.
©Martin Science Publishing. All Rights Reserved.

* Corresponding author. Tel.: 02-050-2223743
E-mail address: eissa16@yahoo.com (Alaa Eissa).
1. Introduction

The development of wireless sensor networks offers the promise of a flexible, low cost solution for monitoring critical infrastructure. For example, sensor networks have been proposed for applications such as traffic monitoring, building monitoring and battlefield surveillance [1]. In any application involving critical infrastructure, there is the potential risk of malicious attacks on this infrastructure, either for financial gain or as a terrorist act. The sensor network has a critical role to play in detecting these attacks, and thus can become a target for attack in its own right. A key attraction of sensor networks is their ease of installation and operation. However, security is one of the key challenges to creating a robust and reliable sensor network [2]. Currently, most research on security in sensor networks has focused on prevention techniques, such as secure routing protocols, cryptography and authentication techniques [3]. These security mechanisms are usually the first line of defense. However, experience with the Internet has shown that flaws in these protocols are continuously being found and exploited by attackers [4]. Sensor network protocols are faced with additional challenges due to complexities such as a wireless access medium, unpredictable node movement and unreliable node operation, and the first line of defense is broken through, compromised nodes could extract security sensitive information (e.g. secret key), leading to breaches of security. These challenges create considerable potential to exploit weaknesses in the network. Consequently, we cannot rely on intrusion prevention techniques alone. In practice, Intrusion Detection Systems (IDSs) are needed to detect both known security exploits and even novel attacks that have yet to be experienced.

This concept was originally proposed by Anderson (1980) two decades ago in a report “Computer Security Threat Monitoring and Surveillance”[5].

Intrusion detection is the problem of identifying misuse of computer systems and networks [6]. Most IDSs apply signature-based techniques. In general, signature based techniques test for features of known network attacks. This raises the question of how to learn these features for known attacks, and how to detect new attacks. It is difficult to use supervised learning in this context, since labeled training data is expensive to produce. More importantly, it is difficult to detect new types of attacks whose signatures may differ from those in its signature set. This has motivated research into unsupervised learning techniques, which do not require labeled data and are able to detect previously ‘unseen’ attacks.

Instead of learning the signature of attack traffic, unsupervised anomaly detection techniques focus on learning the signature of normal traffic. Unsupervised learning techniques do not require the data to be labeled, nor do they require the data to be purely of one type, i.e., normal or attack traffic. This is a significant benefit over the supervised learning approach.

In this survey paper, all detection schemes are divided into two types of detection method: prior-knowledge based, or prior-knowledge free. The prior-knowledge-based detection schemes are better suited to the applications which are biased to detection speed; the prior-knowledge free schemes, on the contrary, are capable of providing applications with stronger detection generality.

In this survey paper, recently proposed detection schemes in WSN are introduced. Because the architecture of a WSN is strongly related to many aspects of designing a suited scheme, these
detection schemes are classified as hierarchical and flat (homogeneous) according to their architectures. In a hierarchical WSN, all sensor nodes are grouped or clustered, where only a single node is elected as the cluster head (possibly equipped with stronger capacity) to conduct the organizational functions within its group or cluster. On the contrary, all sensor nodes equally contribute to any team-functions and participate in internal protocols (e.g. routing protocols) in a flat WSN. For each of the architectures, a number of typical examples are given in terms of the technique category that they belong to.

2. Wireless Sensor Networks

Wireless sensor networks are composed of a large number of tiny nodes that are used to measure some physical or environmental aspect of the hostile environment such as temperature, sound, vibration or motion. These nodes are resource-constrained units that communicate via a wireless medium and forward sensed data to the gateway node. The gateway node, so called base station, is the only connection with the other world apart from the network itself. Wireless sensor networks serve as a bridge from the physical world to the computer system by providing measurements of physical properties of the real world. Although wireless sensor networks were originally designed for the purpose of military application, nowadays their field of application is much wider and they are being used in civilian and industrial areas as well (healthcare applications, traffic control, home automation, industrial process monitoring or wildlife monitoring).

Wireless sensor networks (WSNs) are homogeneous distributed ad-hoc networks. However, there are several important differences from the classical ad-hoc networks.

A WSN is highly distributed and its nodes are independent, self-configurable, capable of establishing the routing subsystem without any prior given infrastructure and able to cooperate with each other.

A sensor network topology may be flat or hierarchical. In case of a hierarchical topology, the network is divided into clusters. Each cluster has a cluster head node which is usually a more powerful node. Cluster heads usually take some responsibilities for network maintenance as for example intrusion detection systems can be installed on these nodes. However, they become a single point of failure. In order to prevent this problem, flat sensor networks are deployed.

2.1 Differences between Ad Hoc and Sensor Networks

Wireless sensor networks (WSNs) are homogeneous distributed ad-hoc networks, but there are some differences between them as below:

- the number of nodes in a sensor network can be much larger than in ad hoc networks;
- the traffic flows in sensor networks are from the sensors to a centralized base station (or vice versa), rather than irregular one-to-one communication between nodes in ad hoc networks;
- sensor nodes can be exposed to hostile environments, and can have a higher likelihood of failure;
• many sensor network applications have more severe resource constraints, which makes communication a more expensive task in contrast to computation.
• Sensor nodes are prone to failures.
• The topology of a sensor network changes very frequently.
• Sensor nodes are limited in power, computational capacities, and memory.
• Sensor nodes may not have global identification (ID) because of the large amount of overhead and large number of sensors.

2.2 Wireless Sensor Networks Applications

WSNs encourage several novel and existing applications such as environment monitoring, infrastructure management, public safety, medical and health care, home and office security, transportation, and military applications. The sensor nodes in WSN are very compact and autonomous. They have limited computation and communication capabilities, and limited power supply. The following are some of its applications:

**General engineering:** It can be used for automotive driving, fingertip accelerometer virtual keyboards, sensing and maintenance in industrial plants, aircraft drag reduction, smart office space management, tracking of goods in retail stores, tracking of containers and boxes in shipping companies, social studies on human behavior, commercial and residential security.

**Agriculture and environmental monitoring:** It can used in crop and livestock management and precision control, planetary exploration, geophysical monitoring, monitoring of freshwater quality, habitat monitoring, disaster detection, can contaminate transport.

**Civil engineering:** It can be used for monitoring of structures, urban planning, and disaster recovery.

**Military applications:** It can be used for asset monitoring and management, surveillance and battle-space monitoring, urban warfare, protection to buildings.

**Health Monitoring and Surgery:** It can be applied in medical sensing, and micro-surgery.

2.3 Software

Wireless sensor networks run specialized operating systems for which applications can be written. Contiki, Mantis, BTNUT, SOS and Nano-RK are some of the operating systems that allow for writing application programmes in the C language [7]. The most commonly used operating system in wireless sensor networks is TinyOS [8]. The TinyOS operating system is an embedded, open-source, component-based operating system for wireless sensor networks.

It is written in a dialect of the C language – the NesC programming language. NesC provides an event-based programming model where programmes are composed of event handlers and tasks. The TinyOS project [9] was rooted at the University of California, Berkeley.
Nowadays, an international consortium, the TinyOS Alliance, supports the academic and industrial community of TinyOS developers and contributors. The current version of TinyOS is 2.1.0.

For the purpose of the WSN development and simulating TinyOS networks, a discrete event simulator, called TOSSIM, was created at the University of California, Berkeley. TOSSIM scales to thousands of motes and compiles directly from the TinyOS source code. The applications written in NesC for TinyOS are built into the compilation. TOSSIM is run on personal computers. Power TOSSIM was created originally at the Harvard University [10]. It is used to simulate power consumption of each node in the simulation.

There comes the Dissemination protocol (DIP) and the Collection tree protocol (CTP) as a part of the TinyOS distribution. The first one is used for establishing eventual consistency on a variable shared by all the nodes in the network. The CTP is used to collect data at the base station (root of the tree) from any node. If all the nodes send data periodically, the CTP creates heavy traffic as every internal node has to forward each packet it receives up the tree. A data aggregation protocol can be used on top of or instead of the CTP in order to decrease the number of sent packets and hence prolong the battery lifetime. However, no data aggregation protocol implementation can be found in the current TinyOS distribution.

2.4 Attacks

There are different types of attackers and many types of attacks they are able to perform. We focus on active external and internal attackers (insiders) as they are able to run more convenient attacks and the intrusion detection system is deployed to defend against these attacks. An IDS is used to differentiate among trusted nodes and attackers as they might form a legitimate part of the network. Symptoms of attacks are very important for the study of intrusion detection systems for WSNs. An IDS may determine an internal attacker in the network based on the pre-defined symptoms of known attacks.

- **Jamming:** Jamming is interfering with the radio frequency used by nodes for their communication. It is performed by deliberate transmission of radio signals. It is used to conduct a denial of service attack as nodes cannot communicate at all while a jamming attack is ongoing. Nodes consider their communication media to be in use, or they believe some node is transmitting and so they remain in a receiving mode the whole time. A jamming attack is caused by a device which is usually referred to as a jammer. It might be a sensor node or some other device able to interfere with the radio frequency of the wireless sensor network. We may distinguish among various types of jamming attacks and jammers. Among the ones that may be the most effective are constant, deceptive, random and reactive jammers [11]. A constant jammer continually emits a radio signal without respecting any medium access protocol. In this case, other nodes never find the medium idle. A deceptive jammer uniformly injects regular packets without any gap so other nodes stay in the receiving mode most of the time. A random jammer emits or is asleep to reduce battery consumption. It switches these two states in a random manner. Random jamming may be implemented by both constant and deceptive jammers. A reactive jammer emits only when there is communication on the medium. It is harder to detect than the previous techniques and again it may be implemented by both constant and
deceptive jammers. Several symptoms might be used to identify jamming. A short overview of these techniques is given in the next paragraph. However, they are not always suitable for every type of jammer. More details on identifying jamming attacks can be found in [11, 12]. A received signal strength indicator might be used to detect jamming because the distribution of signal strength is affected by the jammer being active. The basic approach where an average signal strength value is compared to the threshold calculated from the ambient noise level is rather limited. A more convenient technique uses signal strength spectral discrimination. Unfortunately, statistics based on signal strength magnitude are only suitable for identifying constant and deceptive jammers. Signal strength distribution is not affected in such a manner by reactive and random jammers as they alter the sleeping and emitting states which simulate the behaviour of a normal node. A carrier sensing protocol is used to tell whether a node is allowed to transmit over the media. If a node never finds the media idle, it cannot transmit and may assume that the network is being jammed. A carrier sensing time metric is suitable only when the media access protocol of the sensor network tells whether a channel is idle upon a fixed threshold. The packet delivery ratio drops suddenly to almost zero when a node is jammed. In the case of congestion, it does not drop so suddenly and even though the delivery ratio is very low in a congested network, it’s not as close to zero as it is in the case of a jammed network. The delivery ratio may tell congestion from jamming and is useful for revealing all the jamming scenarios. However, it is prone to be inaccurate due to battery failure or other network dynamics which may suddenly lead to packet delivery inability. The delivery ratio can be estimated both as received acknowledgements per sent packets on the side of a sender or as the number of packets which passed the cyclic redundancy check per received packets on the receiver’s side. In order to eliminate falsely announced jammings, a combination of the methods described above can be used. Energy exhaustion may lead to a false alarm in the case of the packet delivery ratio symptom. If the packet delivery ratio method is combined with the signal strength consistency check, false positives are reduced. Very low packet delivery ratio and low signal strength imply that a node’s neighbour is malfunctioning due to battery depletion. However, when there is a packet delivery ratio close to zero and on the other hand signal strength is high, a jamming attack is ongoing in the wireless sensor network with highest probability. The last method, actually the combination of two of them, should provide the most reliable way to reveal jamming. It is even suitable for all of the jamming scenarios defined in this section. Unfortunately, the delivery ratio is meant to be compared with a predefined threshold (a number close to zero). From this point of view, the neighbour-based technique would not be convenient to use (it compares the delivery ratio values among the nodes in the neighbourhood and that is not intended by the packet delivery ratio method). The neighbour-based detection scheme is determined to serve as a global agent, it monitors the behaviour of its neighbourhood (not the node running the IDS agent itself). That is why detection based on carrier sensing time cannot also be implemented. Carrier sensing time is a statistic appropriate for monitoring by local agents. We exclude monitoring of the distribution of signal strength because of the same reason. A packet sending rate is easy to monitor and simple assumption would suggest that a node which sends an abnormal amount of packets is one that produces jamming [13]. We can definitely detect deceptive jammers by monitoring the packet sending rates of the
neighbouring nodes. However, it depends on the jammer’s implementation and the network protocol used whether detection of random and reactive jammers is possible.

- **Hello flood attack**: Routing protocols usually prefer the shortest or the most reliable path to the base station. Hello packets (sometimes also referred to as advertisements or beacons) are sent out by a new node in the network in order to inform other nodes that they can possibly route their messages via the new node. If a malicious node possesses a long-range antenna, it can broadcast hello packets claiming good connection to the base station. These hello packets will be received by the nodes which cannot reach the adversary back as they do not have such a strong antenna. The affected part of the network becomes paralysed as no messages are routed out of it. Nodes which are close to the attacker may notice that received signal strength indicator values of messages delivered from the attacker are abnormally high. This detection method can be implemented using the neighbour-based detection technique. Nodes will keep statistics of average signal strengths of received messages from their neighbours and compare them with the averaged value of these statistics. A node having its average signal strength significantly higher will be announced as the hello flood attacker.

- **Selective forwarding**: A compromised node (an attacker) drops packets instead of forwarding them further in a multi-hop routing system in case of a selective forwarding attack. An attacker may drop all of the incoming packets (also denoted as black hole attack) or selectively drop only specific packets (coming from a specific source, having a certain destination, containing certain payload data, etc.). In the second case, it is harder to detect and several statistics have to be stored by an IDS to check. A high packet dropping rate can be used to identify nodes that conduct selective forwarding [14]. It is estimated as the ratio between the number of sent packets and the number of received packets over a period of time. This ratio may be kept as an overall number of transmitted packets or just for packets coming from a specific source, having a certain destination, etc. The approach from [15] does not only take into consideration packets being dropped by a malicious node. At the same time, the intrusion detection system checks whether a packet is sent to a legitimate neighbour of the monitored node. Otherwise, it assumes that the packet was forwarded by a malicious node to a mismatched location on purpose. In order to make this technique work, the form of a hello packet (used to establish routing tables when a new node is added to the network) has to be changed so each node is able to derive its 2-hop neighbours from it. This requirement makes this mechanism harder to implement than in the case of packet dropping rate monitoring. The protocol for finding out node’s 2-hop neighbours would mean another communicational and computational overhead for tiny nodes. The described neighbour-based intrusion detection is supposed to be a lightweight agent and is not considered to provide such a functionality. On the other hand, the packet dropping rate is an appropriate metric that can be used to monitor the network behaviour of a node’s neighbourhood in the neighbour-based IDS. Nodes that are in spatial correlation (according to the insider attacker detection) should be dealing with a similar traffic load and network dynamics. The packet dropping rate should be similar as well and if a node drops packets in extreme numbers, it is probably malicious.
- **Sinkhole attack:** A sinkhole node is one where most of the traffic is reflected to. According to a routing protocol, it is the one claiming extremely good connection to the base station in its neighbourhood. An attacker tries to create a sinkhole node from the one that is captured by them. Afterwards, more serious attacks can be run using this node. Depending on which routing algorithm is used, an attacker tries to fake routing protocol’s metrics which define the best path to the gateway so most of its neighbours, preferably all, set the captured node as their parent node. An IDS may identify nodes which claim a suspiciously high-quality connection to the gateway and are the only such nodes in their neighbourhood. This technique cannot identify a sinkhole which is started at the beginning of a network’s existence because the sinkhole’s neighbours will claim good connection via the sinkhole node as well. The extreme difference in the apparent quality of the connection will not be noticeable. If the packet receiving rate of some node is extremely high, it might be suspected of being an attacker [13]. However, this symptom does not differentiate natural sinkholes which may exist depending on network topology. Furthermore, the packet receiving rate of surrounding nodes will become high as well because they will gain very good connection to the base station via the sinkhole node. This approach is considered very limited and not suitable for intrusion detection techniques. Although a malicious node may claim that its connection to the gateway is better than it actually is, nodes in its neighbourhood do not have to change their parents straight away (this depends on a routing protocol, e.g. the CTP in TinyOS is designed this way). An attacker has to downgrade the quality of other nodes’ connections in this case. A malicious node may spoof fake root update packets impersonating its neighbours. The authors of [16] suggest that this form of sinkhole attack may be revealed by an IDS which observes whether senders of root update packets are in the neighbourhood of the node running the IDS. If not or if they are even sent by the node running the IDS itself (possible when the packet’s header is altered), some node is running the sinkhole attack in the network. This technique effectively reveals attackers that try to downgrade the quality of other nodes’ connections. If some node finds out that another node spoofs packets, it should alert other nodes about this immediately. It does not matter what the common behaviour of the neighbourhood is. From this point of view, this technique is not suitable for the neighbour-based intrusion detection. Unfortunately, no other technique that could be used is known.

- **Sybil attack:** A sybil node is one that is claiming multiple identities. An attacker that owns these identities may take advantage in voting protocols or create routing paths for their own benefits. Communication with sybil nodes may be direct or indirect. In direct communication, the compromised node communicates with the other nodes in charge of all of its identities. In indirect communication, the sybil node claims that it communicates with nodes that actually do not exist. Sybil identities might be fabricated or stolen. A sybil attack can be performed simultaneously or non-simultaneously depending on whether the sybil node uses its identities at once or over time. A sybil node may be revealed by location testing which is based on the principle that some number of cooperating nodes are able to estimate another node’s location based on some measurements. If they find out that two nodes are located at the same position, a sybil attack is most likely being conducted by an attacker. The technique using the received signal strength indicator is described in [17, 18]. Three cooperating nodes measure the signal strength upon receiving
a message. After exchanging obtained values, the ratio is measured from them and stored in a database of neighbours. A unique ratio value determines the unique position of a node. A similar approach is published in [19] too. Location testing is based on measuring the time difference of arrival of packets among cooperating nodes instead of the received signal strength. Both location testing methods need a group of nodes to exchange some information. This information is used for a simple computation whose result is a determination of the location of the node which they exchanged the information about. Each node has a table of such locations of the nodes in its neighbourhood. Both techniques do not look for a property of network behaviour which should not vary noticeably for a node in some group of neighbouring nodes. This is the reason why it cannot be used for the studied neighbour-based IDS. Unfortunately, we do not know any other technique that could be implemented.

- **Packet alteration:** An attacker might be interested in spoofing or altering packets of other nodes in order to misuse a routing algorithm, have an advantage in voting protocols or change measured values sent by sensor nodes to the base station. Monitoring of spoofed packets can be provided in a similar way as dedicated to sinkhole attacks and revealing fake beacons. The basic assumption is that a node should be able to hear only packets that have originated in its neighbourhood. If they have originated elsewhere, they are spoofed packets. In order to detect alteration of data, an IDS has to store overheard packets in the buffer, wait until appropriate nodes forward them and compare whether the payloads are the same for the forwarded packets and the packets stored in the buffer. The presence of an attacker that alters or spoofs packets is noticed immediately. There is no need to use techniques such as the neighbour-based detection technique to find out if it is common to deliberately alter packets in the node’s neighbourhood. Deliberate alteration should always be considered as an attack.

- **Fabricated information attack:** A malicious node might send fallacious measured values which would not reflect the reality of its surroundings to the base station. There is an assumption that these values provided by nodes from a close neighbourhood should usually vary just slightly. When values in node’s surroundings are compared and an IDS finds out that the node provides extremely different results, it is suspected of being captured by an attacker. Simulations of both insider attacker and group-based intrusion detection schemes were run to monitor usage of this statistic. The description and results of the simulations can be found in [13, 12] respectively.

- **Neighbour-based detection of the attacks:** Several symptoms that can be used by the detection component of an IDS to identify jamming, hello flood, selective forwarding, sinkhole, sybil, packet alteration and fabricated information attacks were presented in the previous sections. Unfortunately, not all of them are suitable for the neighbour-based detection technique. The reasons for this assumption were given above. In order to reveal the jamming attack, the packet sending rate is an appropriate statistic for the neighbour-based IDS discussed in this work. The hello flood attack can be detected by monitoring the received signal strength indicator readings of nodes in the neighbourhood of the node running the IDS agent. The packet dropping rate may identify a malicious adversary that selectively drops packets instead of forwarding them according to the routing protocol.
Regrettably, no appropriate statistics for neighbour-based detection were found for disclosing the sinkhole or sybil attack. Although the packet alteration attack is easy to recognize, the means by which this is done is not suitable for this work. Finally, the attack based on providing fallacious measured values has already been implemented by the author of [13]. The implementation of the appropriate statistics depicted here is described in detail in the next chapter where our neighbour-based IDS is presented.

3. Intrusion Detection

This concept was originally proposed by Anderson (1980) two decades ago in a report “Computer Security Threat Monitoring and Surveillance”. Intrusion detection is defined as the process of monitoring the events occurring in a computer system or network and analyzing them for any signs of possible incidents, which are violations or imminent threats of violation of computer policies, acceptable use policies, or standard practices [21]. However, anomaly detection [22], a branch of intrusion detection, is best suited to WSN because its methodology is flexible and resource friendly in general. Anomaly detection is defined as the process of comparing definitions of activity that is considered normal against observed events in order to identify significant deviations. Moreover, an anomaly in a dataset is defined as an observation that appears to be inconsistent with the remainder of the dataset [23].

Intrusion detection is the process of discovering, analyzing, and reporting unauthorized or damaging network or computer activities. It discovers violations of confidentiality, integrity, and availability of information and resources. Intrusion detection demands as much information as the computing resources can possibly collect and store. It requires experienced personnel who can interpret network traffic and computer processes. It needs constant improvement of technologies and processes to match pace of Internet innovation. Intrusion can provide digital forensic data to support post-compromise law enforcement actions. It can identify host and network miss-configurations, improve management and customer understanding of the Internet's inherent hostility. Also, it is able to learn how hosts and networks operate at the operating system and protocol levels.

There are not many papers working on general intrusion detection techniques for wireless sensor networks, relatively more works on intrusion detection for specific kind of attacks, like wormhole attacks, routing holes, or to particular operations, like routing, localization, etc. One paper proposes an anomaly approach based on self-organized criticality (SOC) and Hidden Markov models to detect data inconsistencies [24]. This approach is developed based on the structure of naturally occurring events. With the acquired knowledge distilled from the self-organized criticality aspect of the deployment region, it applies a hidden Markov model. It lets sensor networks adapt to the norm of the dynamics in its natural surrounding so that any unusual activities can be singled out.

Another paper formulates the attack-defense problem by game theory and use Markov Decision Process to predict the most vulnerable sensor nodes [25]. It formulates attack-defense problem as a two-player, nonzero-sum, non-cooperative game between an attacker and a sensor network. It shows that this game achieves Nash equilibrium and thus leading to a defense strategy for the network. Then, it uses Markov Decision Process to predict the most vulnerable sensor
node. Finally, it uses an intuitive metric (node’s traffic) and protects the node with the highest value of this metric.

Indeed, wireless sensor networks are susceptible to many forms of intrusion. In wired networks, traffic and computation are typically monitored and analyzed for anomalies at various concentration points. This is often expensive in terms of network’s memory and energy consumption, as well as its inherently limited bandwidth. Wireless sensor networks require a solution that is fully distributed and inexpensive in terms of communication, energy, and memory requirements. In order to look for anomalies, applications and typical threat models must be understood. It is particularly important for researchers and practitioners to understand how cooperating adversaries might attack the system. The use of secure groups may be a promising approach for decentralized intrusion detection [26].

Apart from those more general approaches, some papers provide intrusion detection techniques for particular operations. In [27], it describes a distributed algorithm, BOUNDHOLE, to build routes around the routing holes, which are connected regions of the network with boundaries consisting of all the stuck nodes. It shows that hole-surrounding routes can be used in geographic routing, path migration, information storage mechanisms and identification of regions of interest. The paper [28] proposes a general scheme to detect localization anomalies that are caused by adversaries. It formulates the problem as an anomaly intrusion detection problem, and proposes a number of ways to detect localization anomalies. In [29], it describes an intrusion detection technique that uses information about both the network topology and the positioning of sensors to determine what can be considered malicious in a particular place of the network. The technique relies on an algorithm that automatically generates the appropriate sensor signatures. It applies that approach to an intra-domain distance-vector protocol and reports the results of its evaluation. Moreover, there are some papers applying fault-tolerant technologies in providing network security. In [30], secure multi-path routing to multiple destination base stations is designed to provide intrusion tolerance against isolation of a base station. Also, anti-traffic analysis strategies are proposed to help disguise the location of the base station from eavesdroppers. The paper in [31] targets the identification of faulty sensors and detection of the reach of events in sensor networks with faulty sensors. It proposed two algorithms for faulty sensor identification and fault-tolerant event boundary detection. These algorithms are localized and scalable for WSNs.

Intrusion detection paradigms include the following:

- Anomaly Detection - the AI approach
- Misuse Detection - simple and easy
- Burglar Alarms - policy based detection
- Honey Pots - lure the hackers in
- Hybrids - a bit of this and that

Among all, anomaly detection and misuse detection are the most common traditional intrusion detection techniques. The following are the details of the two techniques:

1. **Anomaly detection**
   - Establish a profile of the subject’s normal activities
Consider a deviation of a subject’s observed activities from its norm profile
A subject – a user, file privileged program, host machine, network
Disadvantages: May treat all anomalies as attacks, false alarms are anticipated

2. Misuse detection (pattern recognition)
Identify and store signature patterns of known attacks
Match observed behavior with known patterns of attack signatures
Attack signatures – e.g. Strings, even sequences, activity graphs, attack scenarios
(event sequences, preconditions, target compromised states)
Disadvantages: cannot detect novel or unusual attacks whose signatures are unknown; have to update the attack signature patterns constantly

Intrusion detection model consists of six main components [32]:
2. Objects: Resources managed by the system-files, commands, devices, etc.
3. Audit records: Generated by target systems in responses to actions performed or attempted by subjects on objects-user login, command execution, file access, etc.
4. Profiles: Structures that characterize the behavior of subjects with respect to objects in terms of statistical metrics and models of observed activity. Profiles are automatically generated and initialized from templates.
5. Anomaly records: Generated when abnormal behavior is detected.
6. Activity rules: Actions taken when some condition is satisfied, which update profiles, detect abnormal behavior, relate anomalies to suspected intrusions, and produce reports.

Observed behavior is characterized in terms of a statistical metric and model. A metric is a random variable \( x \) presenting a quantitative measure accumulated over a period. The statistical models may be an operational model, mean and standard deviation model, multivariate model, Markov process model, time series model, etc [32].

The paper [33] analysis the characteristics of the activity graphs, detects and reports violations of the stated policy. It uses a hierarchical reduction scheme for the graph construction, which allows it to scale to large networks. An early prototype of GrIDS has successfully detected a worm attack.

The paper [34] shows that the architecture for better intrusion detection in wireless ad hoc networks should be distributed and cooperative. A statistical anomaly detection approach should be used. The trace analysis and anomaly detection should be done locally in each node and possibly through cooperation with all nodes in the network. Further, intrusion detection should take place in all networking layers in an integrated cross layer manner.

In [35], it makes use of geometric random graphs induced by the communication range constraints of nodes and presents the necessary and sufficient conditions for detection and defending against wormholes. Their defense mechanism is based on local broadcast keys.
The paper [36], we introduce a secure routing protocol called JANUS, which focuses on the establishment of secure routes between the base station and mobile devices, and the secure routing of the data.

In [37], it presents the rushing attack as a new attack which results in denial-of-service when used against all previous on-demand ad hoc network routing protocols. It then develops Rushing Attack Prevention (RAP), a generic defense against the rushing attack for on-demand protocols. Also, the same authors present a generic mechanism, called packet leashes, for detecting and thus defending against wormhole attacks in [38].

The paper [39] considers ad hoc networks with multiple, mobile colluding intruders. It investigates the placement of the intrusion detection modules for misuse intrusion detection. It mathematically formulates different detection objectives, and shows that computing the optimal solution is NP-hard in each case.

Another paper [40] written by the same authors consider the signature detection technique and investigate the ability of various routing protocols to facilitate intrusion detection when the attack signatures are completely known. They show that reactive ad-hoc routing protocols suffer from a serious problem due to which it might be difficult to detect intrusions even in the absence of mobility. Mobility makes the problem of detecting intruders harder.

3.1 Requirements for Intrusion Detection in WSN

we elaborate on the requirements that an IDS system for sensor networks should satisfy. To do so, one has to look at the specific characteristics of these networks. Each sensor node has limited communication and computational resources and a short radio range. Furthermore, each node is a weak unit that can be easily compromised by an adversary [20], who can then load malicious software to launch an insider attack.

In this context, a distributed architecture, based on node cooperation is a desirable solution. In particular, we require that an IDS system for sensor networks must satisfy the following properties:

1. Localize auditing. An IDS for sensor networks must work with localized and partial audit data. In sensor networks there are no centralized points (apart from the base station) that can collect global audit data, so this approach fits the sensor network paradigm.
2. Minimize resources. An IDS for sensor networks should utilize a small amount of resources. The wireless network does not have stable connections, and physical resources of network and devices, such as bandwidth and power, are limited. Disconnection can happen at any time. In addition, the communication between nodes for intrusion detection purposes should not take too much of the available bandwidth.
3. Trust no node. An IDS cannot assume any single node is secure. Unlike wired networks, sensor nodes can be very easily compromised. Therefore, in cooperative algorithms, the IDS must assume that no node can be fully trusted.
4. Be truly distributed. That means data collection and analysis is performed on a number of locations. The distributed approach also applies to execution of the detection algorithm and alert correlation.
5. Be secure. An IDS should be able to withstand a hostile attack against itself. Compromising a monitoring node and controlling the behavior of the embedded IDS agent should not enable an adversary to revoke a legitimate node from the network, or keep another intruder node undetected.

3.2 IDS Classification

The descriptions of different kinds of IDS classification based on [41] are summarized as
There are generally two types of intrusion detection – anomaly detection and signature (sometimes denoted as misuse) detection. A difference can be seen in the way they discover malicious nodes. Any unusual behavioural deviation in the network opposed to its normal behaviour is announced as an anomaly in case of the anomaly detection. An IDS of such a type has to be able to learn about the normal behaviour of the network. There is usually a start-up phase, often denoted as a training phase, of IDS for this purpose and the IDS only gathers information about normal flow for some period of time. It has to be ensured that no intruders exist in the network during this phase which might be hard to achieve. “Signature based detection techniques match the known attack profiles with suspicious behaviours” as stated in [42]. For this purpose, attack footprints have to be defined for each type of the attack that should be recognized by the IDS. Both anomaly and signature based IDSs have their pros and cons. A signature based detection is very effective in revealing known attacks whose patterns are defined in the IDS. However, it fails completely to uncover unknown attacks. They can be recognized by an anomaly detection though. Unfortunately, such an IDS requires training to learn what a normal traffic flow looks like and if network’s dynamics have changed, the IDS has to be re-trained. Employing both detection techniques should provide an effective detection mechanism for a sensor network. Additionally, a specification detection is sometimes introduced as the third type of IDSs. It is very similar to an anomaly detection, however, the set of rules is defined a priori and so no training phase is involved. This work deals with the neighbour-based intrusion detection which is a specific type of the anomaly detection.

From another point of view, intrusion detection systems might be classified depending on their collaboration abilities into collaborative and non-collaborative (also referred to as distributed and stand-alone respectively). In case of a distributed ID, a false information filtering system should be implemented as the IDS may collaborate with IDSs running on nodes captured by an adversary. Reputation schemes are often employed for this purpose. Stand-alone detection systems do not suffer with these problems. On the other hand, there might not be enough information gathered locally to decide some types of attacks. We designed and implemented both collaborative and non-collaborative modifications of an ID for flat wireless sensor networks.

Collaborative IDSs are categorized as peer-to-peer and hierarchical. Peer-to-peer IDSs create high communication overhead. Hence, information should be distributed just in a small neighbourhood around a node. Hierarchical IDSs assume existence of nodes which take responsibilities of cluster heads which brings the problem of a single point of failure.

An attacker who captures just a few nodes which are actually the cluster heads paralyses functioning of an IDS for the whole network.
Moreover, there is a question on which node an IDS should be actively running at some point in time. The authors of [43] suggest to use the method of spontaneous watchdogs. Then, an IDS is installed on every node. When there is communication on a medium, one of the possible watchdogs for the communication is chosen to be active. A set of possible watchdogs is composed of all the nodes which are able to hear the communication. The selection of the active watchdog is implemented by a random choice in [43]. However, other implementations let possible watchdogs monitor the network in turn.

3.3 Wireless Network’s IDS Architectures

Furthermore, Brutch and Ko divide wireless network IDS architectures into three categories. This classification can be adjusted to the needs of WSN IDS.

1. Stand-alone: In this category each node operates as an independent IDS and is responsible for detecting attacks only for itself. Such an IDS does not share any information or cooperate with other systems. This architecture implies that all the nodes of the network are capable of running an IDS.
2. Distributed and Cooperative: Here, all nodes still are running their own IDS, but the IDS cooperate in order to create a global intrusion detection mechanism.
3. Hierarchical: In this case the network is divided into clusters with cluster-head nodes. These nodes are responsible for routing within the cluster and accept all the accusation messages from the other cluster members indicating something malicious. Additionally, the cluster-head nodes may also detect attacks against the other cluster-head nodes of the network, as they constitute the backbone of the routing infrastructure.

3.4 Detection Techniques for WSN

A. Statistical-Based Approaches: Statistical-based approaches are the earliest approaches to deal with the problem of intrusion detection. The statistical detection techniques are essentially model-based techniques. They assume or estimate a statistical (probability distribution) model which captures the distribution of the data and evaluate data instances with respect to how well they fit the model. A data instance is declared as an anomaly if the probability of the data instance to be generated by this model is very low. The modeling techniques can work in an unsupervised mode, where a statistical model can be determined if it fits majority of the observations while small amounts of anomalies exist in the data. The statistical-based approaches are categorized into parametric and non-parametric based on how the probability distribution model is built.

1. Parametric-Based Approaches: Parametric techniques assume availability of the knowledge about underlying data distribution, i.e., the data is generated from a known distribution. It then estimates the distribution parameters from the given data. Based on type of distribution assumed, these techniques are further categorized into Gaussian-based models and non-Gaussian-based models. In Gaussian models, the data is assumed to be normally distributed.

- Gaussian-based models. Wu et al. [44] present two local techniques for identification of outlying sensors as well as identification of event boundary in
sensor networks. These techniques employ the spatial correlation of the readings existing among neighboring sensor nodes to distinguish between outlying sensors and event boundary. In the technique for identifying outlying sensors, each node computes the difference between its own reading and the median reading from its neighboring readings. Then it standardizes all differences from its neighborhood. A node is considered as an outlying node if the absolute value of its reading’s deviation degree is sufficiently larger than a pre-selected threshold. The technique of event boundary detection is based on the previous results of outlying sensor identification and determines a node as an event node if the absolute value of the node’s deviation degree in one geographical region is much larger than that in another region. Accuracy of these intrusion detection techniques is not relatively high due to the fact that they ignore the temporal correlation of sensor readings. Bettencourt et al. [45] present a local intrusion detection technique to identify errors and detect events in ecological applications of WSNs. This technique can distinguish between erroneous measurements and events by using the spatio-temporal correlations of sensor data. Each node learns the statistical distribution of difference between its own measurements and each of its neighboring nodes, as well as between its current and previous measurements. The procedure can be based on a priori knowledge of data distribution or a non-parametric density estimation. A measurement is identified as anomalous if its value in the statistical significance test is less than a user-specified threshold. The detected anomalous measurement may be considered as event if it is likely to be temporally different from its previous measurements but spatially correlated. The drawback of this technique is that it relies on the choice of the appropriate values of the threshold. Hida et al. [46] design a local technique to make simple aggregation operations, such as MAX or AVG, more reliable under presence of faulty sensor readings and failed nodes. This technique relies on the spatio-temporal correlations of sensor data and uses two statistical tests to locally detect intrusion. Each incoming sensor value is compared against the current value and the previous values of all nodes in the neighborhood. If the incoming value passes the two statistical tests, it is allowed to be aggregated as usual; otherwise (if the incoming value is outside of 2.5 standard deviations of the mean) it is declared as an outlier and will be eliminated from the analysis. Drawbacks of this technique include the fact that it only deals with one-dimensional outlier data and too much memory is required for a node to store historical values of all its neighboring nodes.

- Non-Gaussian-based models. Jun et al. [47] present a statistical-based technique, which uses a symmetric $\alpha$-stable (S$\alpha$S) distribution to model intrusions being in form of impulsive noise. The technique utilizes the spatiotemporal correlations of sensor data to locally detect intrusions. Each node in a cluster first detects and corrects temporal intrusions by comparing the predicted data and the sensing data. Then the cluster head collects the rectified data from all other nodes in the cluster and further detects spatial intrusions that deviate remarkably from other normal data. This technique reduces the communication cost due to local transmission and also reduces computational cost as the cluster-heads carry out most of the computation tasks. However, the S$\alpha$S distribution may not be suitable for real
sensor data and the cluster-based structure may be susceptible to dynamic changes of network topology.

2. Non-Parametric-Based Approaches: Non-parametric techniques do not assume availability of data distribution. They typically define a distance measure between a new test instance and the statistical model and use some kind of thresholds on this distance to determine whether the observation is an intrusion. Two most widely used approaches in this category are histograms and kernel density estimator. Histogramming models involve counting frequency of occurrence of different data instances (thereby estimating the probability of occurrence of a data instance) and compare the test instance with each of the categories in the histogram and test whether it belongs to one of them. Kernel density estimators use kernel functions to estimate the probability distribution function (pdf) for the normal instances. A new instance that lies in the low probability area of this pdf is declared as an intrusion.

- **Histogramming.** Sheng et al. [48] present a histogram based technique to identify global intrusions in data collection applications of sensor networks. This technique attempts to reduce communication cost by collecting histogram information rather than collecting raw data for centralized processing. The sink uses histogram information to extract data distribution from the network and filters out the non-intrusions. Intrusions can be identified by recollecting more histogram information from the network. The identification of intrusions is achieved by a fixed threshold distance or the rank among all intrusions. Drawbacks of this technique include the fact that recollecting more histogram information from the whole network will cause too much communication overhead and the technique only considers one-dimensional data.

- **Kernel functions.** Palpanas et al. [49] propose a kernel based technique for online identification of outliers in streaming sensor data. This technique requires no a priori known data distribution and uses kernel density estimator to approximate the underlying distribution of sensor data. Thus, each node can locally identify intrusions if the values deviate significantly from the model of approximated data distribution. A value is considered as an intrusion if the number of values being in its neighborhood is less than a user-specified threshold. This technique can also be extended to high-level nodes for identification of intrusion in a more global perspective. The main problem of this technique is its high dependency on the defined threshold, while choice of an appropriate threshold is quite difficult and a single threshold may also not be suitable for intrusion detection in multi-dimensional data. Furthermore, the technique does not consider maintaining the model while sensor data is frequently updated. Subramaniam et al. [50] further extend the work of Palpanas et al. [49] and solve the two previous problems of insufficiency of a single threshold for multidimensional data and maintaining the data model built by kernel density estimator. They propose two global intrusion detection techniques for complex applications. One technique allows each node to locally identify intrusions using the same technique as Palpanas et al. [49] and then transmit the intrusions to its corresponding parent to be checked until the sink
eventually determines all global intrusions. In the other technique, each node employs more robust technique called LOCI [51] to locally detect global intrusions by having a copy of global estimator model obtained from the sink. Experimental results show that these techniques achieve high accuracy in terms of estimating data distribution and high detection rate while consuming low memory usage and message transmission. A remaining problem with this technique is its inability to detect spatial intrusions due to the fact that it does not consider the spatial correlations among neighboring sensor data.

**Evaluation of Statistical-Based Techniques:** Statistical based approaches are mathematically justified and can effectively identify intrusions if a correct probability distribution model is acquired. Moreover, after constructing the model, the actual data on which the model is based on is not required. However, in many real-life scenarios, no a priori knowledge of the sensor stream distribution is available. Thus parametric approaches may be useless if sensor data does not follow the preset distribution. Non-parametric techniques are appealing due to the fact that they do not make any assumption about the distribution characteristics. Histogramming models are very efficient for univariate data but are not able to capture the interactions between different attributes of multivariate data. Also, it is not easy to determine an optimal size of the bins to construct the histogram. Kernel functions can scale well in multivariate data and are computationally cheap.

**B. Nearest Neighbor-Based Approaches:** Nearest neighbor-based approaches are the most commonly used approaches to analyze a data instance with respect to its nearest neighbors in the data mining and machine learning community. They use several well-defined distance notions to compute the distance (similarity measure) between two data instance ([52], [53]). A data instance is declared as an intrusion if it is located far from its neighbors. Euclidean distance is a popular choice for univariate and multivariate continuous attributes. Branch et al. [54] propose a technique based on distance similarity to identify global outliers in sensor networks. This technique attempts to reduce the communication overhead by a set of representative data exchanges among neighboring nodes. Each node uses distance similarity to locally identify intrusions and then broadcasts the intrusions to neighboring nodes for verification. The neighboring nodes repeat the procedure until all of the sensor nodes in the network eventually agree on the global intrusions. This technique can be flexible in respect to multiple existing distance-based intrusion detection techniques. However, the technique does not adopt any network structure so that every node uses broadcast to communicate with other nodes in the network, which will cause too much communication overhead. Consequently, it does not scale well to the large-scale networks. Zhang et al. [55] propose a distance-based technique to identify n global intrusions in snapshot and continuous query processing applications of sensor networks. This technique reduces communication overhead as it adopts the structure of aggregation tree and prevents broadcasting of each node in the network [54]. Each node in the tree transmits some useful data to its parent after collecting all the data sent from its children. The sink then roughly figures out top n global intrusions and floods these intrusions to all the nodes in the network for verification. If any node disagrees on the global results, it will send extra data to the sink again for intrusion detection. This procedure is repeated until all the nodes in the network agree on the global results calculated by the sink. This technique considers only one-dimensional data and the aggregation tree used may not be
stable due to the dynamic changes of network topology. Zhuang et al. [56] present two in-
network intrusion cleaning techniques for data collection applications of sensor networks.
One technique uses wavelet analysis specifically for intrusions such as noises or occasionally
appeared errors. The other technique uses dynamic time warping (DTW) distance-based
similarity comparison specifically for intrusions that are erroneous and last for a certain time
period. In this technique, each node transforms raw data into the wavelet time-frequency
domain and identifies the high-frequency data measurements as intrusions and corrects them
using proper wavelet coefficients. The long segmental intrusions can be detected and
removed by comparing the similarity of two sensing series of the neighboring nodes within
two forwarding hops. The proposed techniques take advantage of spatio-temporal correlations
of sensor data for identifying intrusions. A drawback of this technique, however, is its
dependency of a suitable pre-defined threshold that is not obvious to define.

**Evaluation of Nearest Neighbor-based Techniques:** Nearest neighbor-based approaches do
not make any assumption about data distribution and can generalize many notions from
statistical-based approaches. However, these techniques suffer from the choice of the
appropriate input parameters. Additionally, in multivariate data sets it is computationally
expensive to compute the distance between data instances and as a result these technique lack
scalability.

C. **Clustering-Based Approaches:** Clustering-based approaches are popular approaches within
the data mining community to group similar data instances into clusters with similar behavior.
Data instances are identified as intrusions if they do not belong to clusters or if their clusters
are significantly smaller than other clusters. Euclidean distance is often used as the
dissimilarity measure between two data instances. Rajasegarar et al. [57] propose a global
intrusion detection technique based on clustering technique to identify anomalous
measurements in sensor nodes. This technique minimizes the communication overhead by
clustering the sensor measurements and merging clusters before communicating with other
nodes. Initially, each node clusters the measurements and reports cluster summaries rather
than transmitting the raw sensor measurements to its parent. The parent then merges cluster
summaries collected from all of its children before sending them to the sink. An anomalous
cluster can be determined in the sink if the cluster’s average inter-cluster distance is larger
than one threshold value of the set of inter-cluster distances. Determining the parameter \( k \) (the
\( k \) nearest neighbor clusters), which is used to compute the average inter-cluster distance is not
always easy. The parameter of cluster width may also not be defined appropriately.

**Evaluation of Clustering-Based Techniques:** Clustering-based approaches do not require a
priori knowledge of the data distribution and are capable of being used in an incremental
model, i.e., new data instance can be fed into the system and being tested to find outliers.
However, these techniques suffer from the choice of an appropriate parameter of cluster
width. Additionally, computing the distance between data instances in multivariate data is
computationally expensive.

D. **Classification-Based Approaches:** Classification approaches are important systematic
approaches in the data mining and machine learning community. They learn a classification
model using the set of data instances (training) and classify an unseen instance into one of the
learned (normal/ intrusions) class (testing). The unsupervised classification-based techniques
require no knowledge of available labeled training data and learn the classification model
which fits the majority of the data instance during training. The one-class unsupervised techniques learn the boundary around the normal instances while some anomalous instance may exist and declare any new instance falling outside this boundary as an intrusion. The classifier may need to update itself to accommodate the new instance that belong to the normal class. In existing intrusion detection techniques for WSNs, classification-based approaches are categorized into support vector machines (SVM)-based and Bayesian network-based approaches based on type of classification model they use.

1. **Support Vector Machine-Based Approaches:** SVM techniques separate the data belonging to different classes by fitting a hyper plane between them which maximizes the separation. The data is mapped into a higher dimensional feature space where it can be easily separated by a hyper plane. Furthermore, a kernel function is used to approximate the dot products between the mapped vectors in the feature space to find the hyper plane. Rajasegarar et al. [58] propose a SVM-based technique for intrusion detection in sensor data. This technique uses one-class quarter-sphere SVM to reduce the effort of computational complexity and locally identify outliers at each node. The sensor data that lies outside the quarter sphere is considered as an outlier. Each node communicates only summary information (the radius information of sphere) with its parent for global outlier classification. This technique identifies intrusions from the data measurements collected after a long time window and is not performed in real-time. The technique also ignores spatial correlation of neighboring nodes, which makes the results of local intrusions inaccurate.

2. **Bayesian Network-Based Approaches:** Bayesian network-based approaches use a probabilistic graphical model to represent a set of variables and their probabilistic independencies. They aggregate information from different variables and provide an estimate on the expectancy of an event to belong to the learned class. They are categorized as naive Bayesian network, Bayesian belief network, and dynamic Bayesian network approaches based on degree of probabilistic independencies among variables. Naive Bayesian networks techniques capture spatio-temporal correlations among sensor nodes. Bayesian belief network techniques consider the correlations among the attributes of the sensor data. Dynamic Bayesian networks techniques consider the dynamic network topology that evolves over time, adding new state variables to represent the system state at the current time instance.

- **Naive Bayesian Network models:** Elnahrawy and Nath [59] present a Bayesian model-based technique to discover local intrusions and detect faulty sensors. This technique maps the problem of learning spatio-temporal correlations to the problem of learning the parameters of the Bayesian classifier and then uses the classifier for probabilistic inference. Each node locally computes the probabilities of each of its incoming readings being in all subintervals (classes) divided from the whole values interval. If the probability of a sensed reading in its class is smaller than that of being in other classes, it is considered as an outlier. The technique requires no user-specified threshold to determine intrusions and can also be used to approximate the missing readings occurred in the network. It, however,
does not specify how to decide a specific spatial neighborhood under the dynamic change of network topology. Also, it only deals with one dimensional data.

- **Bayesian Belief Network models.** Janakiram et al. [60] present a technique based on Bayesian belief network (BBN) to identify local intrusions in streaming sensor data. This technique uses BBN to capture not only the spatio temporal correlations that exist among the observations of sensor nodes but also conditional dependence among the observations of sensor attributes. Each node trains a BBN to detect intrusions based on behaviors of its neighbors’ readings as well as its own reading. An observation is considered as intrusions if it falls beyond the range of the expected class. Compared to naive Bayesian networks, this technique improves the accuracy in detecting intrusions as it considers conditional dependencies among the attributes. Accuracy of a BBN depends on how the conditional dependence among the observations of sensor attributes exists. This technique may not work well in presence of the dynamic network topology change.

- **Dynamic Bayesian Network models.** Hill et al. [61] present two techniques based on dynamic Bayesian networks (DBNs) to identity local intrusions in environmental sensor data streams. This technique uses DBNs to fast track changes in dynamic network topology of sensor networks. One technique assumes that there is only a measured state variable existing in the multivariate data and the current state can be determined only depending on its historical state. This technique identifies intrusions by computing the posterior probability of the most recent data values in a sliding window. The data measurements that fall outside the expected value interval are considered as intrusions. The other technique uses a more complex DBN including two measured state variables for intrusions detection. This technique makes it possible to operate on several data streams at once.

**Evaluation of Classification-based Techniques:** Classification-based approaches provide an exact set of intrusions by building a classification model to classify. However, a main drawback of SVM-based techniques is their computational complexity and the choice of proper kernel function. Learning the accurate classification model of a Bayesian network is challenging if the number of variables is large in deployed WSNs.

**E. Spectral Decomposition-Based Approaches:** Spectral decomposition-based approaches aim at finding normal modes of behavior in the data by using principle components. Principal component analysis (PCA) is a technique that is used to reduce dimensionality before intrusion detection and finds a new subset of dimension which capture the behavior of the data. Specifically, the top few principal components capture the build of variability and any data instance that violates this structure for the smallest components is considered as an intrusion. Chatziigianakis et al. [62] propose a PCA-based technique to solve data integrity and accuracy problem caused by compromised or malfunctioning sensor nodes. This technique uses PCA to efficiently model the spatio-temporal data correlations in a distributed manner and identifies local intrusions spanning through neighboring nodes. Each primary node offline builds a model of the normal condition by selecting appropriate principal components (PCs) and then obtains sensor readings from other nodes in its group and
performs local real-time analysis. The readings that significantly vary from the modeled variation value under normal condition are declared as intrusion. The primary nodes eventually forward the information about intrusion data to the sink. The offline procedure of selecting appropriate PCs is computationally very expensive.

**Evaluation of Spectral Decomposition-Based Techniques:** Principal component analysis-based approaches tend to capture the normal pattern of the data using the subset of dimensions and can be applied to high-dimensional data. However, selecting suitable principle components, which is needed to accurately estimate the correlation matrix of normal patterns, is computationally very expensive [63-140].

4. **Conclusion**

As far as the technique categories, statistical techniques, data mining, and computational intelligence are employed most widely. Statistical techniques consist of statistical distribution (Palpanas et al., 2003; Subramaniam et al., 2006; Liu et al., 2007; Dallas et al., 2007; Li et al., 2008a; Tiwari et al., 2009), statistical measure (e.g. mean, variance, self-defined, etc.) (Zhang et al., 2008; Pires et al., 2004; Onat and Miri, 2005a,b; Li et al., 2008b), and statistical model (e.g. auto regression) (Curia et al., 2007). Computational intelligence is closely linked to machine learning and remotely linked to data mining. Conceptually, machine learning is more concerned with design and development of the algorithms that enable computers to learn from large-scale datasets. Data mining, however, principally focuses on discovering patterns, associations, changes, anomalies, and statistically significant structures and events in datasets. Under the technique category of data mining and computational intelligence, a couple of examples are introduced, including clustering algorithms (Rajasegarar et al., 2006; Masud et al., 2009; Wang et al., 2009), support vector machine (SVM) (Rajasegarar et al., 2007), artificial neural network (ANN) (Wang et al., 2009), self-organizing map (SOM) (Wang et al., 2009), genetic algorithm (GA) (Rahul et al., 2009), and association rule learning (Yu and Tsai, 2008). Game theory is dedicated to build up smart strategies for identifying vulnerable areas in WSN (Agah et al., 2004a,b). There is only a case that concentrates online king detection with prevention together to protect a hierarchical WSN from both internal and external attacks (Su et al., 2005). Graph-based techniques specialize in modeling a graph with the network flow (Ngai et al., 2006, 2007), which allows applying a few of graph algorithms (such as tree construction, depth-first search, etc.) to detect anomaly. Finally, rule-based techniques, which often build up on prior knowledge such as assumption and experience, are prefer red in flat WSNs (Silva et al., 2005; Yu and Xiao, 2006; Ioannis et al., 2007; Ho et al., 2009).

Two of the most common topology of WSNs, Flat-based Wireless Sensor Network (FWSN) and Cluster-based Wireless Sensor Network (CWSN) (Bace & Mell, 2001), has a large amount of the information is generated by multi-hop communication and the energy consumption is raised in FWSN, such as SPIN (Heinzelman, Kulik, & Balakrishnan, 1999). CWSN is a popular network topology in WSN. In a CWSN, all SNs are clustered, and a cluster head (CH) is elected to manage the operation of its own cluster (Heinzelman et al., 1999; Lindsey & Raghavendra, 2002; Manjeshwar & Agrawal, 2001, 2002). CH should aggregate data from all SNs sensed from a specific target. Therefore, CWSN efficiently reduces the amount of information in the entire network. The advantages of CWSN are a decrease in energy consumption, an increase in the network scale, and a prolonged network lifetime. Many protocols of CWSN have been proposed,
such as LEACH (Heinzelman, Chandrakasan, & Balakrishnan, 2000), TEEN (Manjeshwar & Agrawal, 2001), APTEEN (Manjeshwar & Agrawal, 2002), and PEGASIS (Lindsey & Raghavendra, 2002).

References


