A

Detecting Malicious Data Injections in Wireless Sensor Networks: a Survey

VITTORIO P. ILLIANO and EMIL C. LUPU, Imperial College London

Wireless Sensor Networks are widely advocated to monitor environmental parameters, structural integrity of the built environment and use of urban spaces, services and utilities. However, embedded sensors are vulnerable to compromise by external actors through malware but also through their wireless and physical interfaces. Compromised sensors can be made to report false measurements with the aim to produce inappropriate and potentially dangerous responses. Such malicious data injections can be particularly difficult to detect if multiple sensors have been compromised as they could emulate plausible sensor behaviour such as failures or detection of events where none occur. This survey reviews the related work on malicious data injection in wireless sensor networks, derives general principles and a classification of approaches within this domain, compares related studies and identifies areas that require further investigation.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Unauthorised Access*

General Terms: Security, Algorithms, Measurement

Additional Key Words and Phrases: Wireless Sensor Networks, Security, Correlation

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are an attractive solution to the problem of collecting data from physical spaces, thanks to their flexibility, low cost and ease of deployment. Applications of WSNs include a broad variety of tasks in both shared and personal environments. In shared environments, applications include monitoring infrastructures such as the water network, improving road traffic, monitoring environmental parameters and surveillance. In personal environments, applications include monitoring homes for energy efficiency, user activity such as exercise and sleep, and physiological parameters for healthcare through both wearable and implantable sensors.

In some aspects, WSNs are similar to traditional wired and wireless networks, but they also differ in some others, such as the sensors' limited computational and power resources. Sensors need to be cheap, be physically small, communicate wirelessly and have low-power consumption whether to monitor a human body or a large flood plain and therein lie their main advantages. But these characteristics are also their main limitations as they lead to more frequent failures, poor physical protection, limited degree of redundancy and processing, and limited ability to carry out complex operations.

Wireless sensors carry a much higher risk of being compromised. Their deployments are often unattended and physically accessible, and use of tamper-resistant hardware is often too expensive. The wireless medium is difficult to secure and can be com-

Author's addresses: Vittorio P. Illiano and Emil C. Lupu , Computer Science Department, Imperial College London, Department of Computing, Huxley Building 180 Queen's Gate, South Kensington Campus, London SW7 2AZ, UK; email: {v.illiano13, e.c.lupu}@imperial.ac.uk.

DOI: http://dx.doi.org/10.1145/0000000.0000000

promised at all layers of the protocol stack. Cryptographic operations and key management consume valuable computational and power resources and cannot provide a solution once a node has been compromised. Yet, despite this, WSNs are increasingly used to monitor critical infrastructures and human health where malicious attacks can lead to significant damage and even loss of life.

Faced with the challenge of securing WSNs, researchers have proposed new security solutions for these platforms. The literature is rich and we can only cite a few examples such as [Karlof and Wagner 2003; Perrig et al. 2004; Du et al. 2005; Liu and Ning 2008; Khan and Alghathbar 2010]. Most studies focus on proposing solutions against physical-level and network-level threats, such as jamming attacks, attacks against the routing protocols, confidentiality and integrity of the data in transit. Another body of work is that of *software attestation*, that assesses the node integrity and in particular checks that the nodes run the expected software [Seshadri et al. 2004; Park and Shin 2005; Seshadri et al. 2006; Zhang and Liu 2010].

Despite such solutions, many attacks remain possible against wireless sensor nodes. For example an attacker may compromise a node through its physical interfaces, or tamper with the node hardware itself in order to introduce wrong measurements in the network. This would defeat many of the solutions presented in the literature as cryptographic material present on a compromised sensor would (in the absence of trusted hardware) be available to the attacker. Even when the sensors are hard to reach or to tamper with, an attacker may also seek to compromise the measurements by locally manipulating the sensed environment to induce malicious readings, for example using a lighter to trigger a fire alarm. We refer to all this kind of attacks as *malicious data injections*. Their aim is to compromise the mission of the WSN by producing a picture about the sensed phenomenon, which is different from the real one with potentially devastating effects. In particular, an attacker may seek to:

— **elicit an inappropriate system response**. For example, triggering an overload on a power grid, leading to partial shutdown.
— **masking a desired system response**. For example, silencing an intrusion alarm.

Protecting from such attacks becomes essential because of their potential impact and this survey focuses on solutions proposed that could address this problem. The main challenge for detecting malicious data injections is finding sufficient evidence of the attack. A possible approach is to look for evidence of tampering with the sensor itself through software attestation, as mentioned above. However, software attestation is difficult to deploy in practice (e.g. because of timeliness constraints and device hardware restrictions [Castelluccia et al. 2009]). Attacks that locally modify the sensed environment are also still possible. Another approach is to look for evidence of changed traffic patterns in the communication between the sensors e.g., through traffic analysis [Buttyan and Hubaux 2008]. Whilst effective for detecting network-level attacks, in particular on routing, such approaches often cannot detect malicious data injections since an attacker may modify the values reported by the sensors without changing the traffic patterns of the communications between sensors.

For these reasons, we focus in this article on techniques that look for evidence of compromise in the sensor measurements themselves, *regardless of how they may have been compromised*. Thus, we include in the scope of this survey techniques that perform data analyses on such measurements to detect malicious interference. In addition, we include papers that aim to detect generic anomalies in WSNs, but that are still based on the collected measurements. In contrast, anomaly-based techniques that operate on network parameters such as packet transmission rate, packet drop rate, transmission power etc., are beyond the scope of this survey. Indeed, a key aspect of the detection of malicious data injections is the construction of the *data expectation* *model*, i.e. the model that allows to define expectations about the sensors' measurements. In this context, anomalies arise in the correlation structures that are natively present in the data itself, which cannot be found in network parameters, and may occur without any disruption to the network parameters.

All the papers reviewed in this survey assume that the attacker aims to cause noticeable undesired effects and injects measurements that differ in some detectable way from the correct values that should be reported at that point in time and space. This is the assumption that enables the use of data analysis to detect data injections. However, note that the real value that *should* be reported by compromised sensors is not observable directly. Instead, it can only be characterised from indirect information such as values reported by other sensors, which may or may not be sufficient to detect the compromise. The problem is even more difficult as the indirect information may itself not be correct due to the presence of faults or naturally occurring events. *Faults* refer to any kind of genuine errors, transient or not, and may be difficult to distinguish from a malicious injection. *Events* refer to substantial changes in the sensed phenomenon like a fire, an earthquake etc. We refer to the problem of distinguishing malicious data injections from events and faults as *diagnosis* and review the state-of-the-art approaches to the problem. Another cause for unreliable indirect information is the presence of *colluding sensors* i.e. when multiple compromised sensors produce malicious values in a coordinated fashion. In such scenarios the attacker's leverage on the system increases, and opens the possibilities to new and more effective attacks.

Detecting and diagnosing malicious data injections is a subset of the more general problem of ensuring the integrity of the sensed data, which may have been corrupted by failures or in other ways. This is reflected in the studies surveyed, where many techniques designed for, e.g., detecting faulty sensors or faulty data are also advocated for malicious data injections. Comparatively, only a small proportion of the papers explicitly focus on malicious data injections. However, there is a significant difference between faults and maliciously injected data since the latter is *deliberately* created in sophisticated ways to be difficult to detect. Therefore, there is a need for a survey that (1) analyses the achievements and shortcomings of the work targeted to malicious data injections and that also (2) reviews the state-of-the art techniques proposed for non-malicious data compromise and evaluates their suitability to this problem.

Within the context of WSN, the applicable state of the art studies broadly follow two types of approaches: *anomaly detection* techniques starting from about [2005] ([Tanachaiwiwat and Helmy 2005]) and *trust management* techniques from about [2006] ([Zhang et al. 2006]). We review the state of the art for both approaches and compare the studies surveyed according to their:

— adopted approach
— ability to detect malicious data injections
— results and performance

The remainder of this article is organised as follows. In Sect. 2, we describe existing surveys related to the one we present here. In Sect. 3 we recap concepts useful for understanding the rest of the paper. In Sect. 4 we analyse possible ways of defining an expected behaviour for sensors measurements and analyse the different approaches adopted in the state-of-the art techniques. In Sect. 5 we analyse the state-of-the art detection algorithms. In Sect. 6 we describe two aspects that are important to tackle malicious data injections beyond detection: diagnosis and characterisation of the attack. In Sect. 7 we give comparison tables for the techniques surveyed and their experimental results, together with a brief discussion. Finally, in Sect. 8, we present our conclusions and the open issues that emerged from this study.

2. RELATED SURVEYS

To the best of our knowledge, there are no previous surveys of techniques to detect malicious data injections in WSNs. Several surveys are however related and we discuss them in this section.

Boukerche et al. [2008] analyse techniques for secure localisation algorithms in WSNs. There are some similarities between malicious data injections and attacks on localisation systems, since the sensors' location can be regarded as a particular physical phenomenon being sensed. However, many aspects of the techniques described in [Boukerche et al. 2008], are specific to the localisation problem. In particular, constraints on the topology, the radio transmission power and delay provide a clear criterion to check the consistency of the information reported by the sensors. In contrast, we focus on techniques that do not require a-priori knowledge of the physical phenomena monitored to check data consistency but examine and infer correlations from the data itself.

Rajasegarar et al. [2008] review eleven state-of-the-art papers about anomaly detection in WSNs. Although they focus on detecting intrusions, the survey also covers eliminating erroneous readings and reducing power consumption. The detection algorithms surveyed consider sensor measurements as well as network traffic and power consumption. In contrast, we focus on a more specific target: the detection of malicious data injections. We cover a broader spectrum of papers since we include techniques other than anomaly detection, describe further steps for detecting malicious data and include a significant amount of literature published since then.

Xie et al. [2011] survey anomaly detection in WSNs, with a focus on the WSN architecture (Hirearchical/Flat) and the detection approach (statistical, rule based, data mining etc.). They describe the detection procedure in a similar way to us: definition of a "normal profile", which we refer to as normal or expected behaviour, and test to decide whether it is an anomaly or not, or to what extent. However, our survey is structured based on the approach to both the definition of the normal behaviour and the detection based on it, while [Xie et al. 2011] focus only on the latter. This choice allows us to pinpoint the motivation for the use of a particular detection technique, based on how the data normally looks like. Moreover, the diagnosis process that classifies an anomaly as an attack is not analysed in [Xie et al. 2011] whereas it forms an important part of this survey.

Several surveys discuss trust management for security in WSNs (e.g. [Lopez et al. 2010; Özdemir and Xiao 2009; Sang et al. 2006]). However, they focus on attacks conducted through the network layer, while malicious data injections are given little attention. Yu et al. [2012] lists all the threats that can be mitigated by trust management, including "Stealthy attacks" – a kind of malicious data injection – but these are not analysed in detail. Similarly, Zahariadis et al. [2010a] build a taxonomy of trust metrics, which includes consistency of reported values/data, but they focus mostly on the other network-related metrics. Also Shen et al. [2011] survey defensive strategies against attacks to the network layer. In particular, such strategies are derived from game theory and take into account the strategies that can be adopted by the attacker to balance the profit and loss of reputation coming from the attack; in our survey instead, we focus on techniques to assign and maintain such reputation.

The closest survey to the one presented here is [Jurdak et al. 2011]. It describes anomaly detection strategies for detecting faults due to environmental factors (e.g. obstructions near the sensor) or node hardware/software. Their description of anomaly detection is similar to ours but the two surveys differ notably in the nature of the anomalies considered: attacks in our case, faults in theirs. Jurdak et al. [2011] also claim that anomalies can be detected by spatial or temporal comparisons between sensors, since it is unlikely that many sensors will exhibit a calibration skew or failure at the same time (assuming there are no group failures). This assumption considers anomalies (faults) as independent but does not hold in the presence of malicious data injections, in particular when there is collusion between the compromised sensors.

3. PRELIMINARIES

We describe in the following how sensors measurements are generally gathered in a WSN. We also introduce the two approaches used to detect malicious data injections so far: *anomaly detection* and *trust management*.

3.1. Data Aggregation Schemes and Their Consequences

The typical workflow of a WSN starts with measuring a physical phenomenon through sensing devices connected to a wireless node that propagates the measurements through the network towards data sinks. Measurements collected and aggregated by data sinks (e.g., base stations) can then be interpreted or transmitted to a remote station. However, data can also be aggregated in the network by the intermediate transmitting nodes, with many possible variations on the aggregation architecture. The choice between the different schemes is based on criteria that optimise power efficiency, number of devices, coverage of the physical space etc. Finding the optimal architecture based on such criteria remains an important research challenge.

Early work considered that all raw measurements are collected at the base station, which performs data fusion and other computations [Shepard 1996; Singh et al. 1998]. Later on, especially after the introduction of the LEACH protocol [Heinzelman et al. 2000], architectures became increasingly hierarchical. LEACH applies a one-level hierarchy where sensors are organised in clusters and communicate with the cluster-head, which, in turn, communicates with the base station, as shown in Fig. 1. Cluster-based protocols, and especially those where the clusters change dynamically in time [Heinzelman et al. 2000], have proven to be more energy efficient when communication with the base station requires multi-hop transmissions [Heinzelman et al. 2000].

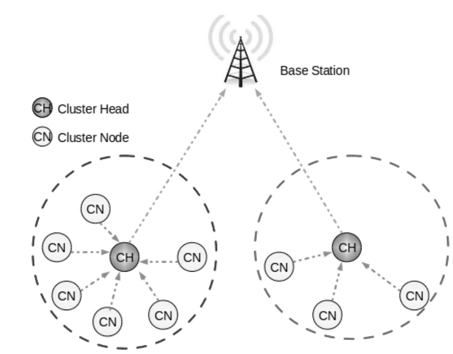

Fig. 1. LEACH measurements collection architecture.

The one-level hierarchy introduced in LEACH can be generalised to tree-based structures as described in [Fasolo et al. 2007]. Intermediate tree nodes may simply merge the packets generated by different sources into a single packet without processing the data. This is referred to as *in-network aggregation without size reduction* [Fasolo et al. 2007]. Alternatively, they process the sensor measurements by applying aggregation operators (e.g. mean, minimum, maximum), which is referred to as *in-network aggregation with size reduction* [Fasolo et al. 2007]. So, cluster heads assume the burden of the additional computation in order to minimise the data transmitted. In essence, this trades the power costs of computation for those of communication, but since in WSNs communication consumes much more power, the trade is usually favourable.

Information about the WSN architecture and where data aggregation is carried out, is important for allocating the detection task to the WSN nodes. For instance, if in-network aggregation with size reduction is used, the base station cannot analyse all the measurements and the aggregating nodes must assist the base station in the detection task. In this case the integrity of the aggregation process at these nodes must also be ascertained [Przydatek et al. 2003; Ganeriwal and Srivastava 2004; Roy et al. 2014].

3.2. Relationship to Anomaly Detection and Trust Management

Detection of malicious data injections has been addressed with two main approaches so far: *anomaly detection* (e.g., [Tanachaiwiwat and Helmy 2005; Liu et al. 2007; Sun et al. 2013]) and *trust management* (e.g., [Atakli et al. 2008; Bao et al. 2012; Oh et al. 2012]). While anomaly detection defines normal behaviours to infer the presence of anomalies, *trust management* evaluates the confidence level (trustworthiness) that a sensor's behaviour is normal. Compromised sensors are then expected to get low trust values when they deviate from their expected behaviour. Although anomaly detection is also based on the definition of an expected behaviour –"Anomaly detection refers to the problem of finding abnormalities in the data that do not conform to expected behaviour" Chandola et al. [2009]– the two approaches differ in how deviations are interpreted. In trust management, the sensors measurements are analysed with the granularity of a sensor, and each sensor has a trust value that is incrementally updated in time. Anomaly detection approaches, instead, can be applied with no restrictions in granularity from the single measurement to the whole system, and generally work by defining a boundary for expected behaviour such that everything outside that boundary is abnormal.

Given the similarities and differences between the two approaches, we structure the following two sections as follows: in the next section we describe how to gather information about expected data, regardless of whether it is for anomaly detection or trust management. In Sect. 5 instead, we describe how to detect deviations from the expected data, treating anomaly detection and trust management separately.

4. MODELLING EXPECTED DATA

In our context, *expected data* refers to a set of properties characterising the measurements that are free of malicious injections. Given that no previous surveys focus on this issue, we start by introducing a generic formulation of WSN sensing. This enables us to analyse different models for the expected data and describe the related work with a coherent terminology as the terms used often differ from one article to another.

4.1. A Characterisation of the Problem

We focus on interpreting the data and abstract from implementation-related issues such as synchronisation between sensors, and network related issues such as packet loss or delays. We consider a deployment region D, in which a set of N sensors are placed. Every sensor measures a physical attribute such as temperature, wind, water quality, power, gas flows. The sensors' measurement process is characterised by a degree of uncertainty, which may be due to noise, faults and also malicious data injections. It is desirable to remove this uncertainty, so we introduce an ideal function φ, which represents the value of a sensor's reading in the absence of any source of uncertainty. The independent variables of such function are the point in space s and the time instant t to which the readings correspond, as shown in Eq.1:

$$\varphi(s,t) \quad s \in D, \forall t \tag{1}$$

We refer to this function as the *physical attribute function*. The reading produced at time t by a generic sensor i, deployed at position s_i, is some approximation of the physical attribute function evaluated at (s_i, t). A generic sensor's reading can then be modelled as a function S_i, that adds a generic measurement error $\epsilon(s_i, t)$ to the physical attribute, which may change with time and space. Eq. 2 defines the function S_i:

$$S_i(t) = \varphi(s_i, t) + \epsilon(s_i, t) \quad i \in 1, ..., N \tag{2}$$

Note that the sensors' readings are the only observable quantities; both the physical attributes and the measurement errors are not observable directly. When malicious data injections occur, some of the sensors' readings also become unobservable, since the attacker substitutes fabricated measurements for the real ones. There is then the need to describe the real measurements with related information from some observable quantities. This process is effective if such related information allows us to discriminate injections and is itself not susceptible to injections.

Describing the unobservable real measurement in terms of observable properties is a modelling process, that makes assumptions about how data can be described. For instance, the measurements produced by a sensor can be modelled as samples from a normal distribution [Zhang et al. 2006]. Assuming compromised nodes do not produce data compliant with a normal distribution, the model can then discriminate compromised nodes [Zhang et al. 2006].

The relation that links the problem to a model is a one-to-many relation. Different models of the same problem are not equivalent and choosing a good model is essential for good performance. In particular, a good model should be characterised by:

— **Accuracy** – No model is perfect and every model is in fact an approximation. An accurate model minimises the approximation error.
— **Adaptability** – Physical attributes measured by the sensors change in time. As a consequence, models should adapt to the dynamically changing environment.
— **Flexibility** – Good models should be applicable in a flexible way, regardless of the application. Such models should abstract as many details as possible and capture only those properties that are needed.

These desirable characteristics conflict with each other: accuracy may be better achieved with context-specific details, which limit flexibility and compromise adaptability. A particular adaptability requirement which significantly affects accuracy and flexibility is the sensors' *mobility*, as when sensor nodes migrate to new locations, previous expectations are invalidated. Indeed sensor migrations correspond to a change in s_i in Eq. 2, which potentially changes all the measurements time series, leaving sensor specific noise as the only invariant.

Support to Mobility. Even though mobility is an aspect that is not directly addressed in the detection of malicious data injections, some techniques are more suited to support mobile sensors than others. In particular, anomaly detection techniques that compare the measurements within a neighbourhood without considering past behaviour (e.g. [Handschin et al. 1975; Ngai et al. 2006; Liu et al. 2007; Wu et al. 2007; Guo et al. 2009]), can generally accomodate mobility, since for every time instant, new expectations are extracted. However, such techniques also need to become aware of topology changes in the presence of mobility.

Trust-management techniques with exchanges of trust information (e.g., [Bao et al. 2012; Huang et al. 2006; Ganeriwal et al. 2003; Momani et al. 2008]) are also suited for mobility, since a sensor i which migrates to a new area and becomes a neighbour of j, can benefit of recommendations from sensors which have been j's neighbours in the past [Zahariadis et al. 2010b]. So far, exchanges of trust informations have been considered without investigating the effects of mobility, therefore sensor i will generally maintain indirect information about sensor j only if there is interaction between i and j, and i cannot observe j's behaviour (e.g., it is not in the wireless communication range). When sensors are mobile instead, even if i and j never interacted, they may interact in the future if they get closer. Only at that time, recommendations for j become of i's interest, and a criterion to request such recommendations is needed.

The existing studies analysed in the remainder of this work, by and large, ignore mobility aspects. We conclude, in light of the considerations above, that more work is required to deal with the problems arising from the sensors' mobility.

4.2. Exploiting Correlation

Since the original measurements substituted with fabricated ones cannot be observed directly, they need to be characterised indirectly with related information. The relationship between two pieces of information is a *correlation*, which can be calculated online, with historical data, or modelled a-priori. In either case, coexistence of genuine and compromised measurements may cause disruptions in the correlation, assuming that the correlations have not changed between the moment when they are calculated and the moment when they are used.

We refer here to *correlation* in a broad sense, meaning that there is some kind of continuous dependency, as opposed to Pearson's correlation coefficient, which is the most commonly used correlation metric. Referring to E, μ and σ as the expected value, the mean and the standard deviation respectively, the Pearson correlation coefficient ρ_{XY} between two random variables X and Y is given by:

$$\rho_{XY} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{3}$$

Note that this coefficient measures only linear dependency between two variables, while non-linear dependencies may be missed.

In Wireless Sensor Networks we can generally consider correlations across three different domains: *temporal*, *spatial* and *attribute* domain [Rassam et al. 2013].

— **Temporal correlation** is the dependency of a sensor's reading on its previous readings. It models the coherence in time of the sensed physical process.
— **Spatial correlation** is the dependency in readings from different sensors at the same time. It models the similarities in how the sensed phenomenon is perceived by different sensors.
— **Attribute correlation** is the dependency in readings that are related to different physical processes. It models physical dependencies among heterogeneous physical quantities such as temperature and relative humidity.

Usually a combination of these different kinds of correlation is used. We now analyse how they contribute to the definition of expected data.

4.3. Temporal Correlation

Variations in time of the sensed data can be modelled as a random process [Boukerche 2009], where the random variables at different time are correlated. As Eq. 2 shows, the variation of a sensor's measurements in time depends on both the variations introduced by the physical attribute and the measurements' error. The variation of the physical attribute in time is subject to constraints, such as the presence of gradual changes, or the alternation of some patterns, since the phenomenon observed usually follows the laws of physics. So, if the measurements are gathered with sufficiently high frequency, consecutive measurements would be subject to similar constraints. This simple observation justifies a procedure that identifies errors (including malicious injections) when temporal variations do not respect these constraints. However, there are two main difficulties in applying this observation to assess deviations: the time evolution of the process is subject to uncertainty factors and the measurements are subject to noise.

When using Kalman Filters [Kalman 1960] to model time series, these two factors are known respectively as *process noise* and *measurement noise*. The measurement noise is typically modelled as a Gaussian process. The process noise, instead, comes from the imperfections of the model used to describe the process dynamics. For example, when modelling the process as a discrete Markov process, the value at time t_1 can be written as:

$$\varphi(t_1) = F(\varphi(t_0)) + w(t_0) \tag{4}$$

where F models the expected evolution of the time process and w is the process noise.

The use of a Markovian process, modelled with a Kalman filter, forms the basis of the Extended Kalman Filter (EKF) based algorithm by Sun et al. [2013]. Here, each sensor builds up a prediction for its neighbours as a function of the neighbours' previous reading. The difference between the predicted and the actual value forms a deviation that is used to detect malicious data injections. However, the authors point out that an attacker can elude the EKF algorithm by introducing changes that are sufficiently small. To address this shortcoming, the authors apply the CUSUM GLR algorithm, which considers the cumulative deviation across more time samples and tests it to be zero-mean. This property, makes it more difficult for attackers to introduce deviations that achieve their goal.

[Subramaniam et al. 2006] also define expected data with temporal correlation. Here, the authors fit the Probability Density Function (PDF) of the measurements inside a temporal window, through kernel density estimators. Given a new measurement p, the PDF gives information about the expected number of values falling in $[p-r, p+r]$ (with parameter r dependent on the application).

4.4. Spatial Correlation

In the presence of sudden events, the dynamics of a physical process can change rapidly. Often detecting such events, such as a forest fire, a volcanic eruption, a cardiac attack is the very purpose of the WSN. However, the occurrence of the event may disrupt temporal correlations, giving rise to false anomalies. Nevertheless, different sensor nodes generally are affected by the event and produce measurements that are spatially correlated to the event source: as a consequence, the measurements of different sensors are correlated during the manifestation of the event [Boukerche 2009]. This phenomenon is known as spatial correlation.

Several techniques make use of spatial correlations by relating the measurements from different sensors in the same time interval – this is equivalent to fixing t in Eq. 2 and letting the parameter i vary. The most widespread spatial correlation model is also the simplest: it assumes that all sensors would produce the same measurements in the absence of errors and noise i.e., they measure the same value, and we refer to this model as *spatially homogeneous* [Zhang et al. 2006; Ngai et al. 2006; Wu et al. 2007; Liu et al. 2007; Bettencourt et al. 2007]. In terms of the physical attribute model given in Eq. 1, $\varphi(s,t)$ is actually a function of time only. In this scenario, the sensors' measurements can be described by a Gaussian distribution. This is because they are independent observations of random variables with a well-defined expected value and well-defined variance, and according to the *central limit theorem* their values will be approximately normally distributed [Rice 2007]. Detecting sensors with abnormal readings becomes then a simple matter of detecting deviations from the spatial measurements' distribution and the accuracy of the distribution estimation increases with the number of sensors.

The homogeneous model is suitable only for regions of space which are small enough and free of obstacles. However, when the deployment topology and characteristics of the physical phenomena violate the homogeneity assumption, the spatial propagation rules can still induce spatial correlations. In many applications, such propagation can be assumed monotonic [Guo et al. 2009]. This implies that the values of the physical attribute at a point in space, should either increase or decrease as the distance from that point increases. For example, when monitoring for forest fires the temperature decreases monotonically as the distance from the fire increases. To ascertain whether this property holds, Guo et al. [2009] divide the deployment space into sections, called faces. For each face, the authors construct a "distance sequence", corresponding to the sequence of sensors ordered by the distance from that face. While sensing the phenomenon, the sensors readings are sorted to generate the *estimated sequence*, which is then compared to all possible distance sequences, as shown in Fig. 2. The sensors measurements are consistent with the expectation if the estimated sequence corresponds exactly to one of the distance sequences. This condition is then relaxed to cope with noisy measurements which degrade the validity of the monotony assumption, but the main factor undermining its validity is the presence of multiple simultaneous events [Guo et al. 2009].

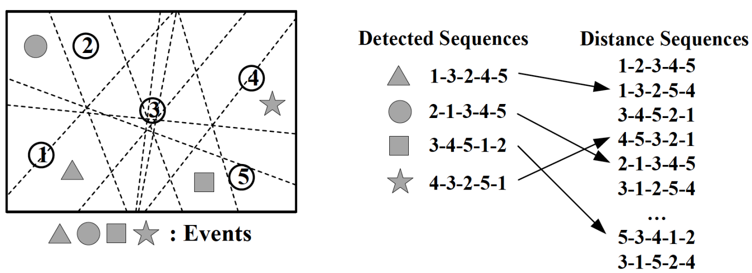

Fig. 2. Detection of measurements which do not comply with the monotonicity assumption, from [Guo et al. 2009].

Instead of considering a strict assumption like the monotonicity of the measurements, it is possible to model correlations between the sensors' readings as a function of their spatial positions. An example of such a model is the *variogram*, defined as the variance of the difference between values of a physical phenomenon at two locations. In our notation, the variogram between two points s_1 and s_2 is defined as $var(\varphi(s_1,t) - \varphi(s_2,t))$. When the physical phenomenon is assumed to be isotropic, the variogram is expressed as a function of the distance only and Zhang et al. [2012] have applied it, to compute an expected measurement as a function of the measurements from other sensors. Note that in the presence of obstacles, the variogram is not only a function of the distance, but also depends on the absolute positions.

Rather than considering distances between sensors, spatial correlation can be calculated as a function of the sensor values themselves. This choice caters for sensors at the same distance, but subject to different noise or obstacles in space. However, it comes at the price of correlation updates when sensors are mobile. For example, Sharma et al. [2010] express a sensor's measurement as a linear combination of the measurements from the other sensors, extract the function's parameters and derive expected sensor readings. Dereszynski and Dietterich [2011] instead, derive expected readings by fitting the joint probability distribution of the measurements from N sensors, after assuming it is an N-variate Gaussian distribution. Note that this approach also implicitly assumes a linear model, as the covariance between two random variables captures linear dependencies (we have mentioned in Sect. 4.2 that this is true for the Pearson correlation coefficient, which is just a normalisation of the covariance index).

Not infrequently, spatial correlation is used in conjunction with temporal correlation, since they capture different kinds of deviations. For example, Bettencourt et al. [2007], propose an outlier detection technique based on two kinds of differences: between a sensor's reading and its own previous reading (temporal correlation) and between readings of different sensors at the same time (spatial correlation). A distribution for both differences is used to check if data samples are statistically significant as related to the temporal domain as well as to the spatial domain.

4.5. Attribute Correlation

In the same WSN, sensors observing different physical attributes such as light, vibrations, temperature etc., may coexist. Some of these attributes may be correlated because of the physical relationship between them e.g., temperature and relative humidity. Commonly, at every deployment location, s_i different sensors in charge of measuring different physical processes are connected to a single sensor node. As described by Eq. 5, for a fixed point in space and time we have a set of A physical attributes. We define attribute correlation as the correlation between them.

$$\varphi^a(s,t) \quad a \in 1,...,A \tag{5}$$

We expect attribute correlations to also be observable in the measurements reported by the sensor nodes. Note, however, that attribute correlations between sensors belonging to the same node are not informative as an attacker who has compromised a node may tamper with all the measurements collected on that node. However, attribute-based expectations are very useful in conjunction with spatial correlations, when spatial redundancy is limited. For example, body sensor networks for healthcare have limited redundancy since it is impractical to cover the patient with several sensors. We can then still exploit correlation among different physiological values (the attributes) measured by different sensor nodes.

An example in the healthcare domain is presented by Salem et al. [2013], who exploit spatial-attribute correlations together with temporal correlations. Based on a Discrete Wavelet transform, they decompose the attribute signals into average and fluctuation signals. Abrupt temporal changes in the energy of the fluctuation signal are detected by a Hampel filter, which flags outlying attributes. This technique has been proposed for fault-tolerant event-detection, based on the observation that multiple attributes are expected to be flagged simultaneously in the presence of an event, due to their attribute correlations. Then, if the minimum number of outlying attributes is not reached, the sensors reporting the outlying readings are considered faulty. However, in the context of malicious data injections, this technique would not prevent an attacker to deliberately inject measurements that subvert the event-detection.

Table I. Correlation Types

Correlation Type	Information Captured	Variations
Temporal	$\text{corr}(\varphi^a(s,t_1),\varphi^a(s,t_2))$	— Time-series evolution model — Time memory (the maximum value of W for which the correlation is modelled)
Spatial	$\text{corr}(\varphi^a(s_1,t),\varphi^a(s_2,t))$	— Spatial model, e.g. homogeneous, monotonic, variogram, linear dependency — Correlation variational model, e.g. distance-dependent, sensors-dependent, fixed — Neighbourhood selection criterion
Attribute	$\text{corr}(\varphi^{a_1}(s,t),\varphi^{a_2}(s,t))$	— Correlation extraction process, e.g. from physical laws, temporal/spatial analysis etc.

4.6. Overview of Techniques for Extracting Expected Data

In the previous sections we have analysed different types of correlations, the information they capture, and variations in the exploitation of the same correlation across the techniques proposed in literature. In Table I we summarise this analysis.

5. DETECTING DEVIATIONS FROM EXPECTED DATA

Expectations about the actual measurements can be used to calculate the deviation of the reported measurements from them. Both anomaly detection and trust management require an expectation, but they use different criteria to cope with abnormal data. Specifically, anomaly detection uses the expectation to discriminate between anomalous and normal data. Trust management instead, uses a criterion to map the deviation from expected data to a trust value. Since the two techniques differ in how they interpret deviation, we will consider them separately in this section.

5.1. Anomaly Detection Techniques

Anomaly detection is a method to characterise data as normal or anomalous. In contrast to Rajasegarar et al. [2008] who consider outlier detection and anomaly detection as equivalent, we instead consider outlier detection as one of the techniques belonging to the anomaly detection category. The reason is that outlier detection identifies the samples that are unlikely to manifest. However, the measurements could be anomalous with respect to other criteria, that cannot be reduced to the problem of finding outliers. Consider for example the case where a sensor is experiencing a *stuck at* fault, i.e., it always produces the same measurement. An outlier detection technique applied on a subset of the last measurements from that sensor will detect no outlier. However, an anomaly still exists and could be detected by considering, for instance, the low variance in the measurements' distribution. To clarify this aspect, we present statistical tests for anomaly detection and highlight their differences with more traditional outlier detection techniques. Then we delve into techniques for outlier detection, which is still the most commonly adopted technique for anomaly detection.

5.1.1. Statistical Tests. Techniques based on statistical tests assume a probabilistic data distribution. Real data is then checked against this distribution to verify its compliance to it. Techniques based on statistical tests are more general than outlier detection because they check the compliance of both outliers and non-outliers to the distribution whereas outlier detection focuses on the *classification* of single data samples.

For example, Rezvani et al. [2013] use a technique based on statistical tests to detect malicious colluding nodes. They assume spatial homogeneity and model sensor measurements as a ground-truth value plus some noise. The ground truth is estimated as a weighted average of measurements and the difference between the estimated value and each measurement is assumed to be normally distributed. This assumption is in keeping with the application of the *central limit theorem* [Rice 2007] – errors are assumed to be due to a large number of independent factors and thus to follow a normal distribution. Compliance with the normal distribution is then assessed with the Kolmogorov-Smirnov test, which quantifies the distance between an empirical distribution (the errors distribution) and a reference distribution (the normal distribution).

5.1.2. Outlier Detection. Outlier detection methods consider as anomalous data that lies outside of the space where most data samples lie. This technique can identify malicious data injections reasonably effectively as long as maliciously injected values are a minority in the dataset and deviate significantly from the other data.

Historically, outlier detection has been proposed in WSN for different purposes, sometimes with opposing goals: in some cases the techniques aim to filter out outliers, in others the outliers represent the main interest. For example, outliers are filtered out to increase data accuracy [Janakiram et al. 2006] and for energy saving [Rajasegarar et al. 2007]. Applications where outliers are the main interest include fault detection [Paradis and Han 2007], event detection [Bahrepour et al. 2009; Zhang et al. 2012] and detection of malicious data. We describe below different approaches to the outlier detection problem independently of the application context, but we focus on those techniques that can be applied to detecting malicious data injections.

Nearest-Neighbour-Based Outlier Detection. In nearest-neighbour based outlier detection, an outlier is a data sample with a narrow neighbourhood, where a neighbourhood comprises the data samples within a certain distance. Most nearest-neighbour based techniques in WSNs are inspired from the well-known LOCI method [Papadimitriou et al. 2003], which calculates for every sample, the number of neighbours in a data space characterised by the radius αr, where α is a parameter used to reduce computational complexity. The relative difference with the average number of neighbours, i.e. the samples within a radius r in the data space, constitutes the *Multi-Granularity Deviation Factor* (MDEF). The MDEF is compared to a threshold equal to 3 times the MDEF standard deviation to ensure that less than 1% values are above the threshold when the distances between data samples follow a Gaussian distribution (the percentage increases up to 10% for other distributions). Note that this method is applicable to malicious data injections by considering the sensors' measurements as the data samples. However, the research community seems to have somewhat lost interest in approaches based on nearest-neighbour since they have large computational overheads due to the calculation of the neighbours for each new data sample.

Clustering-Based Outlier Detection. Clustering is another technique often used for outlier detection. Here the outliers are the elements distant from the others, after organising close elements into clusters. For example, Rajasegarar et al. [2006] identify a cluster as anomalous if its distance to other clusters is more than one standard deviation of the distance of the cluster elements from the mean.

PCA-Based Outlier Detection. Principal component analysis (PCA) [Marsland 2009] is a common data analysis technique, that has also been applied to find outliers [Chatzigiannakis and Papavassiliou 2007]. PCA is based on a projection of the k-dimensional data space onto another k-dimensional data space, where the variables describing the data samples are linearly uncorrelated. This transformation is defined in such a way that the projected variables are sorted with descending variance. The first p out of k variables are defined as the *principal components* and can be projected back to the original data space to obtain a prediction vector y_{norm} [Jackson and Mudholkar 1979], also referred to as *normal data* [Chatzigiannakis and Papavassiliou 2007]. The difference between original and normal data constitutes the *residual vector* y_{res}. Residual vectors that are large in magnitude (i.e., when the squared prediction error $SPE = \|y_{res}\|^2$ of the residual vector is greater than a threshold) are interpreted as deviations from the predicted (normal) vector and considered as outliers [Chatzigiannakis and Papavassiliou 2007]. PCA can be applied to k-dimensional datasets e.g., made up of the measurements time series of k sensors [Chatzigiannakis and Papavassiliou 2007]. In this case y_{res} reflects changes in spatial correlation but the same idea can also be applied to the temporal or attribute domains.

Classification-Based Outlier Detection. Traditional classification techniques learn how to recognise samples from different classes. Anomaly detection considers two classes: anomalous and normal, however, anomalous data samples are rarely observable compared to the normal ones. Therefore, classification for anomaly detection is generally reduced to a one-class classification problem, based on the observation of normal samples only.

Normal and anomalous samples can be viewed as points within two different regions of the data space. Finding the boundary that separates the two regions may be infeasible, because the regions overlap and, even when a boundary exists, it may have a complex shape. Support Vector Machine (SVM) are a classification technique that can overcome this limitation by projecting the data samples into a higher dimensional space. In the projected data-space, a boundary that separates normal from anomalous points may exist even if it does not exist in the original space, or may have a simpler shape. For example, the normal samples could be contained within a sphere in the projected data space. When the data space contains only positive values, this problem reduces to a special type of SVM called *one-class quarter-sphere SVM* [Laskov et al. 2004], which is represented in Fig. 3. With this approach, the classification problem

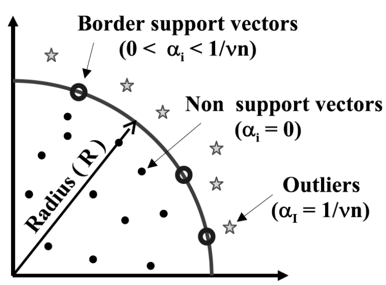

Fig. 3. One-class quarter-sphere support vector machine, from [Rajasegarar et al. 2007].

reduces to finding the sphere's radius. Depending on how the WSN dataset is given in input to quarter-sphere SVM, the classification can be made across its time domain [Rajasegarar et al. 2007], attribute domain, or both [Shahid et al. 2012] .

Bayesian networks have also been applied in WSNs to detect outliers with a classification-based approach. A Bayesian network defines the relations of conditional independence between random variables through a network graph. In WSNs, the random variables can be different values in space and time of the physical attributes.

An example of application of Bayesian networks to WSNs is given by Dereszynski and Dietterich [2011]. The physical attribute $\varphi(s_i, t_k)$ is modelled as a random variable which depends on $\varphi(s_i, t_{k-1})$ (1^{st}-order Markov relationship) and on values at different locations $\varphi(s_{j\neq i}, t_k)$. The aim is to find the state of a sensor, modelled by a random variable with two possible values: *working* and *broken*. The posterior probability of the measurements, which depends on both the physical attribute and on the sensor state variable, is maximised with respect to the state variables to identify faulty nodes. Dereszynski and Dietterich [2011] evaluated their approach assuming that faulty sensors have an high increase their measurements' variance (by 10^5), motivated by the observation that the measurements of faulty sensors appear more noisy. Though reasonable in the case of faults, this assumption does usually not hold for data injections, where an attacker can choose the measurements distribution arbitrarily and wishes in most cases to remain undetected.

Statistical Outlier Detection. Statistical outlier detection identifies outlying data samples through statistical characterisation of the tail of the samples' probability distribution, as shown in Fig. 4.

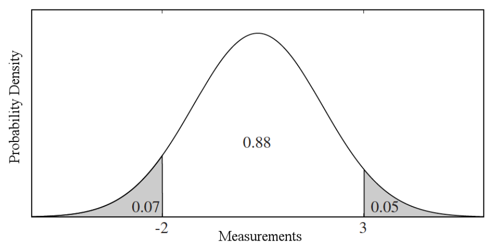

Fig. 4. Statistical characterisation of the sensed data for outlier detection, from [Bettencourt et al. 2007].

Note that this approach differs from anomaly detection based on statistical tests, as it does not test the samples' compliance to their expected distribution, but only identifies the outliers that lie on the tails of the distribution. For example, outliers can be defined as samples far from the mean. Ngai et al. [2006] have applied this idea to measurements from different sensors, thus exploiting spatial correlation. The *spatial* sample mean $\hat{\mu}_S$ of measurements from N different sensors is defined as:

$$\hat{\mu}_S = \frac{1}{N}\sum_{j=1}^{N} S_j(t) \tag{6}$$

Ngai et al. [2006] use it to evaluate the deviation of sensor j from the spatial mean, compared to the magnitude of the mean itself with the metric: $f(j,t) = \sqrt{\frac{(S_j(t)-\hat{\mu}_S)^2}{\hat{\mu}_S}}$.

Similarly Tanachaiwiwat and Helmy [2005], use the metric $t^* = \frac{S_i(t)-(\mu_{T_i}\pm\delta)}{S_{T_i}/\sqrt{W}}$, where μ_{T_i} and S_{T_i} are respectively i's temporal mean and sample standard deviation in a window of size W and δ is a priorly known variation between sensor i and j due to the observed phenomenon's spatial propagation. Considering the model in Sect. 4.1, a generic sensor j calculates its *temporal* sample mean in the W-wide time window $[t_{K-W+1}, t_K]$ as:

$$\hat{\mu}_{T_j} = \frac{1}{W}\sum_{n=0}^{W-1} S_j(t_{K-n}) \tag{7}$$

The *temporal* standard deviation is instead calculated as:

$$S_T = \sqrt{\frac{1}{W-1}\sum_{n=0}^{W-1}(S_j(t_{K-n}) - \hat{\mu}_{T_j})} \tag{8}$$

The value of t^* is then compared with a threshold, that is set to 3 since, in normally distributed data, this accounts for approximately 99.7% of the population (the percentage decreases to 90% for other distributions).

In some cases the median is preferred to the mean, since the former has the advantage of being insensitive to outliers. Indeed, a problem in outlier detection is how to find the general (non-outlying) trend from data affected by outliers. The mean is sensitive to outliers, since it is proportional to the magnitude of each operand. The median takes instead one element to represents all the others. Wu et al. [2007] use the median operator to aggregate sensors measurements in a neighbourhood. We can refer to it as a *spatial median*. If we order the N sensors measurement at time t such that $S_1(t) \leq S_2(t) \leq ... \leq S_N(t)$, the median in the spatial domain is calculated as:

$$\tilde{\mu_S} = \begin{cases} S_{(N+1)/2}(t) & \text{if N is odd} \\ S_{N/2}(t) & \text{if N is even} \end{cases} \tag{9}$$

After calculating the difference between the median and each value, there are two possibilities: comparing each difference to the measurements magnitude, or comparing it to the general distribution of the differences. Yang et al. [2006; Wu et al. [2007] detect outliers in the differences, assuming they are normally distributed. Instead of relying on the assumption of a Gaussian distribution, the probability distribution can also be estimated from the data [Bettencourt et al. 2007].

When sensing multiple physical attributes, the distribution of the measurements across all attributes can be considered, rather than a separate distribution for each one. This approach can potentially detect outliers that a separate approach would fail to detect. Liu et al. [2007] combine different attributes using the Mahalanobis distance, which is based on the inter-attribute correlation and defines how the data is statistically distributed in the attribute space. This scheme is shown in Fig. 5.

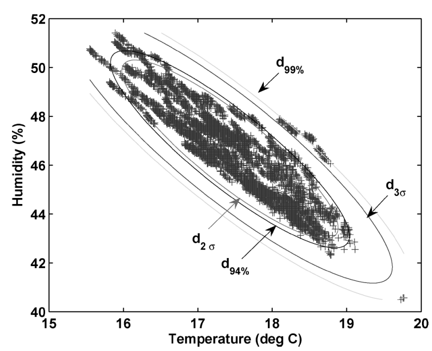

Fig. 5. Statistical distribution in the attribute space made up by temperature and humidity. Points with Mahalanobis distance greater than d are treated as outliers, from [Rajasegarar et al. 2009].

5.2. Trust-Management Based Techniques

Trust-management considers the trustworthiness between two classes of entities: a trustor and a trustee. The trustor assigns each trustee a trustworthiness value, based on how much the trustee's behaviour matches an expectation. Trustworthiness values are usually in the range $[0, 1]$, decreasing when the trustee exhibits deviations from the expected behaviour and increasing when the trustee's behaviour matches the expectation.

Trust-management can be usefully applied in WSNs to reduce the influence of the compromised sensor nodes that inject malicious data. Indeed, if the expected behaviour accurately characterises genuine nodes, compromised nodes would be assigned a low trustworthiness when deviating from it. Since trust values are a continuous metric defined inside an interval, there is no direct classification of compromised and genuine nodes. Instead, the trust values are used to apply a penalisation proportional to the confidence that the sensor is compromised. Note that the influence of the compromised nodes become negligible only when the confidence is sufficiently high. Filtering all the sensors with a trustworthiness under a given threshold [Sun et al. 2012], could help mitigate this drawback, but requires a method to set the appropriate threshold.

5.2.1. Event-based techniques. Trust-management for sensed data was originally introduced as a complement to network-level trust, i.e. how much nodes can be trusted to perform correctly network-level tasks [Ganeriwal et al. 2003; Raya et al. 2008; Momani et al. 2008] such as communicating routes, participating to the route discovery process, routing incoming packets etc. The behaviour with respect to each of these tasks can be of two kinds: cooperative and uncooperative.

The first examples of trust management for sensed data use a similar binary evaluation to build the trustworthiness, defined with respect to an event detection process. Initially, a *decision logic* establishes the presence of the event by combining the sensed data and the trust values). Then, the sensed data is compared to the final decision to measure the sensor's cooperativeness and update the trust values. This criterion is based on the assumption that nearby sensors are expected to agree about the event presence, which is a form of spatial correlation (see Sect. 4.4).

One of the first techniques to adopt this approach is described in [Atakli et al. 2008]. As shown in Fig. 6, the reading of a generic sensor i, $S_i(t)$, which can take the values 0 and 1 (absence/presence of an event), are relayed to a *forwarding node*. This node computes $\sum_{n=1}^{N} W_n S_n(t)$, where $W_{n\,:\,n \in 1 \ldots N}$ denote the trust weights.

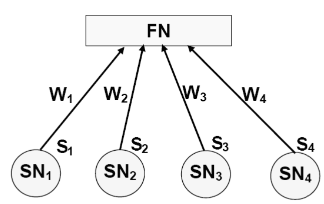

Fig. 6. Trust-weighted aggregation for event detection. FN is a forwarding node, which collects reports from the sensor nodes SN, from [Atakli et al. 2008].

The result is used to decide about the ground truth E. Afterwards, weights are updated with the following rule:

$$W_n = \begin{cases} W_n - \theta r_n, & \text{if } S_n(t) \neq E \\ W_n, & \text{otherwise} \end{cases} \tag{10}$$

where r_n is the ratio of sensors giving different output over the total number of sensors and θ is a penalty weight that determines a trade-off between the detection time and accuracy. In summary, the trustworthiness values, which coincide with the weights, are calculated based on the measurements consistency with the aggregated value. The latter is considered more reliable than the single readings, since sensors which exhibited inconsistent (e.g. malicious) readings in the past contribute less to the aggregation process. Finally, malicious nodes are detected by comparing the weights to a threshold, which the authors heuristically set to 0.4. Note that the algorithm is vulnerable to the *on-off attack*: a node that performs well for a time period, acquires high trustworthiness, then suddenly starts malfunctioning [Sun et al. 2006].

To counteract the on-off attack Oh et al. [2012] and Lim and Choi [2013] propose to penalise $S_n(t) \neq E$ by a quantity α and reward $S_n(t) = E$ by a quantity β with $\beta < \alpha$. As $\frac{\alpha}{\beta}$ grows bigger, faulty and malicious nodes are filtered out faster. However, sensors with transient faults are also filtered out, even though they may report correct measurements later on. To avoid this, the ratio $\frac{\alpha}{\beta}$ needs to also consider the probability of transient faults and their duration distribution. Therefore, this operation just reduces the frequency with which an attacker can switch between "good" and "bad" behaviour in an on-off attack.

When the sum of all trust weights is equal to 1, the weighted sum of sensors reading corresponds to a weighted mean. As described in the previous section, the mean has the drawback of being directly proportional to extreme readings. So in trust-based aggregation as well, the median could be used as a more robust aggregation operator. A trust weighted median has been applied by Wang et al. [2010] in the context of acoustic target localisation, where the median allows to filter out faulty measurements. The advantages of using the weighted median increase when an element with high weight has an extreme value. Indeed, while the weighted mean would be biased towards that value, the weighted median would still filter it out, if the other values are not extreme and the sum of their weights is bigger than the weight of the extreme value. This property reduces the efficacy of an on-off attack.

Another aspect to take into account is the uncertainty in the event's presence. Raya et al. [2008] deal with this aspect by using a decision logic based on Dempster-Shafer Theory (DST), which expresses the belief about the event presence as a combination of individual beliefs from sensor nodes. DST combines the sensors information supporting the event with the information non-refuting the event (the uncertainty margin which may comply with the event presence).

5.2.2. Anomaly-based techniques. Rather than analysing the compliance with the output of on an event decision logic, other trust-management techniques look for anomalous behaviours with techniques similar to anomaly detection ones.

In fact, the output of anomaly detection itself can be used to define a cooperative/uncooperative behaviour [Ganeriwal et al. 2003], but a more flexible approach, that does not restrict the observations to a binary value, is to update trust values based on an anomaly score. An example is given by Bankovic et al. [2010], using self-organizing maps (SOM). SOM are a clustering and data representation technique, that maps the data space to a discrete 2D neuron lattice. Bankovic et al. [2010] build two SOM lattices: one in the temporal domain and another in the spatial domain. The trust values are assigned based on two anomaly scores: the distance between the measurement and the SOM neuron and the distance between the neuron to which the measurement has been assigned and other SOM neurons. The main disadvantage of this algorithm is its computational time. For better accuracy, SOM require many neurons, but the computational time increases noticeably [McHugh 2000].

Another example is given by Zhang et al. [2006], who use a statistical-test approach (see Sect. 5.1.1) to assign reputation values to the sensors. The measurements gathered in time are assumed to approximately follow a normal distribution. The normal and actual measurements distribution are compared with the Kullback-Leibler divergence D_n, which evaluates the information lost when a probability distribution is used in lieu of another. The divergence is then used to update the trust values, with the following expression:

$$W_n = \frac{1}{1 + \sqrt{D_n}} \tag{11}$$

5.2.3. Using Second Hand Information. In the trust-management schemes previously analysed, each sensor's trust values are computed and updated by the device with the trustor role, typically a forwarding node. However, when the trustor is not in the transmission range of its trustee i, it may rely on information from its neighbours N_i to calculate its trustworthiness. Bao et al. [2012] deal with this problem by introducing two different trust update criteria:

$$T_{ij}(t) = \begin{cases} (1-\alpha)T_{ij}(t-\delta t) + \alpha T_{ij}(t) & \text{if } j \in N_i \\ avg_{k \in N_i}\{(1-\gamma)T_{kj}(t-\delta t) + \gamma T_{kj}(t)\} & \text{otherwise} \end{cases} \tag{12}$$

The calculations of the second case represent node j's *recommendation*, i.e. the trustworthiness extracted from relayed information. Eventually, recommendations depend on trustworthiness from the viewpoint of direct neighbours. However, such trustworthiness can be manipulated by malicious nodes to bad-mouth or good-mouth other nodes. Bao et al. [2012] mitigate this problem by controlling the impact of recommendations through parameter γ, set to $\frac{\beta T_{ik}(t)}{1+\beta T_{ik}(t)}$. Thus, if a sensor has little trust compared to the parameter β, the contribution of its recommendation will be small. However, sensors conducting an on-off attack can give false recommendations for a short while and then behave correctly again without being detected.

Even when direct information is available, recommendations can be used as second hand information and combined with direct information to obtain a *reputation*. Second hand information speeds up the convergence of trust values but adds network traffic overhead and introduces new problems, such as the weighting criterion for recommendations and the recommendation exchange frequency [Huang et al. 2006]. Ganeriwal et al. [2003] follow this approach and treat reputation as a probability distribution, updated as a combination of direct and indirect reputation. Direct reputation is updated based on a watchdog module, while indirect reputation is updated with recommendations, i.e. reputation from other nodes. The framework's scheme is shown in Fig.7. Note that such definition of reputation introduces a loop: indirect reputations come from reputations given by other sensors, which in turn depend on indirect reputations. To avoid the information loop, the recommendations need to be taken only from direct observers.

Modelling the reputation as a single value does not consider the uncertainty that a sensor has in trusting another sensor. This information is particularly useful with recommendations, as recommendations from sensors with high uncertainty should contribute less. To consider uncertainty, the reputation can be modelled with a probability distribution, whose choice is dictated mainly from the trust evaluation and update criteria. For example, Ganeriwal et al. [2003] use the *beta distribution* since it is the posterior distribution when the binary interactions between nodes are modelled with a binomial distribution. Momani et al. [2008] apply a normal distribution to model

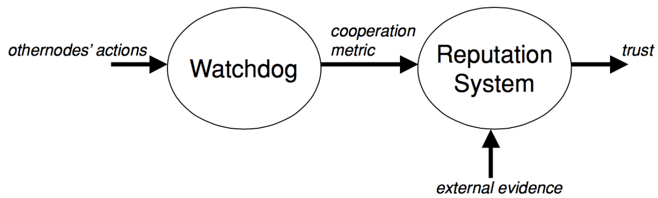

Fig. 7. Combination of direct information and recommendations, from [Ganeriwal et al. 2003].

the differences between the measurements of two sensors (spatial homogeneity is assumed, see Sect. 4.4).

6. DIAGNOSIS AND CHARACTERISATION OF MALICIOUS DATA INJECTIONS

Detecting the deviation of the measurements from the expected behaviour is usually not sufficient to infer the presence of a data injection attack. In the case of outlier detection, for example, we have seen that measurements are only classified as outlying or non-outlying, but malicious data injection is only one of the possible causes for outlying data. In general, regardless of the technique that detects the deviation from an expected behaviour, the cause for that deviation needs to be found. We refer to this task as *diagnosis*. Generally, it is not a trivial task, because different causes such as faults or genuine events may have similar effects.

Additionally, even when the presence of an attack can be ascertained with confidence, further information is needed to determine the course of action to be taken. For example, there is the need to know the attack's effects and the system area (nodes) affected by the attack. We refer to this other task as the *characterisation* of the attack.

In the following we analyse the state-of-the-art for diagnosis and characterisation of malicious data injections in WSNs.

6.1. Diagnosis

Diagnosis of malicious data injections in WSNs consists of distinguishing them from two main phenomena that can produce similar deviations from expected behaviour: faults and events of interest. Faults represent generic unintentional errors introduced e.g., by obstacles in the environment, sensors' battery depletion, pollution, fouling etc. Events of interests represent environmental conditions that seldom manifest, but are interesting as they can reveal an alarm scenario e.g., heart attacks, fires, volcanic eruptions etc.

Information about the cause of an anomaly or of an untrustworthy sensor can be precious. With fine-grained knowledge about the nature of the problem, an appropriate response can be initiated to address it. Unfortunately, in the papers analysed so far, an exhaustive diagnosis phase is still lacking. Most of the attention has focussed on diagnosing events as opposed to faults. The general assumption used to distinguish between them is that faults are likely to be stochastically unrelated, while event measurements are likely to be spatially correlated [Luo et al. 2006; Shahid et al. 2012]. Note that this assumption excludes common-mode failures from the analyses. Based on this assumption, after detecting deviations from expected data with temporal [Bettencourt et al. 2007; Shahid et al. 2012] or attribute [Shahid et al. 2012] correlations, it is possible to diagnose whether the deviation was caused by a fault or an event, by exploiting spatial correlation. When there is a consensus among a set of sensors about the presence of an event, discording sensors are considered faulty [Luo et al. 2006; Shahid et al. 2012; Bettencourt et al. 2007]. Similarly, some sensed attributes (e.g. human vital signs, such as glucose level, blood pressure, etc.) can be assumed heavily correlated in the absence of faults, which instead disrupt attribute correlations. Then, if we further assume that events would cause a minimum number of outlying attributes, faults can be identified when the minimum is not reached [Salem et al. 2013].

Fewer advances have been made towards diagnosing malicious interference as opposed to faults and events – we summarise them in the following sections.

6.1.1. Distinguishing Malicious Interference from Events. In the literature, malicious interference is distinguished from events through an agreement-based strategy [Liu et al. 2007; Atakli et al. 2008; Wang et al. 2010; Oh et al. 2012; Lim and Choi 2013; Sun et al. 2013], i.e. the sensor's information is first used to decide about the presence of an event and then sensors which did not support the final decision are identified as malicious. This approach is based on the assumption that sensors are sufficiently spatially correlated to correctly detect events. However, multiple compromised nodes can also collude in the attack to keep the spatial correlations consistent between themselves. This complicates discriminating between genuine events and malicious data injections, and allows an attacker to fabricate false events or to mask genuine ones. This aspect is discussed in more detail in Sect 6.2.

6.1.2. Distinguishing Malicious Interference from Faults. Criteria to distinguish malicious data injections from faults are less remarked. Two main approaches can be identified: delegating the diagnosis to intrusion-detection techniques and leveraging fault statistics.

Intrusion Detection. One of the main challenges in detecting attacks with anomaly-based techniques, is that such techniques abstract the means through which an attack is conducted. This choice comes from their objective to detect new attacks with unknown patterns, as opposed to intrusion detection techniques which are based on recognising known attack signatures. The framework proposed by Ngai et al. [2006] is a trade-off between an anomaly-detection technique and an intrusion detection system, since the detection is carried out through anomaly detection achieving high detection rate, while the diagnosis is carried out with intrusion detection. Clearly this approach provides diagnosis only for known attacks and cannot distinguish between an unknown attack and a fault.

Fault Statistics. The statistical characterisation of faults can also be used to distinguish them from malicious interference. Oh et al. [2012] and Lim and Choi [2013] use the expected frequency of transient faults to avoid excluding from the system sensors subject to transient faults. Indeed, their trust management algorithm allows such sensors to recover trustworthiness, by allowing temporary misbehaviour. Only sensors misbehaving with higher frequency, including malicious sensors and sensors with permanent faults will then be excluded.

6.2. Characterisation

If *detection* and *diagnosis* of malicious data injections answers the question "Is there an attack?", *characterisation* answers questions such as "Which are the compromised sensors?" and "How is the attack performed?" The difference is perhaps more evident in event-detection tasks. For example, *after* detecting the presence of an event, the event's spatial boundary can be characterised using the methodology proposed in [Wu et al. 2007], which finds the areas where the difference between the measurements from different sensors is high, indicating a discontinuity introduced by the event boundary. In this case, characterisation is triggered by detection, but is a separate task.

6.2.1. Collusion and its effects. In malicious data injections, detection, diagnosis and characterisation are often addressed simultaneously, since the information characterising the attack can be precious to improve the detection. In particular, when multiple sensors have been compromised and *collude* in the attack, they act in concert to change the measurements whilst evading, if possible, any anomaly detection applied. Therefore, identifying which sensors are more likely genuine and which sensors are more likely compromised becomes an integral part of detecting the attack itself.

In *collusion attacks* compromised sensors follow a joint strategy that reduces the advantages of spatial correlation, since the compromised nodes co-operate to form credible spatially correlated data [Tanachaiwiwat and Helmy 2005]. In the presence of collusion, diagnosis is also significantly more complex. Tanachaiwiwat and Helmy [2005] point out that when a genuine outlier (for example related to an event) occurs, extreme readings from the colluding nodes could be hidden. The problem becomes increasingly difficult as the percentage of (colluding) compromised sensors increases. Ultimately, when the number of colluding sensors increases to the point of exceeding genuine sensors, the attack may still be detectable, but it may be impossible to distinguish which nodes are genuine and which nodes are compromised. Tanachaiwiwat and Helmy [2005] evaluate their anomaly detection algorithm against colluding nodes and find that performance noticeably decreases when more than 30% nodes are colluding. A similar result is reported by Chatzigiannakis and Papavassiliou [2007].

Bertino et al. [2014] describe a new attack scenario applicable when the trustworthiness is calculated through an *iterative filtering* algorithm. While in generic (non-iterative) trust-evaluation techniques, trust weights are updated based on data from the current time instant and the weights calculated at the previous time instant, in iterative filtering the weights are iteratively updated with data of the same time instant until a convergence criterion is satisfied. In this context the authors introduce a new attack scenario where all colluding nodes but one, produce noticeable deviations in their readings. The remainder compromised node reports, instead, values close to the aggregated value of all the readings (including malicious ones). Eventually, this node acquires a high trust value, while all the others acquire low trust values. The aggregated value, in turn, quickly converges to a value far from that of the genuine nodes. The authors show that this attack is successful when the sensors are assigned equal initial trustworthiness. They therefore propose to calculate the initial trustworthiness as a function that decreases as the error variance increases. The error is defined as the distance from an estimated physical attribute value $\varphi(t)$, and is the same for all the sensors.

In [Rezvani et al. 2013] the same authors proposed another technique that detects collusion rather than counteracting it. This technique is based on the assumption that deviations from the aggregated values are normally distributed for genuine nodes. This assumption comes from the observation that the deviations of non-compromised nodes, even if large, come from a large number of independent factors, and thus must roughly have a Gaussian distribution. For colluding nodes instead, they assume that this condition does not hold. Then, by running the Kolmogorov-Smirnov test to check compliance to the normal distribution, they discriminate colluding nodes from genuine nodes.

In summary, while many studies propose propose new anomaly detection algorithms to cater for a broad range of scenarios, comparatively fewer address specifically malicious data injections in a way that can cater for more sophisticated attacks involving collusion between sensors. Such scenarios will need to be explored further in the future.

6.2.2. Characterisation Architectures: Centralised vs. Distributed. To detect, diagnose and characterise the nodes injecting malicious measurements, different architectures can be employed with different degrees of distribution. We discuss the properties of different solutions below.

In WSNs there is always at least one entity which eventually collects the measurements for the analyses, decisions and actions that the system needs to carry out: the base station. The base station is usually assumed free of compromise and therefore can be used to characterise the compromised nodes. In this case we have a centralised architecture such as in [Chatzigiannakis and Papavassiliou 2007; Atakli et al. 2008; Oh et al. 2012; Lim and Choi 2013; Rezvani et al. 2013].

Even when the base station is the only trusted entity in the network, distributed characterisation is possible. Indeed, as proposed in [Bao et al. 2012], the sensor nodes can be assessed in a hierarchical structure where each node assesses the trustworthiness of nodes below it in the hierarchy. The base station thus trusts nodes when a chain of trust can be established from that node to the base station.

When the distribution principle is taken to the extreme, each node acts as a watchdog for all its neighbours and reports alerts to the base station (or the next node in the hierarchy) [Ganeriwal et al. 2003; Tanachaiwiwat and Helmy 2005; Liu et al. 2007]. After all the reports are collected, a decision is taken based on algorithms such as majority voting [Hinds 2009]. The drawbacks of this approach are: that it lacks global knowledge and for this reason it is less robust to collusion attacks and that it introduces significant network overhead given by the watchdog reports. Tanachaiwiwat and Helmy [2005] propose to overcome these problems by deploying multiple reliable tamper-resistant sensor nodes that probe suspicious nodes. This solution, however, requires additional expensive hardware, which undermines the cost advantages of WSNs.

7. DISCUSSION

In the previous sections we have seen how different techniques can be applied to detect malicious data injections, how they leverage measurements' correlations and the assumptions on which such correlations are based. We have examined the different detection techniques and how they find deviations from the expected behaviour. We have highlighted the importance of distinguishing between different sources of deviations and presented the main directions of work towards this objective so far.

We now combine these analyses by building direct comparison tables, which summarise their main characteristics. A summary of the results reported by each of the techniques mentioned is provided in the following section.

7.1. Comparison of Approaches

We divide our comparison of the approaches analysed so far into Tables II and III, containing the anomaly detection and trust management techniques respectively. The content of the columns from left to right is: technique name and reference; correlation used to define expected data; assumptions about the spatial model if any; detection criterion used; possible sources of anomalies (as mentioned in their paper) and for which of them diagnosis criteria are given, e.g. {Event},{Malicious or Faulty} means that the authors give a criterion to discern between anomalies arising from events and from malicious or faulty sensors.

We observe that spatial correlation is most often exploited, and this under the frequent assumption of a homogeneous space. The situation is particularly evident for papers considering the presence of malicious data injections and probably a consequence of the fact that, generally, only a minor subset of sensors is assumed to be compromised. Therefore, in the spatial domain there is always a significant set of genuine measurements that can be exploited to detect the malicious ones.

Assuming spatial homogeneity makes the calculations significantly simpler, since the sensors are considered to measure the same value. However, it also significantly restricts the applicability of the techniques in real cases. When the physical phenomenon is observed with low precision, e.g. overall temperature across a large open space area, this assumption is still valid if the spatial variations are absorbed by the error term in Eq. 2. However, this allows an attacker to introduce malicious data that are within the error bounds yet still deviate significantly from the real values. While this assumption is generally appropriate in small areas, small areas also typically include fewer sensors which have higher risk of an attacker compromising them all.

When multiple types of correlation are considered, temporal correlations are generally exploited along spatial ones. Use of attribute correlations is rather infrequent, probably due to the fact that understanding them requires knowledge about their physical significance and this is application specific. The tables highlight even more the lack of diagnosis and characterisation (see Sect. 6.1). Few papers consider specifically malicious injections with collusion and even fewer papers deal with the problem of distinguishing them from other causes of deviations. While distinguishing events from faults is the diagnosis more frequently considered, distinguishing attacks from faults is undoubtedly more challenging and still rather rare.

7.2. Comparing Reported Evaluation Results

In the previous sections, we have considered techniques that could be applied to the problem of detecting, diagnosing and characterising malicious data injections. For those techniques that focus specifically on malicious data injections we now present the experimental evaluation set-up used by the authors and compare the reported results. None of these techniques has been tested on real attacks scenarios. This is not surprising as finding real attack data in existing WSN deployments is difficult. In fact, two approaches have been broadly adopted to evaluate the algorithms for detection of malicious data injections: *simulation* [Sun et al. 2013; Liu et al. 2007; Rezvani et al. 2013; Atakli et al. 2008; Bankovic et al. 2010; Oh et al. 2012; Bao et al. 2012; Lim and Choi 2013] and *injection of attacks* in real datasets [Tanachaiwiwat and Helmy 2005; Chatzigiannakis and Papavassiliou 2007].

Table IV summarises all the results achieved, together with all the relevant simulation parameters. The last three columns express the false positive rate (FPR) when the detection rate (DR) is respectively 0.90, 0.95 and 0.99. DR is, by definition, the number of attack instances that are correctly detected, divided by the total number of attack instances. FPR is, by definition, the number of times normal data instances are misclassified as attacks, divided by the total number of normal data instances. The relationship between DR and FPR is known as the Receiver Operating Characteristic (ROC). Column 2 reports information about the size of the dataset used in the experiments. Column 3 reports the percentage of either malicious nodes or malicious measurements. Column 4 reports the input size for the algorithm; for example in an experiment with 100 nodes, where the nodes are clustered in groups of 10 and the algorithm is run on clusters, the algorithm input size is 10.

Generally, in each paper, the tests are conducted in scenarios with different assumptions. For instance, Liu et al. [2007] generate data with a normal distribution for normal sensors and another normal distribution for malicious sensors. The results are excellent, but depend a lot on the difference between the two distributions. Another important assumption, which has noticeable impact on the results, is the spatial model. As pointed out in Sect. 4.4, most papers assume that the sensors' readings are homogeneous in the space; in other words the measurements are expected to be equal to each other, apart from noise and errors. The consequence of this assumption is that, by increasing the number of sensors, the information redundancy also increases and the

Table II. Anomaly detection techniques

Work	Correlation exploited	Spatial model	Detection method	Classes considered	Inter-class discrimination
EKF, CUSUM GLR [Sun et al. 2013]	Temporal	None	Change in the distribution of error from estimate	Event, Malicious, Faulty	{Event}, {Malicious or faulty}
MGDD [Subramaniam et al. 2006]	Temporal	None	Measurement probability	Event, Fault	None
[Ngai et al. 2006]	Spatial	Homogeneous	Difference with neighbours	Suspicious of Sinkhole attack	None
[Wu et al. 2007]	Spatial	Homogeneous	Difference with neighbours	Event	None
FIND [Guo et al. 2009]	Spatial	Monotonic WRT event source	Spatial monotonicity disruptions	Fault	None
[Salem et al. 2013]	Attribute-temporal	None	Energy of fluctuations	Event, Fault	{Event} {Faulty}
STIOD [Zhang et al. 2012]	Spatio-temporal	Variogram	Difference with estimate	Event, Error	{Event} {Error}
MAP+HBST [Ni and Pottie 2012]	Spatio-temporal	Linear spatial trend	Difference with estimate	Fault	None
[Liu et al. 2007]	Spatial	Homogeneous	Difference with neighbours	Malicious, Event	{Malicious}, {Event}
ART [Tanachaiwiwat and Helmy 2005]	Spatial	Homogeneous	Difference with neighbours	Compromised, Uncalibrated Sybil	{Compromised or Faulty}, {Uncalibrated}, {Sybil}
[Rajasegarar et al. 2007]	Spatio-temporal	Homogeneous	Values outside a quarter-sphere	None	None
STA-QS-SVM [Shahid et al. 2012]	Spatio-temporal and Spatio-attribute	Homogeneous	Values outside a quarter-sphere	None	None
[Chatzigiannakis and Papavassiliou 2007]	Spatial	High Pearson correlation	Changes in correlation	Fault, Malicious	{Point failure or malicious node}, {Group failure or Collusion}
[Bettencourt et al. 2007]	Spatio-temporal	Homogeneous	Distribution of temporal and spatial differences	Event, Fault	{Event}, {Point failure}
[Handschin et al. 1975]	Spatial	Linear combination of state variables	Difference with estimate	Fault	None
Robust IF [Rezvani et al. 2013]	Spatial	Homogeneous	Distribution of distance from estimation	Fault, Malicious	None

number of sensors taken into account is decisive. Recall from Sect. 4.4 that the sensing space can be approximately homogeneous only if we consider a small portion of space where there are no obstacles. In works like [Chatzigiannakis and Papavassiliou 2007] and [Bankovic et al. 2010], where this assumption is not present, the FPR is higher, but the algorithm has wider applicability. Tanachaiwiwat and Helmy [2005] rely on the spatial homogeneity assumption, and apply their technique to a large neighbourhood (100 nodes). The FPR is better but still not negligible (more than 20%). Atakli et al. [2008] also rely on this assumption and apply their algorithm on very large neighbourhoods. With 100 nodes the FPR for DR=0.90 is 3%, but for DR=95 and DR=99 the FPR increases by an order of magnitude. In contrast, [Oh et al. 2012; Bao et al. 2012; Lim and Choi 2013], are successful in keeping the FPR low even for high DR. Note that with a larger number of nodes the FPR of the technique described in [Atakli et al. 2008] increases. This result contrasts with the consideration that we made about the the spatial homogeneity assumption. The reason behind that, lies probably in the

Table III. Trust based detection techniques

Work	Correlation exploited	Spatial model	Detection method	Classes considered	Inter-class discrimination
[Zhang et al. 2006]	Spatio-temporal	Homogeneous	Distance from mean of top-trust sensors	Malicious	None
WTE [Atakli et al. 2008]	Spatial	Homogeneous	Trust under a threshold	Malicious	None
[Momani et al. 2008]	Spatial	Homogeneous	Trust under a threshold	Faulty, Malicious	None
[Wang et al. 2010]	Spatial	Homogeneous	Difference with aggregated value	Faulty, Event	{Faulty}, {Event}
[Bankovic et al. 2010]	Spatio-Temporal	Heterogeneous	Difference with learnt pattern	Malicious	None
Trust-based IDS [Bao et al. 2012]	Spatial	Homogeneous	Trust under a threshold	Malicious, Event	{Malicious}, {Event}
DWE [Oh et al. 2012]	Spatial	Homogeneous	Trust under a threshold	Malicious, Permanent Fault, Transient Fault, Event	{Malicious or Permanent Fault}, {Event}
Dual threshold [Lim and Choi 2013]	Spatial	Homogeneous	Trust under a threshold	Malicious, Permanent Fault, Transient Fault, Event	{Malicious or Permanent Fault}, {Event}

Table IV. Detection performances, independent attacks

Work	Dataset size	Dataset malicious percentage	Input size for each algorithm execution	FPR for DR=0.90	FPR for DR=0.95	FPR for DR=0.99
EKF [Sun et al. 2013]	10000 samples	50% samples, same node	6	0.22	0.42	0.7
[Liu et al. 2007]	4096 nodes	10-25% nodes	10	0.01	0.01	0.07
ART [Tanachaiwiwat and Helmy 2005]	100 nodes	30-50% samples, random selection of malicious nodes	100	0.25	0.22	0.21
[Chatzigiannakis and Papavassiliou 2007]	40 nodes	10% nodes	40	0.67	0.69	0.7
[Chatzigiannakis and Papavassiliou 2007]	40 nodes	40% nodes	40	0.48	0.5	0.6
WTE [Atakli et al. 2008]	100 nodes * 200 samples	0-25% nodes	100	0.03	0.41	0.78
WTE [Atakli et al. 2008]	400 nodes * 200 samples	0-25% nodes	400	0.10	0.44	0.78
[Bankovic et al. 2010]	2000 nodes * 2500 samples (1000 are used for training)	5% nodes	2000	0.5	0.5	0.5
Trust-based IDS [Bao et al. 2012]	900 nodes	N/A	N/A	0.001	0.05	N/A
DWE [Oh et al. 2012]	200 samples	20% nodes	20	0.01	0.01	0.02
Dual threshold [Lim and Choi 2013]	100 samples	10% nodes	12	N/A	N/A	0.001
Dual threshold [Lim and Choi 2013]	100 samples	20% nodes	12	0.18	0.14	0.10

Table V. Detection performances, colluding attacks

Work	Dataset size	Colluding percentage	Input size for each algorithm execution	FPR for DR=0.90	FPR for DR=0.95	FPR for DR=0.99
ART [Tanachaiwiwat and Helmy 2005]	100 nodes	30-50% samples	100	0.25	0.22	0.21
[Chatzigiannakis and Papavassiliou 2007]	40 nodes	10% nodes	40	0.67	0.69	0.7
[Chatzigiannakis and Papavassiliou 2007]	40 nodes	40% nodes	40	0.76	0.78	0.8
Robust IF [Rezvani et al. 2013]	20 nodes per 400 samples	40% nodes	20	N/A	0.021	0.021

inaccuracy of the empirical ROC curve calculation. Another possible cause is that the algorithm is sensitive to the absolute number of compromised nodes rather than to its ratio to total nodes. For example 80 out 400 compromised nodes may be harder to detect than 20 out of 100, event though the percentage of malicious nodes is 20% in both cases.

In Table V, we report the results for the cases considering collusion. The results reported in [Chatzigiannakis and Papavassiliou 2007] show non negligible FPR values (above 60%). The results reported in [Tanachaiwiwat and Helmy 2005] have a better FPR (around 20%). Rezvani et al. [2013] instead, achieve very good results (FPR less than 5%). Nevertheless, recall that this technique is applicable only when the the spatial homogeneity assumption among the 20 sensors is reasonable. In scenarios where the sensors readings cannot be assumed to share the same physical attribute function, the results may degrade substantially. This is the case for physical attributes like vibration, light, wind etc., where the correlation of the attribute measured at different locations rapidly decreases with the event propagation.

7.3. Comparing Techniques Overhead

The applicability of a technique to a real WSN does not only depend on the relationship between the detection rate and the false positive rate, but also on the overhead introduced. We analyse computational and communication overhead for the techniques discussed in the previous section, and summarise their asymptotic complexity in table VI. As usual, N is the number of sensors, while N_n is the average number of neighbours and W is the temporal memory, i.e. the number of past samples used.

From table VI, we note that anomaly detection techniques generally introduce more computational overhead than trust management techniques. The reason behind this result is that trust management iteratively refines its confidence about a sensor's trustworthiness, whereas anomaly detection builds such confidence from scratch at each iteration. On the other hand, this is also the main reason why trust-management algorithms are vulnerable to on-off attacks (see Sect. 5.2).

Another noticeable result is that communication overhead is always kept lower than computational overhead – this result is to be expected since network communication is more expensive in terms of energy and leads to faster battery depletion. In anomaly detection techniques the communication overhead comes from the execution of consensus-like protocols which decide about the maliciousness of nodes after anomalies are detected. Trust management techniques instead, delegate such decisions to the nodes that are higher in a WSN hierarchy (e.g. the forwarding nodes, cluster heads, base station). Thus communication overhead is introduced in trust management techniques only when recommendations are enabled (such as in [Bao et al. 2012]).

Table VI. Techniques overhead

Class	Work	Computational Overhead	Communication Overhead
Anomaly Detection	ART [Tanachaiwiwat and Helmy 2005]	$O(W * N_n)$	$O(1)$
	[Liu et al. 2007]	$O(N_n^2)$	$O(N_n)$
	[Chatzigiannakis and Papavassiliou 2007]	$O(WN_n^2 + N_n^3)$	0
	EKF [Sun et al. 2013]	$O(1)$	$O(N_n)$
	Robust IF [Rezvani et al. 2013]	$O(WN^2)$	0
Trust management	WTE [Atakli et al. 2008]	$O(N_n)$	0
	[Bankovic et al. 2010]	$O(N_n^2) + O(W^2)$	0
	Trust-based IDS [Bao et al. 2012]	$O(N_n)$	$O(N_n)$
	DWE [Oh et al. 2012]	$O(N_n)$	0
	Dual threshold [Lim and Choi 2013]	$O(N_n)$	0

8. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Malicious data injections are a considerable threat for WSNs. We reviewed state-of-the-art techniques that can detect malicious data injections by defining an expected behaviour and then detecting deviations from it. We classified these approaches into two main families: *anomaly detection* and *trust management*. They differ in the assessment of an anomalous condition, but both rely on the definition of an expected behaviour. We analysed and compared the techniques by their definition of expected behaviour and noted that expectations can come from correlations: *a*) *in time:* different time, same sensor, same attribute; *b*) *in space:* same time, different sensors, same attribute; *c*) *across different physical attributes:* same time, same sensor, different attributes; or *d*) their combination.

While many techniques can be applied, comparatively few target explicitly malicious data injections, especially when collusion between compromised sensors is considered. Most techniques aim to detect erroneous measurements, either to improve the quality of the measuring process (e.g. [Subramaniam et al. 2006; Bettencourt et al. 2007]), or to reduce the power associated with the transmission of the measurements (e.g. [Wang et al. 2010; Salem et al. 2013].

Work aimed detecting malicious data injections, generally uses spatial correlation in constructing the expectations (e.g. [Zhang et al. 2006; Liu et al. 2007; Chatzigiannakis and Papavassiliou 2007]) in keeping with a general assumption that only a subset of sensors has been compromised. In this case, a non-void set of genuine measurements is always present in the spatial domain.

We discussed the different assumptions that characterise the spatial domain, and analysed how they impact the performance of the detection algorithms. More precisely, we observed a substantial decrease in performance when moving away from a homogeneous space model, where all sensors perceive similar measurements, to heterogeneous space models, where different measurements are expected at different locations. This result is visible, for example, in the difference between the results achieved in [Tanachaiwiwat and Helmy 2005] and [Rezvani et al. 2013], who assume a homogeneous space, and those achieved by [Chatzigiannakis and Papavassiliou 2007], who only assume some degree of correlation between the sensors. The results, in the latter case, show noticeable higher false positive rates. We conclude that more research is needed to achieve better results when the spatial domain is heterogeneous. This will also improve the general applicability of the algorithms in real life deployments.

We explored different approaches to the detection phase, where the deviation from the expected behaviour is assessed, and noted a clear preference in the literature for outlier-detection techniques (e.g. [Ngai et al. 2006; Liu et al. 2007; Sun et al. 2013]). In this case, the expectation of a measurement is compliant with a generalisation of the measurements behaviour. This approach is independent from the context and is preferred to more context-specific techniques based on model checking (e.g. [Handschin et al. 1975]).

Finally, to complete the detection of malicious data injections, we identified two main aspects that need to be addressed: *diagnosis* and *characterisation*. These are, by and large, insufficiently studied in the literature.

Diagnosis consists of identifying the cause of the detected anomaly which, besides malicious data injections, may lie in faults or events of interest. Both these phenomena can produce deviations from expected behaviour similar to malicious injections. Whilst partial diagnosis is investigated in, e.g., [Tanachaiwiwat and Helmy 2005; Bettencourt et al. 2007; Chatzigiannakis and Papavassiliou 2007; Oh et al. 2012], an exhaustive diagnosis phase is still lacking. Fault-related anomalies may be handled separately from malicious data injections, as fault models are relatively well categorised and understood. However, event-related anomalies cannot be considered separately (like in [Liu et al. 2007]), since an attacker may inject malicious measurements that depict a fabricated event or conceal a real event. Therefore, in WSNs that monitor the occurrence of events, malicious injections and events should be addressed together, to produce a compromise-resistant detection and characterisation of events.

Similarly, further investigation of the *attack characterisation*, is needed, in particular to identify the compromised sensors in the presence of collusion. This aspect adds more complexity to the problem since colluding sensors can reduce data inconsistencies introduced in the attack, especially in the spatial domain.

Across all of the above, a good model of expected system behaviour plays a central role and determines both the applicability of the algorithms for detecting malicious data injections as well as their performance.

9. ACKNOWLEDGMENT

We wish to thank the anonymous reviewers for their comments and suggestions, which have made a valuable contribution to this paper. We are grateful to Dr. Igor Muttik from Intel and to the colleagues in our research group, who have contributed to this work through many useful discussions. This work was funded as part of the Intel Collaborative Research Institute on Sustainable Connected Cities.

REFERENCES

Idris M. Atakli, Hongbing Hu, Yu Chen, Wei-Shinn Ku, and Zhou Su. 2008. Malicious node detection in wireless sensor networks using weighted trust evaluation.. In *SpringSim*, Hassan Rajaei, Gabriel A. Wainer, and Michael J. Chinni (Eds.). SCS/ACM, 836–843. http://dblp.uni-trier.de/db/conf/springsim/springsim2008.html#AtakliHCKS08

Majid Bahrepour, Yang Zhang, Nirvana Meratnia, and Paul JM Havinga. 2009. Use of event detection approaches for outlier detection in wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on*. IEEE, 439–444.

Zorana Bankovic, José Manuel Moya, Álvaro Araujo, David Fraga, Juan Carlos Vallejo, and Juan-Mariano de Goyeneche. 2010. Distributed intrusion detection system for wireless sensor networks based on a reputation system coupled with kernel self-organizing maps. *Integrated Computer-Aided Engineering* 17, 2 (2010), 87–102. http://dblp.uni-trier.de/db/journals/icae/icae17.html#BankovicMAFVG10

Fenye Bao, Ing-Ray Chen, Moonjeong Chang, and Jin-Hee Cho. 2012. Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection. *IEEE Transactions on Network and Service Management* 9, 2 (2012), 169–183. http://dblp.uni-trier.de/db/journals/tnsm/tnsm9.html#BaoCCC12

Elisa Bertino, Aleksandar Ignatovic, and Sanjay Jha. 2014. Secure Data Aggregation Technique for Wireless Sensor Networks in the Presence of Collusion Attacks. *IEEE Transactions on Dependable and Secure Computing* 99, PrePrints (2014), 1. DOI:http://dx.doi.org/10.1109/TDSC.2014.2316816

Luís M. A. Bettencourt, Aric A. Hagberg, and Levi B. Larkey. 2007. Separating the Wheat from the Chaff: Practical Anomaly Detection Schemes in Ecological Applications of Distributed Sensor Networks.. In *DCOSS* (2007-07-09) *(Lecture Notes in Computer Science)*, James Aspnes, Christian Scheideler, Anish Arora, and Samuel Madden (Eds.), Vol. 4549. Springer, 223–239. http://dblp.uni-trier.de/db/conf/dcoss/dcoss2007.html#BettencourtHL07

Azzedine Boukerche. 2009. *Algorithms and Protocols for Wireless Sensor Networks.* Wiley-IEEE Press.

Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. 2008. Secure localization algorithms for wireless sensor networks. *Communications Magazine, IEEE* 46, 4 (2008), 96–101.

Levente Buttyan and Jean-Pierre Hubaux. 2008. *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. Cambridge University Press.

Claude Castelluccia, Aurélien Francillon, Daniele Perito, and Claudio Soriente. 2009. On the difficulty of software-based attestation of embedded devices.. In *ACM Conference on Computer and Communications Security* (2009-11-25), Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis (Eds.). ACM, 400–409. http://dblp.uni-trier.de/db/conf/ccs/ccs2009.html#CastellucciaFPS09

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 3 (2009). http://dblp.uni-trier.de/db/journals/csur/csur41.html#ChandolaBK09

Vassilis Chatzigiannakis and Symeon Papavassiliou. 2007. Diagnosing anomalies and identifying faulty nodes in sensor networks. *Sensors Journal, IEEE* 7, 5 (2007), 637–645.

Ethan W. Dereszynski and Thomas G. Dietterich. 2011. Spatiotemporal Models for Data-Anomaly Detection in Dynamic Environmental Monitoring Campaigns. *TOSN* 8, 1 (2011), 3. http://dblp.uni-trier.de/db/journals/tosn/tosn8.html#DereszynskiD11

Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod K. Varshney, Jonathan Katz, and Aram Khalili. 2005. A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans. Inf. Syst. Secur.* 8, 2 (2005), 228–258. http://dblp.uni-trier.de/db/journals/tissec/tissec8.#DuDHVKK05

Elena Fasolo, Michele Rossi, Jörg Widmer, and Michele Zorzi. 2007. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Commun.* 14, 2 (2007), 70–87. http://dblp.uni-trier.de/db/journals/wc/wc14.html#FasoloRWZ07

Saurabh Ganeriwal, Laura Balzano, and Mani B. Srivastava. 2003. Reputation-based framework for high integrity sensor networks. *TOSN* 4, 3 (2003). http://dblp.uni-trier.de/db/journals/tosn/tosn4.html#GaneriwalBS03

Saurabh Ganeriwal and Mani B. Srivastava. 2004. Reputation-based framework for high integrity sensor networks.. In *SASN* (2006-02-15), Sanjeev Setia and Vipin Swarup (Eds.). ACM, 66–77. http://dblp.uni-trier.de/db/conf/sasn/sasn2004.html#GaneriwalS04

Shuo Guo, Ziguo Zhong, and Tian He. 2009. FIND: faulty node detection for wireless sensor networks.. In *SenSys* (2009-11-18), David E. Culler, Jie Liu, and Matt Welsh (Eds.). ACM, 253–266. http://dblp.uni-trier.de/db/conf/sensys/sensys2009.html#GuoZH09

Edmund Handschin, Fred C. Schweppe, Jurg Kohlas, and Armin Fiechter. 1975. Bad data analysis for power system state estimation. *Power Apparatus and Systems, IEEE Transactions on* 94, 2 (1975), 329–337.

Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. 2000. Energy-Efficient Communication Protocol for Wireless Microsensor Networks.. In *HICSS* (2002-01-03). http://dblp.uni-trier.de/db/conf/hicss/hicss2000-8.html#HeinzelmanCB00

Cheryl V. Hinds. 2009. Efficient detection of compromised nodes in a wireless sensor network.. In *SpringSim*, Gabriel A. Wainer, Clifford A. Shaffer, Robert M. McGraw, and Michael J. Chinni (Eds.). SCS/ACM. http://dblp.uni-trier.de/db/conf/springsim/springsim2009.html#Hinds09

Lei Huang, Lei Li, and Qiang Tan. 2006. Behavior-Based Trust in Wireless Sensor Network.. In *APWeb Workshops* (2005-12-23) *(Lecture Notes in Computer Science)*, Heng Tao Shen, Jinbao Li, Minglu Li, Jun Ni, and Wei Wang (Eds.), Vol. 3842. Springer, 214–223. http://dblp.uni-trier.de/db/conf/apweb/apweb2006w.html#HuangLT06

J. Edward Jackson and Govind S. Mudholkar. 1979. Control Procedures for Residuals Associated with Principal Component Analysis. *Technometrics* 21, 3 (1979), pp. 341–349. http://www.jstor.org/stable/1267757

Dharanipragada Janakiram, Vanteddu Adi Mallikarjuna Reddy, and A.V.U. Phani Kumar. 2006. Outlier Detection in Wireless Sensor Networks using Bayesian Belief Networks. In *Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on*. 1–6. DOI:http://dx.doi.org/10.1109/COMSWA.2006.1665221

Raja Jurdak, X.Rosalind Wang, Oliver Obst, and Philip Valencia. 2011. Wireless Sensor Network Anomalies: Diagnosis and Detection Strategies. In *Intelligence-Based Systems Engineering*, Andreas Tolk and LakhmiC. Jain (Eds.). Intelligent Systems Reference Library, Vol. 10. Springer Berlin Heidelberg, 309–325. DOI:http://dx.doi.org/10.1007/978-3-642-17931-0_12

Rudolph Emil Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering* 82, Series D (1960), 35–45.

Chris Karlof and David Wagner. 2003. Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks* 1, 2-3 (2003), 293–315. http://dblp.uni-trier.de/db/journals/adhoc/adhoc1.#KarlofW03

Muhammad Khurram Khan and Khaled Alghathbar. 2010. Cryptanalysis and Security Improvements of 'Two-Factor User Authentication in Wireless Sensor Networks'. *Sensors* 10, 3 (2010), 2450–2459. DOI:http://dx.doi.org/10.3390/s100302450

Pavel Laskov, Christin Schäfer, Igor V. Kotenko, and Klaus-Robert Müller. 2004. Intrusion Detection in Unlabeled Data with Quarter-sphere Support Vector Machines. *Praxis der Informationsverarbeitung und Kommunikation* 27, 4 (2004), 228–236. http://dblp.uni-trier.de/db/journals/pik/pik27.html#LaskovSKM04

Sung Yul Lim and Yoon-Hwa Choi. 2013. Malicious Node Detection Using a Dual Threshold in Wireless Sensor Networks. *J. Sensor and Actuator Networks* 2, 1 (2013), 70–84. http://dblp.uni-trier.de/db/journals/jsan/jsan2.html#LimC13

An Liu and Peng Ning. 2008. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks.. In *IPSN* (2008-05-07). IEEE Computer Society, 245–256. http://dblp.uni-trier.de/db/conf/ipsn/ipsn2008.html#LiuN08

Fang Liu, Xiuzhen Cheng, and Dechang Chen. 2007. Insider Attacker Detection in Wireless Sensor Networks.. In *INFOCOM* (2007-06-27). IEEE, 1937–1945. http://dblp.uni-trier.de/db/conf/infocom/infocom2007.html#LiuCC07

Javier Lopez, Rodrigo Roman, Isaac Agudo, and M. Carmen Fernández Gago. 2010. Trust management systems for wireless sensor networks: Best practices. *Computer Communications* 33, 9 (2010), 1086–1093. http://dblp.uni-trier.de/db/journals/comcom/comcom33.#LopezRAG10

Xuanwen Luo, Ming Dong, and Yinlun Huang. 2006. On Distributed Fault-Tolerant Detection in Wireless Sensor Networks. *IEEE Trans. Computers* 55, 1 (2006), 58–70. http://dblp.uni-trier.de/db/journals/tc/tc55.html#LuoDH06

Stephen Marsland. 2009. *Machine Learning - An Algorithmic Perspective.* CRC Press. I–XVI, 1–390 pages.

John McHugh. 2000. Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Trans. Inf. Syst. Secur.* 3, 4 (2000), 262–294. http://dblp.uni-trier.de/db/journals/tissec/tissec3.#McHugh00

Mohammad Momani, Subhash Challa, and Rami Alhmouz. 2008. Can we trust trusted nodes in wireless sensor networks?. In *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*. IEEE, 1227–1232.

Edith CH Ngai, Jiangchuan Liu, and Michael R Lyu. 2006. On the intruder detection for sinkhole attack in wireless sensor networks. In *Communications, 2006. ICC'06. IEEE International Conference on*, Vol. 8. IEEE, 3383–3389.

Kevin Ni and Gregory J. Pottie. 2012. Sensor network data fault detection with maximum a posteriori selection and bayesian modeling. *TOSN* 8, 3 (2012), 23. http://dblp.uni-trier.de/db/journals/tosn/tosn8.html#NiP12

Seo Hyun Oh, Chan O. Hong, and Yoon-Hwa Choi. 2012. A Malicious and Malfunctioning Node Detection Scheme for Wireless Sensor Networks. *Wireless Sensor Network* 4, 3 (2012), 84–90.

Suat Özdemir and Yang Xiao. 2009. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Computer Networks* 53, 12 (2009), 2022–2037. http://dblp.uni-trier.de/db/journals/cn/cn53.#OzdemirX09

Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. 2003. LOCI: Fast Outlier Detection Using the Local Correlation Integral.. In *ICDE*, Umeshwar Dayal, Krithi Ramamritham, and T. M. Vijayaraman (Eds.). IEEE Computer Society, 315–326. http://dblp.uni-trier.de/db/conf/icde/icde2003.html#PapadimitriouKGF03

Lilia Paradis and Qi Han. 2007. A Survey of Fault Management in Wireless Sensor Networks. *J. Network Syst. Manage.* 15, 2 (2007), 171–190. http://dblp.uni-trier.de/db/journals/jnsm/jnsm15.html#ParadisH07

Taejoon Park and Kang G. Shin. 2005. Soft Tamper-Proofing via Program Integrity Verification in Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* 4, 3 (2005), 297–309. http://dblp.uni-trier.de/db/journals/tmc/tmc4.html#ParkS05

Adrian Perrig, John Stankovic, and David Wagner. 2004. Security in wireless sensor networks. *Commun. ACM* 47 (June 2004), 53–57. Issue 6. DOI:http://dx.doi.org/10.1145/990680.990707

Bartosz Przydatek, Dawn Xiaodong Song, and Adrian Perrig. 2003. SIA: secure information aggregation in sensor networks.. In *SenSys* (2006-02-15), Ian F. Akyildiz, Deborah Estrin, David E. Culler, and Mani B. Srivastava (Eds.). ACM, 255–265. http://dblp.uni-trier.de/db/conf/sensys/sensys2003.html#PrzydatekSP03

Sutharshan Rajasegarar, James C. Bezdek, Christopher Leckie, and Marimuthu Palaniswami. 2009. Elliptical anomalies in wireless sensor networks. *TOSN* 6, 1 (2009). http://dblp.uni-trier.de/db/journals/tosn/tosn6.html#RajasegararBLP09

Sutharshan Rajasegarar, Christopher Leckie, and Marimuthu Palaniswami. 2008. Anomaly detection in wireless sensor networks. *IEEE Wireless Commun.* 15, 4 (2008), 34–40. http://dblp.uni-trier.de/db/journals/wc/wc15.html#RajasegararLP08

Sutharshan Rajasegarar, Christopher Leckie, Marimuthu Palaniswami, and James C Bezdek. 2006. Distributed anomaly detection in wireless sensor networks. In *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on*. IEEE, 1–5.

Sutharshan Rajasegarar, Christopher Leckie, Marimuthu Palaniswami, and James C. Bezdek. 2007. Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks.. In *ICC* (2009-04-15). IEEE, 3864–3869. http://dblp.uni-trier.de/db/conf/icc/icc2007.html#RajasegararLPB07

Murad A. Rassam, Anazida Zainal, and Mohd Aizaini Maarof. 2013. Advancements of Data Anomaly Detection Research in Wireless Sensor Networks: A Survey and Open Issues. *Sensors* 13, 8 (2013), 10087–10122. DOI:http://dx.doi.org/10.3390/s130810087

Maxim Raya, Panos Papadimitratos, Virgil D Gligor, and Jean-Pierre Hubaux. 2008. On data-centric trust establishment in ephemeral ad hoc networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE.

Mohsen Rezvani, Aleksandar Ignjatovic, Elisa Bertino, and Sanjay Jha. 2013. A robust iterative filtering technique for wireless sensor networks in the presence of malicious attacks.. In *SenSys*, Chiara Petrioli, Landon P. Cox, and Kamin Whitehouse (Eds.). ACM, 30. http://dblp.uni-trier.de/db/conf/sensys/sensys2013.html#RezvaniIBJ13

John A. Rice. 2007. *Mathematical Statistics and Data Analysis* (3rd ed.). Duxbury Press, Belmont, CA.

Sandip Roy, Marco Conti, Sanjeev Setia, and Sushil Jajodia. 2014. Secure data aggregation in wireless sensor networks: Filtering out the attacker's impact. *Information Forensics and Security, IEEE Transactions on* 9, 4 (2014), 681–694.

Osman Salem, Yaning Liu, and Ahmed Mehaoua. 2013. A lightweight anomaly detection framework for medical wireless sensor networks.. In *WCNC*. IEEE, 4358–4363. http://dblp.uni-trier.de/db/conf/wcnc/wcnc2013.html#SalemLM13

Yingpeng Sang, Hong Shen, Yasushi Inoguchi, Yasuo Tan, and Naixue Xiong. 2006. Secure Data Aggregation in Wireless Sensor Networks: A Survey.. In *PDCAT* (2007-01-15). IEEE Computer Society, 315–320. http://dblp.uni-trier.de/db/conf/pdcat/pdcat2006.html#SangSITX06

Arvind Seshadri, Mark Luk, Adrian Perrig, Leendert van Doorn, and Pradeep K. Khosla. 2006. SCUBA: Secure Code Update By Attestation in sensor networks.. In *Workshop on Wireless Security* (2008-02-07), Radha Poovendran and Ari Juels (Eds.). ACM, 85–94. http://dblp.uni-trier.de/db/conf/ws/ws2006.html#SeshadriLPDK06

Arvind Seshadri, Adrian Perrig, Leendert van Doorn, and Pradeep K. Khosla. 2004. SWATT: SoftWare-based ATTestation for Embedded Devices.. In *IEEE Symposium on Security and Privacy* (2004-06-21). IEEE Computer Society, 272–. http://dblp.uni-trier.de/db/conf/sp/sp2004.html#SeshadriPDK04

Nauman Shahid, Ijaz Haider Naqvi, and Saad B. Qaisar. 2012. Quarter-Sphere SVM: Attribute and Spatio-Temporal correlations based Outlier & Event Detection in wireless sensor networks.. In *WCNC*. IEEE, 2048–2053. http://dblp.uni-trier.de/db/conf/wcnc/wcnc2012.html#ShahidNQ12

Abhishek B. Sharma, Leana Golubchik, and Ramesh Govindan. 2010. Sensor faults: Detection methods and prevalence in real-world datasets. *TOSN* 6, 3 (2010). http://dblp.uni-trier.de/db/journals/tosn/tosn6.html#SharmaGG10

Shigen Shen, Guangxue Yue, Qiying Cao, and Fei Yu. 2011. A Survey of Game Theory in Wireless Sensor Networks Security. *JNW* 6, 3 (2011), 521–532. http://dblp.uni-trier.de/db/journals/jnw/jnw6.html#ShenYCY11

Timothy J. Shepard. 1996. A Channel Access Scheme for Large Dense Packet Radio Networks.. In *SIGCOMM* (2002-12-09). 219–230. http://dblp.uni-trier.de/db/conf/sigcomm/sigcomm1996.html#Shepard96

Suresh Singh, Mike Woo, and C. S. Raghavendra. 1998. Power-Aware Routing in Mobile Ad Hoc Networks.. In *MOBICOM*, William P. Osborne and Dhawal B. Moghe (Eds.). ACM, 181–190. http://dblp.uni-trier.de/db/conf/mobicom/mobicom1998.html#SinghWR98

Sharmila Subramaniam, Themis Palpanas, Dimitris Papadopoulos, Vana Kalogeraki, and Dimitrios Gunopulos. 2006. Online Outlier Detection in Sensor Data Using Non-Parametric Models.. In *VLDB* (2006-09-27), Umeshwar Dayal, Kyu-Young Whang, David B. Lomet, Gustavo Alonso, Guy M. Lohman, Martin L. Kersten, Sang Kyun Cha, and Young-Kuk Kim (Eds.). ACM, 187–198. http://dblp.uni-trier.de/db/conf/vldb/vldb2006.html#SubramaniamPPKG06

Bo Sun, Xuemei Shan, Kui Wu, and Yang Xiao. 2013. Anomaly Detection Based Secure In-Network Aggregation for Wireless Sensor Networks. *IEEE Systems Journal* 7, 1 (2013), 13–25. http://dblp.uni-trier.de/db/journals/sj/sj7.html#SunSWX13

Yan Sun, Hong Luo, and Sajal K. Das. 2012. A Trust-Based Framework for Fault-Tolerant Data Aggregation in Wireless Multimedia Sensor Networks. *IEEE Trans. Dependable Sec. Comput.* 9, 6 (2012), 785–797. http://dblp.uni-trier.de/db/journals/tdsc/tdsc9.html#SunLD12

Yan Lindsay Sun, Zhu Han, Wei Yu, and K. J. Ray Liu. 2006. A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks.. In *INFOCOM* (2007-12-06). IEEE. http://dblp.uni-trier.de/db/conf/infocom/infocom2006.html#SunHYL06

Sapon Tanachaiwiwat and Ahmed Helmy. 2005. Correlation Analysis for Alleviating Effects of Inserted Data in Wireless Sensor Networks.. In *MobiQuitous* (2006-01-25). IEEE Computer Society, 97–108. http://dblp.uni-trier.de/db/conf/mobiquitous/mobiquitous2005.html#TanachaiwiwatH05

Xue Wang, Liang Ding, and Daowei Bi. 2010. Reputation-Enabled Self-Modification for Target Sensing in Wireless Sensor Networks. *IEEE T. Instrumentation and Measurement* 59, 1 (2010), 171–179. http://dblp.uni-trier.de/db/journals/tim/tim59.html#WangDB10

Weili Wu, Xiuzhen Cheng, Min Ding 0001, Kai Xing, Fang Liu, and Ping Deng. 2007. Localized Outlying and Boundary Data Detection in Sensor Networks. *IEEE Trans. Knowl. Data Eng.* 19, 8 (2007), 1145–1157. http://dblp.uni-trier.de/db/journals/tkde/tkde19.html#WuCDXLD07

Miao Xie, Song Han, Biming Tian, and Sazia Parvin. 2011. Anomaly detection in wireless sensor networks: A survey. *J. Network and Computer Applications* 34, 4 (2011), 1302–1325. http://dblp.uni-trier.de/db/journals/jnca/jnca34.html#XieHTP11

Yi Yang, Xinran Wang, Sencun Zhu, and Guohong Cao. 2006. SDAP: : a secure hop-by-Hop data aggregation protocol for sensor networks.. In *MobiHoc* (2006-12-22), Sergio Palazzo, Marco Conti, and Raghupathy Sivakumar (Eds.). ACM, 356–367. http://dblp.uni-trier.de/db/conf/mobihoc/mobihoc2006.html#YangWZ06

Yanli Yu, Keqiu Li, Wanlei Zhou, and Ping Li. 2012. Trust Mechanisms in Wireless Sensor Networks: Attack Analysis and Countermeasures. *J. Netw. Comput. Appl.* 35, 3 (May 2012), 867–880. DOI:http://dx.doi.org/10.1016/j.jnca.2011.03.005

Theodore Zahariadis, Helen-Catherine Leligou, Panagiotis Trakadas, and Stamatis Voliotis. 2010a. Mobile Networks Trust management in wireless sensor networks. *European Transactions on Telecommunications* 21, 4 (2010), 386–395. http://dblp.uni-trier.de/db/journals/ett/ett21.html#ZahariadisLTV10

Theodore Zahariadis, Helen C. Leligou, Panagiotis Trakadas, and Stamatis Voliotis. 2010b. Trust management in wireless sensor networks. *European Transactions on Telecommunications* 21, 4 (2010), 386–395. DOI:http://dx.doi.org/10.1002/ett.1413

Dazhi Zhang and Donggang Liu. 2010. DataGuard: Dynamic data attestation in wireless sensor networks.. In *DSN*. IEEE, 261–270. http://dblp.uni-trier.de/db/conf/dsn/dsn2010.html#ZhangL10

Wei Zhang, Sajal K. Das, and Yonghe Liu. 2006. A Trust Based Framework for Secure Data Aggregation in Wireless Sensor Networks.. In *SECON*. IEEE, 60–69. http://dblp.uni-trier.de/db/conf/secon/secon2006.html#ZhangDL06

Yang Zhang, Nicholas AS Hamm, Nirvana Meratnia, Alfred Stein, M van de Voort, and Paul JM Havinga. 2012. Statistics-based outlier detection for wireless sensor networks. *International Journal of Geographical Information Science* 26, 8 (2012), 1373–1392.