CHAPTER 1

# Introduction

This aim of this introductory chapter is to motivate the extensive research work carried in this book, highlighting the existing solutions and their limitations, and putting in context the innovative work and ideas described in this book.

## 1.1 OVERVIEW

**Supervisory Control and Data Acquisition** (SCADA) systems have been integrated to control and monitor industrial processes and our daily critical infrastructures such as electric power generation, water distribution and waste water collection systems. This integration adds valuable input to improve the safety of the process and the personnel and to reduce operation costs (Boyer, 2009). However, any disruption to SCADA systems can result in financial disasters or may lead to loss of life in a worst case scenario. Therefore, in the past, such systems were secure by virtue of their isolation and only proprietary hardware and software were used to operate these systems. In other words, these systems were self-contained and totally isolated from the public network (e.g., the Internet). This isolation created the myth that malicious intrusions and attacks from the outside world were not a big concern and that such attacks were expected to come from the inside. Therefore, when developing SCADA protocols, the security of the information system was given no consideration.

In recent years, SCADA systems have begun to shift away from using proprietary and customized hardware and software to using Commercial-Off-The-Shelf (COTS) solutions. This shift has increased their connectivity to the public networks using standard protocols (e.g., TCP/IP). In addition, there is decreased reliance on a single vendor. Undoubtedly, this increases productivity and profitability but will, however, expose these systems to cyber threats (Oman et al., 2000). According to a survey published by the SANS Institute (Bird and Kim, 2012), only 14% of organizations carry out security reviews of COTS applications that are being used, while over 50% of other organizations do not perform security assessments and rely only on vendor reputation or the legal liability agreements, or they have no policies at all regarding the use of COTS solutions.

The adoption of COTS solutions is a time- and cost-efficient means of building SCADA systems. In addition, COST-based devices are intended to operate on

*SCADA Security: Machine Learning Concepts for Intrusion Detection and Prevention*,
First Edition. Abdulmohsen Almalawi, Zahir Tari, Adil Fahad and Xun Yi.

traditional Ethernet networks and the TCP/IP stack. This feature allows devices from various vendors to communicate with each other, and also helps to remotely supervise and control critical industrial systems from any place and at any time using the Internet. Moreover, wireless technologies can efficiently be used to provide mobility and local control for multivendor devices at a low cost for installation and maintenance. However, the convergence of state-of-the-art communication technologies exposes SCADA systems to all the inherent vulnerabilities of these technologies. In what follows, we discuss how the potential cyber-attacks against traditional IT can also be possible against SCADA systems.

- **Denial of Services (DoS) attacks.** This is a potential attack on any Internet-connected device where a large number of spurious packets are sent to a victim in order to consume excessive amounts of endpoint network bandwidth. A packet flooding attack (Houle et al., 2001) is often used as another term for a DoS attack. This type of attack delays or totally prevents the victim from receiving the legitimate packets (Householder et al., 2001). SCADA networking devices that are exposed to the Internet such as routers, gateways and firewalls are susceptible to this type of attack. Long et al. (2005) proposed two models of DoS attacks on a SCADA network using reliable simulation. The first model was directly launched to an endpoint (e.g., controller or a customer-edge router connecting to the Internet), while the second model is an indirect attack, where the DoS attack is launched on a router (on the Internet) that is located in the path between the plant and endpoint. In this study, it was found that DoS attacks that were launched directly (or indirectly) cause excessive packet losses. Consequently, a controller that receives the measurement and control data late or not at all from the devices deployed in the field will make a decision based on old data.
- **Propagation of malicious codes.** Such types of attack can occur in various forms such as viruses, Trojan horses, and worms. They are potential threats to SCADA systems that are directly (or indirectly) connected to the Internet. Unlike worms, viruses and Trojans require a human action to be initiated. However, all these threats are highly likely as long as the personnel are connected to the Internet through the corporate network, which is directly connected to the SCADA system, or if they are allowed to plug their personal USBs into the corporate workstations. Therefore, a user can be deceived into downloading a contaminated file containing a virus or installing software that appears to be useful. Shamoon (Bronk and Tikk-Ringas, 2013), Stuxnet (Falliere et al., 2011), Duqu (Bencsáth et al., 2012), and Flame (Munro, 2012) are examples of such threats targeting SCADA systems and oil and energy sectors.
- **Inside threats.** The employees who are disgruntled or intend to divulge valuable information for malicious reasons can pose real threats and risks that should be taken seriously. This is because employees usually have unrestricted access to the SCADA systems and also know the configuration settings of these systems. For instance, the attack on the sewage treatment system in Maroochy Shire, South-East Queensland (Australia) in 2001 (Slay and Miller, 2007) is an

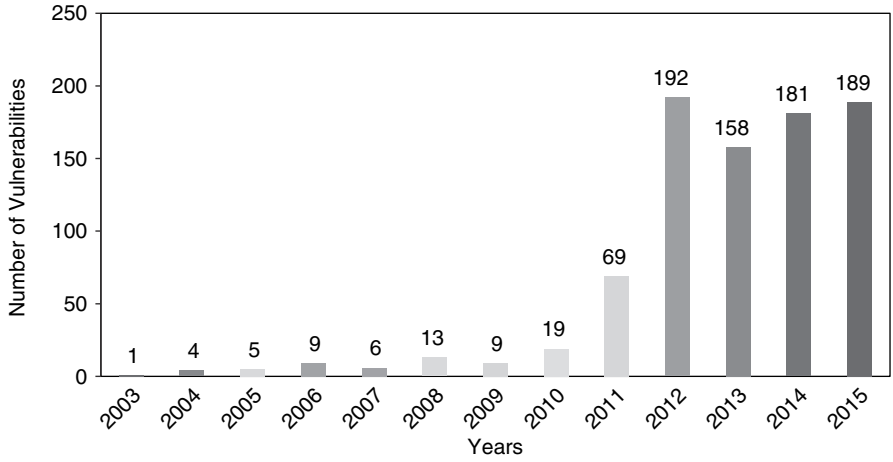


**Figure 1.1** SCADA vulnerabilities revealed since 2001 in OSVDB.

example of an attack that was launched by a disgruntled employee, where the attacker took over the control devices of a SCADA system and caused 800,000 litres of raw sewage to spill out into local parks and rivers.

- **Unpatched vulnerabilities.** The existence of vulnerabilities is highly expected in any system and it is known that hackers always exploit unpatched vulnerabilities to obtain access and to control the targeted system. Even though the vendors immediately release the patches for the identified vulnerabilities, it is challenging to install these patches on SCADA systems that run twenty-four-by-seven. Therefore, such systems will remain vulnerable for weeks or months. As depicted in Figure 1.1, and according to the independent and Open Source Vulnerability DataBase (OSVDB)[1] for the security community, vulnerabilities targeting SCADA systems have substantially increased over the past three years since 2011.
- **Nontechnical (social engineering) attacks.** This type of attack can bypass state-of-the-art security technologies that cost millions of dollars. In general, the attackers initially try to obtain sensitive information such as the design, operations, or security controls of the targeted SCADA system. There are a number of ways to gather such information. If the network access credentials of ex-employees are not immediately disabled, they can be revealed to another party in order to profit from the information, or as a desire for revenge. In another way, such critical information can be easily obtained from current employees as long as they are known by building a trust relationship or by knowing some information about a naive employee who is allowed to remotely control and monitor the systems via the Internet, all of which can help the attacker to answer the expected questions when calling up the central office

[1] http://osvdb.org/

to tell them that s/he forgot the network access credentials and assistance is needed to connect to the field network.

The security concepts that have been extensively used in traditional IT systems (e.g., *management*, *filtering*, *encryption*, and *intrusion detection*) can be adapted to mitigate the risk of the aforementioned potential threats against SCADA systems. However, these concepts cannot be directly applied without considering the *nature* of SCADA systems. For instance, the resource constraints of SCADA devices, such as low bandwidth, processing power, and memory, complicate the integration of complex cryptography, especially with legacy devices. All the SCADA protocols were developed without any consideration given to information security and, therefore, they lack authentication and integrity. Two solutions to secure the SCADA communications are: placing the cryptographic technologies at each end of the communication medium (American Gas Association (AGA), 2006; Tsang and Smith, 2008), or directly integrating them into the protocol, such as a secure DNP3 that protects the communication between master stations and outstations such as PLCs, RTUs, and IEDs (Majdalawieh et al., 2006).

Apart from the efforts to authenticate and encrypt SCADA communication links, it is still an ***open research challenge*** to secure the tens of SCADA protocols that are being used or to develop security modules to protect the communication link between two parties. AGA (American Gas Association (AGA), 2006) highlighted the challenges in building security modules that can be broadly summarized into two points: (i) the additional latency can be introduced by a secure protocol and (ii) the sophisticated key management system requires high bandwidth and additional communication channels that SCADA communication links are lacking.

Similarly, the traffic filtering process between a SCADA network and a corporate network using firewalls is a considerable countermeasure to mitigate the potential threats. However, although modern firewalls are efficient for analysing traditional IT traffic, they are incapable of in-depth analysis of the SCADA protocols. To design firewalls tailored to SCADA systems, the UK governments National Infrastructure Security Co-ordination Center (NISCC) published its guidelines for the appropriate use of firewalls in SCADA networks (Byres et al., 2005). It was proposed that a microfirewall should be embedded within each SCADA device to allow only the traffic relevant to the host devices. However, the computational power of SCADA devices can be a challenging issue to support this type of firewall.

Firewalls can be configured using restrict-constrained rules to control traffic in and out of the SCADA network; however, this will conflict with the feature allowing remote maintenance and operation by vendors and operators. Additionally, firewalls are assumed to be physically placed between the communication endpoints to examine each packet prior to passing it to the receiver. This may cause a latency that is not acceptable in real-time networks. Since firewalls do not know the "normal" operational behavior of the targeted system, they cannot stop malicious control messages, which may drive the targeted system from its expected and normal behavior, when they are sent from a compromised unit that is often used to remotely control and monitor SCADA networks. Moreover, it is beyond the ability of firewalls when the

attacks are initiated internally using an already-implanted malicious code or directly by an employee. Stuxnet (Falliere et al., 2011), Duqu (Bencsáth et al., 2012), and Flame (Munro, 2012) are the recent cyber-attacks that were initiated from inside automation systems. Therefore, the reliance only on firewalls is not sufficient to mitigate the potential threats to SCADA systems. Hence, an additional defense needs to be installed to monitor already predefined (or unexpected) patterns for either network traffic or system behavior in order to detect any intrusion attempt. The system using such a method is known in the information security area as an *Intrusion Detection System* (IDS).

There is no security countermeasures that can completely protect the target systems from potential threats, although a number of countermeasures can be used in conjunction with each other in order to build a robust security system. An IDS (Intrusion Detection System) is one of the security methods that has demonstrated promising results in detecting malicious activities in traditional IT systems. The source of audit data and the detection methods are the main, salient parts in the development of an IDS. The network traffic, system-level events and application-level activities are the most usual sources of audit data. The detection methods are categorized into two strategies: *signature-based* and *anomaly-based*. The former searches for an attack whose signature is already known, while the latter searches for activities that deviate from an expected pattern or from the predefined normal behavior.

Due to the differences between the nature and characteristics of traditional IT and SCADA systems, there has been a need for the development of SCADA-specific IDSs, and in recent years this has become an interesting research area. In the literature, they vary in terms of the information source being used and in the analysis strategy. Some of them use SCADA network traffic (Linda et al., 2009; Cheung et al., 2007; Valdes and Cheung, 2009), system-level events (Yang et al., 2006), or measurement and control data (values of sensors and actuators) (Rrushi et al., 2009b; Fovino et al., 2010a, 2012; Carcano et al., 2011) as the information source to detect malicious, uncommon or inappropriate actions of the monitored system using various analysis strategies which can be signature-based, anomaly-based or a combination of both.

It is believed that modeling of measurement and control data is a promising means of detecting malicious attacks intended to jeopardize a targeted SCADA system. For instance, the Stuxnet worm is a sophisticated attack that targets a control system and initially cannot be detected by the antivirus software that was installed in the victim (Falliere et al., 2011). This is because it used zero-day vulnerabilities and validated its drivers with trusted stolen certificates. Moreover, it could hide its modifications using sophisticated PLC rootkits. However, the final goal of this attack cannot be hidden since the manipulation of measurement and control data will make the behavior of the targeted system deviate from previously seen ones. This is the **main motivation of this book**, namely to explain in detail ***how to design SCADA-specific IDSs using SCADA data (measurement and control data)***, thus enabling the reader to build/implement an information source that monitors the internal behavior of a given system and protects it from malicious actions that are intended to sabotage or disturb the proper functionality of the targeted system.

As previously indicated, the analysis/modeling method, which will be used to build the detection model using SCADA data, is the second most important part after the selection of the information source when designing an Intrusion Detection System (IDS). It is difficult to build the "normal" behavior of a given system using observations of the raw SCADA data because, firstly, it cannot be guaranteed that all observations represent one behavior as either "*normal*" or "*abnormal*", and therefore domain experts are required for the labeling of each observation, and this process is prohibitively expensive; secondly, in order to obtain purely "normal" observations that comprehensively represent "normal" behavior, this requires a given system to be run for a long period under normal conditions, and this not practical; and, finally, it is challenging to obtain observations that will cover all possible abnormal behavior that can occur in the future. Therefore, we strongly argue that the design of a SCADA-specific IDS that uses **SCADA data** as well as **operating in unsupervised mode**, where the labeled data is not available, has great potential as a means of addressing the aforementioned issues. The unsupervised IDS can be a time- and cost-efficient means of building detection models from unlabeled data; however, this requires an efficient and accurate method to differentiate between the normal and abnormal observations without the involvement of experts, which is costly and prone to human error. Then, from observations of each behavior, either normal or abnormal, the detection models can be built.

## 1.2 EXISTING SOLUTIONS

A layered defense could be the best security mechanism, where each layer in the computer and network system is provided with a particular security countermeasure. For instance, organizations deploy firewalls between their private networks and others to prevent unauthorized users from entering. However, firewalls cannot address all risks and vulnerabilities. Therefore, an additional security layer is required. The last component at the security level is the IDS, which is used to monitor intrusive activities (Pathan, 2014). The concept of an IDS is based on the assumption that the behavior of intrusive activities are noticeably distinguishable from the normal ones (Denning, 1987). Since the last decade, compared to other security countermeasures, the deployment of IDS technology has attracted great interest from the traditional IT systems domain (Pathan, 2014). The promising functionalities of this technology have encouraged researchers and practitioners concerned with the security of SCADA systems to adopt this technology while taking into account the nature and characteristics of SCADA systems.

To design an IDS, two main processes are often considered: first, the selection of the information source (e.g., network-based, application-based) to be used, through which anomalies can be detected; second, the building of the detection models using the specified information source. SCADA-specific IDSs can be broadly grouped into three categories in terms of the latter process: *signature-based detection* (Digitalbond, 2013), *anomaly detection* (Linda et al., 2009; Kumar et al., 2007; Valdes and Cheung, 2009; Yang et al., 2006; Ning et al., 2002; Gross et al., 2004), and *specification-based*

*detection* (Cheung et al., 2007; Carcano et al., 2011; Fovino et al., 2010a; Fernandez et al., 2009). Recently, several signature-based rules (Digitalbond, 2013) have been designed to specifically detect particular attacks on SCADA protocols. The rules can perfectly detect *known attacks* at the SCADA network level. To detect *unknown attacks* at the SCADA network level, a number of methods have been proposed. Linda et al. (2009) suggested a window-based feature extraction method to extract important features of SCADA network traffic and then used a feed-forward neural network with the back propagation training algorithm for modeling the boundaries of normal behavior. However, this method suffers from the great amount of execution time required in the training phase, in addition to the need for relearning the boundaries of normal behavior upon receiving new behavior.

The model-based detection method proposed in Valdes and Cheung (2009) illustrates communication patterns. This is based on the assumption that the communication patterns of control systems are regular and predictable because SCADA has specific services as well as interconnected and communicated devices that are already predefined. This method is useful in providing a border monitoring of the requested services sand devices. Similarly, Gross et al. (2004) proposed a collaborative method, named "selecticast", which uses a centralized server to disperse among ID sensors any information about activities coming from suspicious IPs. Ning et al. (2002) identify causal relationships between alerts using prerequisites and consequences. In essence, these methods fail to detect *high-level control attacks*, which are the most difficult threats to combat successfully (Wei et al., 2011). Furthermore, SCADA network level methods are not concerned with the operational meaning of the process parameter values, which are carried by SCADA protocols, as long as they are not violating the specifications of the protocol being used or a broader picture of the monitored system.

Thus, analytical models based on the full system's specifications have been suggested in the literature. Fovino et al. (2010a) proposed an analytical method to identify critical states for specific-correlated process parameters. Therefore, the developed detection models are used to detect malicious actions (such as high-level control attacks) that try to drive the targeted system into a critical state. In the same direction, Carcano et al. (2011) and Fovino et al. (2012) extended this idea by identifying critical states for specific-correlated process parameters. Then, each critical state is represented by a multivariate vector, each vector being a reference point to measure the degree of criticality of the current system. For example, when the distance of the current system state is close to any critical state, it shows that the system is approaching a critical state. However, the critical state-based methods require full specifications of all correlated process parameters in addition to their respective acceptable values. Moreover, the analytical identification of critical states for a relatively large number of correlated process parameters is time-expensive and difficult. This is because the complexity of the interrelationship among these parameters is proportional to their numbers. Furthermore, any change in the system brought about by adding or removing process parameters will require the same effort again. Obviously, human errors are highly expected in the identification process of critical system states.

Due to the aforementioned issues relating to analytical methods, SCADA data-driven methods have been proposed to capture the mechanistic behavior of SCADA systems without a knowledge of the physical behavior of the systems. It was experimentally found by Wenxian and Jiesheng (2011) that operational SCADA data for wind turbine systems are useful if they are properly analyzed to indicate the condition of the system that is being supervised. A number of SCADA data-driven methods for anomaly detection have appeared in the literature. Jin et al. (2006) extended the set of invariant models by *a value range model* to detect anomalous values in the values for a particular process parameter. A predetermined threshold is proposed for each parameter and any value exceeding this threshold is considered as anomalous. This method can detect the anomalous values of an individual process parameter. However, the value of an individual process parameter may not be abnormal, but, in combination with other process parameters, may produce abnormal observation, which very rarely occurs. These types of parameter are called *multivariate parameters* and are assumed to be directly (or indirectly) correlated. Rrushi et al. (2009b) applied probabilistic models to estimate the normalcy of the evolution of values of multivariate process parameters. Similarly, Marton et al. (2013) proposed a data-driven method to detect abnormal behaviour in industrial equipment, where two multivariate analysis methods, namely principal component analysis (PCA) and partial least squares (PLS), are combined to build the detection models. Neural network-based methods have been proposed to model the normal behavior for various SCADA applications. For instance, Gao et al. (2010) proposed a neural-network-based intrusion detection system for water tank control systems. In a different application, this method has been adapted by Zaher et al. (2009) to build the normal behaviour for a wind turbine to identify faults or unexpected behavior (anomalies).

Although the results for the aforementioned SCADA data-driven methods are promising, they work only in supervised or semisupervised modes. The former method is applicable when the labels for both normal/abnormal behavior are available. Domain experts need to be involved in the labeling process but it is costly and time-consuming to label hundreds of thousands of data observations (instances). In addition, it is difficult to obtain abnormal observations that comprehensively represent anomalous behavior, while in the latter mode a one-class problem (either normal or abnormal data) is required to train the model. Obtaining a normal training data set can be done by running a target system under normal conditions and the collected data is assumed to be normal. To obtain purely normal data that comprehensively represent normal behavior, the system has to operate for a long time under normal conditions. However, this cannot be guaranteed and therefore any anomalous activity occurring during this period will be learned as normal. On the other hand, it is challenging to obtain a training data set that covers all possible anomalous behavior that can occur in the future.

Unlike supervised, semisupervised, and analytical solutions, this book is about **designing unsupervised anomaly detection methods**, where experts are not required to prepare a labeled training data set or analytically define the boundaries of normal/abnormal behavior of a given system. In other words, this book is interested

in developing a **robust unsupervised intrusion detection system** that automatically identifies, from unlabeled SCADA data, both normal and abnormal behavior, and then extracts the proximity-detection rules for each behavior.

## 1.3 SIGNIFICANT RESEARCH PROBLEMS

In recent years, many researchers and practitioners have turned their attention to SCADA data to build data-driven methods that are able to learn the *mechanistic* behavior of SCADA systems without a knowledge of the physical behavior of these systems. Such methods have shown a promising ability to detect anomalies, malfunctions, or faults in SCADA components. Nonetheless, it remains a relatively **open research area** to develop unsupervised SCADA data-driven detection methods that can be time- and cost-efficient for learning detection methods from unlabeled data. However, such methods often have a low detection accuracy. The focus of this book is about the **design of an efficient and accurate unsupervised SCADA data-driven IDS**, and four main research problems are formulated here for this purpose. Three of these pertain to the development of methods that are used to build a robust unsupervised SCADA data-driven IDS. The fourth research problem relates to the design of a framework for a SCADA security testbed that is intended to be an evaluation and testing environment for SCADA security in general and for the proposed unsupervised IDS in particular.

1. **How to design a SCADA-based testbed that is a realistic alternative for real SCADA systems so that it can be used for proper SCADA security evaluation and testing purposes.** An evaluation of the security solutions of SCADA systems is important. However, actual SCADA systems cannot be used for such a purpose because availability and performance, which are the most important issues, are most likely to be affected when analysing vulnerabilities, threats, and the impact of attacks. To address this problem, "real SCADA testbeds" have been set up for evaluation purposes, but they are costly and beyond the reach of most researchers. Similarly, small real SCADA testbeds have also been set up; however, they are still proprietary and location-constrained. Unfortunately, such labs are not available to researchers and practionners interested in working on SCADA security. Hence, the design of a SCADA-based testbed for that purpose will be very useful for evaluation and testing purposes. Two essential parts could be considered here: *SCADA system components* and a *controlled environment*. In the former, both high-level and field-level components will be considered and the integration of a real SCADA protocol will be devised to realistically produce SCADA network traffic. In the latter, it is important to model a controlled environment such as smart grid power or water distribution systems so that we can produce realistic SCADA data.
2. **How to make an existing suitable data mining method deal with large high-dimensional data.** Due to the specific nature of the unsupervised SCADA

systems, an IDS will be designed here based on SCADA data-driven methods from the unlabeled SCADA data which, it is highly expected, will contain anomalous data; the task is intended to give an anomaly score for each observation. The *k*-Nearest Neighbour (*k*-NN) algorithm was found, from an extensive literature review, to be one of the top ten most interesting and best algorithms for data mining in general (Wu et al., 2008), and, in particular, it has demonstrated promising results in anomaly detection (Chandola et al., 2009). This is because the anomalous observation is assumed to have a neighborhood in which it will stand out, while a normal observation will have a neighborhood where all its neighbors will be exactly like it. However, having to examine all observations in a data set in order to find *k*-NN for an observation *x* is the main drawback of this method, especially with a vast amount of high dimensional data. To efficiently utilize this method, the reduction of computation time in finding *k*-NN is the aim of this research problem that this book endeavors to address.

3. **How to learn clustering-based proximity rules from unlabeled SCADA data for SCADA anomaly detection methods.** To build efficient SCADA data-driven detection methods, the efficient proposed *k*-NN algorithm in problem 2 is used to assign an anomaly score to each observation in the training data set. However, it is impractical to use all the training data in the anomaly detection phase. This is because a large memory capacity is needed to store all scored observations and it is computationally infeasible to compute the similarity between these observations and each current new observation. Therefore, it would be ideal to efficiently separate the observations, which are highly expected to be consistent (normal) or inconsistent (abnormal). Then, a few proximity detection rules for each behavior, whether consistent or inconsistent, are automatically extracted from the observations that belong to that behavior.
4. **How to compute a global and efficient anomaly threshold for unsupervised detection methods.** Anomaly-scoring-based and clustering-based methods are among the best-known ones that are often used to identify the anomalies in unlabeled data. With anomaly-scoring-based methods (Eskin et al., 2002; Angiulli and Pizzuti, 2002; Zhang and Wang, 2006), all observations in a data set are given an anomaly score and therefore actual anomalies are assumed to have the highest scores. The key problem is how to find the near-optimal cut-off threshold that minimizes the false positive rate while maximizing the detection rate. On the one hand, clustering-based methods (Portnoy et al., 2001; Mahoney and Chan, 2003a; Portnoy et al., 2001; Jianliang et al., 2009; Münz et al., 2007) group similar observations together into a number of clusters, and anomalies are identified by making use of the fact that those anomalous observations will be considered as outliers, and therefore will not be assigned to any cluster, or they will be grouped in small clusters that have some characteristics that are different from those of normal clusters. However, the detection of anomalies is controlled through several parameter choices within each used detection method. For instance, given the top 50% of the observations that have the highest anomaly scores, these are assumed as anomalies. In this case, both

detection and false positive rates will be much higher. Similarly, labeling a low percentage of largest clusters as normal in clustering-based intrusion detection methods will result in higher detection and false positive rates. Therefore, the effectiveness of unsupervised intrusion methods is sensitive to parameter choices, especially when the boundaries between normal and abnormal behavior are not clearly distinguishable. Thus, it would be interesting to identify the observations whose anomaly scores are extreme and significantly deviate from others, and then such observations are assumed to be "abnormal". On another hand, the observations whose anomaly scores are significantly distant from "abnormal" ones will be assumed to be "normal". Then, the ensemble-based supervised learning is proposed to find a global and efficient anomaly threshold using the information of both "normal"/"abnormal" behavior.

## 1.4 BOOK FOCUS

This section summarizes the important lessons learned from the development of robust unsupervised SCADA data-driven Intrusion Detection Systems (IDSs), which are detailed in the various chapters of this book. The first lesson relates to the design of a SCADA security testbed through which the practicality and efficiency of SCADA security solutions are evaluated and tested, while, the remaining three aspects focus on the details of the various elements of a robust unsupervised SCADA data-driven IDS.

- The evaluation and testing of security solutions tailored to SCADA systems is a challenging issue facing researchers and practitioners working on such systems. Several reasons for this include: privacy, security, and legal constraints that prevent organizations from publishing their respective SCADA data. In addition, it is not feasible to conduct experiments on actual live systems, as this is highly likely to affect their availability and performance. Moreover, the establishment of a real SCADA Lab can be costly and place-constrained, and therefore unavailable to all researchers and practitioners. In this book, a framework for a SCADA security testbed is described to build a full SCADA system based on a hybrid of emulation and simulation methods. A real SCADA protocol is implemented and therefore realistic SCADA network traffic is generated. Moreover, a key benefit of this framework is that it is a realistic alternative to real-world SCADA systems and, in particular, it can be used to evaluate the accuracy and efficiency of unsupervised SCADA data-driven Intrusion Detection Systems (IDSs).
- Unsupervised learning for anomaly-detection methods is time- and cost-efficient since they can learn from unlabeled data. This is because human expertise is not required to identify the behavior (whether normal or abnormal) for each observation in a large amount of training data sets. Anomaly scoring methods are believed to be promising automatic methods for assigning an anomaly degree to each observation (Chandola et al., 2009). The $k$-NN method

is one of the most interesting and best methods for computing the degree of anomaly based on neighborhood density of a particular observation (Wu et al., 2008). However, this method requires high computational cost, especially with large and high-dimensional data that we expect to have in the development of an unsupervised SCADA data-driven IDS. Therefore, this book describes an efficient *k*-nearest neighbor-based method, called *k*NNVWC (*k*-Nearest Neighbor approach based on Various-Widths Clustering), which utilizes a novel various-width clustering algorithm and triangle inequality.

- It is not feasible to retain all the training data in SCADA data-driven anomaly detection methods, especially when these are built from a large training data set. This is because such detection methods will be used for on-line monitoring, and therefore the more information retained in the detection methods, the larger the memory capacity required and the higher the computation cost required. To address this issue, this book describes a clustering-based method to extract proximity-based detection rules, called SDAD (SCADA Data-Driven Anomaly Detection), which are assumed to be a tiny portion compared to the training data, for each behavior (normal and abnormal). Each rule comprehensively represents a subset of observations that represent only one behavior.
- Unsupervised learning for anomaly-detection methods are based mainly on assumptions to find the near-optimal anomaly detection threshold. Therefore, the accuracy of the detection methods is based on the validity of the assumptions. This book, however, describes an efficient method, called GATUD (Global Anomaly Threshold to Unsupervised Detection), which firstly identifies observations whose anomaly scores significantly deviate from others to represent "abnormal" behavior. On the other hand, a tiny portion of observations whose anomaly scores are the smallest are considered to represent "normal" behavior. Then an ensemble-based decision-making method is described, which aims to find a global and efficient anomaly threshold using the information of both "normal"/"abnormal" behavior.

## 1.5 BOOK ORGANIZATION

The remainder of the book is structured as follows. Chapter 2 gives an introduction to readers who do not have an understanding of SCADA systems and their architectures, and the main components. This includes a description of the relationship between the main components and three generations of SCADA systems. The classification of a SCADA IDS based on its architecture and implementation is described.

Chapter 3 describes in detail SCADAVT, a framework for a SCADA security testbed based on virtualization technology. This framework is used to create a simulation of the main SCADA system components and a controlled environment. The main SCADA components and real SCADA protocol (e.g., Modbus/TCP) are integrated. In addition, a server, which acts as a surrogate for water distribution systems, is introduced. This framework is used throughout the book to simulate a realistic SCADA

system for supervising and controlling a water distribution system. This simulation is mentioned in the other chapters to evaluate and test anomaly detection models for SCADA systems.

Chapter 4 describes in detail *k*NNVWC, an efficient method that finds the *k*-nearest neighbors in large and high-dimensional data. In *k*NNVWC, a new various-widths clustering algorithm is introduced, where the data is partitioned into a number of clusters using various widths. Triangle inequality is adapted to prune unlikely clusters in the search process of *k*-nearest neighbors for an observation. Experimental results show that *k*NNVWC performs well in finding *k*-nearest neighbors compared to a number of *k*-nearest neighbor-based algorithms, especially for a data set with high dimensions, various distributions, and large size.

Chapter 5 describes SDAD, a method that extracts proximity-based detection rules from unlabeled SCADA data, based on a clustering-based method. The evaluation of SDAD is carried out using real and simulated data sets. The extracted proximity-based detection rules show a significant detection accuracy rate compared with an existing clustering-based intrusion detection algorithm.

Chapter 6 describes GATUD, a method that finds a global and efficient anomaly threshold. GATUD is proposed as an add-on component that can be attached to any unsupervised anomaly detection method in order to define the near-optimal anomaly threshold. GATUD shows significant and promising results with two unsupervised anomaly detection methods.

Chapter 7 looks at the authentication aspects related to SCADA environments. It describes two innovative protocols which are based on TPASS (Threshold Password-Authenticated Secret Sharing) protocols; one is built on two-phase commitment and has lower computation complexity and the other is based on zero-knowledge proof and has less communication rounds. Both protocols are particularly efficient for the client, who only needs to send a request and receive a response. Additionally, this chapter provides rigorous proofs of security for the protocols in the standard model.

Finally, Chapter 8 concludes with a summary of the various tools and methods described in this book to the extant body of research and suggests possible directions for future research.