A Taxonomy of Supervised Learning for IDSs in SCADA Environments

Jakapan Suaboot, Adil Fahad, Zahir Tari, John Grundy, Abdun Naser Mahmood, Abdulmohsen Almalawi, Albert Zomaya, Khalil Drira

A Taxonomy of Supervised Learning for IDSs in SCADA Environments

Jakapan Suaboot, RMIT University, Australia
Adil Fahad, University of Albaha, Saudi Arabia
Zahir Tari, RMIT University, Australia
John Grundy, Monash University, Australia
Abdun Naser Mahmood, La Trobe University, Australia
Abdulmohsen Almalawi, King Abdulaziz University, Saudi Arabia
Albert Y. Zomaya, The University of Sydney, Australia
Khalil Drira, University of Toulouse, France

Abstract

Supervisory Control and Data Acquisition (SCADA) systems play an important role in monitoring industrial processes such as electric power distribution, transport systems, water distribution, and wastewater collec- tion systems. Such systems require a particular attention with regards to security aspects, as they deal with critical infrastructures that are crucial to organizations and countries. Protecting SCADA systems from intru- sion is a very challenging task because they do not only inherit traditional IT security threats but they also include additional vulnerabilities related to field components (e.g., cyber-physical attacks). Many of the exist- ing intrusion detection techniques rely on supervised learning that consists of algorithms that are first trained with reference inputs to learn specific information, and then tested on unseen inputs for classification pur- poses. This article surveys supervised learning from a specific security angle, namely SCADA-based intrusion detection. Based on a systematic review process, existing literature is categorized and evaluated according to SCADA-specific requirements. Additionally, this survey reports on well-known SCADA datasets and testbeds used with machine learning methods. Finally, we present key challenges and our recommendations for using specific supervised methods for SCADA systems.

Key Words: SCADA security, network intrusion, machine learning, supervised learning

1 INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems have been integrated into our critical infrastructures such as electric power generation, transport systems, water distribution and wastewater collection systems to control and monitor such industrial processes. Since industrial manufacturing and power distribution sites are geographically distant and involve potentially extreme hazards, the integration of SCADA allows operators to maximize cost-effective operations and safety of their personnel [22]. For example, using a Smart Grid application with SCADA technology, power outages can be quickly diagnosed and temporarily fixed remotely and securely from the control center.

A typical SCADA system is operated through a *control center*, consisting of a complex of computers, networks, and databases. The databases store values gathered by Master Terminal Unit (MTU) from sensors, e.g., voltage, current, valve pressure, and so on. The control center sends control commands to actuators, such as Programmable Logic Controller (PLC), Remote Terminal Unit (RTU) or Intelligent Electronic Device (IED), to control the industrial processes [22]. A SCADA system is pervasive, and its components are interconnected using both wireless and wired communication. On the one hand, SCADA is connected to traditional IT and networking technologies (e.g., operating systems and Internet protocols). On the other hand, it is also connected to SCADA-specific technologies, such as industrial devices and communication protocols such as standard protocols (e.g., IEC 60870-5-101 or 104, and DNP3) and proprietary protocols (i.e., Modbus RTU, RP-570, Profibus and Conitel) [54]. However, any disruption to SCADA systems can result in significant financial loss or even lead to loss of life. In the past, such systems were secure by virtue of their isolation from corporate networks and the Internet, and due to the use of proprietary hardware and software. In other words, they were self-contained and totally isolated from the public networks. This isolation created the myth that malicious intrusions and attacks from the outside world were not a big concern, and such attacks were expected to only come from the inside. Therefore, when developing SCADA protocols, the security of the information systems and the significance of loss due to denial of service was given little consideration.

In recent years, SCADA systems have begun to shift away from using proprietary and customized hardware and software in favor of using Commercial-Off-The-Shelf (COTS) solutions [85]. Among the well-known vendors are ABB, Siemens, General Electric (GE), Alstom, SEL, Toshiba, and Schneider Electric [137]. This shift has increased their connectivity to public networks (Internet) using standard protocols, e.g., TCP/IP. In addition, there is a decreased reliance on one vendor. Undoubtedly, this increases productivity and profitability by reducing capital expenditure (Capex). However, this also now exposes such systems to more diverse, intelligent cyber attacks [87]. The convergence of state-of-the-art communication technologies exposes SCADA systems to all the inherent vulnerabilities of these technologies. According to the research result published in 2017 by Kaspersky Lab on the threats landscape for industrial automation systems [58], various indus- try sectors are affected by serious vulnerabilities that operate remotely using traditional network connectivity, see Figure 1.

Vulnerabilities targeting SCADA systems have increased significantly since 2009, according to the Open Source Vulnerability Database (OSVDB) [7]. Different actors have different motivations

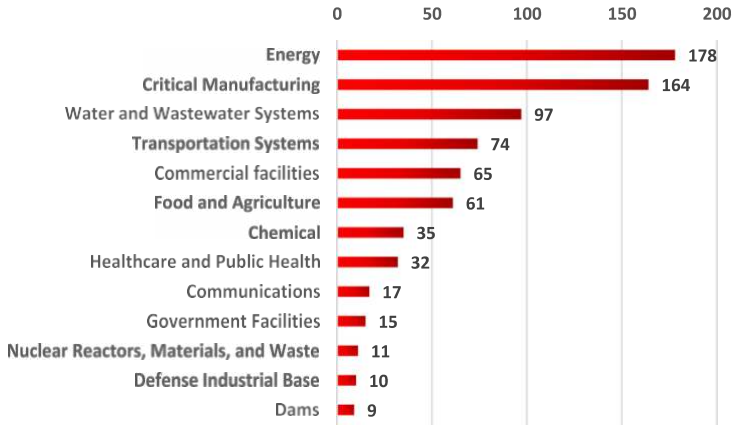

Fig. 1. The number of vulnerabilities identified in various industries, published in 2017 [58].

for identifying and exploiting SCADA vulnerabilities. For example, state-based actors are interested in vulnerabilities related to the critical infrastructure, and criminals or hackers are more interested in stealing intellectual property and sabotaging industrial processes. Therefore, the nature of SCADA attacks has shifted from small scale, insider-based attacks to external threat and large-scale attacks. For example, prior to the year 2000, the majority of the reported incidents impacting SCADA networks were either due to accidents or due to disgruntled insiders acting maliciously. Since 2009, there has been a sharp increase in the total number of reported incidents and most of them (above 70%) are attacks originating from the Internet [8].

Numerous malicious attacks have posed serious and evolving security threats to SCADA systems. Practically, there is no security countermeasure that can completely protect a target system from potential threats. An Intrusion Detection System (IDS) [38] is one of the security solutions 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 the IDS. The network traffic, system-level events and application-level activities are the 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 a predefined model of normal behavior of the system.

Due to the differences between the operational nature and characteristics of traditional IT and SCADA systems, there is a need for the development of SCADA-specific IDS. As a consequence, a number of IDS schemes based on machine learning approaches have been investigated, proposed, and developed by the research community and the networking industry over the past ten years. We have used Microsoft Academic search engine [109] to estimate the popularity in machine learning based IDS approaches since 2007.

Figure 2 shows the growth over time in the number of publications in *classification/supervised*, *clustering/unsupervised* and *feature selection*. The researchers have made great efforts in developing advanced supervised classification approaches—compared to unsupervised classification and feature selection approaches—in order to improve the accuracy of SCADA-specific IDS. This is due to their higher performance and accuracy compared with unsupervised classification. Thus, supervised machine learning techniques are widely used across a wide range of SCADA-specific IDS approaches.

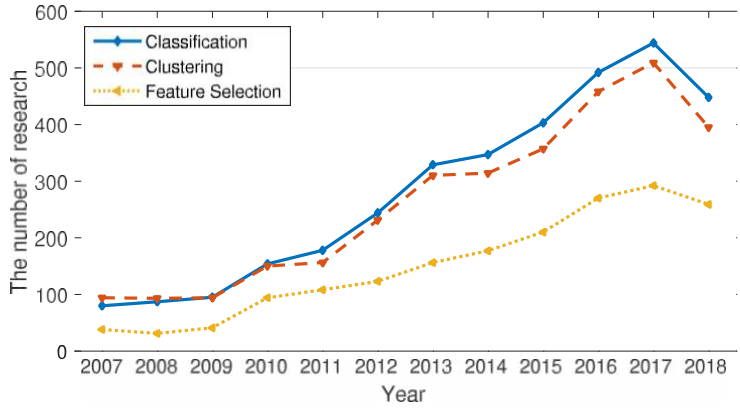

Fig. 2. Evolution of IDS schemes based on machine learning approaches (between 2007-2018 and ongoing).

Many articles have been written that survey IDSs for critical infrastructure systems from different perspectives. For example, the survey [140] emphasizes the architectures of IDS for SCADA systems. The research in [78] focuses on classifying IDS solutions based on detection techniques and audit materials. The review in [50] gathers information about the testbeds used for industrial control systems. In contrast to previous reviews, the focus of this article is on *Supervised Learning for SCADA-Specific IDSs*. Additionally, this article analyzes existing literature based on holistic perspectives, including detection architecture, detection technique, auditing sources, and feasibility of applying the proposed IDS to the various real SCADA systems.

This survey attempts to review the field of supervised machine learning techniques used for SCADA-specific IDS and achieve the following objectives:

- Propose a framework that systematically groups a collection of existing SCADA-specific IDS techniques into appropriate categories and compares their advantages and drawbacks;
- Present a comprehensive taxonomy of supervised machine learning techniques specifically used for SCADA-based IDSs; and
- Propose an evaluation metric to allow theoretical analysis of the most representative supervised machine learning algorithms from each category, with respect to the requirements of SCADA-specific IDSs.

In summary, the proposed survey presents a taxonomy of supervised machine learning techniques used for SCADA-specific IDS and proposes a categorizing framework that covers major factors in the selection of a suitable technique for SCADA-specific IDS.

The rest of this paper is organized as follows. Section 2 provides useful background for readers on the basics of SCADA systems. Section 3 presents a taxonomy of SCADA-specific IDSs. Section 4 provides a brief overview and a taxonomy of supervised-machine learning-based IDSs. Section 5 provides the evaluation of SCADA-based IDS supervised learning, and theoretically analyze the most representative supervised machine learning algorithms for SCADA-specific IDSs. Section 6 discusses the research gaps. We conclude this article in Section 7 and discuss future research.

2 BACKGROUND

This section presents an introduction to the key architectural concepts of a SCADA system as well as the classification of SCADA-based IDSs. In order to distinguish between the SCADA and a

generic ICT system, we also discuss threats and vulnerabilities of the SCADA that lead to specific requirements for the IDS of such system.

2.1 SCADA Systems

SCADA is widely used in an industrial system that continuously monitors and controls many different sections of industrial infrastructure. Applications include but are not limited to oil refineries, water treatment and distribution systems, transport systems, manufacturing plants, and electric power generation plants. A SCADA system gathers measurements—collects data from the deployed field devices—and sends them to a central site (normally a control center) for further processing and analysis. The information and status of the supervised and monitored processes are displayed at the base station or at the utility center. As such industrial systems are large and complex, a central master unit continuously and remotely monitors and controls different sections of the plant to guarantee their proper functioning.

SCADA Components

The main components of a typical SCADA system include the following: Master Terminal Unit (MTU), Programmable Logic Controller (PLCs), Remote Terminal Unit (RTU), Intelligent Electronic Device (IED), Human Machine Interface (HMI), and Communication Media [7].

- *Master Terminal Unit (MTU)*: This is the core of a SCADA system that gathers information from the distributed RTUs and analyzes it to control the various processes [7]. A plant's data is analyzed through histogram generation, standard deviation calculation, and plot- ting one parameter with respect to another. Based on the performance results, an operator may decide to monitor any channel more frequently, change the limits, or shut down the terminal units. The software can be designed according to the applications and the type of analysis required. The operator may be interested in finding the best operating steps for a plant which will minimize the overall operating cost. To solve this problem, engineers often employ different optimization techniques on the collected data to determine the best operating process. The operating process is then converted to appropriate operating signals and then sent to the remote terminal units RTUs through communication pathways (e.g., radio links, leasedline, or fiber optic [44]).
- *Field Devices (RTUs, PLCs, and IEDs)*: These are computer-based components deployed at a remote site to gather data from sensors and to control actuators [7]. Each field device may be connected to one or more sensors and actuators that are directly connected to physical equipment (e.g., pumps, valves, motors). The main function of such devices is to convert the electrical signals received by sensors into digital values in order to send them to the MTU for further processing and analysis using some specific communication protocol (e.g., Modbus [53]). On the other hand, they can convert a digital command message received by MTU into an electrical signal in order to control actuators that are controlling some physical aspects of the controlled system. Different field-level devices, e.g., RTUs, PLCs, and IEDs, that are deployed at remote sites perform different functions. RTUs collect data from sensors and send it back to the MTU, and then the MTU takes a decision based on this data and sends commands to different actuators. In addition to the same function of RTUs, PLCs can collect data from sensors and based on the collected data, they can send commands directly to actuators using a local loop. In other words, PLCs can processes the data locally and take the decision without contacting MTU. IEDs are part of control systems such as transformers, circuit breakers, sensors, and the like, and they can be controlled via PLCs or RTUs.

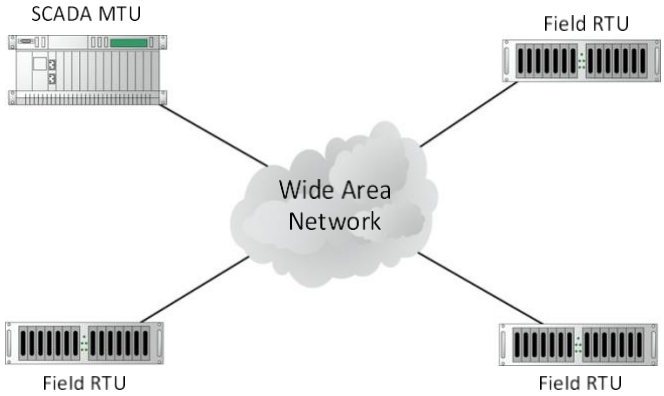

Fig. 3. The first generation of SCADA architecture.

- *Human-Machine Interface (HMI)*: This provides an efficient human-machine interface through which an operator can monitor/control end devices, such as sensors and actuators. An HMI graphically displays information on the current state of the supervised and controlled process including alerts, warnings, and urgent messages. In addition, the HMI allows the user to interact with the system using switches, buttons, and other controls [7].
- *Historian*: This is a database that stores the different types of data gathered by the SCADA system, such as measurement and control data, events, alarms, and operator activities. This data is used for historical, auditing, analysis, and operational purposes [7].

SCADA Architecture

A SCADA network provides the communication infrastructure for different field devices (e.g., PLCs and RTU) of a plant. These field devices are remotely monitored and controlled throughout the SCADA network. To make the network communication more efficient and secure, many modern computing technologies have evolved from a monolithic system to a distributed system and to the current inter-networked systems [7].

Monolithic Systems (1st Generation). The *monolithic* SCADA system is considered to be the first generation SCADA system [7]. At that time, networks were non-existent in general, and therefore a SCADA system was deployed as a stand-alone system, and no connectivity to other systems existed. As can be seen in Figure 3, a SCADA master used Wide Area Networks (WANs) to communicate with field devices using communication protocols that were developed by vendors of field devices. In addition, these protocols had limited functionality—they could only do scanning and controlling over points within certain types of RTUs. The communication between the master and field devices (e.g., RTUs) was carried out at the communication bus level using proprietary adapters. To avoid system failures, two identically equipped mainframe systems were used: one as the primary and the other as a backup. The latter was designed to take over when a failure of the primary system was detected. Figure 3 illustrates the typical architecture of this type of SCADA architecture.

Distributed Systems (2nd Generation). Figure 4 depicts a typical second generation SCADA architecture. With the development of Local Area Networking (LAN) technologies, the second generation of SCADA system distributes the processing to multiple systems and assigns a specific function for each station [7]. In addition, multiple stations could be connected to a LAN to share information with each other in real time. For instance, the communication server can be set up

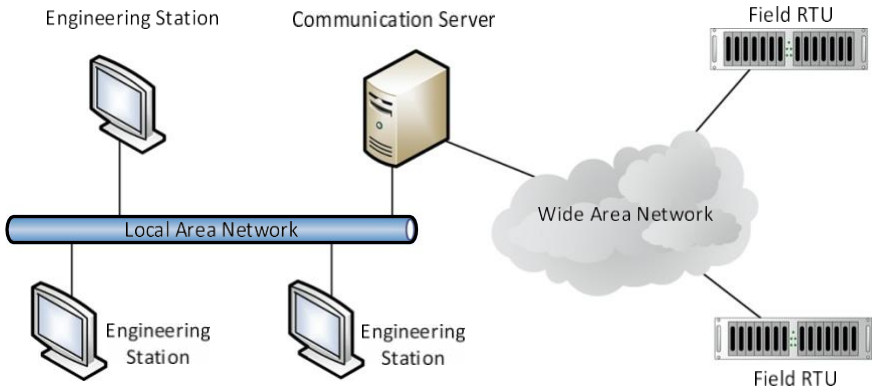

Fig. 4. The second generation of SCADA architecture.

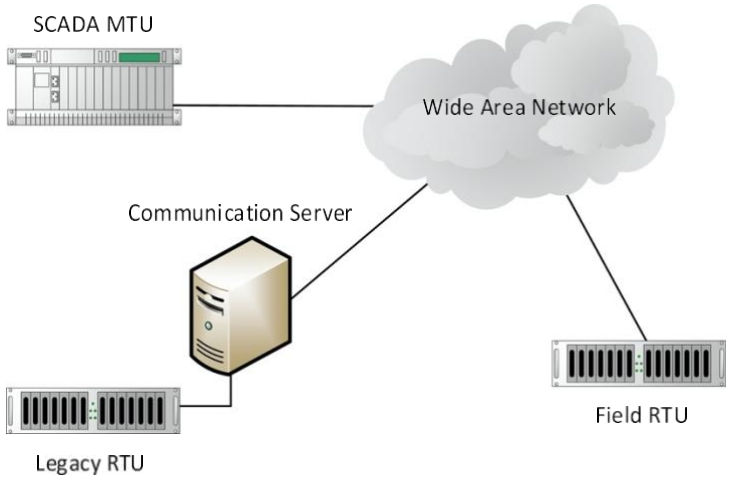

Fig. 5. The third generation of SCADA architecture.

to communicate with field devices such as PLCs and RTUs. Some of the distributed stations can be an MTU, a Historian, or an HMI server. The distribution of system's functionalities across the network-connected systems increases the processing power, reduces the redundancy, and improves the reliability of the system as a whole. In this generation, system failures are addressed by keeping all stations on the LAN in an online state during the operation time, and if one station (e.g., HMI station) fails, then another HMI station takes over.

Networked Systems (3rd Generation). Unlike the second generation, a third generation SCADA system is based on an open system architecture, rather than vendor-controlled proprietary solutions. One of the major differences is that the third generation can utilize open standard protocols and products. Consequently, the SCADA functionalities can be distributed across a WAN and not justa LAN [7]. For instance, most field devices can be connected directly to the MTU over an Ethernet connection. This open system architecture allows various products from different vendors to be integrated with each other to build a SCADA system at low cost. In addition, a remote field device can be supervised and controlled from any place and at any time using the Internet. Figure 5 shows the architecture of a typical third generation networked SCADA system.

2.2 Different Types of SCADA-Based Intrusion Detection Systems

An IDS is an autonomous hardware, software, or combination of both systems that can detect threats in a SCADA system by monitoring and analyzing network or device activities from both internal and external attackers. In traditional IT systems, IDS can be classified into network-based and host-based IDS [34] depending on the location of the collected data and logs. However, due to the different nature of SCADA systems in terms of architecture, functionalities, and operating devices, SCADA-based IDS, within the scope of this article, are categorized based on only the source of collected data: *SCADA network-based IDS* and *SCADA application-based IDS*.

SCADA Network-Based IDSs

A SCADA network-based IDS [48, 69, 86, 119] captures the data packets that are communicated between devices such as point-to-point between RTU/PLC, and between RTU/PLCs and the MTU. If a packet is a suspicious one, the security team will be sent an alarm for further investigation. An advantage of a SCADA network-based IDS is their lower computation costs, as only information in the packet's header is needed during the investigation process, and therefore a SCADA network packet can be analyzed on-the-fly. Consequently, traffic from larger networks can be inspected within a short period of time [69]. When there is high network traffic however, a SCADA network-based IDS may experience issues in monitoring all the packets and might miss some attacks.

However, the key weakness is that the operational behavior of the underlying SCADA processes cannot be inferred from the information provided at the network level (e.g., IP address, protocol, port,and so on). For example, if the payload of the SCADA network packet contains a malicious message, which is crafted at the application level, the SCADA network-based IDS cannot detect it, particularly when this is not violating the specifications of the protocol being used, or the communication pattern between SCADA networked devices [7, 24, 40, 41].

SCADA Application-Based IDSs

SCADA data, which comprises the measurements and control data generated by sensors and actuators, represents the majority of the information. Using this data, the operational behavior of a given SCADA system can be inferred [24, 41, 99, 126, 136]. In contrast to SCADA network-based IDSs that only inspect network level information, a SCADA application-based IDS can inspect high-level data (i.e., SCADA data) to detect the presence of unusual behavior. For example, SCADA network-based IDSs are often unable to detect high-level control attacks [124] from packet headers; which can be detected by analyzing SCADA data [99].

Since the information source of a SCADA application-based IDS can be gathered from different remote field devices, the following are the various ways to deploy a SCADA application-based IDS [7]: (i) It can be deployed in the historian server, as this server is periodically updated by the MTU server which acquires, through field devices, such as PLC and RTU, the information and status of the monitored system for each time period. However, this type of deployment raises a security issue when the information and status stored in the historian differ from the realtime data in the field. This could occur when the MTU server has been compromised or the data has been changed using False Data Injection attacks [11, 14, 121]; (ii) It can also be deployed in an independent security hardened server, which from time to time acquires information and statuses from the monitored field devices [40]. Consequently, the large number of requests from this server might increase the network overheads resulting in degraded performance of the IDS; (iii) Each adjacent field device can be connected with a server running a SCADA application-based IDS, which is similar to the work proposed in [5] and [6]. However, the key issue is that SCADA data is directly/indirectly correlated, and therefore sometimes there is an abnormality in a parameter

not because of itself, but due to anomalous value in another parameter [24, 41]. Therefore, it would be appropriate to identify and monitor correlated parameters, such as sensor readings related to a single process.

Signature-Based vs. Anomaly-Based SCADA IDS Approaches

Many types of SCADA-based IDS have been proposed in the literature, and these fall into two broad categories in terms of the detection strategy [7]: *signature-based detection* [26, 35] and *anomaly-based detection* [3, 48, 64, 69, 86, 119, 128].

A signature-based IDS detects malicious activities in a SCADA system's network traffic or in its application events. It does this by using pattern matching techniques to detect telltale events against a database of signatures [35] or fingerprints [26] of known attacks. The false positive rate (i.e., incorrectly identifying a normal event as an attack) in this type of IDS is very low and can ap- proach zero. Moreover, the detection time can be fast because it is based only on the use of a match- ing process during the detection phase. Despite the aforementioned advantages of signature-based IDSs, they typically fail to detect new attacks (e.g., zero-day) whose signatures are not known, or which do not exist in its database. Therefore, the database must be constantly updated with pat- terns of new attacks.

An anomaly-based IDS assumes that the behaviors of intrusive activities are noticeably distinguishable from normal behavior [34]. The "normal model" is created using a realistic training set using advanced mathematical/statistical techniques. Any significant deviation from this model is flagged as an anomaly or potential attack. For example, normal SCADA network traffic can be obtained over a period of *normal* operations, and then a modeling technique is applied to build the normal SCADA network profiles. During the detection phase, the deviation degree between the current network traffic and the created normal network profile is computed: if the deviation exceeds the predefined threshold, the current network traffic will be flagged as an intrusive activity. The primary advantage of anomaly-based IDSs compared to signature-based ones is that new or unknown attacks can be detected, although it generally suffers from a higher false positive rate (i.e., detecting normal behavior as malicious).

2.3 SCADA Threats and Vulnerabilities

When SCADA was initially suggested, the focus was on efficiency and effectiveness without considering the potential security issues it might encounter in the future. As security concerns became more critical, it was discovered that it was not easy to address such issues, since an upgrade or replacement of a vital SCADA network in an old industrial system can disrupt the production or management of existing critical processes and services [54]. SCADA was also originally developed for *isolated* systems. Modern critical infrastructure has since been interconnected via the Internet network to increase scope and capability and we have thus seen various new attacks on the systems. An example is the Havex malware that allows attackers to remote access and controls the system using a backdoor channel. Such malware affected victims in numerous industries, including sections of energy, aviation, and defense to name a few. The Stuxnet worm that targets PLC devices and gives unexpected commands to the infected control device [132]. This threatprimarily targets Iran's nuclear program. The SQL Slammer worm that exploits buffer overflow vulnerability and performs DoS on the infected system (i.e., Davis-Besse—the American nuclear power plan). We categorize vulnerabilities of the SCADA into three aspects as follows:

Hardware. The SCADA system is geographically distributed (i.e., covering regions of cities), and many low-level controllers/sensors are wirelessly interconnected. As a result, it is hard to prevent attackers from accessing SCADA components, wirelessly or even physically. For instance,

attackers could intercept the wireless communication signal using tools like Aircrack-NG [21] to gain access to the network. In the field, it is even possible that attackers intrude into the station and direct access to the control device (e.g., using the USB drive malware).

The physical component also has a tight relationship with its software counterpart. Hence, an attack with a series of malicious commands could severely affect the hardware. An example is the Aurora Generator Test showing the destruction of the electric generator by remotely attack from the network [125]. Furthermore, the inclination to use COTS devices in SCADA system makes the system more vulnerable. Since the COTS equipment has a generic design and protocol standards, it has become a target of exploitation [54].

Software. SCADA-specific protocols use plaintext messages to communicate between sensors-actuators (e.g., Modbus, DNP-3, IEC 60870-5-101, and IEC 60870-5-104). These simple communications can be easily manipulated by false data injection attacks [10–13]. Here attackers inject (e.g., using man-in-the-middle (MITM) technique) a fake measurement into a closed loop control system. This can disrupt or even stop the critical system.

Patching the newly discovered vulnerability can be complicated for a SCADA system. Since the distributed control components are mostly Windows- or Linux-based computers, inherited vulnerabilities are inevitable. Unfortunately, with the availability requirement and diversity of system components, appropriate security patches might take several months to arrive [7]. This is because system components might come from different manufacturers, using various standards or proprietary protocols. In some cases, software/hardware is being used for the extended period of time after the end of the manufacturer-supported warranty [84].

Security Administration and Insider Attacks. Apart from the security technology, attackers could harvest information using social engineering to attack employees of the targeted organization. A bad security practice, such as weak passwords or bad configurations, could make the SCADA system vulnerable. On the other hand, an angry former employee could hack into the system and cause devastating damages to the system. For instance, there was an incident in Queensland, Australia in which a a former staff member of the Maroochy Water Services flooded millions liters of sewage water into parks and rivers by using a radio transmitter and a computer from his car [1].

2.4 Requirements of SCADA-Based IDSs

SCADA is distinguished from a traditional IT system by such key characteristics as availability and reliability . It also includes a wide range of proprietary COTS components, i.e., the cyber-physical components that are tightly coupled. Hence, the key requirements for a SCADA-specific IDS can be listed as follows:

(1) *Availability and Robustness*: According to the availability constraint, the IDS technology does not only require anomaly detection that covers both known and unknown attacks, it also requires support for a model updating mechanism that minimizes downtime. With regard to robustness, the IDS has to be compatible with incomplete features gathered from different platforms to allow components from diverse manufacturers to be used. Furthermore, the security system has to deal with training/detecting data that contains a little portion of attacks event and lots of noises [73].

(2) *Scalability and Resilience*: Despite a large number of logs that are continuously generated from a number of sensors/controllers, the detection module should not slow down the manufacturing or control process. Conversely, it should give timely automated protection and alarms. Since a decentralized or distributed IDS could be an answer to the scalability issue, the design of algorithms should also account for the possibility that detection

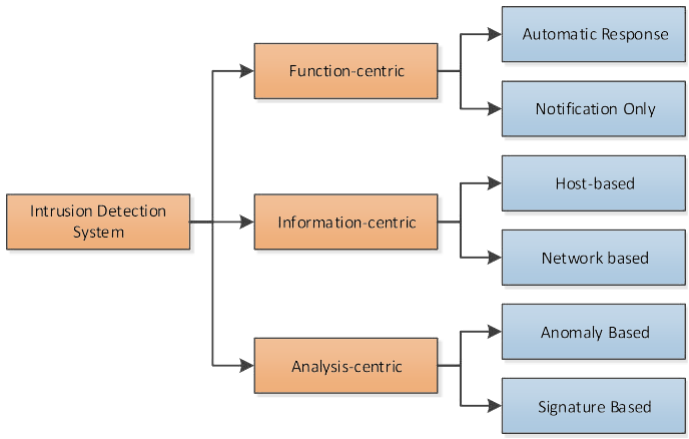

Fig. 6. A taxonomy of IDSs for SCADA systems.

devices may themselves be compromised or fail [46]. To extend scalability without scarifying availability, resilience is also an important requirement.

(3) *Information Aggregation and Correlation*: Since a SCADA system has a tight cyber-physical relationship [7], the IDS should be able to make use of multiple attributes in detecting anomalies. The SCADA system integrates Operational Technology (OT), such as sensors and actuators, as well as Information Technology (IT), such as databases, servers, firewalls, and routers. As a result, the SCADA-specific IDS has to support aggregation and correlation of variables from multiple sources. Besides, manipulation of either cyber or physical world could affect one another. For instance, a false-data injection attack [11] that infiltrates fake measurement data into the system to cause physical disruption. Hence, it is important to monitor anomaly from a holistic perspective, including cyber perspective (i.e., network communication, application behavior) and physical perspective (e.g., signals from sensors/controllers).

(4) *Feasibility*: As the nature of the SCADA system can be different depending on the application (e.g., electric power distribution, water supply system, and manufacturing process control system), the practicality study of the proposed system is crucial. In addition, the completeness of assessing the proposed model is also important for the critical infrastructure system. Despite the limited research and development cost, the same system can act differently on different evaluation environment (e.g., tested on a simulator, SCADA test-bed and implementing the real system).

3 TAXONOMY OF SCADA-BASED IDSS

Protecting a SCADA system from intrusion is a challenging task because it not only inherits many of the existing ICT vulnerabilities but also includes vulnerabilities from the OT field components. In addition, the implementation of countermeasures suffers due to the limited computational resources on the OT side. Thus, there is a need to study IDS systems used for SCADA. Motivated by the work proposed in [45], [55], [78], [120], and [140], the existing approaches can be classified using three different categorizations based on the requirements of SCADA-specific IDSs. Figure 6 shows our taxonomy of SCADA-based IDSs.

- *Function-Centric*: As part of its operations, a SCADA system generates alarms for processes that go beyond their operating parameters, e.g., due to an expected or unexpected change in

the underlying physical processes. While this may not always correspond to an attack, alert-based responses address inherent problems in SCADA operations, which are otherwise not possible to capture using traditional IT-based IDS. Based on the requirements of the strict availability, the IDS can respond immediately to unusual situations [56, 63, 69, 91–93, 95, 132, 133, 135, 138, 141], or provide a delayed notification summarizing similar alarms [52, 74, 79, 88, 100, 114, 122, 123].

- Information-Centric: If we examine the information used for the detection, then IDS systems can be further categorized into Host-based Intrusion Detection (HID) and Network-based Intrusion Detection (NID). Host-based methods detect intrusions by examining data gathered from hosts, such as device memory, application logs [62, 90, 94, 123, 132, 138, 141], the change of system configuration [79], Network-based methods collect data from either a network, a hub or a router and detect anomalies at the source, destination, protocol and payload from network data [9, 31, 51, 63, 72, 88, 111, 113, 122].
- *Analysis-Centric:* This category focuses on different analysis techniques for detecting outliers. As discussed in Section 2.2, this has two subgroups, namely *signature-based* and *anomaly-based* approaches. The scope of this review is *anomaly-based approach*, which employs various kinds of machine learning techniques.

As three previously discussed categories classify existing literature from different perspectives, a single publication can be included into three categories. For instance, References [132], [138], and [141] are grouped as *automatic response*, *host-based*, and *anomaly-based* according to their functions, detecting sources and analysis methods.

However, since this review focuses on supervised learning algorithm, the number of reviewing literature mostly consists of *anomaly based* approaches. This is because the anomaly-based detection focuses on finding deviations (outliers) from the trained normal or abnormal models rather than using the pre-defined patterns of attacks to detect abnormal events, e.g., patterns of spoof Address Resolution Protocol (ARP) messages or sequence of Application Programming Interface (API) called by known malicious programs. A few numbers of signature-based approaches were in- cluded as they employed supervised learning algorithms to dynamically generate sets of signature for detecting suspicious incidents.

4 SUPERVISED LEARNING TECHNIQUES

Supervised learning algorithms use labeled training data to formulate detection models, e.g., set of rules [101], separation plane [75], decision trees [39], neural network [61]. Later on, these mod- els are used to detect anomalies. Different classification techniques or classifiers are used to pre- dict anomaly, e.g., One-class Support Vector Machine OCSVM [133], the Hidden Markov Model HMM [138], ANN [61], to name a few. The rest of this section presents an overview of supervised learning-based IDSs followed by a comprehensive summary of different classification techniques.

4.1 Overview of Supervised Learning-Based IDSs

Although various IDS approaches have different processes, we describe the generic process in this section to give a general idea of how the detection model is trained and used to detect attacks. Figure 7 depicts an overview of the implementation of the IDS approach in practice. In particular, such systems consist of five main processes: (A) data collection, (B) feature extraction and selection, (C) tagging, (D) training, and (E) anomaly detection. Data collection (see Figure 7(A)) represents the measurement step, where the input data is collected, e.g., logs of events or system states, traffic trace or Net-flow data from a network monitor and so on. Figure 7(B.1) shows the feature extraction and feature selection processes, discriminating features are extracted and selected into a form

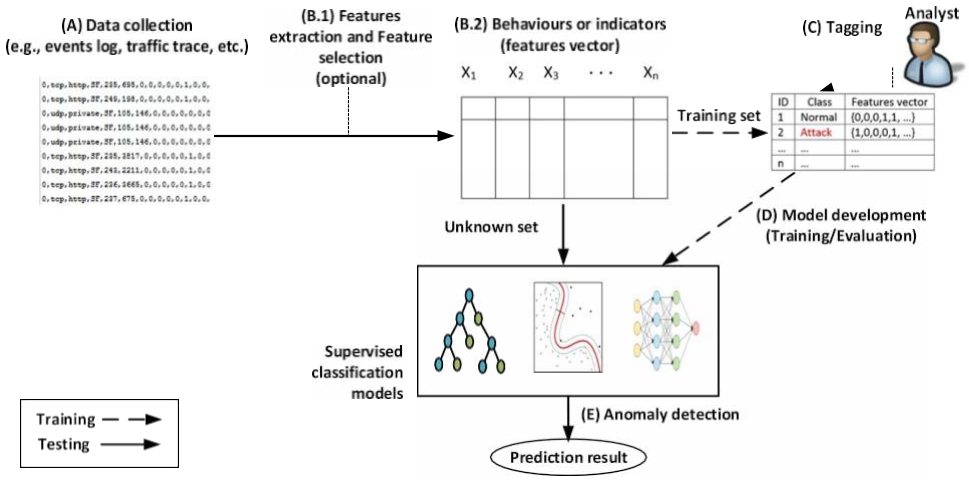

Fig. 7. The process of the supervised learning-based IDS approach.

that is usable in the classification process. The feature selection step might not be required for some machine learning approaches (i.e., ANN [61], that include network pruning and contrasting procedures which automatically select the discriminant features). In Figure 7(B.2), each data point (i.e., record) is represented using a feature vector which consists of attributes $x_1, x_2, \ldots, x_n$ (e.g., behaviors or indicators). Sometimes values of features require to be normalized at this stage to prevent feature with a large range (e.g., payload size) to overweight feature with a relatively small range (e.g., binary). For instance, Min-Max normalization [4] can be used to transform the numeric value v of feature x to v' that ranges between [0, 1] as follows:

$$v' = \frac{v - min_x}{max_x - min_x}, \tag{1}$$

where min_x and max_x are the minimum and the maximum value of the feature x. Note that, each data point may not always be a tuple with a specific number of features, some instances of a sample could also be a sequence of features. We elaborate this in more detail in Section 4.2. In Figure 7(C), each training record must be labeled to identify the class (i.e., *normal* or *attack* in the context of the IDS) that the data instance belongs to, either manually or automatically, using input from an analyst or expert. In several cases, the attack incident is hard to simulate or collect from the field. For these scenarios, IDS designers use only *normal* dataset to train the model (e.g., [32], [71], [81], [83], [123], and [133]). Figure 7(D) shows the model development process. The labeled dataset is used to train and evaluate the classification model. In order to improve the detection accuracy, parameters of the model could be adjusted, and the training/evaluation process is repeated until the anticipating efficiency is satisfied.

Once model training is complete, the model can be used to classify new instances of data. Figure 7(E) shows the process of detecting an anomaly using the classification model. Each unknown data item is labeled as either benign or malicious (most systems will provide lower-level detail for malicious points). Finally, the output of this module will be presented to the administrator to notify or take a response action.

4.2 Taxonomy of Supervised Learning Approaches

Even though there exist several machine learning classification methods/algorithms to detect anomalies [49], there is a lack of a taxonomy of these methods in the literature. A proper taxonomy

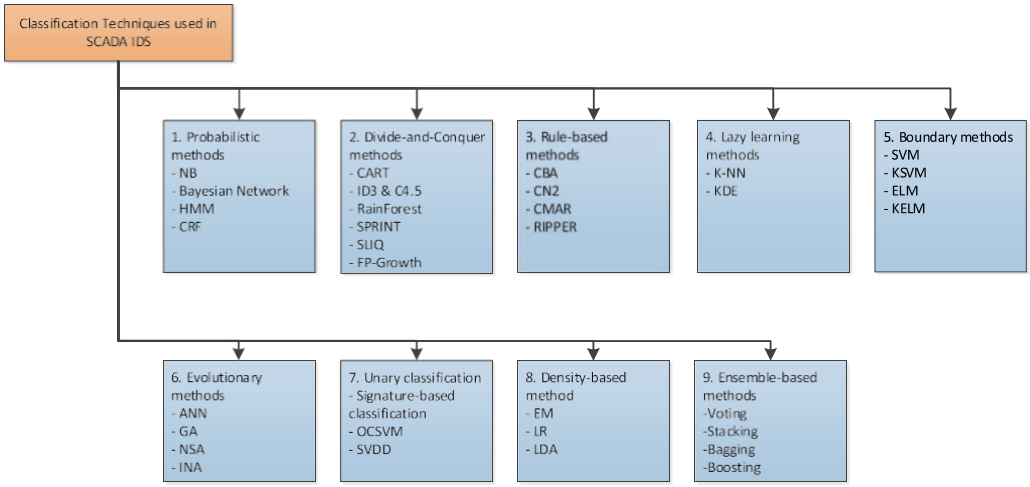

Fig. 8. Categories of classification algorithms used in SCADA-based IDSs.

is not only important to develop the evaluation matrix, but this also helps the separation between computational and architectural complexity of SCADA IDS systems. Hence, in this section, we present a structured view of classification approaches based on nine (9) categories (see Figure 8). Below, we describe each category in detail.

Probabilistic Method. This category predicts a class of unseen data based on probabilities. During the training phase, the presence of certain features of each class is used to calculate the distribution of joint or conditional possibility [112]. This number reflects the probability that a particular input falls into one of the predefined classes of objects to be classified. Examples of techniques in this group are Naïve Bayes (NB), Bayesian Network, Hidden Markov Model (HMM), and Conditional Random Field (CRF).

A basic probabilistic method like NB [117] learns conditional probability of each feature from the training data based on the assumption that every feature can be seen independently (i.e., conditionally independent). NB predicts the class of the input based on the highest joint likelihood. Bayesian Network technique [43] generalizes NB classifier by combining Bayesian variables with directed acyclic graph structure. The vertex in the graph depicts the variable (i.e., features and class), whereas the edge represents a probabilistic relationship between these variables. The graph is used to compute posterior probability of each class given all evidence.

On the other hand, some probabilistic models work on a sequence of features instead of the fixed number of features. For instance, HMM [111] inherits the concept of NB to predict the class of the datapoint based on a series of observable features over time. The assumption of HMM is that an observable feature is generated by the particular hidden system states. That is explained as a Markov process. The hidden state at time t depends on the previous state t_1. Once the joint probability of the future state is calculated, the likelihood of the observable sequence is determined, using knowledge of the predicted hidden states. HMM is used to classify labels of events based on sequences of features that are traced over the monitoring period. For instance, messages from a communication log are considered a sequence of features.

CRF is similar to HMM but more complicated. Instead of relying on a joint probability, the CRF works based on conditional probability [60]. As a result, CRF is more flexible. In other words, it includes a wide variety of overlapping features. However, this makes CRF less efficient in terms of computational complexity issue compared to HMM. Apart from that, when the newer data

becomes available, CRF model does not support re-training of the model. Therefore, it is not suitable for attacks that change over time.

Divide & Conquer Method. The divide and conquer technique includes (but not limited to) a broad category of algorithms known as the *decision tree*. It formulates a tree-like data structure from a set of attributes of each training tuple. Using a decision tree a rule set is easily derived, which can be used to classify the input into a particular group [79, 90, 101]. There are several algorithms in the group, such as Classification and Regression Tree (CART), Iterative Dichotomiser (ID3), C4.5 (an improved ID3), Frequent Pattern (FP) Growth, Supervised Learning in Quest (SLIQ), SPRINT, and Random Forest (RF).

CART consists of a set of decision rules, which is described using a set of if-else rules using a structure of *binary tree* [65]. The *node* of the tree represents a feature that is used to classify data points, while the *edge* represents decision paths. The path that connects more than one class is called a *sub-tree*. The path that yields only one class is called the *leaf node* and represents the final outcome. In reality, the classification node gives a mixture of instances from different categories; hence the feature that best discriminates the input data is chosen (so-called *splitter*). After that, the recursive process repeats at the next level of the tree until the classification reaches a leaf node. Since the method of finding the best splitter is based on the greedy approach (non-backtracking), the output decision tree might not be optimized. The tree pruning technique is the most commonly used [49] to reduce the size of the tree.

The advanced decision tree techniques like ID3 and C4.5 build a decision tree in the same way, but C4.5 has been improved over ID3 in various aspects. Specifically, C4.5 handles both continuous and discrete attributes. If the attribute $X \in \mathbb{R}$, the *splitting threshold* will be used to divide the data into two groups instead of the exact value of the discrete X. Apart from that, C4.5 allows the training data with some missing attribute values, and it supports tree pruning after creation. According to the research [79], C4.5 is faster and more flexible compared to its precedent ID3; therefore, it is a better choice for the context of the SCADA IDS that produces variety and a large amount of the monitoring inputs data. Scalability is an important issue of tree based classifiers as the size of tree is growing when the training data is larger [76]. Advanced decision tree techniques improve scalability and accuracy compared to the classic methods. For instance, IBM proposes SLIQ [76] and SPRINT [104] algorithms, which are based on a scalable method that splits parts of the tree from the memory to the database on a hard drive to address the scalability issue.

Aside from the decision tree techniques, the divide-and-conquer approach also includes a technique that works and is based on a frequent pattern, for example, the FP-Growth algorithm. Pan et al. [90] use FP-Growth to discover common communication patterns of the smart grid system. Then, the infrequently seen pattern is identified as an anomaly. The FP-Growth uses *FP-Tree* data structure to summarize patterns of events (i.e., nodes) that frequently occur together (i.e., edges of the tree). Although the FP-Growth technique is efficient, the data structure might be too large to fit in the main memory.

Rule-Based Method. This method uses a set of rules to determine classes of inputs. Undeniably, the rule is easier for a human to understand the reason behind a decision, compared to the probabilistic and numerical models. A rule generally contains condition and conclusion parts. The *conclusion* is the output of the classification, whereas the *condition* is the features of particular object that are being classified. The efficiency of rule-based method is measured by *coverage* and *accuracy*. Coverage shows how many tuples in the dataset satisfy by the *condition*. The *accuracy* determines the number of tuples that apply the *conclusion*. A definition of rules set is either specified by specialists of the particular system or mined from the training data using various supervised learning techniques.

With respect to automate rules discovery approach, a set of rules can be extracted from the training data. This process can be done by *sequential covering* algorithms or *decision tree* methods [49]. The sequential covering approach extracts rules from training data one-by-one in sequence, whereas the decision tree formulates a multiple tree-like structure of decision rules. There are many different rules mining approaches, such as Classification Based on Associations (CBA) [70], CN2 [27], Multiple Association Rules (CMAR) [67] and Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [28].

The simple method like CBA [70] selects the most effective rules that identify a class of each tuple from the training set. The CBA algorithm learns a Class Association Rule (CAR) from the training set D, defined as $x_i \rightarrow C_i$, where the feature x_i implies a class C_i. The rule holds support $sup\% = |x_i| / |D|$ and confident $conf\% = |x_i \cap C_i| / |D|$, where $|x_i|$ depicts total number of tuples with the attribute x_i, $|x_i \cap C_i|$ means total number of attribute value x_i that implies the class value C_i, and D denotes total number of tuples in the training set. These are minimum thresholds (i.e., $sup\%$ and $conf\%$) that determine if the rule should be selected for the classification. Although the selected rules are the most effective, they might not be the best discrimination. In addition, a scalability issue is an important limitation. That is, the rule-based method scales poorly with the large training data, especially when outliers cannot be avoided compared to the decision tree method (i.e., C4.5) [28].

The more sophisticated approaches like CMAR and RIPPER, on the other hand, aim to increase the robustness of the approach by eliminating impurities from the training data and reducing the number of rules generated from the training phase. CMAR [67] extends FP-Growth technique [90] to formulate a Class-Association Rules (CR)-Tree and minimizes memory space requirement (e.g., by using tree pruning technique); hence, CMAR has a better classification accuracy and scalability compared to the classic CBA and C4.5 algorithms. RIPPER [91] aims to address the impurity issue by proposing an iterative pruning technique for reducing error and allow a large and noisy training dataset to be used. More specifically, the training data is divided into *growing* and *pruning* sets. The growing part is used to formulate a rule-set, whereas the new set of rules is pruned immediately with the pruning dataset until there is no improvement from the pruning step. The new set of rules is integrated with a rule-set from the previous iteration.

Lazy Learning Method. This method differs from others in terms of the learning process. The main idea behind the technique is to update the training model as late (hence, lazy) as possible. For example, the training data is stored in the memory without building a prediction model like an eager learning technique. The prediction model is only built during the classification step. It offers both advantages and disadvantages. Even though the lazy learning method offers the best performance at the learning step, this advantage is a tradeoff of computational complexity when it is making a prediction or classification. However, it is more flexible than others because the trained model can be incrementally improved [108, 114].

The most commonly used lazy learning technique is the k-Nearest-Neighbor (k-NN) technique [105]. It compares the input with k data points using Euclidean distance function. An advanced method, like Kernel Density Estimation (KDE), implements k-NN-based approach (e.g., nearest neighbors, reverse neighbors, and shared neighbors [102]) as a kernel function that is used to formulate the density function (i.e., multivariate Gaussian [114]) of the *normal* data; hence, the boundary of the *normal* can be described using parameters of the density function. Given that one data point contains n features, the anomaly is detected by projecting the data point into n-dimension space of features. The data point that resides beyond the *normal* boundary is considered an anomaly.

Boundary Method. Unlike the lazy learning method, this method considered an eager learner. Support Vector Machine (SVM) [29] is the most common technique in this category. It transforms the training data into a decision boundary (so-called *hyperplane*). The idea of hyperplane is that the training data with *non-linear mapping* is transformed into an adequate higher dimension, where two classes can be linearly separated by the hyperplane. Also, the algorithm aims to maximize the width of the separation plane. Despite the training time being very long, this approach dominates other methods by being less prone to over-fitting on the training data. The SVM-based solution is quite common for SCADA-IDS. In practice, it is combined with other techniques (such as *pruning* and *kernel-trick*) to archive a higher detection accuracy [75], lower false positive and negative alarm [92], minimize an offline learning phase [93], or to apply with environment where input features are limited [122].

Another example of the boundary method includes technique in the research [74]. The proposed technique is used to identify anomaly for proprietary communication (e.g., vehicle CAN bus protocol). Since the communication protocol is not disclosed by manufacturer, it is challenging to detect attacks on those critical system. Markovitz and Wool [74] identify field boundaries based on the pre-defined knowledge of control variables and the actual control signals from various operations situations. After that, they are able to train a classifier, which is used to detect anomaly. Although the proposed technique is specific to the particular control system, the same principle could be applied when communication specification is unknown.

Evolutionary Method. The evolution of a living organism inspires the development of artificial intelligent techniques. Various techniques have been proposed for SCADA-IDS (e.g., [61], [68], [69], and [100]). There are two well-known techniques, namely, Artificial Neural Network (ANN) and Genetic Algorithm (GA). ANN uses a brain-like structure (i.e., a network of neural cells) for classification. The neural network consists of three different layers of nodes, namely *input layer*, *hidden layer(s)*, and *output layer*. The input layer can contain several nodes depending on the input data (e.g., number of features of each tuple). Every input node has connections with every nodes in the hidden layer. The connections have different weight values; these weights are fitted during the training period in order to provide the most accurate output according to the supervised knowledge. There can be more than one hidden layer of the neural network. Even though ANN gives a high prediction accuracy in various applications (e.g., image/voice recognition, misuse behavior and anomaly detection), the explanation of how ANN works is still controversial [20]. Researchers do not fully understand how the ANN classifies the input data, and designing of the ANN topology (i.e., number of nodes and hidden layers) is still trial and error. The output layer could be only one node in case of a binary classification problem. That is, giving output between 0 and 1, which serves as a likelihood of being a particular class. For multi-class classification problem, the number of output could be k nodes, represent the prediction of k-classes. Each output node indicates likelihood of each class. Important limitations of ANN technique are the requirement of a large training dataset and the trained model might be over-fitted to the training set; hence, it is not applicable in some contexts. For example, recording all possible attack data from the real SCADA system.

The GA algorithm [66], on the other hand, simulates the process of selective survival in the evolution theory. It assumes that the knowledge (e.g., anomaly detection rules) can be represented as a *chromosome* of the living things. These chromosomes can be optimized according to *evaluation objectives*. In context of IDS, design of *evaluation function* is crucial to optimize the *fitness* of the outcome. GA is well known for minimizing the effect of erroneous training sets and sometimes overcomes the problem of multiple local optima. Other aspects of nature are also applied in anomaly detection. Negative Selection Algorithm (NSA) or Immune Network Algorithm (INA)

[68] formulates the detector module for the classification between normal and abnormal data. The idea was initially derived from the organic process where immune cells detect harmful agents in our body.

Unary Classification. Unary or one-class classification (OCC) solves the binary or multi-class classification problem by primarily learning from a training set of one class only (i.e., *normal* system states); hence, the anomaly can be detected as the class of *others*. This approach advantages other solution when one class of the training data can be clearly observed, while information of other classes is severely hard to record.

The most common algorithm in this category is One-Class Support Vector Machine (OCSVM) (so-called *Support Vector Data Description: SVDD*), e.g., [32], [71], [81–83], [123], and [133]. Therefore, the basic principle of the SVDD technique is similar to the SVM technique, discussed previously. Instead of separating between two classes using a linear hyperplane. In higher dimensional space of the data, SVDD formulates boundary function of a spherically shaped that minimally covers the complete population of the target class [115]. Let the sphere is described by the center a and radius $R \geq 0$, SVDD aims to minimize volume of the sphere by minimizing value of R^2 described in the *error function* as,

$$\begin{aligned} &\text{minimize } F(R, a) = R^2, \\ &\text{subject to } \quad \|x_i - a\|^2 \leq R^2, \forall i, \end{aligned} \tag{2}$$

where x_i denotes distance between the data point i and the center a. However, the perfect spherical shape could also include outliers from the training set (hence, not optimized). According to [115], the separation plane is formulated by resolving the following optimization problem:

$$\begin{aligned} &\text{minimize } \quad \frac{1}{2}\|\omega\|^2 + \frac{1}{vr}\sum_{i=1}^{r} \xi_i - \rho, \\ &\text{subject to } \quad (\omega \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0, i = 1 \cdots r, \end{aligned} \tag{3}$$

where $x_i \in X$ is a data point out of total r samples in the training set X, and $\Phi : X \rightarrow H$ denotes the mapping function from raw to the high-dimensional space. The normal vector and compensation parameter of the hyperplane H is denoted by ω and ρ, respectively; $v = (0\ 1)$ is a tradeoff parameter that controls proportion of support vectors in the training set. Lastly, ξ_i represents the slack variable that allows some training samples to be incorrectly classified. This optimization problem is solved using Lagrange multiplier, which can find detail in the further reading [115].

Density-Based Method. This method builds an estimate function from statistical data based on the training set. It can be used for both clustering and classification problems. Examples of algorithms in this group are Expectation Maximization (EM), Logistic Regression (LR) and Linear discriminant analysis (LDA). EM is an iterative approach, and it works well on incomplete training data [33]. In the training period, it adjusts parameters of *likelihood function* based on the data point from the previous round. The aim is to optimize settings of the likelihood function. This process repeats on the next round until the stopping criteria are met (e.g., difference of output is zero or remain unchanged). Since the EM technique maximizes the accuracy of the classification function, the advanced IDS solution [135] incorporates EM technique into one-class classifier to reduce the outlier from the training data set in the pre-processing phase to get the more accurate classification model.

Logistic regression (LR) [59], on the other hand, is mainly used for binary classification problem. It maps value of each feature, e.g., payload size $x = (1\ 100)$ of the network traffic, with likelihood of the predicting class (say *malicious* and *benign*). Basically, the technique tries to fit all data

points in the sigmoid curve by shifting the line and re-calculate the likelihood until the maximum likelihood value is found. An important limitation of LR technique is when classes have a *complete separation*. That is, some features can separate two classes; hence, the binary function cannot be used to classify the input.

The more advanced technique like LDA [77] is suitable for classifying data with multiple classes and multiple features (so-called *dimension*). Data point x with n features can be plotted on a n-dimensional space to find a separation point, separation line, or separation plane, when $n = 1, 2, 3$ respectively. In reality, data might have more than 3 dimensions. Hence, it is complicated to calculate the separation plane directly. LDA resolves this problem by reducing complexity of data dimensions. In case of binary classification, it creates a new axis and projects all data points onto the new axis. Regardless of the feature dimension, data points of two classes can be separated on the new axis. The new axis is formulated from the training data by maximizing the combination of *means* of data points in each class and minimizing combination of *between and within scatters*. The density-based method, however, performs well only if a larger number of training data points are available.

Ensemble-Based Method. The idea behind this method is to combine two or more methods to improve the accuracy of the individual methods. The most popular methods are *Voting*, *Stacking*, *Bagging,* and *Boosting*. This method requires other techniques to be used as base methods. These can be any of the aforementioned approaches, different algorithms, or their variations.

Voting is the simplest technique to decide a final prediction result from multiple voters. The majority of the votes will be chosen as a final decision. Votes are gathered from a collection of classifiers. *Stacking* is a more advanced form of voting. Instead of taking the majority vote, it uses a meta-learner to justify the best result based on the supervised knowledge. In other words, outputs from a collection of first-level classifiers are fed to a second-level learning algorithm. The meta-learner is trained to optimize the final prediction [107].

Take the research in [47] as an example. The decision-tree-based classifier and the rule-based classifier are stacked to detect any anomaly from a large network traffic log efficiently. First, the C4.5 decision tree is used to detect known network attack events from all communication traffic. The attack events are classified using C4.5 technique. Meanwhile, the normal traffic is further examined by a CBA classifier, which is trained with normal data. Therefore, the well-known anomaly is quickly filtered out and the unknown log is examined at the second stage using the different classifier.

On the other hand, *Bagging* and *Boosting* focus on distribution of the training data. This is because the combination of independent bases can dramatically improve the efficiency of final prediction. *Bagging* (so-called *Bootstrap Aggregating*) obtains a data subset for different baselines by using the bootstrap sampling technique [36]. The different base classifiers are aggregated using the *voting* and *averaging* technique to improve prediction accuracy. *Boosting* [139] improves the accuracy by building the stronger classifier from an existing weak classifier. Suppose a distribution of data D consists of three parts X_1, X_2, and X_3, and we only have a weak classifier that correctly predicts only X_1 and X_2. Let the wrong classification X_3 be denoted by h_1. In order to correct the mistake made by h_1, the boosting technique derives a new distribution D^J from D. For example, the researcher should focus more on instances in X_3 and then train a classifier h_2 from D^J. Suppose that the new classifier has the correct classification in X_1 and X_3. We can now combine classifiers h_1 and h_2 to get a stronger classifier. The process is repeated by adjusting the distribution's parameters until no improvement can be made.

Despite the complexity of incorporating several methods, this technique has been used in a large number of works (e.g., [23], [31], [39], [62], [63], [94], [72], [106]). The algorithms included not

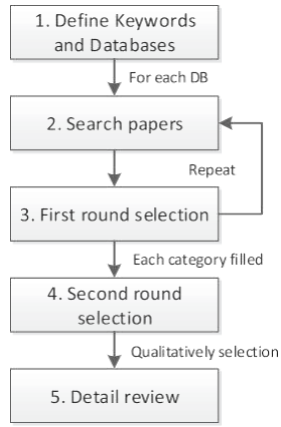

Fig. 9. Paper Selection Process.

only classification-based but also clustering-based techniques. The selected paper in each category is evaluated in Section 5 based on the proposed holistic-analytical evaluation method.

5 EVALUATION OF SUPERVISED LEARNING APPROACHES FOR SCADA-BASED IDS SYSTEMS

This section provides a detailed review of SCADA-based IDSs in each of the categories presented in 4.2. We describe the paper selection process, followed by the evaluation criteria of each selection. We collected information of proposed solutions with regard to quantity (i.e., the number of publications that used a similar approach) and quality (e.g., benefits, drawbacks and constraints). This information is used to evaluate supervised learning approaches with respect to holistic perspectives from each criterion.

5.1 Paper Selection Process

In order to select the relevant and important papers from SCADA-specific IDS research, we followed the literature review protocol illustrated in Figure 9. In step 1, we chose key databases in information technology publications to find papers, namely Association for Computing Machinery (ACM) Digital Library, IEEEXplore (IEE/IEEE), SCOPUS (Elsevier) and Web of Science. Then we defined keywords in three topics to locate potential papers in each database as follows:

SCADA: *SCADA, Smart grid, Critical Infrastructure, Industrial Control System, ICS*
Intrusion detection: *Intrusion Detection, IDS, Anomaly Detection*
Machine Learning technique: *Machine learning, ML, classification, classifier*

For each database in steps 2 and 3, we searched and selected the papers using the snowball sampling method [19], which starts with a small set of highly relevant papers, and then follows its references to build a larger pool of papers to review. The snowball sampling approach was chosen because some important and relevant research might not be indexed by the databases or the search engines that were used. As illustrated earlier in Figure 2, the number of publications on SCADA-specific IDS based on supervised learning solution has risen significantly since 2007, which reflects the renewed and increasing public interest in the protection of critical infrastructure. Hence, we limited the time range from 2007 to present. We formulated search strings suitable for each database based on the keywords above. Next, we ran the queries, assembled the located

Table 1. Evaluation Criteria for SCADA-Based IDSs

Criterion	Reasons
Algorithm	Review the most and least popular classification techniques for SCADA-based IDSs studied in the literature with associated benefits, drawbacks, and constraints.
Approach	Compare the literature from the accuracy and flexibility point of view. Although the signature-based method precisely detects attacks, the anomaly-based approach is more flexible in detecting future attacks such as the zero-day exploits.
Architectural properties	The architecture design directly reflects how well it is tailored for SCADA systems. To be specific, we examine the system based on scalability, availability, distribution capability, and robustness of the systems.
Auditing sources	Show how far IDS can cover the different types of attacks on the resource constrained environment. Some works detect the anomaly based on the communication behavior only, while others consider the physical state or multiple variables in combination.
Application domain	Indicate the holistic design of the security system that ranges from anomaly detection and supported decision making to investigation support.
Feasibility	Show the distance between simulation and reality. As SCADA is deployed on the critical infrastructure, the simulation and the actual environment are different. Therefore, testing on the real world situation or the SCADA-specific simulator is crucial.

papers, and removed the duplicate papers. The paper abstract and keywords in the located papers were chosen to be related to the three main topics defined above. We then sorted the located papers by main supervised learning algorithms described in Section 4.2.

In step 4, the papers were examined more closely for categorization. Key inclusion criteria were relationship to SCADA security; use of IDS; use of supervised learning to train IDS; quality threshold—clear methodology described; technique implemented and evaluated; and paper published in a journal or conference proceedings that we were able to access the full text. If there were several papers in a particular category, the journal publications were considered with a higher priority compared to conference proceedings. This was because the journals have less restriction on the number of pages and multiple revision cycles, hence they discussed the proposed approaches more comprehensively. The conference papers were also selected based on the relevancy and ranks of the conferences suggested by the Computing Research and Education Association of Australasia (CORE) [116].

5.2 Evaluation Criteria

SCADA-specific IDSs differ in their design (i.e., auditing sources and monitoring attacks) and targeted application (i.e., topology and components); therefore, we cannot solely justify the compatibility of IDSs based on the statistical metrics (i.e., accuracy, specificity, sensitivity). For this reason, we have used a qualitative comparison instead.

In order to understand the challenges of SCADA-based IDS with supervised learning techniques, we propose some evaluation criteria to measure the effectiveness and efficiency of the proposed algorithms with respect to the requirements of SCADA-based IDSs discussed in Section 2.4. Table 1 summarizes the key criterion of each measure and the reason behind its selection.

5.3 Categories of SCADA-Based IDS Systems

Rule-Based Method. As shown in Table 2, the rule-based technique is one of the most common technique among all categories from the selected set of publications in this survey. With respect to the detection approach (see Table 2 column *Approach*), more than half of the proposed solutions in the literature use signature-based approach. In most cases, experts are required to establish a set of detection rules, especially, to learn about the known normal/abnormal states associated with SCADA-specific hardware, e.g., IED [95, 130], Generic Object Oriented Substation Events (GOOSE), Sampled Measured Value (SMV) protocols [132], and behavior of automated process [37].

In general, the predefined rules are functions of normal states rather than fixed constants. Take the research in [95] as an example, they define misuse detection rules using multiple alert functions, namely correlation between switching devices, alarms from relay protection function, time-related constraints of critical control commands, and payload length detector. We observed fewer papers in this group (e.g., Rough Sets Classification [56] and RIPPER [91] techniques) that used automated rules set discovery technique compared to the signature-based approach. Apart from that, special "honey" tokens can also be applied to detect tampers with the communication traffic [17].

Since rule-based method is the easiest technique to understand and customize manually by a human, this technique is suitable for specific environments like building an automation control network (i.e., a fire alarm system) [91], SCADA RTUs [30], and micro-phasor measurement units (μPMUs) [56]. These systems require specialists or engineers to define associated rules that describe normal states of the system; therefore, the rule-based technique is more efficient compared to techniques in the other categories. Additionally, the automated detection approaches can also be integrated into the rule-based solution to improve flexibility in detecting unknown malicious incidents.

Ensemble Method. This technique also commonly appears in the literature as ensemble approaches build the classification model by selecting the best result from multiple classifiers. Despite the complexity of this method, a number of works [23, 31, 39, 62, 63, 72, 94, 106] have successfully used it to obtain better detection rate accuracy. Apart from accuracy, the ensemble method also has the advantage of prediction models that are robust and resistant to a system failure from the malicious cyber event [15]. Also, the research [62] overcomes the insufficient training data issue by combining the result from OCSVM and the k-means clustering technique to increase the detection speed and achieve real-time performance. In [94], the network telemetry (e.g., packet size and time of arrival) is used instead of the actual network data. Various telemetry features are taken into account. The boosting technique creates a strong classifier out of several weak ones. On the other hand, multi-class classifiers (e.g., HMM and SVM) favor big and small classes differently. Hence, the bagging method can be used to find the most effective classifier for the particular class, which is chosen to give a label to the detecting event [106].

Unary Classifier. In some cases [32, 71, 83, 123], unary classifiers, such as the OCSVM, have outperformed other classification methods in terms of accuracy. Unary classifiers work well when there is only one class of data available. Since there is a lack of datasets containing real SCADA attacks, the OCSVM classifier can be used to train only using the normal data [134]. Hence, OCSVM is more popular than the original SVM, which requires both normal and abnormal training set [32]. However, a constraint of unary classifier is that the training dataset must not contain attack instances. It is possible in a real-life scenario that, there could be some 0-day attacks that remain undetected [97].

Probabilistic Method. Probabilistic classifiers are useful to provide an estimate a probability distribution over a set of classes, rather than a single class that the observation should belong to.

Table 2. Comparison of Methods for SCADA-Based IDS Groups by Classification Techniques

Category	Authors	Algorithms	Approach			Architecture			Auditing					Application			Feasibility						
			Signature-based	Anomaly-based	Hybrid approach	Scalable	Real time detection	Decentralized/distributed Resilient	Network traffic	SCADA specifc protocol	Application behavior	Physical state	Unified cyber-physical	Detection	Prevention	Investigation	Simulation/testbed	SCADA-testbed	Prototype	Real-world deployment	Open source	Portable
Probabilistic	Tylman W. [117]	NB	✓	✓			✓		✓	✓				✓			✓				✓	
	Zhou C. et al. [138]	HMM	✓	✓			✓		✓	✓				✓	✓		✓					
	Hosic J. et al. [51]	Genetic Programming	✓	✓			✓		✓	✓				✓			✓					
	Zohrevand Z. et al. [141]	HMM	✓	✓				✓				✓		✓	✓		✓					
	Stefanidis K. and Voyiatzis A. G. [111]	HMM		✓	✓		✓		✓	✓				✓			✓					
	Andrysiak T. et al. [9]	MLE	✓				✓		✓					✓						✓		
Divide & conquer	Pan S. et al. [90]	FP-growth			✓	✓			✓			✓		✓			✓					
	Samdarshi et al. [101]	C4.5, RandomForests			✓				✓			✓		✓			✓				✓	
	Moon D. et al. [79]	C4.5	✓						✓	✓				✓					✓			
Rule-based	Coutinho M. P. et al. [30]	Rough Sets Classification		✓			✓					✓		✓			✓					
	Premaratne U. K. et al. [95]	Rules enumeration	✓						✓	✓		✓		✓	✓		✓					
	Yang Y. et al. [130]	IEC 60870-5-104 Signature	✓						✓	✓				✓			✓					
	Asif M. K. and Al-Harthi Y. S. [17]	Signature-based	✓				✓		✓	✓				✓			✓					
	Yang Y. et al. [131]	If-then rules	✓				✓	✓	✓	✓				✓				✓				
	Erez N. and Wool A. [37]	Single Window Classification		✓			✓		✓			✓		✓			✓					
	Yang Y. et al. [132]	Rule-based	✓				✓		✓	✓		✓	✓	✓	✓			✓				
	Pan Z. et al. [91]	RIPPER		✓			✓		✓	✓				✓	✓			✓				
	Jamei M. et al. [56]	Rule-based	✓			✓			✓			✓	✓	✓	✓					✓		
Lazy learners	Silva P. [108]	K-NN		✓					✓	✓				✓				✓				
	Tang B. and He H. [114]	RDOS		✓					✓					✓			✓					
Boundary	Masduki B. W. et al. [75]	SVM		✓					✓					✓			✓					
	Patrascu A. and Patriciu V. [93]	SVM		✓				✓	✓					✓	✓				✓			
	Vijayanand R. et al. [122]	SVM		✓				✓	✓	✓				✓			✓					
	Patel A. et al. [92]	SVM		✓		✓	✓		✓	✓				✓	✓	✓	✓					
	Markovitz M. and Wool A. [74]	Field Boundary		✓			✓		✓	✓				✓						✓		
Evolutionary	Linda O. et al. [69]	IDS-NNM		✓					✓					✓	✓		✓					
	Lima A. D. P. et al. [68]	NSA		✓			✓					✓		✓			✓					
	Kosek A. M. [61]	ANN		✓					✓			✓	✓	✓			✓					
	Shitharth S and Prince Winston D [100]	IWP-CSO with HNA-NN		✓					✓					✓			✓					
Unary based	Yasakethu S. L. P. et al. [133]	OCSVM		✓			✓		✓					✓	✓					✓		
	Nader P. et al. [81]	OCSVM		✓			✓					✓		✓				✓				
	Nader P. et al. [82]	SVDD		✓								✓		✓			✓					
	Maglaras L. A. and Jiang J. M. [71]	OCSVM		✓			✓		✓	✓				✓			✓					
	Nader P. et al. [83]	OCSVM		✓					✓	✓				✓			✓					
	da Silva E. G. et al. [32]	OCSVM		✓		✓	✓		✓	✓				✓					✓			
	Wan M. et al. [123]	OCSVM		✓			✓			✓	✓			✓			✓					
Density based	Yoo H. and Shon T. [135]	EM		✓			✓		✓	✓				✓			✓					
Ensemble	Faisal M. A. et al. [39]	HoeffdingTreeNB, LimAttHoeffdingTreeNBAdaptive, HoeffdingTreeNB and HoeffdingTreeNBAdaptive		✓		✓	✓	✓	✓	✓				✓			✓				✓	
	Branisavljevic N. et al. [23]	PCA, ANN, OCSVM, others (6 methods)		✓			✓					✓		✓					✓			
	Shahir H. Y. et al. [106]	HMM, SVM		✓		✓	✓					✓		✓			✓			✓		
	Ponomarev S. and Atkison T. [94]	REPTree,NB,Simple Logistic, Ripple-Down Rule, Decision Stump, C4.5		✓			✓		✓	✓				✓				✓				
	Maglaras L. A. et al. [72]	IT-OCSVM and SNA		✓			✓	✓	✓	✓				✓				✓				
	Cruz T. et al. [31]	OCSVM, SNA and K-means clustering			✓		✓	✓	✓	✓				✓			✓					
	Kosek A. M. and Gehrke O.[62]	RM-AD		✓					✓			✓	✓	✓			✓					
	Sadhasivan D. K. and Balasubramaninan K. [63]	FCM clustering and RBA		✓			✓	✓	✓	✓				✓	✓		✓					
Benchmarking	Ozgur A. and Erdem H. [89]	SVM, NB, DT		✓					✓	✓				✓			✓					
	Hurst W. et al. [52]	LDC, UDC, QDC, PARZENC and TREEC		✓								✓		✓			✓					
	Swetha R. B. S. and Meena K. G. [113]	DT, K-NN, SVM		✓			✓	✓	✓	✓				✓			✓					
	Junejo K. N. and Goh J. [57]	Multiple		✓		✓	✓					✓		✓			✓					
	Onoda T. [88]	HMM,CRF, OCSVM, SVDD, Rule-based		✓					✓					✓			✓					

HMM is the most popular option in this group. Although HMM is computationally efficient and flexible to retrain the model when the updated data is available, it cannot capture higher order correlation of the data. This is because of a strict states dependency assumption of HMM. For instance, the same sequence of features could indicate different states depending on different contexts or intentions. This issue is known by researchers [88, 138, 141]. A technique like N-gram is used to preserve contextual meaning by grouping N sequence of observations together. Apart from that, multiple independent factors (e.g., *Task and resource models*, *Control data flow models*, and *Critical state of critical processes*) are correlated to increase prediction accuracy and decrease the false alerts.

Other Methods. The remaining techniques proposed in the literature are across various categories of algorithms, such as divide-and-conquer (i.e., [79], [90], and [101]) and boundary methods (e.g., SVM [75, 92, 93, 122] and Field Boundary [74]). However, lazy learning (i.e., [108] and [114]) and density-based (i.e., EM [135]) are occasionally used for supervised SCADA IDS. Some constraints have been learned from applying these solutions. For example, the lazy learning method does not scale enough for the large network [108] and the density method still lacks accuracy to be used in the real field [135].

5.4 Approaches Used to Detect Anomalies

Based on the *approach* column in Table 2, published works are categorized into three types: *signature-based*, *anomaly-based* and *hybrid*.A *signature-based solution* is specifically designed for a particular system or protocol, as discussed in Section 5.3. Due to the constraint of using the rules, only known attacks, such as MITM attack and ARP cache poisoning, are analyzed to evaluate the detection efficiency [131]. Even though these tasks require the knowledge in system protocols, operations, and specific characteristic of attacks, it usually offers a low false alarm rate compared to the machine learning solution, hence more practical for industry. Besides, the signature-based method requires a footprint of attacks to be updated regularly and cannot guarantee the new intrusion threats such as zero-day exploits [73]. Therefore, we observed that most of the proposed works are anomaly-based detection.

It is hard to keep the detection rules up-to-date as new vulnerability is emerging regularly. The *anomaly-based approach* focuses on building a model of normal/abnormal behaviors instead of defining rules. In constructing the model, various machine learning algorithms were used (see column *Algorithms* in Table 2). However, the decision of what algorithm should be used not only depends on characteristic of auditing sources (e.g., network traffic, application usage behavior, or physical state of the actuator; see column *Auditing*) but also vary by the character of data and constraints of applications. With network traffic dataset from DARPA [134], different algorithms are applied. For instance, Onoda [88] used the HMM algorithm to focus on characteristic of sequential behaviors, whereas SVM is used in [75] and [92] with selected features of network communication data (considered as a data-point) to build the optimal separation plane between the center of two classes, which is used to distinguish between benign and anomaly incidents. Meanwhile, Sadhasivan and Balasubramaninan [63] aim to develop an adaptive IDS, which can update the attack information over time. The proposed work fuses ADA (Anomaly Detection Agent) and RBA (Rule-Based Agent) to detect misuse from the network traffic data. Based on the same dataset, an ensemble technique is considered [39] to eliminate noises from the training or testing dataset. With more than one classifier, the boosting technique selects the best prediction results from classifiers, hence accuracy is improved.

The *hybrid method* combines both the system-specific signature and the behavior-based detection model, for example, a defense-in-depth strategy in [111]. The IDS system consists of "Header

subsystem" and "Data subsystem", where the signature of abnormal Modbus protocol header is defined in the first module and the HMM is used to detect (in depth) attacks from traffic data in the second module. Combination of the signature and behaviors models helps to speed up the detection and highlight the specific type of attacks. Similarly, Distributed Intrusion Detection Sys- tem (DIDS) system in [31] handles known exploitations with signature definition while using the anomaly-based approach to prevent the emerging threats. Despite tight coupling to the system and the complexity of combining two detection systems, the number of hybrid-methods is quite small compared to other approaches.

5.5 Architectural Design Properties

The decision between centralized and decentralized architecture is always controversial. On the one hand, the centralized IDS is easy to monitor and make a decision from the central location without limitations of resources; hence, the more complicated detection/classification tasks can be done. On the other hand, the decentralized or distributed IDS architecture is more scalable and resilient compared to the centralized scheme [38]. According to the requirement of SCADA- IDS in Section 2.4, here, the architectural perspective is discussed based on four (4) key aspects:
(1) scalability of the algorithm, (2) real-time performance, (3) decentralization, and (4) resilience, as follows:

Scalability of Algorithm. We have not observed many papers that explicitly state scalability property of their work. For instance, the work in [32] proposes the OCSVM-based algorithm, where the sub-model learns heterogeneous normal training sets to detect outliers of the large and diverse system. Apart from that, the cost of maintenance is important for scalability design. Pan et al. [90] developed a scalable stateful IDS to prevent temporal attacks. Besides, enlarging the existing complex system is a costly and difficult task. The scalable system helps to save lots of money and time for maintenance tasks. Hence, the hybrid approach was used. First, the signature is used to reduce irrelevant events. Second, the anomaly detection model examines the suspect events more closely. The proposed common path mining algorithm mines data from both the physical (synchrophasor) and the logical (system logs) algorithms to formulate signatures of attacks (common paths). This fusion technique identifies attacks from abnormal states which shared in the common paths.

Real-time performance: this is a crucial property. Based on Table 2: column *Architecture*, it is clear that most of the literature focuses on real-time detection characteristics to avoid reduction of a system's availability. For instance, the critical infrastructure like railway traffic control. This system needs to be monitored at real-time. In [9], the traffic control data gathered from WSN (Wireless Sensor Network) is modeled using ARFIMA (Autoregressive Fractional Integrated Moving Average) technique, which analyzes the deviation between parameters of the network traffic and creates a statistical model for the system. The MLE (Maximum Likelihood Estimation) algorithm is used to detect the anomaly in this control system.

Parallel processing is needed to archive real-time performance. NIDS (Network Intrusion Detection System) in [123] has been tested on the simulated Modbus/TCP system. Two OCSVM-based algorithms were proposed to analyze both control and process data simultaneously. The detection process consumed less than 27 seconds to detect attacks with a small, medium, and large num- ber of abnormal function control behavior. On the other hand, as parts of the CockpitCI project, the Domain-specific IDS for SCADA ICS (Industrial Control System) [31] focuses on decreasing the false positives of OCSVM by proposing IT-OCSVM algorithm. By running several OCSVMs in parallel, the final outcome is chosen using the mean value method and SNA (Social Network Analysis). They monitored three layers (data-link, network and transport protocols) to detect attacks, e.g., MITM and DoS (Denial of Service).

Decentralization. The design of distributed IDS is arbitrary and specialized for a particular system. For instance, the IDS for AMI (Advanced Metering Infrastructure) in [39] distributes IDS modules into three components: Meter-IDS, Data Concentrator-IDS, and Headend-IDS, from a small to a large stream of information, respectively. The IDS filters attacks from the smallest unit (Meter-IDS) to the largest (Headend-IDS) serially. However, the proposed system requires the whole network to be isolated and free from noise (i.e., communication between other devices in the network) to achieve the best detection accuracy; hence, it is suitable for Software Defined Network (SDN) SCADA. On the other hand, the Distributed IDS has been used in [31]. With several IDS distributed through the wide network, they are not only able to identify attacks from different parts of the system but also increase system robustness in case of failure that causes sub-networks to disconnected from the central server. The DIDS breaks the whole network traffic data down to subsets of the disjoint dataset. By using the IT-OCSVM technique and weighting technique (i.e., voting between results from the classifiers to get the best prediction outcome), the proposed DIDS is able to increase detection accuracy while minimizing the rate of false alarms.

From a different angle, da Silva et al. [32] points out that the distributed NIDS processes the detection in parallel and is suitable for a large-scale SDN SCADA. The proposed NIDS consists of five (5) components SDN controller, historian server, feature selector, one-class classifier, and NIDS management interface. Each component is responsible for different tasks. For example, the SDN controller monitors anomalous flows in network switches, while the historian server stores snapshots of networks to be able to apply the parallel MapReduce operation. The proposed One-Class Classifier (OCC) works well with a large dataset, it uses only one set of normal network data to detect the DoS attack on Modbus protocol.

Resilience. A formal definition of resilience is the ability of a system to recover from faults and return to its original state or other working states. Interestingly enough, only a limited number of works discussed resiliency of IDS systems. In [93], the time-out has been used to discard unresponsive classifier workers. The proposed work integrates game theory and data streaming classifiers (which work in parallel) to heuristically detect various type of attacks. The IDS feeds abnormal events into the game model to determine win conditions between attacker and defender; hence, the IDS can detect the unknown threats by learning for the existing data. The result shows detec- tion accuracy is better than using KNN and decision trees techniques. Another example has not explicitly specified the resilient characteristic in this article. We assume that the distributed IDS with parallel detection components like [39] resists failures. Since a number of IDS components are duplicated and work independently, some faulty nodes would not affect the whole system.

5.6 Data Sources Used for Anomaly Detection

Several sources of information can be inspected to identify misuse or abnormal events in a SCADA system. Literature in this review is categorized into four (4) groups for auditing data: (1) Network traffic, (2) Physical state, (3) Application usage behavior, and (4) Unification of cyber-physical state. Since most works chose to monitor network traffic, we separate out *SCADA specific protocol* to check if the IDS uses generic communication features or SCADA specific protocols' features. This is shown in column *Auditing* in Table 2.

Most network-based IDS solutions were tested with *SCADA-specific datasets* or a *combination of generic and SCADA-specific datasets*. In [63], both types of communication data are used to train various functionalities detection agents (namely Sniffer Agent, Filter Agent, Rule Mining Agent, Anomaly Detection Agent, and Rule-Based Agent). The proposed multiagents IDS shows a better detection performance on the SCADA-specific dataset [80] compared to the generic KDD CUP 99 dataset [15]. Meanwhile, instead of extracting features from contents of packets (e.g.,

TCP/IP or Modbus), the researchers [94] make use of *telemetry* characteristic. By capturing flows of the transmitting packets between clients and servers, they are able to differentiate between PLC machine of attacker and engineer. This approach makes senses when considering contexts of ICS network since nodes are resource-constrained and connected wirelessly. They claimed the accuracy of telemetry technique is closed to other IDS approach yet harder to evade using encryption techniques [94].

On the other hand, since the SCADA protocol is based on existing Internet standards, some approaches (e.g., [75], [88], and [93]) used *generic traffic data* to evaluate the proposed IDS solutions. In [93], the pre-recorded *pcap* dataset, which contains various network attacks, has been used to verify the game-based multi-agents IDS solution. This dataset has also been used in the Capture The Flag contest at DEFCON event. Masduki et al. [75] focuses on the particular Remote to Local (R2L) attack. The generic KDD 99 Dataset [15] has been used to evaluate the detection accuracy. The probabilistic-based IDS in [88] randomly chose 10,000 normal packets from DARPA dataset [134] to train the proposed model, and they selected another 10,000 random normal and unauthorized connection data for testing.

Physical control states data can be complicated to model and to detect misuse incidents. Some researchers [81] gathered datasets from a Gas Pipeline testbed [80] to develop and test their solution. Various fault injection exploits on the control/sensing signals (e.g., negative pressure value injection, fast change response injection, bust response injection, wave pressure injection, and single packet injection) have been studied. The proposed model was trained by using normal control states, and by properly tuned free parameters for the kernel function (bandwidth of the kernel and the number of eigenvectors), they were able to detect both slow and burst response injection attacks. The IDS solution for water treatment system [57] was evaluated by injecting ten (10) types of attack into their proposed SWaT testbed. The attack disturbed PLC with fault sensor and actuator signals (e.g., data of inflow/outflow rate or level of water in tanks). In their work, various classic classifiers have been compared in detecting the false signals (e.g., SVM, RF, NB, BRTree, BayesNet, and IBK).

Software-behavior-based IDSs have a less significant number of published works compared to the network-based approach. Moon et al. [79] monitors both software- and network-based behavior. They defined software behavior using sets of API calls. For instance, *file copying* action is described using API `CopyFile`, `CopyFileA`, and `MoveFileA`, whereas *file deletion* is defined by calling `DeleteFileA` and `RemoveDirectoryA` functions. On the other hand, network behaviors (e.g., excessive network access, changes of packet delivery on ARP/MAC/IP protocols) also been fused with software activities to formulate decision tree for normal and malicious behaviors. In [123], the behavior of control functions has been model to detect misuse incident. Although the researcher extracted control commands from SCADA control packets, the detection model only focuses on behavior of software usage (commands issued) instead of network protocol properties. In this case, the models of normal behavior are learned by using numbers of commands issued per minute. In the experiment, they injected 280 malicious and 720 normal commands to the simulated Modbus/TCP system. The OCSVM and RE-KPCA (Reconstruction Error based on Kernel Principal Component Analysis) classifiers are trained and used to classify anomaly in a series of control commands.

The *cyber-physical* IDS [61] correlates digital data and physical signal to disclose anomaly. Take Distributed Energy Resources (DER) IDS as an example, the proposed IDS detects anomaly based on context. Variables from photovoltaic (PV) and meteorological data (e.g., solar, wind and temperature properties) are aggregated with contextual variable (i.e., timestamp of each measurement). The ANN classifier is used to detect cyber or physical attacks. The research [61] is extended by incorporating with variables of distance between adjacent nodes in [62].

Table 3. Description of Dataset

Dataset	Data Type	Description	Reference
DARPA	Raw TCP/IP dump files	The dataset published by Defense Advanced Research Projects Agency (DARPA) initiative for evaluation of network-based intrusion detection system.	[134]
KDDCup99	The feature extracted from DARPA raw network data dump.	The dataset based on network communication data has been widely used to test the network intrusion detection system. It originally used for the Data Mining Tools competition.	[15]
NSL-KDD	The improve version of KDDCup99	The improved version of KDDCup99 which removes a number of duplicated records.	[16]
A control system testbed to validate critical infrastructure protection concepts	TCP/IP data communication in various control system	Data is measured from a laboratory-scale gas pipeline, a laboratory-scale water tower, and a laboratory-scale electric transmission system. It is a pre-processed network transaction from 100,000 to 5,000,000 records.	[80]
AIS dataset	Marine vessel movement characteristics	The kinematic and non-kinematic vessel characteristics dataset is a location-based marine movements information collected by U.S. Coast Guard Services. The example of features is latitude, longitude, ground speed, course over ground, rate of turn, and vessel type.	[118]
Behavior-Based Attack Detection and Classification in Cyber Physical Systems Using Machine Learning	Physical state of sensors and actuators.	The states are collected from the physical state of sensors and actuators every second for 28,800 records in total. The state can be used to analyze the effects of cyber-attacks on the physical states.	[57]
Water supply system dataset	Water supply control system status	It contains log files of multiple features such as inflow, outflow, water level, temperature, and running status of water stations. The data was collected from 2011 to 2014 at city of Surrey in BC, Canada.	[141]
BOCISS dataset	Physical state of control system	Physical state of critical control system is simulated from the Siemens Tecnomatix Plant Simulator.	[52]

Furthermore, the holistically monitoring solution in [132] combined knowledge from three (3) sources: (1) physical states, e,g., critical switching signal correlation and key analog signal comparison, (2) protocol specifications, e.g., parameters from Generic Object Oriented Substation (GOOSE) and Simple Measure value (SMV) protocols, and (3) behaviors, such as, Substation Configuration Description (SCD) files and IEC 61850 packet contents. The research [132] aims to detect exploits from malware, namely Havex and Stuxnet.

Table 3 summarizes available datasets referred to by literature in this survey. Various data types are included, namely network traffic, location, physical states, sensors/actuator logs, and states of control devices.

5.7 The Feasibility of the Proposed Work

Table 2 column *Application* compares coverage of IDS design from perspective of security suited, i.e., Security Information and Event Management (SIEM) tools [2, 103]. Despite system's avail-

ability being crucial, only a few researchers included information about their decision support strategies to help system administrator preventing the system from being attacked [96]. Most of the work offers only feature detection in their design.

On the other hand, from aspect of practicality, researchers used various techniques to verify feasibility to deploy the system in the reality range from simulation to physical testbed. Indeed, the simulated software could lack fidelity, especially the physical signal, which is hard to simulate the actual signal from different hardware devices [50]. However, the physical testbed does not scale enough to verify the system, as the physical testbed scale down the real control system. According to Table 2 column *feasibility*, only four (4) out of fifty (50) approaches have deployed in the field [9, 74, 106, 133]. Twelve (12) papers are tested on SCADA-specific testbed or implemented as a testing prototype, whereas the rest are evaluated using various machine learning framework (e.g., MATLAB [100], WEKA [101], ACCORD [111]) or a generic network simulator (e.g., hardware in the loop testbed [90]).

Table 4 lists available testbeds that were used by the literature surveyed in this review including both physical and simulated testbed. As referred to by the papers in this survey, the evaluation testbed includes both SCADA-specific and general-purpose systems.

6 KEY OPEN PROBLEMS

This section presents key research gaps and future research directions of SCADA-based IDSs that use supervised machine learning approaches.

Testbeds **and** *test datasets* **need further research and development**: since it is not practical to train and evaluate a supervised IDS system on a real SCADA system, testbeds and test datasets are crucial for developing a security solution. However, SCADA is widely applied on various control systems, and each of them has the different constraints (e.g., power/water/gas distribution system, manufacturing processes, or railway control system). This makes the construction of high fidelity testbeds costly. Furthermore, some of them are unable to reuse in different contexts of applications. We found that some testbed solutions/datasets in Tables 3 and 4 are either for the general IT system (e.g., [15], [16], [110], and [134]) or are too specific (such as [57] and [118]). Thus, the fidelity can be low compared to the real-world system or not reliable to be used with different settings and scale. With respect to the development cost, the direction of research should focus on developing a high-fidelity simulation testbed and not the more expensive hardware-based solution.

Resilience and Validation of the Security Design Have not Yet Been Sufficiently Explored. Since SCADA is designed for critical control systems, system availability is crucial. Although several articles focused on real-time performance in detecting threats and scalability of the system, *resilience* —which allows the recovery of the security system after the faults (e.g., attacks or natural disaster) to its original or useful state—is often ignored. Besides, most of the research detects anomalies solely based on network traffic from a single source only. Thus, a single point of failure could easily stop the whole security system. For distributed solutions—SCADA DIDS, e.g., [9], [31], [32], [39], [62], [63], [72], [93], [113], [131], and [141], the validation of security and system resilience is a complex task. The formal method, which is used to design the critical system, could also be used to verify the security system (e.g., using the recovery model [18]) and further develop an optimization solution from security and resilience perspectives.

Prevention and Investigation Are not Yet Well Studied. Although the proposed IDS solutions serve as parts of an overall security system, an active SCADA system should rely on a more holistic solution. Some of the research work [56, 63, 69, 91–93, 95, 132, 133, 135, 138, 141] covers both detection and prevention by including automatic critical incident response. However, there is only one work [92] that includes forensic solution in the selected literature, which helps to record and

Table 4. Description of Testbed

Name	SCADA-Specifitc?	Approach	Description	Reference
Fire Alarm System testbed	Y	Physical	Building Automation and Control (BAC) networks or BACnet testbed simulate the operation of fire alarm system using BACnet protocol monitoring module	[91]
SCADA testbed	Y	Physical	It simulates a small SCADA system which composed of Human-Machine Interface (HMI) Station, managed switch and two PLCs. This can be used to simulate the network attack such as TCP port scanning, ARP cache spoofing and denial of service attack.	[72]
CSIT SCADA IDS	Y	Simulated	The testbed consists of SCADA nodes (e.g., HMI, historian, IED), protocols (i.e., IEC 60870-5-103) and malicious host to simulate attacks incident, such as MITM attacks. However, details of the software used in the simulate is undisclosed for security purposes.	[131]
Gas pipeline testbed	Y	Physical	This testbed simulates typical SCADA control units, namely Master Terminal Unit (MTU), Remote Terminal Units (RTU) and Human Machine Interface (HMI).	[81]
Secure Water Treatment (SWaT)	Y	Simulated	This testbed scales down water treatment system. It is designed to develop a security solution for Cyber Physical System (CPS), such as water treatment, electric power generation and distribution. It is composed of networking layers, PLCs, HMIs, SCADA workstation and Historian unit.	[57]
SCADAsim	Y	Simulated	Based on OMNET++, it emulates the network communication of simulated and real devices to analyze the impact of attacks on the devices on SCADA network.	[97]
Cyber-physical test-bed	Y	Simulated	This is based on Cyber-physical build for IEC 61850 based smart substations. The testbed consists of six (6) layers from simulation to substation layer. It supports a number of network-based attacks, such as malformed packet, MITM, address resolution protocol (ARP) spoofing.	[129]
Accord Framework	N	Simulated	Accord provides a well-tested and documented library for constructing various types of algorithms	[110]
CONPOT ICS/SCADA Honeypot	Y	Simulated	It can be used to simulate the network of programmable logic controllers (PLC) units to analyze network telemetry of honeypots and the packages generated by intruders.	[98]

reconstruct the attack event to identify system vulnerability. It is still an open question, how to incorporate these three areas of security measures to deliver the resilience and robustness to the SCADA system.

Distributed IDS Collaboration for the SCADA System Is Still in an Early Age of Development. The DIDS collaborates multiple IDSs to enable scalability to the large network as well as mitigating with the massive parallel attacks. The research [25, 42] shows that aggregation and correlation

between various data sources have potential to detect distributions of malware or exploit. However, the more challenging problem of collaborative IDS is how can the distributed network of IDSs share their knowledge and efficiently improve the detection efficiency. The result from the new research [127] illustrates a potential of improving DIDS efficiency by introducing a distributed learning model. That is, multiple learners (or IDSs) share information about malicious events to improve their own detection models. However, this is still in an early stage of the work of the distributed learners and needs further study.

7 CONCLUSIONS

This survey article looked at emerging research into the application of supervised-learning-based approaches to implementing SCADA-based IDS systems. We have reviewed the development of such systems from research and industry perspectives and provided a comprehensive study of supervised-learning approaches for SCADA-based IDS systems using specific criteria and properties. We have discussed additional issues and challenges for SCADA-based IDS systems using supervised-learning techniques and illustrated the trends to develop such systems.

To identify the future directions in developing new algorithms and to guide the selection of algorithms for SCADA-based IDS systems, we propose a categorizing framework to classify a number of supervised-learning algorithms. This framework is designed from a theoretical viewpoint both to evaluate IDSs of SCADA systems on supervised learning as well to theoretically analyze the most representative supervised machine learning algorithm for SCADA-based IDS systems. Thus, even future SCADA-based IDS systems could be incorporated into the framework according to the proposed criteria and properties. In future work, we will focus on the unsupervised learning algorithms and feature selection for SCADA-based IDS systems.

REFERENCES

[1] Marshall D. Abrams and Joe Weiss. 2008. Malicious control system cyber security attack case study–Maroochy Water Services, Australia. https://www.acsac.org/2008/program/case-studies/Abrams.pdf

[2] Kavita Agrawal and Hemant Makwana. 2015. A study on critical capabilities for security information and event management. *International Journal of Science and Research* 4, 7 (2015), 1893–1896.

[3] Mohiuddin Ahmed, Adnan Anwar, Abdun Naser Mahmood, Zubair Shah, and Michael J. Maher. 2015. An investigation of performance analysis of anomaly detection techniques for big data in SCADA systems. *EAI Endorsed Transactions on Industrial Networks And Intelligent Systems* 2 (2015), 1–16. Issue 3, e5. DOI: https://doi.org/10.4108/inis.2.3.e5

[4] Selim Aksoy and Robert M. Haralick. 2001. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters* 22, 5 (2001), 563–582. DOI: https://doi.org/10.1016/S0167-8655(00)00112-4

[5] Cristina Alcaraz and Javier Lopez. 2014. Diagnosis mechanism for accurate monitoring in critical infrastructure protection. *Computer Standards & Interfaces* 36, 3 (2014), 501–512. DOI: https://doi.org/10.1016/j.csi.2013.10.002

[6] Cristina Alcaraz and Javier Lopez. 2014. WASAM: A dynamic wide-area situational awareness model for critical domains in smart grids. *Future Generation Computer Systems* 30 (2014), 146–154.

[7] Abdul Mohsen Afaf Almalawi. 2014. *Designing Unsupervised Intrusion Detection for SCADA Systems*. Ph.D. Dissertation. RMIT University, School of Computer Science.

[8] Abdul Mohsen Afaf Almalawi, Xinghuo Yu, Zahir Tari, Adil Alharthi Fahad, and Ibrahim Khalil. 2014. An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. *Elsevier Journal on Computers & Security* 46 (2014), 94–110. DOI: https://doi.org/10.1016/j.cose.2014.07.005

[9] Tomasz Andrysiak, Łukasz Saganowski, and Wojciech Mazurczyk. 2016. Network anomaly detection for railway critical infrastructure based on autoregressive fractional integrated moving average. *Springer Journal on Wireless Communications and Networking*, 1 (2016), 245.

[10] Adnan Anwar and Abdun Naser Mahmood. 2014. Vulnerabilities of smart grid state estimation against false data injection attack. *Springer Journal on Renewable Energy Integration*, 411–428.

[11] Adnan Anwar, Abdun Naser Mahmood, and Mohiuddin Ahmed. 2014. False data injection attack targeting the LTC transformers to disrupt smart grid operation. In *International Conference on Security and Privacy in Communication Systems*. Springer International Publishing, Cham, 252–266.

[12] Adnan Anwar, Abdun Naser Mahmood, and Mark Pickering. 2017. Modeling and performance evaluation of stealthy false data injection attacks on smart grid in the presence of corrupted measurements. *Elsevier Journal of Computer and System Sciences* 83, 1 (2017), 58–72.

[13] Adnan Anwar, Abdun Naser Mahmood, and Zahir Tari. 2015. Identification of vulnerable node clusters against false data injection attack in an AMI based smart grid. *Elsevier Journal on Information Systems* 53 (2015), 201–212.

[14] Adnan Anwar, Abdun N. Mahmood, and Zahir Tari. 2017. Ensuring data integrity of OPF module and energy database by detecting changes in power flow patterns in smart grids. *IEEE Transactions on Industrial Informatics* 13, 6 (2017), 3299–3311.

[15] Aditya Ashok, Manimaran Govindarasu, and Jianhui Wang. 2017. Cyber-physical attack-resilient wide-area monitoring, protection, and control for the power grid. *Proceedings of the IEEE* 105, 7 (2017), 1389–1407.

[16] Aditya Ashok, Siddharth Sridhar, A. David McKinnon, Wang Pengyuan, and Manimaran Govindarasu. 2016. Testbed-based performance evaluation of Attack Resilient Control for AGC. In *2016 Resilience Week (RWS)*. IEEE, Chicago, IL, 125–129. DOI: https://doi.org/10.1109/RWEEK.2016.7573319

[17] Muhammad Kamran Asif and Yahya Subhi Al-Harthi. 2014. Intrusion detection system using honey token based encrypted pointers to mitigate cyber threats for critical infrastructure networks. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, San Diego, CA,, 1266–1270. DOI: https://doi.org/10.1109/SMC.2014.6974088

[18] Guillaume Babin, Yamine Aït-Ameur, Neeraj Kumar Singh, and Marc Pantel. 2016. A system substitution mechanism for hybrid systems in Event-B. In *International Conference on Formal Engineering Methods*. Springer International Publishing, Cham, 106–121.

[19] Patrick Biernacki and Dan Waldorf. 1981. Snowball sampling: Problems and techniques of chain referral sampling. *Sociological Methods & Research* 10, 2 (1981), 141–163.

[20] Christopher M. Bishop. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press, New York,.

[21] Thomas d'Otreppe de Bouvette. 2009. Aircrack-ng - Main documentation. Retrieved Apirl 1, 2019 from https://www.aircrack-ng.org/documentation.html

[22] Stuart A. Boyer. 2009. *SCADA: Supervisory Control and Data Acquisition (4th ed.)*. International Society of Automation.

[23] Nemanja Branisavljević, Zoran Kapelan, and Dušan Prodanović. 2011. Improved real-time data anomaly detection using context classification. *IWA Journal of Hydroinformatics* 13, 3 (2011), 307–323.

[24] Andrea Carcano, Alessio Coletta, Michele Guglielmi, Marcelo Masera, Igor Nai Fovino, and Alberto Trombetta. 2011. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics* 7, 2 (May 2011), 179–186.

[25] Zhongqiang Chen, Mema Roussopoulos, Zhanyan Liang, Yuan Zhang, Zhongrong Chen, and Alex Delis. 2012. Malware characteristics and threats on the internet ecosystem. *Journal of Systems and Software* 85, 7 (2012), 1650–1672.

[26] Zhongqiang Chen, Yuan Zhang, Zhongrong Chen, and Alex Delis. 2009. A digest and pattern matching-based intrusion detection engine. *Comput. J.* 52, 6 (2009), 699–723.

[27] Peter Clark and Tim Niblett. 1989. The CN2 induction algorithm. *Machine Learning* 3, 4 (1989), 261–283.

[28] William W. Cohen and Yoram Singer. 1999. A simple, fast, and effective rule learner. In *Proceedings of the 16th National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*. American Association for Artificial Intelligence, Menlo Park, CA,, 335–342.

[29] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3 (1995), 273–297.

[30] Maurilio Pereira Coutinho, Germano Lambert-Torres, Luiz Eduardo Borges da Silva, Jonas Guedes Borges da Silva, Jose Cabral Neto, and Horst Lazarek. 2008. Improving a methodology to extract rules to identify attacks in power system critical infrastructure: New results. In *IEEE Conference on Transmission, Distribution and Exposition*. IEEE, Chicago, IL, 1–6.

[31] Tiago Cruz, Luis Rosa, Jorge Proença, Leandros Maglaras, Matthieu Aubigny, Leonid Lev, Jianmin Jiang, and Paulo Simões. 2016. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Transactions on Industrial Informatics (TII)* 12, 6 (2016), 2236–2246.

[32] Eduardo Germano da Silva, Anderson Santos da Silva, Juliano Araujo Wickboldt, Paul Smith, Lisandro Zambenedetti Granville, and Alberto Schaeffer-Filho. 2016. A one-class NIDS for SDN-based SCADA systems. In *40th IEEE Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 1. IEEE, Atlanta, GA, 303–312.

[33] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

[34] Dorothy E. Denning. 1987. An intrusion-detection model. *IEEE Transactions on Software Engineering* SE-13, 2 (Feb. 1987), 222–232.

[35] Digitalbond.com. 2013. IDS-signatures/modbus-tcp. Retrieved December, 2018 from http://www.digitalbond.com/index.php/research/ids-signatures/modbus-tcp-ids-signatures/.
[36] Bradley Efron and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC Press, New York.
[37] Noam Erez and Avishai Wool. 2015. Control variable classification, modeling and anomaly detection in Modbus/TCP SCADA systems. *Elsevier International Journal of Critical Infrastructure Protection* 10 (2015), 59–70.
[38] Terry Escamilla. 1998. *Intrusion Detection: Network Security Beyond the Firewall*. Vol. 8. John Wiley, New York.
[39] Mustafa Amir Faisal, Zeyar Aung, John R. Williams, and Abel Sanchez. 2012. Securing advanced metering infrastructure using intrusion detection system with data stream mining. *Springer Journal on Intelligence and Security Informatics* 7299 (2012), 96–111.
[40] Igor Nai Fovino, Andrea Carcano, Thibault De Lacheze Murel, Alberto Trombetta, and Marcelo Masera. 2010. Modbus/DNP3 state-based intrusion detection system. In *24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, Perth, Australia, 729–736.
[41] Igor Nai Fovino, Alessio Coletta, Andrea Carcano, and Marcelo Masera. 2012. Critical state-based filtering system for securing SCADA network protocols. *IEEE Transactions on Industrial Electronics* 59, 10 (October 2012), 3943–3950.
[42] Ivo Friedberg, Florian Skopik, Giuseppe Settanni, and Roman Fiedler. 2015. Combating advanced persistent threats: From network event correlation to incident detection. *Computers & Security* 48 (2015), 35–57.
[43] Nir Friedman, Dan Geiger, and Moises Goldszmidt. 1997. Bayesian network classifiers. *Machine Learning* 29, 2–3 (1997), 131–163.
[44] Jingcheng Gao, Jing Liu, Bharat Rajan, Rahul Nori, Bo Fu, Yang Xiao, Wei Liang, and C. L. Philip Chen. 2014. SCADA communication and security issues. *Security and Communication Networks* 7, 1 (2014), 175–194.
[45] Iñaki Garitano, Roberto Uribeetxeberria, and Urko Zurutuza. 2011. A review of SCADA anomaly detection sys- tems. In *6th Springer International Conference on Soft Computing Models in Industrial and Environmental Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 357–366.
[46] Bela Genge, Piroska Haller, and Istvan Kiss. 2016. A framework for designing resilient distributed intrusion detection systems for critical infrastructures. *International Journal of Critical Infrastructure Protection* 15 (2016), 3–11.
[47] Radhika Goel, Anjali Sardana, and Ramesh C. Joshi. 2012. Parallel misuse and anomaly detection model. *International Journal of Network Security* 14, 4 (2012), 211–222.
[48] Philip Gross, Janak Parekh, and Gail Kaiser. 2004. Secure selecticast for collaborative intrusion detection systems. In *3rd International Workshop on Distributed Event-Based Systems (DEBS)*. Institution of Engineering and Technology, Edinburgh, UK, 50–55.
[49] Jiawei Han, Micheline Kamber, and Jian Pei. 2012. *Data Mining: Concepts and Techniques* (3rd ed.). Elsevier, MA.
[50] Hannes Holm, Martin Karresand, Arne Vidström, and Erik Westring. 2015. A survey of industrial control system testbeds. In *20th Nordic Conference on Secure IT Systems (NordSec 2015)*. Springer International Publishing, Stockholm, Sweden, 11–26.
[51] Jasenko Hosic, Jereme Lamps, and Derek H. Hart. 2015. Evolving decision trees to detect anomalies in recurrent ICS networks. In *IEEE World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, London, UK, 50–57.
[52] William Hurst, Madjid Merabti, and Paul Fergus. 2014. Big data analysis techniques for cyber-threat detection in critical infrastructures. In *28th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, Victoria, BC, Canada, 916–921.
[53] Modbus IDA. 2004. Modbus messaging on tcp/ip implementation guide v1. 0a. http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0a.pdf.
[54] Vinay M. Igure, Sean A. Laughter, and Ronald D. Williams. 2006. Security issues in SCADA networks. *Computers & Security* 25, 7 (2006), 498–506.
[55] V. Jaiganesh, S. Mangayarkarasi, and P. Sumathi. 2013. Intrusion detection systems: A survey and analysis of classification techniques. *International Journal of Advanced Research in Computer and Communication Engineering* 2, 4 (2013), 1629–1635.
[56] Mahdi Jamei, Emma Stewart, Sean Peisert, Anna Scaglione, Chuck McParland, Ciaran Roberts, and Alex McEachern. 2016. Micro synchrophasor-based intrusion detection in automated distribution systems: Toward critical infrastructure security. *IEEE Internet Computing* 20, 5 (2016), 18–27.
[57] Khurum Nazir Junejo and Jonathan Goh. 2016. Behaviour-based attack detection and classification in cyber physical systems using machine learning. In *2nd ACM International Workshop on Cyber-Physical System Security (CPSS'16)*. ACM, New York,, 34–43. DOI: https://doi.org/10.1145/2899015.2899016
[58] Andrey Olegovich Kalashnikov and Ekaterina Sakrutina. 2018. Towards risk potential of significant plants of critical information infrastructure. In *International Russian Automation Conference (RusAutoCon)*. IEEE, Sochi, Russia, 1–6.
[59] David G. Kleinbaum, Lawrence L. Kupper, Keith E. Muller, and Azhar Nizam. 1988. *Applied Regression Analysis and Other Multivariable Methods*. Vol. 601. Duxbury Press Belmont, CA, Boston, MA.

[60] Roman Klinger and Katrin Tomanek. 2007. *Classical Probabilistic Models and Conditional Random Fields*. Dortmund University of Technology, Dortmund, Germany.

[61] Anna Magdalena Kosek. 2016. Contextual anomaly detection for cyber-physical security in smart grids based on an artificial neural network model. In *IEEE Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG)*. IEEE, Vienna, Austria, 1–6.

[62] Anna Magdalena Kosek and Oliver Gehrke. 2016. Ensemble regression model-based anomaly detection for cyber-physical intrusion detection in smart grids. In *IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, Ottawa, ON, Canada, 1–7.

[63] Dhanalakshmi Krishnan Sadhasivan and Kannapiran Balasubramanian. 2017. A fusion of multiagent functionalities for effective intrusion detection system. *Security and Communication Networks* 2017, Article 216078 (2017), 15 pages. DOI: https://doi.org/10.1155/2017/6216078

[64] Sathish Alampalayam P. Kumar, Anup Kumar, and S. Srinivasan. 2007. Statistical based intrusion detection framework using six sigma technique. *International Journal of Computer Science and Network Security* 7, 10 (2007), 333–342.

[65] Roger J. Lewis. 2000. An introduction to classification and regression tree (CART) analysis. In *Annual Meeting of the Society for Academic Emergency Medicine*. The Pennsylvania State University, San Francisco, CA, 1–14.

[66] Wei Li. 2004. Using genetic algorithm for network intrusion detection. In *Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference*. Louisiana State University, Kansas City, KS, USA, 24–27.

[67] Wenmin Li, Jiawei Han, and Jian Pei. 2001. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE, San Jose, CA, USA, 369–376. DOI: https://doi.org/10.1109/ICDM.2001.989541

[68] Anna D. P. Lotufo, Fernando P. A. Lima, and Carlos R. Minussi. 2014. Disturbance detection for optimal database storage in electrical distribution systems using artificial immune systems with negative selection. *Elsevier Journal on Electric Power Systems Research* 109 (2014), 54–62. DOI: https://doi.org/10.1016/j.epsr.2013.12.010

[69] Ondrej Linda, Todd Vollmer, and Milos Manic. 2009. Neural network based intrusion detection system for critical infrastructures. In *International Joint Conference on Neural Networks (IJCNN)*. IEEE, Atlanta, GA, 1827–1834.

[70] Bing Liu, Wynne Hsu, and Yiming Ma. 1998. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 80–86.

[71] Leandros A. Maglaras and Jianmin Jiang. 2014. Intrusion detection in SCADA systems using machine learning techniques. In *IEEE Science and Information Conference (SAI), 2014*. IEEE, London, UK, 626–631.

[72] Leandros A. Maglaras, Jianmin Jiang, and Tiago J. Cruz. 2016. Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems. *Elsevier Journal of Information Security and Applications* 30 (2016), 15–26.

[73] Abdun Naser Mahmood, Christopher Leckie, Jiankun Hu, Zahir Tari, and Mohammed Atiquzzaman. 2010. *Network Traffic Analysis and SCADA Security*. Springer Berlin Heidelberg, Berlin, Heidelberg. 383–405 pages.

[74] Moti Markovitz and Avishai Wool. 2017. Field classification, modeling and anomaly detection in unknown CAN bus networks. *Journal on Vehicular Communications* 9 (2017), 43–52. DOI: https://doi.org/10.1016/j.vehcom.2017.02.005

[75] Bisyron Wahyudi Masduki, Kalamullah Ramli, Ferry Astika Saputra, and Dedy Sugiarto. 2015. Study on implementation of machine learning methods combination for improving attacks detection accuracy on Intrusion Detection System (IDS). In *IEEE International Conference on Quality in Research (QiR)*. IEEE, Lombok, Indonesia, 56–64.

[76] Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. 1996. SLIQ: A fast scalable classifier for data mining. In *Springer International Conference on Extending Database Technology (EDBT)*. Springer Berlin, 18–32.

[77] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. 1999. Fisher discriminant analysis with kernels. In *1999 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing IX*. IEEE, Madison, WI, 41–48.

[78] Robert Mitchell and Ing-Ray Chen. 2014. A survey of intrusion detection techniques for cyber-physical systems. *Computing Surveys* 46, 4 (2014), 55.

[79] Daesung Moon, Hyungjin Im, Ikkyun Kim, and Jong Hyuk Park. 2015. DTB-IDS: An intrusion detection system based on decision tree using behavior analysis for preventing APT attacks. *Springer Journal of Supercomputing* 73, 7 (2015), 2881–2895.

[80] Thomas Morris, Anurag Srivastava, Bradley Reaves, Wei Gao, Kalyan Pavurapu, and Ram Reddi. 2011. A control system testbed to validate critical infrastructure protection concepts. *Elsevier International Journal of Critical Infrastructure Protection* 4, 2 (2011), 88–103.

[81] Patric Nader, Paul Honeine, and Pierre Beauseroy. 2013. Intrusion detection in SCADA systems using one-class classification. In *21st European IEEE Signal Processing Conference (EUSIPCO)*. IEEE, Marrakech, Morocco, 1–5.

[82] Patric Nader, Paul Honeine, and Pierre Beauseroy. 2014. l_p-norms in one-class classification for intrusion detection in SCADA systems. *IEEE Transactions on Industrial Informatics (TII)* 10, 4 (2014), 2308–2317.

[83] Patric Nader, Paul Honeine, and Pierre Beauseroy. 2016. Detection of cyberattacks in a water distribution system using machine learning techniques. In *6th IEEE International Conference on Digital Information Processing and Communications (ICDIPC)*. IEEE, Beirut, Lebanon, 25–30. DOI: https://doi.org/10.1109/ICDIPC.2016.7470786

[84] Sajid Nazir, Shushma Patel, and Dilip Patel. 2017. Assessing and augmenting SCADA cyber security: A survey of techniques. *Computers & Security* 70 (2017), 436–454. DOI: https://doi.org/10.1016/j.cose.2017.06.010

[85] Andrew Nicholson, Helge Janicke, and Tim Watson. 2013. An initial investigation into attribution in SCADA systems. In *1st International Symposium on ICS & SCADA Cyber Security Research* (September 16–17). ACM, Leicester, UK, 56–65.

[86] Peng Ning, Yun Cui, and Douglas S. Reeves. 2002. Constructing attack scenarios through correlation of intru- sion alerts. In *9th ACM Conference on Computer and Communications Security (CCS'02)*. ACM, New York, 245–254. DOI: https://doi.org/10.1145/586110.586144

[87] Paul Oman, Edmund Schweitzer, and Deborah Frincke. 2000. Concerns about intrusions into remotely accessible substation controllers and SCADA systems. In *27th Annual Western Protective Relay Conference*, Vol. 160. Citeseer, Spokane, WA, 1–16.

[88] Takashi Onoda. 2016. Probabilistic models-based intrusion detection using sequence characteristics in control system communication. *Springer Journal on Neural Computing and Applications* 27, 5 (2016), 1119–1127. DOI: https://doi.org/10.1007/s00521-015-1984-y

[89] Atilla Özgür and Hamit Erdem. 2012. Intrusion detection classifiers comparison in different operating environments. In *9th International Conference on Electronics Computer and Computation (ICECCO)*, V. Kiray, R. Ozcan, and T. Malas (Eds.). Turgut Ozal Univ, Turkey, 24–27.

[90] Shengyi Pan, Thomas Morris, and Uttam Adhikari. 2015. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid (TSG)* 6, 6 (2015), 3104–3113. DOI: https://doi.org/10.1109/TSG.2015.2409775

[91] Zhiwen Pan, Salim Hariri, and Youssif Al-Nashif. 2014. Anomaly based intrusion detection for building automation and control networks. In *11th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE, Doha, Qatar, 72–77.

[92] Ahmed Patel, Hitham Alhussian, Jens Myrup Pedersen, Bouchaib Bounabat, Joaquim Celestino Jr, and Sokratis Katsikas. 2017. A nifty collaborative intrusion detection and prevention architecture for smart grid ecosystems. *Computers & Security* 64 (2017), 92–109. DOI: https://doi.org/10.1016/j.cose.2016.07.002

[93] Alecsandru Patrascu and Victor-Valeriu Patriciu. 2015. Cyber protection of critical infrastructures using supervised learning. In *20th IEEE International Conference on Control Systems and Computer Science (CSCS)*. IEEE, Bucharest, Romania, 461–468. DOI: https://doi.org/10.1109/CSCS.2015.34

[94] Stanislav Ponomarev and Travis Atkison. 2016. Industrial control system network intrusion detection by telemetry analysis. *IEEE Transactions on Dependable and Secure Computing (TDSC)* 13, 2 (2016), 252–260. DOI: https://doi.org/10.1109/TDSC.2015.2443793

[95] Upeka Kanchana Premaratne, Jagath Samarabandu, Tarlochan S. Sidhu, Robert Beresh, and Jian-Cheng Tan. 2010. An intrusion detection system for IEC61850 automated substations. *IEEE Transactions on Power Delivery* 25, 4 (2010), 2376–2383.

[96] Carlos Queiroz, Abdun Mahmood, and Zahir Tari. 2013. A probabilistic model to predict the survivability of SCADA systems. *IEEE Transactions on Industrial Informatics* 9, 4 (2013), 1975–1985. DOI: https://doi.org/10.1109/TII.2012.2231419

[97] Carlos Queiroz, Abdun Naser Mahmood, and Zahir Tari. 2011. SCADASim-A framework for building SCADA simulations. *IEEE Transactions on Smart Grid (TSG)* 2, 4 (2011), 589–597. DOI: https://doi.org/10.1109/TSG.2011.2162432

[98] Lukas Rift, Johnny Vastergaard, Daniel Haslinger, Andrea Pasquale, and John Smith. 2013. Conpot ICS/SCADA honeypot. Retrieved April 2018 from http://conpot.org.

[99] Julian L. Rrushi. 2009. *Composite Intrusion Detection in Process Control Networks*. Ph.D. Dissertation. Università degli Studi di Milano, Milano, Italy.

[100] S. Shitharth and D. Prince Winston. 2017. An enhanced optimization based algorithm for intrusion detection in SCADA network. *Elsevier Journal on Computers & Security* 70 (2017), 16–26. DOI: https://doi.org/10.1016/j.cose.2017.04.012

[101] Rishabh Samdarshi, Nidul Sinha, and Paritosh Tripathi. 2015. A triple layer intrusion detection system for SCADA security of electric utility. In *IEEE Annual India Conference (INDICON)*. IEEE, New Delhi, India, 1–5.

[102] M. F. Schilling. 1986. Mutual and shared neighbor probabilities: Finite-and infinite-dimensional results. *Advances in Applied Probability* 18, 2 (1986), 388–405.

[103] S. Sandeep Sekharan and Kamalanathan Kandasamy. 2017. Profiling SIEM tools and correlation engines for security analytics. In *IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, Chennai, India, 717–721.

[104] John Shafer, Rakeeh Agrawal, and Manish Mehta. 1996. SPRINT: A scalable parallel classifier for data mining. In *22nd International Conference on Very Large Data Bases (VLDB)*. Citeseer, Mumbai, India, 544–555.

[105] Zubair Shah, Abdun Naser Mahmood, Mehmet A. Orgun, and M. Hadi Mashinchi. 2013. Subset selection classifier (SSC): A training set reduction method. In *16th IEEE International Conference on Computational Science and Engineering (CSE)*. IEEE, Sydney, NSW, Australia, 862–869.

[106] Hamed Yaghoubi Shahir, Uwe Glasser, Amir Yaghoubi Shahir, and Hans Wehn. 2015. Maritime situation analysis framework: Vessel interaction classification and anomaly detection. In *IEEE International Conference on Big Data (Big Data)*. IEEE, Santa Clara, CA, 1279–1289.

[107] Joseph Sill, Gábor Takács, Lester Mackey, and David Lin. 2009. Feature-weighted Linear Stacking. (2009). arXiv:arXiv:0911.0460.

[108] Pedro Silva. 2014. *On the Use of K-NN in Intrusion Detection for Industrial Control Systems*. Master'sthesis. Department of Information Technology, Galway, Ireland.

[109] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. 2015. An overview of Microsoft Academic Service (MAS) and applications. In *24th International Conference on World Wide Web*. ACM, Florence, Italy, 243–246.

[110] Cézar Roberto Souza. 2014. The Accord .NET Framework. Retrieved January, 2017 from http://accord-framework.net.

[111] Kyriakos Stefanidis and Artemios G. Voyiatzis. 2016. An HMM-based anomaly detection approach for SCADA systems. In *IFIP International Conference on Information Security Theory and Practice*. Springer International Publishing, Heraklion, Crete, Greece, 85–99.

[112] Charles Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning* 4, 4 (2012), 267–373.

[113] R. Bala Sri Swetha and K. Goklia Meena. 2015. Smart grid – A network based intrusion detection system. In *International Conference on Innovations in Computing Techniques (ICICT 2015)*. Semantic Scholar, Coimbatore, India, 29–36.

[114] Bo Tang and Haibo He. 2017. A local density-based approach for outlier detection. *Neurocomputing* 241 (2017), 171–180. https://doi.org/10.1016/j.neucom.2017.02.039

[115] David M. J. Tax and Robert P. W. Duin. 2004. Support vector data description. *Machine Learning* 54, 1 (1 Jan 2004), 45–66. DOI: https://doi.org/10.1023/B:MACH.0000008084.60811.49

[116] The Computing Research and Education Association of Australasia (CORE). 2018. CORE Conference Portal. Retrieved August 12, 2018 from http://portal.core.edu.au/conf-ranks/.

[117] Wojciech Tylman. 2013. SCADA intrusion detection based on modelling of allowed communication patterns. In *New Results in Dependability and Computer Systems*. Springer, Heidelberg, Brunów, Poland, 489–500. https://doi.org/10.1007/978-3-319-00945-2_45

[118] U.S. Coast Guard Navigation Center. 2015. Automatic identification system overview. Retrieved August 12, 2018 from http://www.navcen.uscg.gov/?pageName=AISmain.

[119] Alfonso Valdes and Steven Cheung. 2009. Communication pattern anomaly detection in process control systems. In *IEEE Conference on Technologies for Homeland Security (HST)*. IEEE, Boston, MA, USA, 22–29.

[120] Jan Vávra and Martin Hromada. 2017. Evaluation of anomaly detection based on classification in relation to SCADA. In *IEEE International Conference on Military Technologies (ICMT)*. IEEE, Brno, Czech Republic, 330–334.

[121] Jared Verba and Michael Milvich. 2008. Idaho national laboratory supervisory control and data acquisition intrusion detection system (SCADA IDS). In *IEEE Conference on Technologies for Homeland Security*. IEEE, Waltham, MA, 469–473.

[122] R. Vijayanand, D. Devaraj, and B. Kannapiran. 2017. Support vector machine based intrusion detection system with reduced input features for advanced metering infrastructure of smart grid. In *4th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, Coimbatore, India, 1–7. DOI: https://doi.org/10.1109/ICACCS.2017.8014590

[123] Ming Wan, Wenli Shang, and Peng Zeng. 2017. Double behavior characteristics for one-class classification anomaly detection in networked control systems. *IEEE Transactions on Information Forensics and Security (TIFS)* 12, 12 (2017), 3011–3023. DOI: https://doi.org/10.1109/TIFS.2017.2730581

[124] Dong Wei, Yan Lu, Mohsen Jafari, Paul M. Skare, and Kenneth Rohde. 2011. Protecting smart grid automation systems against cyberattacks. *IEEE Transactions on Smart Grid* 2, 4 (2011), 782–795. DOI: https://doi.org/10.1109/TSG.2011.2159999

[125] Joe Weiss. 2016. Aurora generator test. In *Handbook of SCADA/Control Systems Security*. CRC Press, Boca Raton, FL, 107–114. DOI: https://doi.org/10.1201/b13869

[126] Yang Wenxian and Jiang Jiesheng. 2011. Wind turbine condition monitoring and reliability analysis by SCADA information. In *2nd International Conference Mechanic Automation and Control Engineering (MACE)*. IEEE, Hohhot, China, 1872–1875.

[127] Rongjun Xie, Ibrahim Khalil, Shahriar Badsha, and Mohammed Atiquzzaman. 2018. Fast and peer-to-peer vital signal learning system for cloud-based healthcare. *Future Generation Computer Systems (FGCS)* 88 (2018), 220–233.

[128] Dayu Yang, Alexander Usynin, and J. Wesley Hines. 2006. Anomaly-based intrusion detection for SCADA systems. In *5th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies (NPIC&HMIT)*. Semantic Scholar, Albuquerque, NM, USA, 12–16.

[129] Yi Yang, H. T. Jiang, Kieran McLaughlin, L. Gao, Y. B. Yuan, W. Huang, and Sakir Sezer. 2015. Cybersecurity test-bed for IEC 61850 based smart substations. In *2015 IEEE Power & Energy Society General Meeting*. IEEE, Denver, CO, 1–5. DOI: https://doi.org/10.1109/PESGM.2015.7286357

[130] Yi Yang, Kieran McLaughlin, Tim Littler, Sakir Sezer, and Haifeng Wang. 2013. Rule-based intrusion detection system for SCADA networks. In *2nd IET Renewable Power Generation Conference (RPG 2013)*. Institution of Engineering and Technology, Beijing, China, 1–4. DOI: https://doi.org/10.1049/cp.2013.1729

[131] Yi Yang, Kieran McLaughlin, Sakir Sezer, Tim Littler, Eul Gyu Im, Bernardi Pranggono, and Haifeng Wang. 2014. Multiattribute SCADA-specific intrusion detection system for power networks. *IEEE Transactions on Power Delivery* 29, 3 (2014), 1092–1102. DOI: https://doi.org/10.1109/TPWRD.2014.2300099

[132] Yi Yang, Hai-Qing Xu, Lei Gao, Yu-Bo Yuan, Kieran McLaughlin, and Sakir Sezer. 2017. Multidimensional intrusion detection system for IEC 61850-based SCADA networks. *IEEE Transactions on Power Delivery* 32, 2 (2017), 1068–1078. DOI: https://doi.org/10.1109/TPWRD.2016.2603339

[133] S. L. P. Yasakethu, J. Jiang, and A. Graziano. 2013. Intelligent risk detection and analysis tools for critical infrastructure protection. In *IEEE EUROCON Conference*. IEEE, Zagreb, Croatia, 52–59. DOI: https://doi.org/10.1109/EUROCON.2013.6624965

[134] Shen Yin, Xiangping Zhu, and Chen Jing. 2014. Fault detection based on a robust one class support vector machine. *Neurocomputing* 145 (2014), 263–268. DOI: https://doi.org/10.1016/j.neucom.2014.05.035

[135] Hyunguk Yoo and Taeshik Shon. 2015. Novel approach for detecting network anomalies for substation au- tomation based on IEC 61850. *Multimedia Tools and Applications* 74, 1 (2015), 303–318. https://doi.org/10.1007/ s11042-014-1870-0

[136] A. Zaher, S. D. J. McArthur, D. G. Infield, and Y. Patel. 2009. Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy* 12, 6 (2009), 574–593. DOI: https://doi.org/10.1002/we.319

[137] Hossein Zeynal, Mostafa Eidiani, and Dariush Yazdanpanah. 2014. Intelligent substation automation systems for robust operation of smart grids. In *2014 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA)*. IEEE, Kuala Lumpur, Malaysia, 786–790. DOI: https://doi.org/10.1109/ISGT-Asia.2014.6873893

[138] Chunjie Zhou, Shuang Huang, Naixue Xiong, Shuang-Hua Yang, Huiyun Li, Yuanqing Qin, and Xuan Li. 2015. Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 10 (2015), 1345–1360. DOI: https://doi.org/10.1109/TSMC.2015.2415763

[139] Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, New York.

[140] Bonnie Zhu and Shankar S. Sastry. 2010. SCADA-specific intrusion detection/prevention systems: A survey and taxonomy. In *1st Workshop on Secure Control Systems (SCS)*, Vol. 11. Berkeley University of California, Article 8, 7 pages.

[141] Zahra Zohrevand, Uwe Glasser, Hamed Yaghoubi Shahir, Mohammad A. Tayebi, and Robert Costanzo. 2016. Hidden Markov based anomaly detection for water supply systems. In *IEEE International Conference on Big Data*. IEEE, WA, USA, 1551–1560. DOI: https://doi.org/10.1109/BigData.2016.7840763