Egger *et al. Energy Informatics* 2020, **3**(Suppl 1):15
https://doi.org/10.1186/s42162-020-00118-4

Energy Informatics

RESEARCH Open Access

Comparison of approaches for intrusion detection in substations using the IEC 60870-5-104 protocol

Michael Egger[1*], Günther Eibl[2] and Dominik Engel[2]

From The 9th DACH+ Conference on Energy Informatics
Sierre, Switzerland. 29-30 October 2020

*Correspondence:
michael.egger@apg.at
[1]Austrian Power Grid AG, IZD-Tower, Wagramer Str. 19, Vienna, Austria
Full list of author information is available at the end of the article

Abstract

Electrical networks of transmission system operators are mostly built up as isolated networks without access to the Internet. With the increasing popularity of smart grids, securing the communication network has become more important to avoid cyber-attacks that could result in possible power outages. For misuse detection, signature-based approaches are already in use and special rules for a wide range of protocols have been developed. However, one big disadvantage of signature-based intrusion detection is that zero-day exploits cannot be detected.
Machine-learning-based anomaly detection methods have the potential to achieve that. In this paper, various such methods for intrusion detection in substations, which use the asynchronous communication protocol International Electrotechnical Commission (IEC) 60870-5-104, are tested and compared. The evaluation of the proposed methods is performed by applying them to a data set which includes normal operation traffic and four different attacks. While the results of supervised and semi-supervised machine learning approaches are rather encouraging, the unsupervised and signature-based methods suffer from general bad performance and had difficulties to detect some attacks.

Keywords: Intrusion detection, IEC 60870-5-104, SCADA

Introduction

Industrial Control System (ICS) that are used to monitor and control infrastructures, such as electrical power grids, traditionally have mainly consisted of devices specially developed for this specific purpose. Furthermore, they were only used in isolated networks without access to the Internet (Berthier et al. 2010). Since energy supply companies focused on availability and reliability requirements, cyber security measures were often classified as insignificant. Incidents such as *Stuxnet* or the *BlackEnergy* attacks on power grids in the Ukraine, strongly increased awareness towards the security of these critical infrastructures (Ang and Utomo 2017). Particularly with the stronger digitalization

of substations and the networking of control technology components, the dangers have become more diverse and complex. It is therefore important that not only known but also currently unknown attack vectors against critical infrastructures can be recognized (Butt et al. 2019). In order to meet these challenges, additional concepts for ensuring network security and maintaining network stability will be necessary (Yan et al. 2012).

This paper studies the detection of various common attacks using packet-based methods in the context of substations using the asynchronous communication protocol IEC 60870-5-104. Network traffic used in this paper was captured in a test environment of the Austrian Power Grid AG (APG), which is highly similar to the real-world setup. Therefore, the test data can be considered as equivalent to real data from a substation in which the communication is based on the IEC 60870-5-104 protocol, see Fig. 2 in the "Experiments" section. In order to enable reproducible research the full data set, along with the used code, will be made publicly available as stated in "Availability of data and materials" section.

The following research questions should be answered: While the standard signature-based approach is expected to lead to only few false-positives, are the existing rules sufficient to reliably detect various common attacks in the context of substations operating using the asynchronous communication protocol IEC 60870-5-104?

Furthermore, a semi-supervised and an unsupervised approach, which are two anomaly detection settings with the potential to detect new attacks, are compared. Here both, the potential to detect an attack and the price of false-positives, are unknown. A supervised classification approach is also applied for comparison and to answer the next question: Can the principle potential of machine learning methods be shown to detect attacks with low false positive rates, based on the extracted features? Since a new attack cannot be detected in a supervised setting, this should serve as an upper limit against which the two anomaly detection settings can be compared.

Related work

Several methods to detect attacks on Supervisory Control and Data Acquisition (SCADA) systems have already been proposed. These methods can mainly be classified into signature-based and anomaly-based approaches (Phillips et al. 2020). It is these two classes, which are also compared in our paper.

Signature-based approaches consist of a set of deterministic rules that either describe the normal system or patterns which arise for already known attacks. Peterson (Peterson 2009) developed *Project Quickdraw*, which has later been renamed to *Snort* (see "Snort" section), and tried to detect anomalies in legacy SCADA systems by using signatures. Snort got one of the most popular signature based intrusion detection systems nowadays and is also used for misuse detection in IEC 60970-5-104 SCADA traffic in our paper.

Yang et al. proposed a rule set for IEC 60970-5-104 traffic in Yang et al. (2013), which could be used in conjunction with signature-based intrusion detection systems like Snort. This was one of the most important steps in the evolution of signature-based detection methods for securing critical infrastructures which rely on this communication protocol. Some of these rules have been added to the rule set used to detect anomalies with Snort in this paper. The detection of relay and Man-In-The-Middle (MITM) attacks, which was carried out by Maynard et al. (2014), but is not part of our proposed attack data set, could be considered in further research on IEC 60870-5-104 SCADA networks with Snort.

One big problem of signature-based anomaly detection is that encrypted payloads or unfamiliar protocols cannot be read. In our experiments, this problem prohibited the detection of one of the proposed attacks (see "Misuse detection with snort" section). Hoeve proposed methods to detect intrusions in encrypted control traffic in Hoeve (2013). Patterns in packet series could be found by only looking at the time, size, and direction of packets. Tests on IEC 60870-5-104 SCADA traffic showed that this implementation could work out but the quality of the outcomes depends on the choice of parameters and has to be improved. Therefore, the features which we have selected to detect anomalies in IEC 60870-5-104 traffic could contribute to the improvement of intrusion detection in encrypted control traffic.

In contrast to signature-based approaches, anomaly detection works without knowledge about the attack. Therefore, it has the potential to detect unknown attacks like zero-day attacks. Most of the existing methods are semi-supervised: in this setting, normal data are modeled and big deviations from normal are considered as potential attack. Besides packet-based approaches, time-series methods are also often proposed to check for intrusions. Feng et al. introduced a sophisticated model which combined a packet-based anomaly detection using Bloom filters with a prediction using Long Short Term Memory (LSTM) network classifiers in Feng et al. (2017). If the next packet's signature was not among the top k signatures, the packet was considered as an anomaly. The method showed good results when applied to a real data set created from a gas pipeline SCADA system, but in comparison to our paper Modbus was used as transmission protocol instead of IEC 60870-5-104.

Another method targeting industrial control systems models the command and data sequences as coupled chains of a dynamic Bayesian network was developed by Yoon and Ciocarlie (2014). Probabilistic suffix trees are used as a variable-order representation of each of the two chains that can be learned incrementally. Additionally, false positives are reduced by introduction of possible missing elements. The method was evaluated for data from a Modbus network test-bed. Lin et al. proposed a Probabilistic Suffix Tree in Yoon and Ciocarlie (2014) to model and predict inter-arrival times of IEC 60870-5-104 spontaneous events. The authors discovered that events with good predictability might help developing intrusion detection methods.

In the most difficult, unsupervised variant, an unlabeled mixture of normal and anomalous data is available. Jiang et al. applied a one-class Support Vector Machine (SVM), which is also used for anomaly detection in our paper, with RBF kernel to a SCADA communication traffic time-series, where the created feature is the value relative to the previous one in Jiang and Yasakethu (2013). Experiments were carried out on simulated data from telecommunication networks, but no further hint to the used protocol was provided. Therefore, we try to find out, if anomaly detection is also possible, when a one-class SVM gets applied to IEC 60870-5-104 traffic.

The attack scenarios, used in this paper and which are described in more detail in "Attacks" section, are also representative for real world attacks on critical infrastructures. The *Triton* attack framework, which was built to interact with Triconex Safety Instrumented System (SIS) controllers, probes for specific ports using custom tools could be seen. User Datagram Protocol (UDP) port 1502 was used for sending broadcasts to discover remote systems (CISA 2019). Furthermore, the sending of unauthorized command messages, which happens during the Fuzzy Scan, had been observed in states 3 and

4 by *Stuxnet*, where two network bursts with instructions for the frequency converter drives were sent out (Falliere et al. 2011). Also various Denial of Service (DoS) attacks on SCADA systems happened before. *Industroyer*, a sophisticated piece of malware used in the *BlackEnergy* Attack 2015 in the Ukraine, was designed to cause an impact to the working processes of ICS used in electrical substations by blocking serial COM channels temporarily causing a denial of control (Cherepanov 2017).

Background

The rapid technical progress of the past years in information and communication technologies had a noticeable influence on the control technology components used in substations. Initially, the operators of the high-voltage networks coordinated necessary switching operations by telephone to the responsible control rooms in the substations. In the course of digitalization, data acquisition systems with automatically recorded measurement data, have been installed. Decentralized control technology components, also called Remote Terminal Units (RTU), are used to record and forward data (Vadari 2020). The information flow is illustrated in Fig. 2 in "Experiments" section. Operators of the SCADA System communicate with the RTUs of primary devices like transformers or circuit breakers. Communication also happens locally in the substation between the RTUs and those primary devices themselves.

Since 1995, data transmission in European substations has mainly been based on the asynchronous communication protocol IEC 60870-5-104 (Czechowski et al. 2015). It shows significant security gaps due to its age and its adaptation to the needs of ICS in the 1990s, such as low bandwidth (Pliatsios et al. 2020). Mostly, control or setting commands are sent to the primary devices. Measured values, as well as positioning data of circuit breakers get transmitted from the primary devices to the control system. The main purpose of the RTUs, which are located in so-called control cabinets in the substations, is to transmit control signals directly to the motors of the primary devices, which can be seen in Fig. 2 in "Experiments" section.

IEC 60870-5-104

To better understand the extraction process of the different field values for anomaly detection also covered in "Experiments" section, a brief overview of the IEC 60870-5-104 protocol is covered here. The standard specification for the IEC 60870-5-104 protocol combines the application layer of its predecessor IEC 60870-5-101 and a selection of the Transmission Control Protocol (TCP)/Internet Protocol (IP) protocol suite, which can be seen in Fig. 1.

<table>
<tr><th>Selected application functions</th><th>User process</th></tr>
<tr><td>Selection of Application Service Data Units (ASDU) of IEC 60870-5-101 and 104</td><td rowspan="2">Application Layer (L7)</td></tr>
<tr><td>Application Protocol Control Information (APCI)</td></tr>
<tr><td rowspan="4">Selection of TCP/IP Protocol Suite (RFC 2200)</td><td>Transport Layer (L4)</td></tr>
<tr><td>Network Layer (L3)</td></tr>
<tr><td>Link Layer (L2)</td></tr>
<tr><td>Physical Layer (L1)</td></tr>
</table>

Fig. 1 Protocol stack with IEC 60870-5-104 according to (Matoušek 2017)

Table 1 Mapping of Transmission causes to field values

Code	Cause of transmission
1-127	standard definitions
128-135	reserved for message routing
136-255	special usage

The basic frame in IEC 60870-5-104 is called Application Protocol Data Unit (APDU) and occurs in three different formats for data flow and link monitoring, as well as for information transmission. Furthermore, an APDU can be split up in two parts, the Application Protocol Control Information (APCI) and the Application Service Data Unit (ASDU).

The APCI contains basic information like APDU length or sender and receiver sequence numbers and has a fixed packet length of 4 Bytes. Otherwise the ASDU has a variable length to describe detailed attributes such as "Type Identification" or the "Cause of Transmission" (IEC 2006). These field values were also extracted to detect anomalies in the network traffic and get explained below in the tables. Table 1 shows the mapping of transmission causes to code values.

The types of IEC 60870-5-104 packets can be structured into 255 different Type Identification numbers, but like in the case of the cause of transmission field codes, not all of the available numbers have to be implemented (Skoko et al. 2014). Some of the most common values for Type IDs are listed in Table 2.

Snort

Snort is an open source, signature-based, Network Intrusion Detection System (NIDS), capable of performing real-time traffic analysis as well as packet logging on IP-based networks. In this paper, its performance in detecting anomalies in IEC 60870-5-104 SCADA traffic will be compared to machine-learning-based anomaly detection. Signature-based detection typically follows a blacklist approach. Snort basically acts as a packet sniffer and does protocol analysis as well as content matching by using rules which watch for specific fields in a network packet. A variety of attacks can be detected, because known signatures periodically get transformed into rules and can be downloaded from the developers website. Snorts detection engine processes the rules in order to know what fields to look for in the raw network packets. By considering the information gathered from the rules, it is able to detect occurring anomalous values. TCP, UDP, Internet Control Message Protocol (ICMP), and IP are the four protocols which are currently supported by Roesch et al. (2020).

Snort is set up by specifying the detection rules. Snort rules contain a rule header and the rule options part. The header includes the resulting Snort action, protocol value, source and destination IP addresses or port numbers (Roesch et al. 2020). Information about the inspecting part of the packet and alert messages are specified in the rule options

Table 2 Description of common ASDU Types

Type	Description
1	Single point informations
3	Double point information
13	Measured value, short floating point value
36	Measured value, short floating point value with time tag CP56Time2a

section. Snort has a standard basic rule set and additional rule sets which are deactivated by default and need to be activated selectively. In 2016, *Cisco Talos* released an additional rule set consisting of 33 rules, which are able to analyze IEC 60870-5-104 network traffic (Pacho 2016).

Experiments

The data set used in this paper consists out of five network packet traces in .pcap format, measured with Wireshark. The recording of the test data was done in a dedicated test substation operated by APG and a schematic representation can be seen in Fig. 2. As can be seen in the figure, the test substation contains a 110kV and a 220kV switchgear, each of which has its own Local Area Network (LAN). The *General Functions* ring consisted of several facilities important for real substations, like lock functions for primary devices.

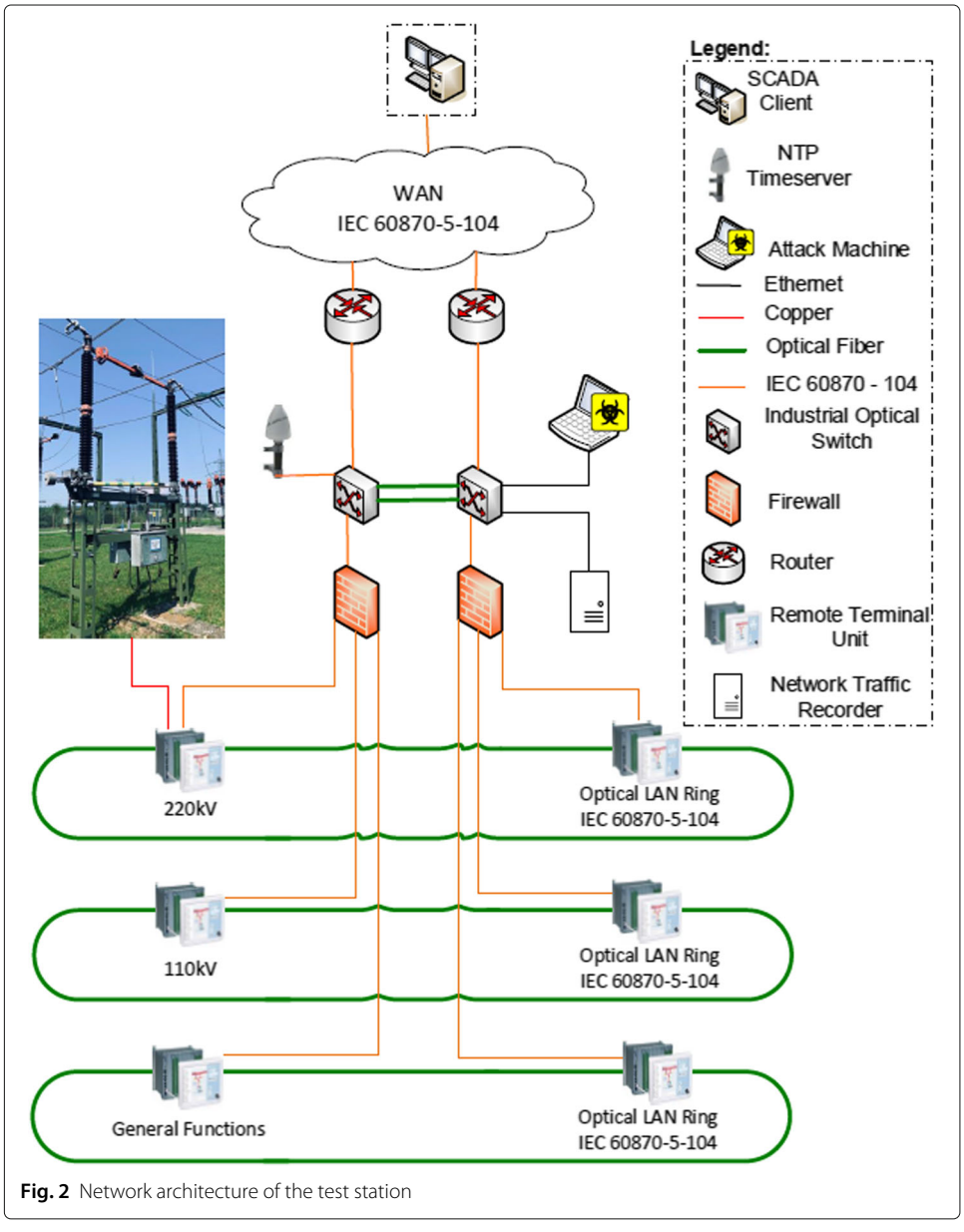

Fig. 2 Network architecture of the test station

The connection from the control center to the RTUs is also shown schematically in Fig. 2. With the help of switches, the RTUs are connected to the APG LAN as well as the firewall, the Network Time Protocol (NTP) server and the SCADA client. Instead of the primary devices normally used in substations, such as the circuit breaker shown in the figure, relays were set up to illustrate the correct control behavior. The station and field LAN areas are based on the IEC 60870-5-104 standard. Furthermore, the LAN itself was also integrated into the APG Wide Area Network (WAN) via IEC 60870-5-104 using a router.

The *Attack Machine* was a laptop capable to run Kali Linux as an operating system. All the different attack described in the Attacks section below were performed from here. Network traffic was recorded by a Raspberry Pi running Wireshark, which is connected on a trunk port at the same switch as the *Attack Machine*.

Attacks

In addition to normal operation, four different common attacks were used to attack the test substation. Table 3 gives a brief overview over the performed attacks.

According to the *Technique Matrix* proposed in the *MITRE ATT&CK for ICS Framework*, the types of attacks performed in this paper can be classified either as *Discovery* or as *Impair Process Control* which makes them representative for real world attack scenarios (MITRE 2020).

Feature extraction

Currently there are no best practice guidelines for feature extraction of the IEC 60870-5-104 protocol available. The implementation of this protocol is very application-specific, because the users can adjust which values they want to parameterize and some of the fields are only for special use. We extracted features that were considered as potentially discriminating attacks from normal traffic out of the Open Systems Interconnection (OSI) Layers 1 – 4 and the *Application Layer* 7. Features from Layers 1, 2 and 3 are described below in Table 4.

With Wireshark more features could be measured from Layer 4 and they are described in Table 5.

Features that are specific for the IEC 60870-5-104 protocol are shown in Table 6.

Table 3 Description of the performed attacks

Attack name	Description
Portscan	The aim of this scan, performed by the network reconnaissance tool "nmap", was to figure out which services were available by the network devices in the scanned network area. The used command was `nmap -sV -r 192.168.0.0-15`. By the parameter `-sV` it was possible to draw conclusions about open ports and ongoing services.
Vulnerability scan	A vulnerability scan against the top ten vulnerabilities in websites was done over the graphical interface of the vulnerability scanning tool "Nessus", targeting the web interfaces of the RTUs.
Protocol vulnerability scan	So-called *Fuzz-testing* is a popular security evaluation technique, in which hostile inputs are crafted and passed to the target system in order to reveal failures and security bugs. These hostile inputs could also be created by attackers to cause potentially unwanted behavior, like opening circuit breakers, in a SCADA system and so the IEC 60870-5-104 protocol fuzzer Aegis Studio has been applied.
Denial of service	To simulate a denial of service attack against the test substation, the application "hping" was used with the command `hping3 -flood -S 192.168.0/24`. The parameter `-flood` lead the program to send packets as fast as possible while `-S` symbolized the SYN Flood attack.

Table 4 Extracted field values from Layers 1 - 3

Field	Description	OSI Layer
frame_len	Represents the Ethernet frame length. Minimum size is 64 byte	L1
vlan_id	The VLAN ID field marks which VLAN the frame belongs to.	L2
ip_len	Total Length of the IP packet that includes the IP header and the user data.	
ip_flags_df	The *Don't Fragment* flag bit signals that fragmentation of this packet is not permitted.	
ip_ttl	*Time-to-live* tells a network router the time after which a packet should be discarded. Can be set to any value between 1 and 255.	

Finally, these features are enriched by a feature from Wireshark in Table 7.

Feature selection and creation for learning approaches

The whole data set consists of 288277 normal examples and 10000 attack samples (2500 per attack type). The input variables of the data set are a mixture of numerical and categorical ones.

In the first step, several variables are omitted from further analysis: There is a near perfect correlation between the frame-length and the ip_length and the tcp_length (R= 0.9983 and 0.9965), so only the frame length was retained. Similarly, x104_apdu_len was omitted, because it perfectly correlated with tcp_pdu_size. Finally, expert-severity was discarded since it had 97.8% missing values for the normal data.

Initial analyses showed that a very high number of values are missing. The reasons for missing values are likely systematic, because nearly all IP-variables, all TCP-variables and all IEC104-variables had the same number of missing values, respectively (see Table 8).

Since removing missing data would result in no samples left, the frame len was treated by introducing 4 new binary features describing, if Virtual Local Area Network (VLAN), IP-data, TCP-data and the tcp_pdu_size is missing, respectively. For numerical variables missing values were then replaced by the median values. Then they were standardized to the unit interval using the minimum and maximum values. Categorical variables with a small number of occurring attributes were generally coded as dummy variables: VLAN was coded into 4 variables according to the information, if VLAN had one of the values 20, 30, 40 or 50 (or missing, as already described above). In an analogous way this was

Table 5 Extracted field values from layer 4

Field	Description	OSI Layer
tcp_srcport	TCP Source Port is the port number used by the client sending the TCP segment. It is usually a number below 1024. IEC 60870-5-104 messages are sent over Port 2404.	
tcp_dstport	TCP Destination Port is the port number used by the client receiving the TCP packet. It is usually a number below 1024. The IEC 60870-5-104 default port is set to 2404.	L4
tcp_len	Total length of the TCP Segment, includes header and data.	
tcp_hdr_len	SLength of the TCP header can range from 20 bytes to 60 byte.	
tcp_window _size_value	The TCP window size, is an indication of how much bytes the receiving device is willing to receive at any point in time. In situations when the receiver is overwhelmed, it will advertise a zero window size.	
tcp_pdu_size	Equals tcp-len, but the value can only be dissected when it is a IEC 60870-5-104 packet.	

Table 6 Extracted features specific for x104 from Layer 7

Field	Description	OSI Layer
x104apci_apdulen	According to above there are packets with fixed length and with variable length containing ASDU. Fixed length packets contain control information and have the size of 4 Bytes.	
x104apci_rx	Number of the last packet the sender received.	L7
x104apci_tx	Send Sequence number of the transmitted frame. This value gets increased by one for every frame which gets sent to a specific IP address.	
x104asdu _typeid	Type IDs are described in Table 2.	
x104asdu _causetx	Cause of Transmission values are described in Table 1	

done for the other categorical variables IP-flag (0 or 1), asduTypeid (13 or 36) and asduCause (1 or 3). The numerical variable tcp_hdr_len was treated as being either 20 or not. Source and destination ports needed a special coding that involves both variables at a time. First, descriptive analysis showed that port 2404 is occurring in 53.9% of cases with other port Identification numbers being comparably rare. Additionally source and destination port are not the same. This information was coded as two binary variables portFeatureSrc2404 (source-port is 2404 and destination-port is not 2404) and portFeatureDst2404 (destination-port is 2404 and source-port is not 2404). The third case occurring (TCP-information is missing) was already coded above.

The resulting 21 features are like follows: 4 numerical variables (frame length, IP Ttl, TCP winSize and TCP pdu size); 5 binary *missing-value* variables (VLAN, IP, TCP, TCP pdu size, x104); 10 dummy variables for VLAN (4), IP-flag (1), xTcpTdrLen (1), asduTypeID (2) and asduCause (2); the 2 special variables for the ports.

Note that this way of coding was done based on descriptive analysis of only normal data. Therefore, this removes a possible advantage of the supervised approach where single variables could be turned into features additionally based on their class-separating ability. While this was also tried, in this case it resulted in nearly the same variables with very similar classification results. Therefore, this part is omitted. Finally, it is important to mention that feature generation was done in a blinded way by a person not familiar with the system or any Snort rules.

Detection of attacks

Four types of methodologies will be compared: supervised classification, semi-supervised anomaly detection, unsupervised anomaly detection and signature-based anomaly detection using Snort. The signature-based approach is expected to have a small or zero false positive rate, but the ability to detect the attacks is unknown. For the methods of semi-supervised and unsupervised anomaly detection, the attacks are unknown during learning so they have the potential to detect zero-day attacks. The question is, how close these methods can get to the signature-based and the classification based method. Since

Table 7 Extracted feature from Wireshark

Field	Description	Wireshark
x_ws_expert _severity	Expert information value from Wireshark. The application keeps track of problems like malformed packets.	

Table 8 NAN-statistics

Variable	vlan-id	IP	TCP	x104 rest	TCP-pdu-size	expert-severity
% missing	1.5%	3.7%	18.6%	71.3%	56.7%	97.8%

anomaly-based methods typically suffer from detecting too many false attacks (false positives), which limits their practical usability, the focus is especially laid on decreasing the false positive rates.

Misuse detection with snort

For misuse detection five packet capture files were extracted from the whole data set to be analyzed by Snort. The first file contained 50,000 normal examples and the attacks were split up into four files with equal size of 500 packets.

In order to use Snort, the rules must be specified. The NIDS only looks at the field values of network packets which are defined in the rules, if the fields do not contain the exact value defined in one of the rules, the rule will not trigger, but if a match is detected, an alarm will be raised either in the command line or in a separate log file. In addition to the default rule sets, the additional IEC 60870-5-104 rules were activated. In the course of this work, some rules were modified as follows: The rule to detect the port scan was modified from a default Snort rule and controlled if TCP traffic with set SYN, ACK or RESET flags entered the network. Additionally, the port scan rule was configured to only trigger if the packet had a Time To Live (TTL) of 64. This modification enables to distinguish the port scan from a denial of service attack, which showed different, lower TTLs. This explicit setting has the consequence that the rules would also trigger if a legitimate connection attempt, which includes a TCP handshake, to some network gear would happen. Such a connection attempt could occur, for example, in case of configuration changes. While this process was not part of the data set, this change could result in false positives in a productive environment.

The application of Snort on the five packet capture files with the rules as specified above leads to the following confusion matrix (Table 9). The result shows that no rule triggered when Snort analyzed the normal traffic which resulted in zero false positives.

Furthermore, the detection of the port scan done with *nmap* and the DoS Attack done with Syn Flood delivered very good results and nearly all packets were classified correctly. In the beginning five false negatives were found in the data of the Syn Flood attack, but after a further examination of the data set it turned out, that two of these packets were actually Domain Name System (DNS) requests from the attackers laptop. Since DNS

Table 9 Statistics for Misuse Detection with Snort

	Predicted class	
	Normal	**Attack**
True class		
normal	50000	0
Nmap	0	500
Syn Flood	3	497
Nessus	354	146
Fuzzy	466	34

requests should not appear in the normal traffic, the rule for detecting them was deactivated and the false positives could be eliminated. The remaining three packets contained Transport Layer Security (TLS) encrypted traffic and did not match any of the rules, so it was classified as normal.

The packets of the Nessus attack remained largely undetected. An investigation of possible reasons revealed a high amount of TLS encrypted packets. This seems legitimate because Nessus tested the web interfaces of the RTUs against several vulnerabilities. Since Snort is not able to decrypt these packets, most of the traffic was not classified as an attack. The detection rate of the fuzzy attack was also low. This can be assigned to the anatomy of the attack itself. During the Fuzzy Scan mixed packets from normal condition to malformed are sent into the network to find vulnerabilities in the protocol. In first tests, only 11 packets were classified as attacks. By adding additional custom rules as stated above the accuracy could only be improved by a small amount.

Considering the short time range of only 500 packets, the detection of some attacks was good, but the classification of the attack against the IEC 60870-5-104 protocol itself was significantly worse than expected.

Supervised intrusion detection: classification

For training of the classifiers 50,000 normal and 1,750 attack samples were randomly chosen using the features described above. Classification was performed with Matlab's tool Classification Learner app. As for anomaly detection the primary goal is to separate attacks from the normal cases, the outcome is binary: either normal or attack. The remaining samples comprise the test set which was used only for evaluation purposes. Several classifiers were tried: classification trees of various sizes, linear discriminant analysis, SVM (with linear and Gaussian kernels), k-nearest neighbor classifiers and tree ensembles trying both bagging and boosting. Since the goal was a principle assessment of the classification ability, no extensive fine tuning and parameter optimization was done (mainly the default settings were used) and the best model was chosen using standard 5-fold cross-validation.

For the two-class classification setting, a *medium sized* tree (maximum 20 splits allowed, using the Gini index as splitting criterion) showed the best performance (Table 10).

Without any normal case treated as an attack, all but 31 of the 8251 attack samples were detected. Thus, a precision of 100% and a recall of 99.62% was reached. The 31 undetected attack samples consist of 2 Syn Flood, 2 port scan, 7 vulnerability scan and 20 fuzzy samples. Note that several classifiers (for example a k-nearest neighbor classifier with k=5 for data re-standardized to standard normal distribution) reached a similar performance.

If the goal is to additionally distinguish between the different attacks, one needs to consider the multi-class-setting. Also there, the medium tree had slightly the best result

Table 10 Results of the classification approach: confusion matrix for a medium tree applied to test data for the twoclass-case

	Predicted class	
	Normal	**Attack**
True Normal	238277	0
Attack	31	8220

Table 11 Results of the classification approach: confusion matrix for a medium tree applied to test data for the multi-class case, where the goal is to distinguish the attacks, too

	Predicted class				
	Normal	**Nmap**	**Syn Flood**	**Nessus**	**Fuzzy**
True class					
Normal	238277	0	0	0	0
Nmap	4	2046	0	21	0
Syn Flood	2	0	2053	12	4
Nessus	7	7	455	1597	1
Fuzzy	40	6	1	1	1994

(Table 11). Note that in contrast to the two-class case now 53 instead of 31 attacks are not detected, because the tree must also reach distinguishability between the attacks. Distinguishability is quite good, just Nessus and SynFlood are difficult to distinguish. This pattern can also be seen for other classification models. Note that in a real application one could use the two-class-classifier to find anomalies and then, in a subsequent step, try to identify the attack using the multi-class-classifier. In practice this differentiation could increase the resilience by decreasing the time until proper countermeasures can be employed.

Although the groups are highly unbalanced (attacks are rare), classification results are quite satisfying even without special treatment. That is the reason why no methods like under- or oversampling have been applied.

Semi-Supervised anomaly detection

For training of the anomaly detection the same 50,000 normal samples as in the classification approach with the same features were used. The commonly employed one-class SVM (Schölkopf et al. 2000) was chosen for modeling the normal data. A Gaussian kernel was chosen, because the data are a mixture of numerical and categorical data. Since the data was normalized to a [0,1] interval, a kernel scale of 0.1 was tried out first. Due to the semi-supervised setting where only normal data are analyzed the outlier-fraction was set to zero. For evaluation, the model was applied to all the remaining data including the unused training attack data of the classification setting.

The result is quite encouraging, still nearly all attacks are detected (Table 12). However, in contrast to the classification setting now some false positives exist.

The confusion matrix is shown merely to compare the result with the classification result. The fact that now false positives exist, is of course a big disadvantage. However, it should be stated that the semi-supervised setting, where only the normal cases are presented to the model is by nature much harder. This has two consequences: (i) it cannot really expected that anomaly detection reaches similar result; (ii) anomaly detection results are typically not presented as a confusion matrix but by ROC-plots where the true

Table 12 Results of the semi-supervised anomaly detection approach: confusion matrix resulting when applying the model to test data

	Predicted class	
	Normal	**Attack**
True Normal	238141	136
Attack	56	9945

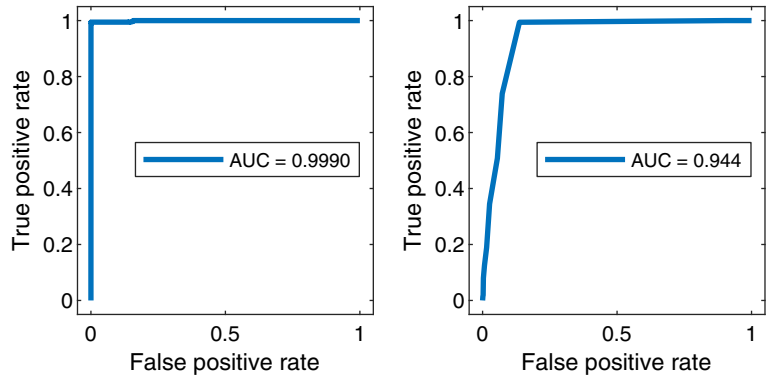

Fig. 3 ROC-plot for semi-supervised anomaly detection (left) and unsupervised anomaly detection (right)

positive rate is plotted against the false positive rate. As a measure for the performance the Area Under Curve (AUC) is usually given. Here, the AUC is 0.9990, which is very near to the optimal value of 1.

Unsupervised anomaly detection

The hardest but maybe most realistic intrusion detection setting is the unsupervised one, where the Intrusion Detection System (IDS) analyzes an unknown mixture of normal and contaminated traffic. For training all samples are unlabeled, and intrusion detection relies on the assumption that contaminated data shows up as anomalies. Since the relative amount of contaminated traffic is not known, for evaluation this information is treated as a parameter that is varied.

In the present case the same one-class SVM model as above is used as an anomaly detector, because it already showed good results for the semi-supervised setting. Therefore, a bad result due to model restrictions is ruled out and a worse choice of parameters can be attributed to the unsupervised setting. As a training set, a random sample of 50,000 samples of all data (both normal traffic and attacks) was randomly chosen. For the experiments the parameter outlier-fraction was varied from 0.01 to 0.3. The arising Receiver Operating Curve (ROC)-plot (Fig. 3, right panel) shows the True Positive Rate (TPR) (TPR=TP/P) dependent on the False Positive Rate (FPR) (FPR=FP/N) depending on the outlier- fraction (with P and N denoting all attack and normal samples, respectively). The ROC-plot and the AUC of 0.944 stated that the model seemed to be good.

While an AUC of 0.944 looks good at first glance, a detailed analysis shows that this is not sufficient for practical purposes. The results of the unsupervised approach are far inferior to the ones of the semi-supervised approach in terms of classification success: Table 13 shows the result of the model when the outlier-fraction is set near the right value of 3.5%.

Table 13 Results of the unsupervised anomaly detection approach: confusion matrix resulting when applying the model to test data

	Predicted class	
	Normal	**Attack**
True Normal	233724	6211
Attack	5463	2880

While the FPR of 2.6% seems to be good, due to high proportion of normal samples, this results in 6211 false positive samples which is too much for a practical IDS. Note this high number of false positives already occurs when still about two third of attacks are undetected.

Discussion of results

This study has several limits and should be considered as doing a first, static step towards intrusion detection. First, the duration where data were gathered, is rather short so time series analysis methods were excluded as detection methods. Additionally, due to the short duration, not all normal situations may be included and therefore generalization may be worse than estimated from the available data. For example, during the normal operation captured in this study only measurements or settings of switches are requested in single cases with the Cause of Transmission value of 3 or cyclically with a Cause of Transmission value of 1. However, situations exist that should also be considered as normal have not been seen. One such example would be an outage of a component where many more requests arise with the Cause of Transmission value of 20 which would be classified as an intrusion based on the existing data. However, while it is not an intrusion it is an anomaly whose detection would be valuable.

The present data set has a huge number of missing values which had to be accounted for by extracting corresponding features. A first analysis of the cause for missing data showed that during the port scan or during the Syn Flood attack, no transmission of measured values took place, so the fields here are not filled. The RTUs in the test facility crashed about two minutes after the start of the test because they could not withstand the number of queries. In a follow-up study, the unknown mechanisms leading to missing values especially for normal operation should be studied in detail before data are gathered. Labeling packets with many missing values as an intrusion would lead to an unacceptable number of false positives, since 821 *normal* packets had all values missing except the frame length. Therefore, many missing values are considered as normal. This has in turn the disadvantage that faking normal operation by setting values simply to missing could be a way to disguise an attack.

While accuracy of predictions is important, understanding the detection mechanisms is also crucial for further improvements of detection methods and the choice of possible countermeasures. In the raw packet data, we found that nmap sends SYN requests with tcp_window_size 1024 and gets ACK responses from the RTU with tcp_window_size 0 because it is overwhelmed. Due to the large number of inquiries and impending overload similar behaviour could be seen in the vulnerability scan data. Thus the tcp_win_size feature seems to be well suited to detect the port and Nessus vulnerability scan because they lead to situations when the receiver is overwhelmed. Then it will advertise a zero window size, which enables detection of these two attacks.

The missing of data can also be caused by attacks: during the Syn Flood attack hping tried to send as many SYN requests to a device in the test station as possible which caused them to crash. There the data mainly showed the SYN requests, which announced a tcp_window_size of 1024 bytes. Due to the overwhelming count of requests, no answers from the RTU were received which resulted in missing x104apci and x104asdu values. These fields were also missing in the Nessus packets because there could only be two IEC 60870-5-104 APCI packets be found in the data. During the attack Nessus

also overwhelmed the network with encrypted traffic and HyperText Transfer Protocol (HTTP) requests.

The x104rx and x104tx counters could be used to detect the fuzzy attack: the x104rx and x104tx counter start at zero and increase with each packet sent or received. They only get reset when a new connection is established. During the fuzzy test, a new connection attempt happens from the attackers laptop to the RTU. Therefore the x104rx and x104tx counter values at the Fuzzy test are very low, in contrast to normal operation mode where connections usually exist for a long time.

Conclusion and future work

In order to facilitate a comparison of the different approaches, Table 14 shows an overview of the results for the two-class setting. As expected Snort had no false positives but misses some attacks. The fact that the supervised approach also showed zero false positives with fewer undetected attacks proved that one can get excellent results using machine learning based on the generated features. However, as a matter of principle the supervised approach was not able to detect new, unknown attacks. The more suitable semi-supervised approach was promising as it is better in detecting attacks than Snort at the cost of some false positives. Although the supervised and the semi-supervised approach showed that the features would suffice to yield good results, the unsupervised approach was clearly the worst solution both in terms of false positives and detection capability.

The application of machine learning methods to IEC 60870-5-104 network traffic demonstrates their promise in addressing security concerns in the ICS domain. With the chosen set of features, and treating missing values as additional binary variables, the results of the evaluation demonstrated the power of machine learning at detecting the considered attacks. The supervised method, with a medium sized decision tree by using the Gini-index as splitting criterion, as well as the semi-supervised approach, based on a one-class SVM with a Gaussian kernel, could reach 100% precision and also extraordinary high recall and a nearly perfect AUC for classifiers, respectively. The results are good enough to be further analyzed in future intrusion detection approaches for ICS.

However, the unsupervised anomaly detection, which can be considered as the most realistic intrusion detection setting and was trained with completely unlabeled data on the proven one-class SVM from the semi-supervised approach, showed a false positive rate of 2.6% by an AUC of 0.944. This seems good, but in a productive environment, the number of generated false positives would definitely be too high.

Despite the overall good performance of the various learners, with the signature-based approach, using Snort as an IDS, the results were much worse than the ones from the unsupervised method. Only the DoS and the portscan attack could be clearly identified by Snort because the signatures for them are very obvious. The NIDS detects an anomaly,

Table 14 Summary of detection results of the four different approaches for the two-class setting

Detection method	False positive packets	Undetected attack packets
Snort	0	823
Supervised	0	31
Semi-supervised	136	56
Unsupervised	6211	2880

if a network packet exactly matches the pattern of one of the activated rules but does not consider other deviations of the normal system, which occur in the cause of the attack, as unusual. To improve the detection rate of the signature-based method in the SCADA data set used in this paper, it would also be necessary to develop additional rules to better meet the requirements of the test system but this requires a tremendous amount of work and extensive knowledge of the network.

As the proposed attacks involve timing, format, and protocol violations, we intend to extend this work to explore additional features that consider these elements in making the intrusion detection decision. Improved performance, especially considering the unsupervised approach, will be necessary in order to maintain the same accuracy in a more complex environment comprised of multiple RTUs and SCADA systems. The data sets only covered short time ranges, therefore more data will be needed to achieve this. In a next step, it will be necessary to also catch traffic during re-configuration or outage of network devices. Furthermore the reasons for missing data points have to be investigated.

The consideration of other protocols than IEC 60870-5-104 was basically out of scope of this paper but because of the promising results that the attack detection showed it could be considered in further researches. Especially an extension to IEC 61850 would be interesting because this specific protocol is widely used for automation tasks by distribution system operators (Khodabakhsh et al. 2020).

The novelty is in the comparison of intrusion detection capabilities between machine learning approaches and a signature based NIDS by applying them to network traffic, which resembles the behaviour of a real substation. This work can be seen as a foundation, and encourage the future exploration of more complex SCADA systems, more difficult attack vectors, and more advanced machine learning methods to discriminate attacks on critical infrastructures based on the IEC 60870-5-104 protocol.

Abbreviations

APCI: Application protocol control information. 5, 14; APDU: Application protocol data unit. 4, 5; APG: Austrian power grid AG. 2, 6; ASDU: Application service data unit. 5, 8; AUC: Area under curve. 12, 13, 15; DNS: Domain name system. 10; DoS: Denial of Service. 3, 10, 15; FPR: False positive rate. 13; HTTP: HyperText transfer protocol. 14; ICMP: Internet control message protocol. 5; ICS: Industrial control system. 1, 4, 7, 15; IDS: Intrusion detection system. 12, 13, 15; IEC: International electrotechnical commission. 1, 8, 10, 11, 14, 16; IP: Internet Protocol. 4, 5, 7, 9; LAN: Local area network. 6; LSTM: Long short term memory. 3; MITM: Man-In-The-Middle. 2; NIDS: Network intrusion detection system. 5, 10, 15, 16; NTP: Network time protocol. 6; OSI: Open systems interconnection. 7; ROC: Receiver operating curve. 13; RTU: Remote terminal units. 4, 6, 7, 10, 14, 16; SCADA: Supervisory control and data acquisition. 2, 7, 15, 16; SIS: Safety instrumented system. 3; SVM: Support vector machine. 3, 11, 13, 15; TCP: Transmission control protocol. 4, 5, 8, 10; TLS: Transport layer security. 10; TPR: True positive rate. 13; TTL: Time to live. 10; UDP: User datagram protocol. 3, 5; VLAN: Virtual local area network. 9; WAN: Wide area network. 6

About this supplement

This article has been published as part of *Energy Informatics Volume 3 Supplement 1, 2020: Proceedings of the 9th DACH+ Conference on Energy Informatics*. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-3-supplement-1.

Authors' contributions

This paper was written by Michael Egger (50%), Günther Eibl (45%) and Dominik Engel (5%). The detailed contributions are as follows: The idea for the paper was developed by Michael Egger (75%) and Dominik Engel (25%). Experiments and data collection were done by Michael Egger (100%). Snort was applied and evaluated by Michael Egger (100%). Features were generated by Günther Eibl (100%), supervised, semi-supervised and unsupervised learning methods were applied and evaluated by Günther Eibl (100%). The corresponding sections were mostly written by them. Writing of the rest of the paper by sections fixing the order (Michael Egger, Günther Eibl, Dominik Engel): Abstract: (50%, 45%, 5%), Introduction (35%, 40%, 25%), Related Work (45%, 45%, 10%), Background (75%, 10%, 15%), Discussion of results (60%, 25%, 15%), Conclusion and future work (20%, 50%, 30%). The author(s) read and approved the final manuscript.

Funding

Günther Eibl and Dominik Engel gratefully acknowledge funding by the Federal State of Salzburg under the WISS2025 program. Publication costs were covered by the DACH+ Energy Informatics Conference Organizers, supported by the Swiss Federal Office of Energy.

Availability of data and materials
The Snort rule files, which were used for misuse detection with Snort, as well as the complete data set, are available on GitHub under the following link: https://github.com/eggermi/Energy-Informatics-2020.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Austrian Power Grid AG, IZD-Tower, Wagramer Str. 19, Vienna, Austria. [2]Salzburg University of Applied Sciences, Center for Secure Energy Informatics, Urstein Süd 1, Puch/Salzburg, Austria.

Published: 28 October 2020

References
Ang CKG, Utomo NP (2017) Cyber security in the energy world. In: 2017 Asian Conference on Energy, Power and Transportation Electrification (ACEPT). IEEE, Singapore
Berthier R, Sanders WH, Khurana H (2010) Intrusion detection for advanced metering infrastructures: Requirements and architectural directions. In: 2010 First IEEE International Conference on Smart Grid Communications. IEEE, Gaithersburg
Butt UJ, Abbod M, Lors A, Jahankhani H, Jamal A, Kumar A (2019) Ransomware threat and its impact on SCADA. In: 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3). IEEE, London
Cherepanov A (2017) Win32/industroyer a new threat for industrial control systems. techreport, ESET. https://www.welivesecurity.com/wp-content/uploads/2017/06/Win32_Industroyer.pdf
CISA (2019) Mar-17-352-01 hatman - safety system targeted malware (update b). techreport, U.S. Department of Homeland Security. https://us-cert.cisa.gov/sites/default/files/documents/MAR-17-352-01%20HatMan%E2%80%94Safety%20System%20Targeted%20Malware_S508C.pdf
Czechowski R, Wicher P, Wiecha B (2015) Cyber security in communication of SCADA systems using IEC 61850. In: 2015 Modern Electric Power Systems (MEPS). IEEE, Wroclaw
Falliere N, Murchu LO, Chien E (2011) W32.stuxnet dossier [Whitepaper]. https://nsarchive2.gwu.edu//NSAEBB/NSAEBB424/docs/Cyber-044.pdf
Feng C, Li T, Chana D (2017) Multi-level Anomaly Detection in Industrial Control Systems via Package Signatures and LSTM Networks. In: 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, Denver. pp 261–272
Hoeve M (2013) Detecting intrusions in encrypted control traffic. In: Proceedings of the First ACM Workshop on Smart Energy Grid Security - SEGS 2013. ACM Press, New York
IEC (2006) Telecontrol equipment and systems - Part 5-104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles. IEC 60870-5-104:2006. IEC, Geneva
Jiang J, Yasakethu L (2013) Anomaly detection via one class SVM for protection of SCADA systems. In: 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE, Beijing
Khodabakhsh A, Yayilgan SY, Houmb SH, Hurzuk N, Foros J, Istad M (2020) Cyber-security gaps in a digital substation: From sensors to SCADA. In: 2020 9th Mediterranean Conference on Embedded Computing (MECO). IEEE, Budva
Matoušek P (2017) Description and analysis of iec 104 protocol. techreport. Faculty of Information Technology BUT, Brno University of Technology
Maynard P, McLaughlin K, Haberler B (2014) Towards understanding man-in-the-middle attacks on IEC 60870-5-104 SCADA networks. In: 2nd International Symposium for ICS & SCADA Cyber Security Research 2014. BCS Learning & Development, Niederösterreich
MITRE (2020) ATT&CK for Industrial Control Systems. https://collaborate.mitre.org/attackics/index.php/Main_Page. Accessed 01 June 2020
Pacho C (2016) IEC 60870-5-104 Protocol Detection Rules. https://blog.snort.org/2016/12/iec60870-5-104-protocol-detection-rules.html. Accessed 22 May 2020
Peterson D (2009) Quickdraw: Generating security log events for legacy SCADA and control system devices. In: 2009 Cybersecurity Applications & Technology Conference for Homeland Security. IEEE, Washington
Phillips B, Gamess E, Krishnaprasad S (2020) An evaluation of machine learning-based anomaly detection in a SCADA system using the modbus protocol. In: Proceedings of the 2020 ACM Southeast Conference. ACM, Tampa
Pliatsios D, Sarigiannidis P, Lagkas T, Sarigiannidis AG (2020) A survey on SCADA systems: Secure protocols, incidents, threats and tactics. IEEE Commun Surv Tutor 22:1–1
Roesch M, Green C, Cisco, Team S (2020) SNORT Users Manual 2.9.16. https://www.snort.org/#documents. Accessed 22 May 2020
Schölkopf B, Williamson R, Smola A, Shawe-Taylor J, Piatt J (2000) Support vector method for novelty detection. In: Advances in Neural Information Processing Systems. Proceedings of the 12th International Conference on Neural Information Processing, Denver. pp 582–588
Skoko V, Atlagic B, Isakov N (2014) Comparative realization of IEC 60870-5 industrial protocol standards. In: 22nd Telecommunications Forum Telfor (TELFOR). IEEE, Belgrade
Vadari M (2020) Electric System Operations : Evolving to the Modern Grid. Artech House, Boston
Yan Y, Qian Y, Sharif H, Tipper D (2012) A survey on cyber security for smart grid communications. IEEE Commun Surv Tutor 14(4):998–1010
Yang Y, McLaughlin K, Littler T, Sezer S, Pranggono B, Wang HF (2013) Intrusion detection system for IEC 60870-5-104 based SCADA networks. In: 2013 IEEE Power & Energy Society General Meeting. IEEE, Vancouver
Yoon M-K, Ciocarlie G (2014) Communication Pattern Monitoring: Improving the Utility of Anomaly Detection for Industrial Control Systems. In: NDSS Workshop on Security of Emerging Networking Technologies, San Diego. pp 1–10. https://doi.org/10.14722/sent.2014.23012

Publisher's Note