SPEAR SIEM: A Security Information and Event Management System for the Smart Grid

Panagiotis Radoglou-Grammatikis[a], Panagiotis Sarigiannidis[a,*], Eider Iturbe[b], Erkuden Rios[b], Saturnino Martinez[b], Antonios Sarigiannidis[c], Georgios Eftathopoulos[d], Ioannis Spyridis[d], Achilleas Sesis[d], Nikolaos Vakakis[e], Dimitrios Tzovaras[e], Emmanouil Kafetzakis[f], Ioannis Giannoulakis[f], Michalis Tzifas[f], Alkiviadis Giannakoulias[g], Michail Angelopoulos[h,i], Francisco Ramos[j]

[a] *Department of Electrical and Computer Engineering,*
University of Western Macedonia, Kozani, Greece
[b] *TECNALIA, Basque Research and Technology Alliance (BRTA),Derio,Spain*
[c] *Sidroco Holdings Ltd,Limassol,Cyprus,3113*
[d] *0INF, Imperial Offices, London, UK, E6 2JG*
[e] *Center for Research and Technology Hellas / Information Technologies Institute,6th km Charilaou-Thermi Road, Thessaloniki,Greece*
[f] *Eight Bells Ltd,Agias paraskevis 23, P.C. 2002, Strovolos, Nicosia,Cyprus*
[g] *European Dynamics,12, Jean Engling str.,Luxembourg,L-1466*
[h] *University of Piraeus*
[i] *Testing Research & Standards Center / Public Power Corporation SA,Leontariou 9, Kantza, Athens, Attica,15351*
[j] *Schneider Electric, Charles Darwin s/n, Edificio Bogaris,Sevilla,Spain,41092*

Abstract

The technological leap of smart technologies has brought the conventional electrical grid in a new digital era called Smart Grid (SG), providing multiple benefits, such as two-way communication, pervasive control and self-healing. However, this new reality generates significant cybersecurity risks due to the heterogeneous and insecure nature of SG. In particular, SG relies on legacy communication protocols that have not been implemented having cybersecurity in mind. Moreover, the advent of the Internet of Things (IoT) creates severe cybersecurity challenges. The Security Information and Event Management (SIEM) systems constitute an emerging technology in the cybersecurity area, having the capability to detect, normalise and correlate a vast amount of security events. They can orchestrate the entire security of a smart ecosystem, such as SG. Nevertheless, the current SIEM systems do not take into account the unique SG peculiarities and characteristics like the legacy communication protocols. In this paper, we present the Secure and PrivatE smArt gRid (SPEAR) SIEM, which focuses on SG. The main contribution of our work is the design and implementation of a SIEM system capable of detecting, normalising and correlating cyberattacks and anomalies against a plethora of SG application-layer protocols. It is noteworthy that the detection performance of the SPEAR SIEM is demonstrated with real data originating from four real SG use case (a) hydropower plant, (b) substation, (c) power plant and (d) smart home.

Keywords: Anomaly Detection, Cybersecurity, Deep Learning, Intrusion Detection, Machine Learning, SCADA, Security Information and Event Management, Smart Grid

1. Introduction

The next-generation electrical grid, also known as Smart Grid (SG), intends to address multiple challenges of the conventional model, such as generation diversification, demand response and the optimal management of the existing resources. In particular, the point of convergence between electrical engineering and the Internet of Things (IoT) creates an intelligent layer over the current model, which allows the development of appropriate business applications offering pervasive control, self-monitoring and self-healing [1]. However, this transition to the SG encloses significant cybersecurity risks that can lead to disastrous consequences [2]. Characteristic examples are the BlackEnergy3 (2015) and Crashoverride (2016) Advanced Persistent Threats (APTs) that caused extensive blackouts in Ukraine [3]. The necessary presence of legacy systems, such as Supervisory Control and Data Acquisition (SCADA)/Industrial Control Systems (ICS) and the advent of IoT increase the attack surface of SG [4]. On the one side, SCADA/ICS use insecure communication protocols, like Modbus and IEC 60870-5-104 [5] that allow the cyberattackers to perform various cyberattacks. On the other side, IoT generates new cybersecurity concerns [6]. First, IoT relies on the Internet model, which is vulnerable by itself. Second, the vast amount of the IoT data, such as the smart metering data constitutes an attractive target for potential cyberattackers. Finally, the capability of the various objects to interact with each other without any human intervention increases the privacy concerns.

Taking into account the critical cybersecurity issues of SG, both academia and industry have investigated possible countermeasures. First, the IEC 62351 standard has defined a set of security controls and guidelines based mainly on existing authentication and authorisation technologies [7–9]. Moreover, the Security Information and Event Management (SIEM) systems constitute an emerging technology organsisng the monitoring, detection and prevention measures of a smart ecosystem, such as SG [10]. In particular, a SIEM can aggregate, normalise and correlate various security events, thus identifying potential security violations [10]. A security event is considered a normalised message related to the security status of the monitored infrastructure [10]. However, the continuous progression of cyberattacks

*Corresponding Author

Email addresses: pradoglou@uowm.gr (Panagiotis Radoglou-Grammatikis), psarigiannidis@uowm.gr (Panagiotis Sarigiannidis), Eider.Iturbe@tecnalia.com (Eider Iturbe), Erkuden.Rios@tecnalia.com (Erkuden Rios), satur.martinez@tecnalia.com (Saturnino Martinez), asarigia@sidroco.com (Antonios Sarigiannidis), george@0infinity.net (Georgios Eftathopoulos), yannis@0infinity.net (Ioannis Spyridis), achilleas@0infinity.net (Achilleas Sesis), nikovaka@iti.gr (Nikolaos Vakakis), tzovaras@iti.gr (Dimitrios Tzovaras), mkafetz@8bellsresearch.com (Emmanouil Kafetzakis), giannoul@8bellsresearch.com (Ioannis Giannoulakis), tzifas@8bellsresearch.com (Michalis Tzifas), Alkiviadis.Giannakoulias@eurodyn.com (Alkiviadis Giannakoulias), m.angelopoulos@dei.com.gr (Michail Angelopoulos), francisco.ramos@se.com (Francisco Ramos)

and malware requires the simultaneous evolution and adoption of the necessary countermeasures. First, the guidelines of IEC 62351 cannot be adopted quickly by the vendors and manufacturers, especially when the corresponding SCADA/ICS operate in real-time since safety issues can arise. On the other side, the current SIEM systems include a limited set of intrusion and anomaly detection mechanisms regarding the SG application-layer protocols [11]. In addition, they are characterised by a lack of understanding between the complicated relations of the real intrusion instances and fake alerts [12]. Therefore, the difficult goal of ensuring intelligent, safe, viable and efficient SG becomes a major need filled with significant and far-reaching challenges.

Based on the aforementioned remarks, this paper presents a SIEM system called Secure and PrivatE smArt gRid (SPEAR) SIEM, which is exclusively focused on the SG ecosystem. The proposed SIEM is focused on detecting, normalising and correlating security events against SG environments and calculating the reputation value of each SG asset (hardware or virtual device), which reflects how secure and trustworthy the functionality of each asset is. To this end, SPEAR SIEM is capable of detecting, normalising and correlating cyberattacks and anomalies against a plethora of SG communication protocols. Moreover, it includes anomaly detection models that process time-series operational data (i.e., raw electricity measurements) of four SG environments, namely (a) hydropower plant, (b) substation, (c) power plant and (d) smart home. The architectural model of SPEAR SIEM consists of five primary components, namely (a) AlienVault OSSIM SIEM [13], (b) SPEAR SIEM Basis, (c) Message Bus, (d) Big Data Analytics Component (BDAC), (e) the Visual-based Intrusion Detection System (VIDS) and (f) Grid Trusted Module (GTM). Section 3 analyses the architecture of SPEAR, detailing the functionality of each component. The contributions of this paper are summarised in the following points.

- **Providing a SIEM system specially designed for SG**: The proposed SIEM can detect, normalise and correlate the security events related to multiple SG application-layer cyberattacks.
- **Providing a set of operational data-based anomaly detection models**: The specific models can detect anomalies based on the operational data (i.e., time series electricity data) of four SG use cases: (a) hydropower plant, (b) substation, (c) power plant and (d) smart home.
- **Implementing a visual-based detection mechanism through ML/DL dimensionality reduction techniques**: Through VIDS, the security administrator can identify potential, undetected security issues.
- **Implementing a reputation mechanism reflecting the trust value of each SG asset**: GTM can calculate the reputation value of each SG asset based on the security events and alerts received.
- **Developing two novel Deep Neural Networks (DNNs), namely SPEAR Stacked Denoising Autoencoder (SDAE) and Payload Text CNN Classi-**

fier: The proposed DNNs are part of BDAC, detecting particular cyberattacks and anomalies, respectively.

- **Evaluating a plethora of ML/DL methods for detecting various cyberattacks in four SG use cases**: The various ML and DL methods of BDAC and VIDS are evaluated in four SG use cases: (a) hydropower plant, (b) substation, (c) power plant and (d) smart home.

The rest of this paper is organised as follows. Section 2 presents relevant works. Section 3 is devoted to the architecture of SPEAR SIEM. Section 4 presents the evaluation analysis. Finally, section 5 concludes this paper. It is noteworthy that SPEAR SIEM was implemented under the H2020 SPEAR project [14].

2. Related Work

Many papers have studied the security and privacy issues of SG. Some of them are listed in [2, 11, 15–20]. In particular, in [11], the authors provide a comprehensive survey regarding the intrusion detection in the SG sector. After providing the necessary background about the SG and IDS, the authors investigate 37 cases related to detecting cyberattacks and anomalies against (a) the entire SG ecosystem, (b) the Advanced Metering Infrastructure (AMI), (c) SCADA systems, (d) substations and (e) synchrophasors. The DiSIEM project in [15] evaluates the efficiency of seven SIEM systems: (a) HP ArcSight, (b) IBM QRadar, (c) Intel McAfee Enterprise Security Manager, (d) Alienvault OSSIM, (e) XL-SIEM, (f) Splunk and (g) Elastic Stack based on various criteria like data sources, data storage, User and Entity Behaviour Analytics (UEBA), risk analysis, exposed APIs, resilience, event management and visualisation. Similarly, in [16], L. Cui et al. examine the detection of False Data Injection (FDI) attacks in SG, utilising ML methods. In particular, the authors focus on FDI attacks against (a) energy consumption data, (b) state estimation and (c) load forecasting. In [17], S. Quincozes et al. provide a survey about the intrusion detection and prevention mechanisms concerning the digital substations. M. Gunduz and R. Das in [18] investigate the various threats in SG, providing the corresponding solutions and directions for future work. In a similar manner, in [2], P. Kumar et al. present a detailed study about the smart metering networks, paying special attention to the security, privacy and open research issues. Accordingly, in [19], M. Hassan et al. present a compilation about the differential privacy techniques for Cyber-Physical Systems (CPS). Finally, in [20], I. Stellios et al. study IoT-based cyberattacks against Critical Infrastructures (CIs), including SG, SCADA and smart home environments. Subsequently, we pay our attention to some specific cases, highlighting the differences with our work. Each paragraph focuses on a dedicated case.

In [21], R. Leszczyna and M. Wrbel review three open-source SIEM systems based on the SG conditions. In particular, the SIEM systems investigated are (a) AlienVault OSSIM [13], (b) Cyberoam iView [22] and (c) Prelude [23]. For the evaluation procedure, the authors adopt the Solution Merit Index (SMI) by B. Sahay and K. Gupta [24]. The proposed methodology relies on (a) primary criteria and (b) secondary criteria. The primary criteria are (a) number of available and compatible sensors, (b) number of the out-of-the-box sensors, (c) diversity of available sensors, (d) real-time performance, (e) range and flexibility of reporting, (f) alert correlation, (g) auto-response capabilities. On the other hand, the secondary criteria are (a) documentation comprehensiveness, (b) complexity of the installation process, (c) complexity of the system configuration, (d) portability and (e) hardware requirements. Based on the primary criteria, the OSSIM performance reaches 97% while the performance of Cyberoam iView and CS Prelude reach 76% and 24.3%, respectively. Concerning the secondary criteria, the Prelude performance approaches 86.8% while OSSIM and Cyberoam iView reach 59.4% and 56.6%. The complete SMI for OSSIM is 81.96% while the SMI of Prelude and Cyberoam iView is calculated at 80.68% and 37.16%. Therefore, according to the authors, OSSIM is a complete SIEM system appropriate for the situational awareness of an SG environment.

In [25], K. Zhang et al. introduce the Backward Influence Factor (BIF) algorithm capable of processing and mining intrusion patterns originating from a sequence of IDS alerts. The proposed algorithm handles efficiently the sequence data analysis issues like random noise, disordering and element missing. In particular, it consists of five phases: (a) normalisation, (b) intrusion action extraction, (c) intrusion session pruning, (d) correlation discovery and (e) dynamic correlation graph construction. During the first phase, the IDS alerts are normalised into a common format. Next, the intrusion action extraction phase follows by discriminating the alerts based on two elements: (a) the source IP address and (b) the destination IP address. Subsequently, the intrusion actions are specified, considering the type and the destination port fields. Next, the intrusion session pruning phase undertakes to separate long intrusion actions into smaller sequences called intrusion sessions. Then, the pruning process starts, removing the sub-patterns from the initial sequence. Next, the correlation discovery phase aggregates all pruned sessions, based on their starting time. The BIF algorithm is responsible for computing the attraction score between two sessions. The attraction score is expressed by the Influence Factor (IF). Finally, the last phase generates a dynamic correlation graph based on the higher IF values.

In [26], M. Albanese et al. provide a probability-based framework, which assesses and quantifies whether the sequence of events is unexplained, considering models of previously learned behaviours. Based on the authors, such events can originate from (a) intrusion detection and (b) alert correlation processes. Although their work can be applied to both processes, it does not aim to overcome or replace them. In contrast, the proposed framework runs on top of them, analysing whether their output is adequately explained. The authors consider that the available intrusion detection models and alert correlation models are ineffective for explaining a sequence of events identified in data streams. The input for the intrusion detection decision is a vector of network packets, while the alert correlation procedure relies on a set of alerts. The proposed framework is actually based on their previous work in [27] related to the cybersecurity settings. In particular, the authors adapt the algorithms of [27] appropriately in order to estimate the probability that a sequence of events is unexplained. The evaluation results demonstrate the efficacy of the proposed framework in terms of accuracy and scalability.

K. Zhang et al. in [28] provide an alert correlation framework called Intrusion Action

Based Correlation Framework (IACF) presenting a similar architecture as in their previous work in [25]. The proposed framework enhances the aggregation of cybersecurity alerts, the intrusion actions association, the extraction of intrusion sessions and finally the intrusion scenarios identification. IACF is composed of three phases: (a) normalisation, (b) intrusion session construction and (c) intrusion scenario construction. First, the cybersecurity alerts are aggregated and divided into two groups based on the source IP and the destination IP address. Thus, the intrusion actions are extracted based on the sequence of alerts displaying an intrinsic correlation. Next, the extraction of intrusion sessions follows, aiming to split long sequences of intrusion actions into smaller intrusion sessions. To this end, two algorithms are used, namely (a) Time-lag based Sequence Splitting (TSS) and (b) Sequence Pruning Algorithm (SPA). Finally, the intrusion scenario construction starts, following the assumption that intrusion sessions presenting a binary relation can compose the intrusion scenario. Finally, a correlation graph is generated, consisting of the intrusion sessions and their binary relations. The evaluation analysis shows the efficacy of IACF in terms of (a) the recognition of multi-step cyberattacks, (b) the performance of the proposed algorithms and (c) accuracy.

In our previous work in [29], we present an IDS called ARIES (smArt gRid Intrusion dEtection System), which focuses on SG. The architecture of the proposed IDS consists of three main modules: (a) Data Collection Module, (b) ARIES Analysis Engine and (c) Response Module. The Data Collection Module is responsible for collecting (a) network flow statistics, (b) Modbus/TCP payload information and (c) operational data. Next, the ARIES Analysis Engine consists of three detection layers related to the aforementioned data types. The first layer focuses on detecting cyberattacks, utilising network flow statistics. In particular, it consists of two complementary detection models: (a) Intrusion Detection Model and (b) Anomaly Detection Model. First, IDM takes place, adopting a decision tree classifier capable of detecting five cyberattacks: (a) File Transfer Protocol (FTP) brute-force attacks, (b) Secure Shell (SSH) brute-force attacks, (c) DoS, (d) bot and (e) port scanning. If the detection outcome of IDM is normal, then ADM is activated, trying to identify a potential anomaly. To this end, an autoencoder is used. Next, the second layer is devoted to detecting potential Modbus anomalies by analysing the Modbus payload through the isolated forest algorithm. Finally, the third layer focuses on electricity-related operational data and adopts the ARIES Generative Adversarial Network (GAN) to recognise relevant anomalies. Finally, the Response Module notifies the security administrator and can generate some automated firewall rules to mitigate the impact of the potential cyberattacks/anomalies. The main novelty of this work lies in the development of the ARIES GAN at the third detection layer. The evaluation results demonstrate the efficacy of ARIES, including a comparison study with multiple ML/DL methods.

In [30], the authors introduce an anomaly-based IDS for the electrical grid, based on operational data of a real power plant. The proposed IDS consists of two primary stages (a) the training stage and (b) the testing stage. In the first stage, the ML training process is carried out, while the testing stage allows real-time anomaly detection, predicting whether an anomaly exists or not. In particular, the training stage includes four modules: (a) Data Collection Module, (b) Pre-Processing Module, (c) Feature Selection module and (d) Training

Module. Accordingly, the testing stage comprises four modules: (a) Data Collection Module, (b) Pre-processing Module, (c) Anomaly Detection Module and (d) Response Module. The main innovation of this work lies in the fact that the Pre-Processing Module (in both stages) adopts a complex data representation, which results in better detection performance. The evaluation analysis demonstrates the efficiency of the complex data representation, comprising a plethora of ML and DL methods, such as Principal Component Analysis (PCA), One-Class Support Vector Machine (SVM), isolation forest, Angle-Based Outlier Detection (ABOD), SOS and autoencoder.

In [31], M. Ali et al. present MALGRA, which constitutes a combined ML and N-Gram malware feature extraction and detection system. The methodology behind MALGRA includes six steps: (a) dynamic analysis, (b) Application Programming Interface (API) call feature extraction, (c) N-Gram creation, (d) feature reduction, (e) N-Gram model preparation and (f) testing using samples. First, the authors follow a dynamic analysis in order to investigate the behaviour of various malware, utilising an Artificial Intelligence (AI) sandbox, called SNDBOX. In particular, the authors investigate two scenarios. The first one focuses on the API calls and their arguments' memory location to construct N-Grams. An N-Gram is a subset of a given data sample with a length of n. In the second scenario, the N-Grams are implemented based on the function calls and their arguments' address. Next, the Term FrequencyInverse Document Frequency (TF-IDF) method is adopted in order to reduce the feature space. TF-IDF is a statistical method assessing how relevant a word is in a document. Finally, the N-Grams are transformed into binary vectors introduced to the ML methods. The evaluation analysis demonstrates the effectiveness of MALGRA. To this end, the authors used four ML methods and 60 malicious samples from the virus share website. The ML methods used are (a) Naive Bayes, (b) Decision Tree, (c) Random Forest and (e) Logistic Regression. Based on the experimental results, the Logistic Regression accomplishes the best detection accuracy.

M. Ghafouri et al. [32] provide a detection and mitigation system against cyber-physical attacks related to a Wide Area Management System (WAM) and its components (i.e., Phasor Measurement Unit (PMU) and Phasor Data Concentrator (PDC)). A voltage stability problem refers to the instability of the power system to maintain and control the appropriate voltage values at all buses during the regular operation or after an electrical disturbance. This situation can lead to various consequences, such as load curtailment, brownouts or even power outage. First, the authors study the cyberattacks against WAM, discriminating two main categories: (a) cyberattacks against communication links and (b) cyberattacks related to the WAM devices and data. Based on this study, an attack generation algorithm is implemented, targeting the voltage stability. The proposed attack generation algorithm relies on the power flow equations, addressing traditional anomaly detection techniques. Next, the authors introduce a detection mechanism adopting the Thevenin Equivalent (TE) parameters. It is worth noting that the proposed detection scheme does not rely on historical data and is capable of detecting the aforementioned cyberattacks. Next, a mitigation framework is presented, allowing the system operator to specify the compromised PMUs or PDCs and recover their proper functionality. The authors evaluate their system with three use cases: (a) 7-bus transmission power system, (b) 39-bus New England system and IEEE

118-bus system. The experimental result confirms the efficiency of the proposed detection and mitigation system.

Undoubtedly, the previous works introduce significant contributions. Based on [21], we use the AlienVault OSSIM as a basis for the proposed SPEAR SIEM. However, AlienVault OSSIM focuses mainly on signature-based techniques without considering the special peculiarities and characteristics of SG. It is noteworthy that the commercial version of AlienVault OSSIM called AlienVault Unified Security Management (USM) [13] includes some correlation rules and directives about SCADA systems. However, both AlienVault OSSIM and AlienVault USM do not utilise ML and DL solutions targeted to the SG application layer protocols. Furthermore, although several research efforts use ML and DL for detecting cyberattacks or anomalies against SG application-layer protocols, they cannot discriminate the exact cyberattack type. For instance, they may detect a DoS attack without describing specifically how this attack is related to the respective application-layer protocol. Moreover, a few papers pay attention to industrial protocols like BACnet and IEC 60870-5-104, without again specifying the exact cyberattack type. Also, it is worth mentioning that the existing works do not correlate the various SG-related security events.

Therefore, based on the aforementioned remarks, we provide a comprehensive SIEM system dedicated to SG, aiming to address the current shortcomings. First, SPEAR SIEM includes a variety of ML and DL detectors capable of discriminating the exact cyberattack type. Next, it introduces visual-based detection mechanisms that allow the security administrator to identify undetected security issues. Next, SPEAR SIEM correlates the security events related to Modbus, thus composing security alerts reflecting actual attack scenarios. Finally, SPEAR SIEM introduces an extra protection level that quantifies the trust value of each SG asset based on the security events received by the various detectors.

3. SPEAR SIEM Architecture

The SPEAR SIEM architecture relies on the ARCADE framework [14] and consists of three layers as illustrated in Fig. 1. First, at the Data Capturing Layer, the SPEAR SIEM Basis collects the necessary data for the intrusion detection processes. Three types of data are captured: (a) network flow statistics, (b) packet payload information and (c) operational data (i.e., time-series electricity data). Then, the Detection Layer follows, where the intrusion and anomaly detection processes take place, generating the corresponding security events. There are four intrusion detection processes: (a) network flow-based detection, (b) packet-based detection, (c) operational data data-based detection and (d) visual-based detection. The first three are implemented by BDAC while VIDS carries out the last. Finally, the correlation layer follows where the security events are correlated. There are two kinds of correlation. The first one is implemented by VIDS through correlation rules for the Modbus/TCP protocol, thus producing alerts reflecting multi-step Modbus-related attack scenarios. The second kind is conducted by GTM, which receives the various security events and calculates each SG asset's reputation value. Fig. 2 illustrates the interactions among the SPEAR SIEM components. First, the OSSIM Sensors (part of AlienVault OSSIM) and the SPEAR Sensors (part of SPEAR SIEM Basis) are distributed throughout the SG infrastructure, thus monitoring, collecting and parsing various data. This information is transmitted then to the OSSIM Server (part of AlienVault OSSIM) and Data Acquisition, Parsing and Storage (DAPS) (part of SPEAR SIEM Basis), respectively. The OSSIM Server normalises this information and uses a MySQL database for the storage, while DAPS uses an Elasticsearch database and distributes this information to BDAC and VIDS. The normalised information stored in the OSSIM server and the detection results of BDAC and VIDS are named ‘security events’. Through the Message Bus, these security events are sent to GTM and VIDS. Finally, the security events originating from BDAC and VIDS, the GTM updated reputation values and the security alerts are visualised by VIDS. The following subsections analyse each component in detail.

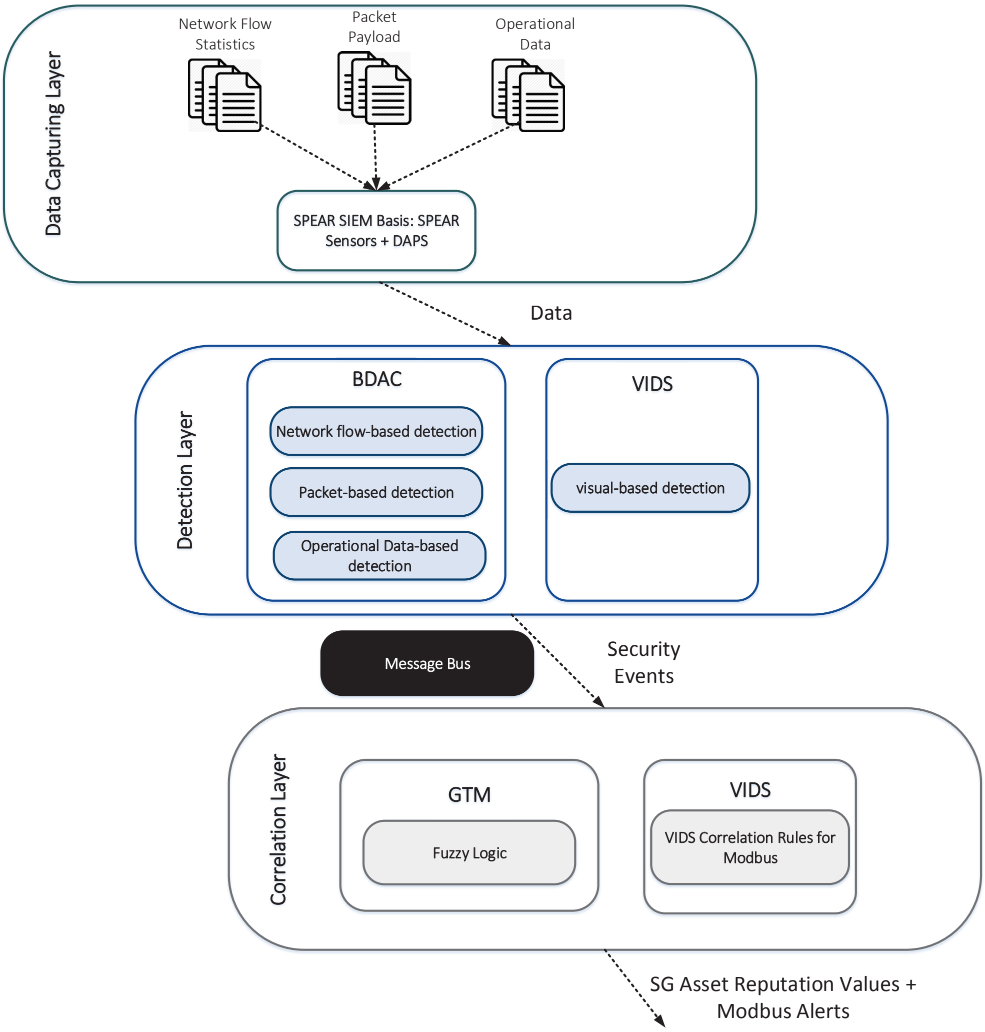


Figure 1: SPEAR SIEM Architecture.

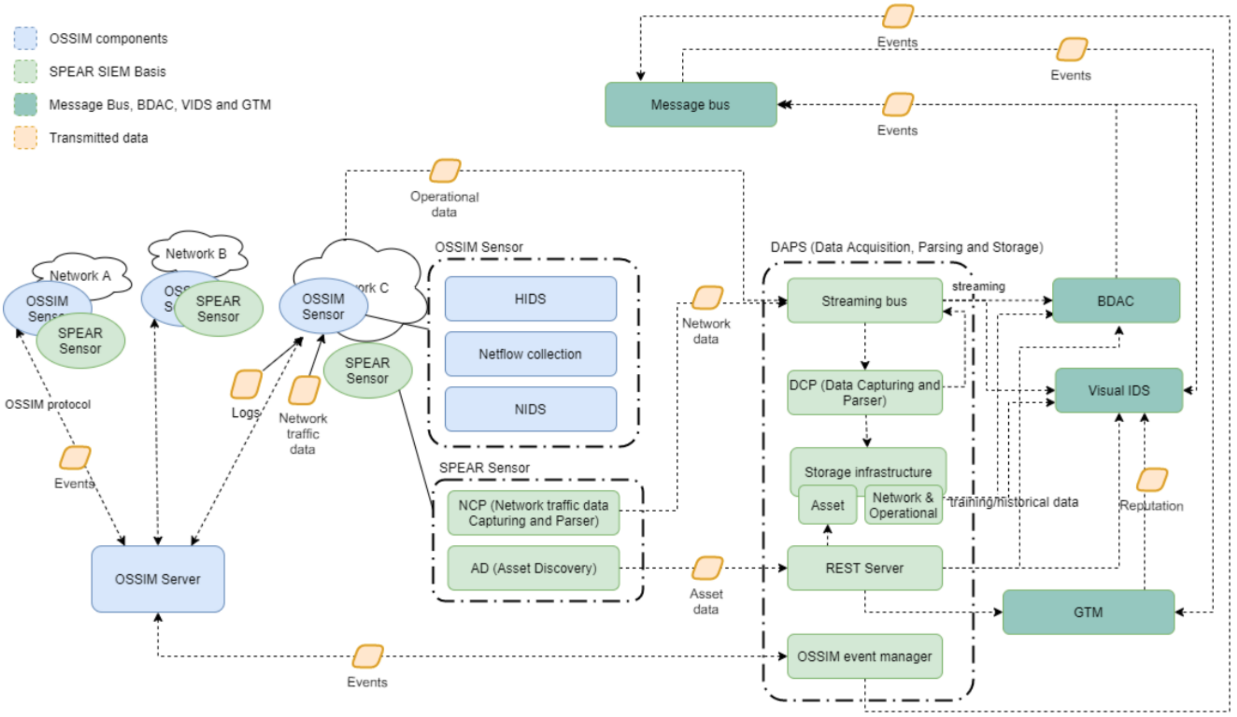


Figure 2: SPEAR SIEM Operation Flow.

3.1. AlienVault OSSIM

AlienVault OSSIM is an open-source SIEM system capable of providing several security capabilities. Its architecture is composed of two main components: (a) OSSIM Server and (b) OSSIM Sensors. The OSSIM Sensors are deployed throughout the SG infrastructure, collecting and normalising security-related information from any asset (hardware or virtual devices). A wide range of OSSIM sensors is available, including firewalls, Host-based Intrusion Detection Systems (HIDS) and Network-based Intrusion Detection Systems (NIDS). Next, the OSSIM Server aggregates and correlates the security information gathered by the OSSIM Sensors, thus composing security alerts. A security alert is defined as a set of security events associated with each other [13]. It is noteworthy that AlienVault OSSIM is already implemented and provided by AT&T. In the context of this paper, we use the AlienVault OSSIM as a signature-based detection, producing the corresponding security events and alerts.

3.2. SPEAR SIEM Basis

SPEAR SIEM Basis follows a server-sensor architecture consisting of two components: (a) SPEAR Sensors and (b) DAPS. Fig. 3 illustrates the SPEAR SIEM Basis architecture, showing the relationship between the SPEAR Sensors and DAPS. In particular, a SPEAR Sensor consists of two main functional elements (a) Network Capturer and Parser (NCP) and (b) Asset Discovery (AD). NCP uses a runtime network analyser to continuously capture, parse and forward network traffic data to DAPS. More detailed, NCP analyses a plethora

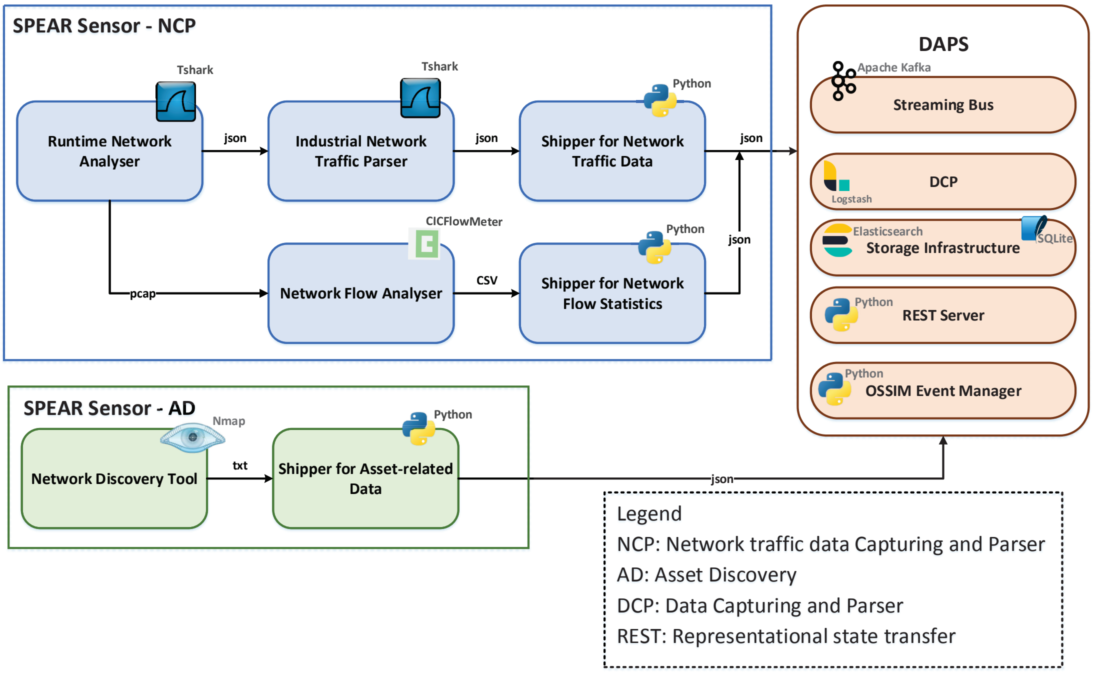


Figure 3: SPEAR SIEM Basis Architecture.

of SG application-layer protocols by isolating specific payload information and relevant network flow statistics used by BDAC and VIDS to detect intrusions/anomalies. To this end, Tshark [33] and CICFlowMeter [34] are adopted. The format of the network flow statistics is defined by CICFlowMeter [34]. Finally, AD utilises periodically Nmap [35] to discover which assets (hardware and virtual devices) are active, thus collecting and delivering relevant information to DAPS.

DAPS is a centralised server consisting of five functional elements: (a) Streaming Bus, (b) Data Capturing and Parser (DCP), (c) Storage Infrastructure, (d) Representational State Transfer (REST) Server and (e) OSSIM Event Manager. First, the Streaming Bus is in charge of providing a near real-time streaming data to BDAC and VIDS in order to detect intrusions/anomalies during the prediction phase. In particular, the Streaming Bus relies on Apache Kafka and transmits (a) specific packet payload information, (b) network flow statistics (c) operational data and (e) honeypot data. The operational data is retrieved directly by DAPS from the corresponding SG use case, while the honeypot data is given by the Honeypot Manager, which is an external component analysed in [14]. The SPEAR honeypots and how the honeypot data is introduced into DAPS is out of the scope of this paper. More details about this content are provided by [36, 37] and [14], respectively. Next, DCP is responsible for importing the data published in the Streaming Bus and storing them in the Storage Infrastructure. In turn, the Storage Infrastructure persists all captured data originating either from the SPEAR Sensors or DAPS. More precisely, the payload information related to the SG application-layer protocols, the network flow statistics, the operational data (i.e., time series electricity data) and the honeypot data are stored into an Elasticsearch database. On the other hand, the asset-related data originating from AD is stored in an SQLite database. Next, the REST server transmits the asset-related to BDAC, VIDS and GTM. Finally, the OSSIM Event Manager is in charge of retrieving OSSIM security events from the OSSIM Server periodically and forwarding them to the Message Bus. The OSSIM security events are retrieved with all the attributes as defined by AlienVault [13] and then they are parsed to match with the SPEAR SIEM security event format (Table A.8).

3.3. BDAC: Big Data Analytics Component

BDAC is a backend component consisting of four main modules: (a) Data Receiving Module, (b) Training Module, (c) BDAC Analysis Module and (d) Security Event Extraction Module. First, the Data Receiving Module is responsible for communicating with the SPEAR SIEM Basis to receive the appropriate data for detecting potential cyberattacks and anomalies. Then, the BDAC Analysis Engine analyses this data, identifying potential cyberattacks and anomalies. The BDAC Analysis Engine includes 24 intrusion and anomaly detection models that analyse appropriately the various data types. The intrusion and anomaly detection models of the BDAC Analysis Engine are updated periodically via the Training Module. In particular, the Training Module is fed by the Data Receiving Module with new normal and malicious data, thereby re-training the current intrusion/anomaly detection models of the BDAC Analysis Engine only whether their accuracy and the F1 score are better compared to the previous ones. Finally, based on the BDAC Analysis Engine's response, the SPEAR Event Extraction Module extracts the corresponding security events. The following subsections provide more details about the architectural components of BDAC. It is noteworthy that all BDAC modules are located in a common place so that the communication interfaces among them are not necessary.

3.3.1. Data Receiving Module

The Data Receiving Module communicates with the DAPS subcomponent of the SPEAR SIEM Basis in order to receive (a) network flow statistics, (b) payload information of the SG application layer protocols, (c) operational data, (d) honeypots' logs and (e) asset-related data. In particular, the Data Receiving Module utilises the DAPS Streaming Bus to monitor the network flow statistics and honeypots' logs, while the payload of the SG application-layer protocols and the operational data are received periodically via the DAPS Storage Infrastructure of DAPS, utilising specific threshold values. According to the network characteristics of each SG use case, these threshold values are defined appropriately. Finally, the asset-related data is received from the DAPS REST Server.

3.3.2. Big Data Analysis Engine

The BDAC Analysis Engine is the core architectural component of BDAC responsible for detecting possible cyberattacks and anomalies. It focuses mainly on detecting cyberattacks and anomalies against the SG application-layer protocols, including Modbus, DNP3, IEC 60870-5-104, IEC 61850 (MMS), BACnet, MQTT, HTTP and SSH. Therefore, the corresponding detection models are formed (e.g., Modbus Intrusion/Anomaly Detection Models).

For each of these protocols, two detection categories are identified: (a) Network Flow-Based Detection Models and (b) Packet-Based Detection Models. The first category (i.e., Network Flow-Based Detection Models) is devoted to identifying cyberattacks and anomalies based on network flow statistics. It is divided into two subcategories: (a) Network Flow-Based Intrusion Detection Models and (b) Network Flow-Based Anomaly Detection Models. In particular, the Network Flow-Based Intrusion Detection Models rely on multiclass classification ML/DL methods in order to identify specific cyberattack types. In contrast, the Network Flow-Based Anomaly Detection Models use outlier/novelty detection to detect potential anomalies. The difference between a cyberattack and anomaly lies in the fact that a cyberattack specifies a particular intrusion type like a Denial of Service Attack (DoS) or a port scan, while an anomaly can originate from an intrusion or another reason like an electrical disturbance. Hence, the second subcategory (i.e., Network Flow-Based Anomaly Detection Models) operates as complementary to the first one (i.e., Network Flow-Based Intrusion Detection Models) based on the flowchart presented in Fig. 4. In particular, by checking the TCP/User Datagram Protocol(UDP) source and destination port of a network flow received by the Data Receiving Module, the corresponding SG application layer protocol is identified. Therefore, the appropriate Network Flow-Based Intrusion Detection Model related to this protocol is activated (e.g., Modbus Network Flow-Based Intrusion Detection Model). Then, if this model detects a specific attack, the corresponding security event is generated via the Security Event Extraction Module. Otherwise, the relevant Network Flow-Based Anomaly Detection Model is activated (e.g., Modbus Network Flow-based Anomaly Detection Model). Similarly, if the specific model identifies an anomaly, the corresponding security event is produced. Otherwise, the TCP/UDP Network Flow-Based Intrusion/Anomaly detection models are used in a similar manner. It should be noted that the last models have been presented in our previous work in [29] and focus on the TCP and UDP protocols of the transport-layer. Hence, if the TCP/UDP Network Flow-Based Intrusion Detection Model detects a specific attack, the respective security event is generated. Otherwise, the TCP/UDP Network Flow-Based Anomaly Detection Model undertakes to discover whether a possible anomaly exists, generating a suitable security event or not. Finally, it should be noted that this process is carried out continuously, always monitoring new network flow statistics.

The second category (i.e., Packet-Based Anomaly Detection Models) identifies potential anomalies based on the payload information of each packet. Fig. 5 illustrates the relevant flowchart of the Packet-based Anomaly Detection Models. First, the information of each packet is received through the Data Receiving Module. Next, the corresponding application layer protocol is identified to execute the appropriate packet-based anomaly detection model. Finally, if an anomaly is detected, the corresponding security event is produced via the Security Event Extraction Module.

Apart from the application-layer protocols, the BDAC Analysis Engine uses operational data (i.e., raw electricity measurements) and honeypots logs in order to identify additional anomalies. Thus, the corresponding models are identified, i.e., Operational Data-Based Anomaly Detection Models and Honeypot-Based Anomaly Detection Models. The operational data originate from the local environment of each SG use case and is captured

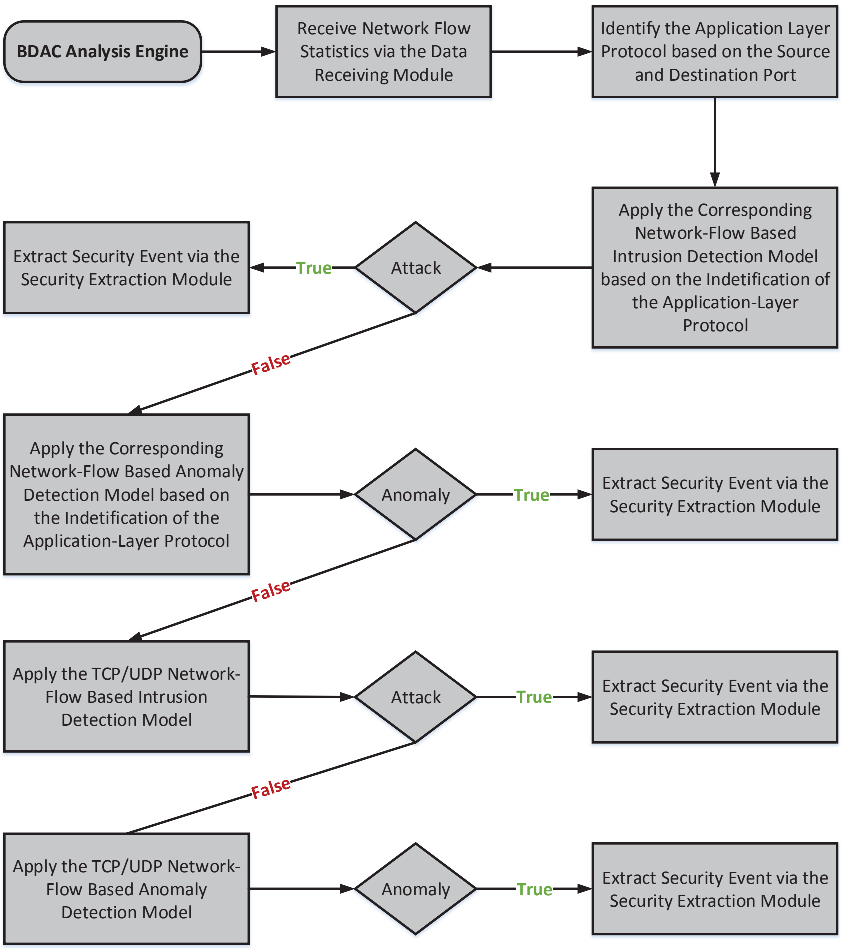

BDAC Analysis Engine
Receive Network Flow Statistics via the Data Receiving Module
Identify the Application Layer Protocol based on the Source and Destination Port
Apply the Corresponding Network-Flow Based Intrusion Detection Model based on the Indetification of the Application-Layer Protocol
Attack
True
Extract Security Event via the Security Extraction Module
False
Apply the Corresponding Network-Flow Based Anomaly Detection Model based on the Indetification of the Application-Layer Protocol
Anomaly
True
Extract Security Event via the Security Extraction Module
False
Apply the TCP/UDP Network-Flow Based Intrusion Detection Model
Attack
True
Extract Security Event via the Security Extraction Module
False
Apply the TCP/UDP Network-Flow Based Anomaly Detection Model
Anomaly
True
Extract Security Event via the Security Extraction Module

Figure 4: Flowchart of the Network Flow based Detection Models.

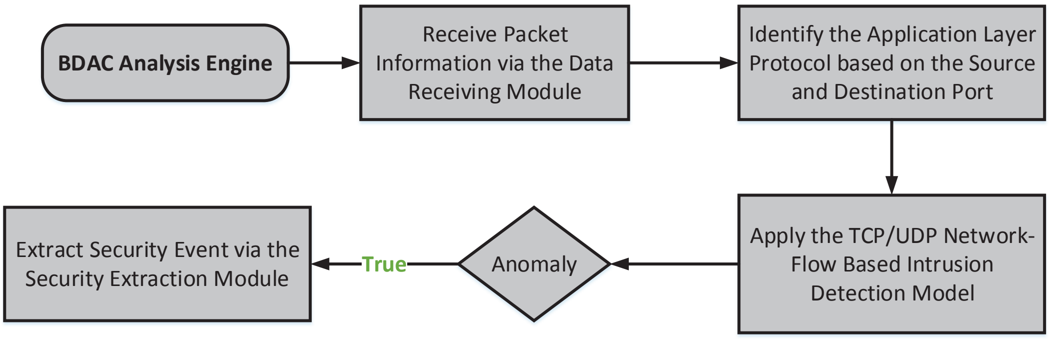

BDAC Analysis Engine
Receive Packet Information via the Data Receiving Module
Identify the Application Layer Protocol based on the Source and Destination Port
Apply the TCP/UDP Network-Flow Based Intrusion Detection Model
Anomaly
True
Extract Security Event via the Security Extraction Module

Figure 5: Flowchart of the Packet based Detection Models.

through the SPEAR SIEM Basis. In particular, four kinds of operational data were considered based on four individual SG use cases, i.e., (a) hydropower plant, (b) substation, (c) power plant and (d) smart home. On the other side, any interaction with a honeypot is considered an anomalous activity since a legitimate user will not interact with it. Figure 6 and Figure 7 show the flowcharts related to the Operational Data-Based Anomaly Detection Models and Honeypot-Based Anomaly Detection Models, respectively. Regarding the Operational Data-Based Anomaly Detection Models, initially, a series of operational data (i.e., electricity measurements) is collected through the Data Receiving Module. Next, the respective Operational Data-Based Anomaly detection model is applied. If an anomaly is recognised, a relevant security event is generated by the Security Event Extraction Module. On the other side, the honeypots logs are received via the Data Receiving Module and are transformed into security events by the Security Event Extraction Module. Therefore, based on the previous remarks, the following subsections analyse the respective intrusion/anomaly detection models per SG application-layer protocol and those related to the operational data and honeypots logs.

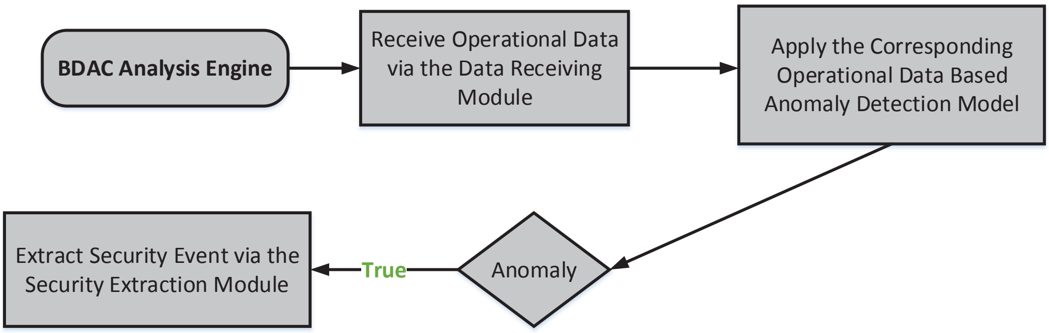

BDAC Analysis Engine
Receive Operational Data via the Data Receiving Module
Apply the Corresponding Operational Data Based Anomaly Detection Model
Anomaly
True
Extract Security Event via the Security Extraction Module

Figure 6: Flow Diagram of the Operational Data-Based Detection Models.

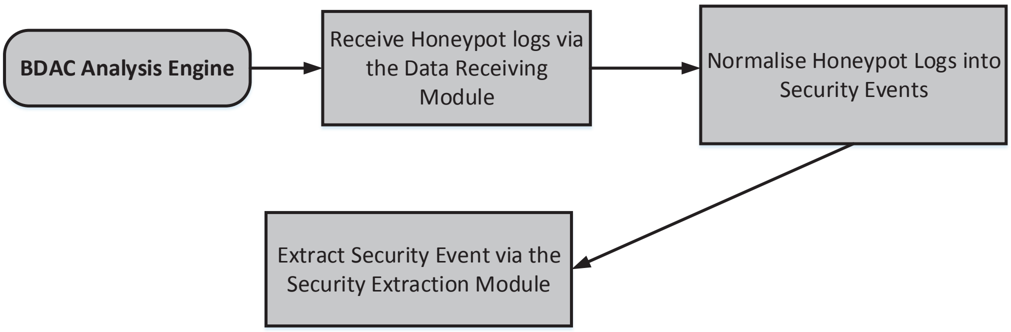

BDAC Analysis Engine
Receive Honeypot logs via the Data Receiving Module
Normalise Honeypot Logs into Security Events
Extract Security Event via the Security Extraction Module

Figure 7: Flow Diagram of the Honeypot-Based Detection Model.

3.3.2.1. Modbus/TCP Intrusion/Anomaly Detection Models. Three Modbus/TCP-related intrusion/anomaly detection models were implemented: (a) Modbus/TCP Network Flow-Based Intrusion Detection Model, (b) Modbus/TCP Network Flow-Based Anomaly Detection Model and (c) Modbus Packet-Based Anomaly Detection Model. The first two rely on Modbus/TCP related network flow statistics. In particular, the Modbus/TCP Network Flow-Based Intrusion Detection Model utilises a Decision Tree Classifier [38] aiming to identify malicious network flows indicating the following Modbus/TCP cyberattacks:

- **modbus/function/readInputRegister (DoS)**: This DoS attack floods the target system with Modbus/TCP Read Input Register packets (Function Code 04).
- **modbus/function/writeSingleCoils**: This unauthorised access attack sends a Modbus/TCP packet (Function Code 05), which changes the status of a single coil either to ON or OFF. Since the Modbus/TCP protocol does not include any authentication or authorisation mechanism, a cyberattacker can send malicious Modbus/TCP commands against the target system.
- **modbus/scanner/getfunc**: This reconnaissance attack enumerates all Modbus/TCP function codes used and supported by the target system.
- **modbus/dos/writeSingleRegister**: This DoS attack floods the target system with Modbus/TCP Write Single Register packets (Function Code 06).
- **modbus/function/readDiscreteInputs (DoS)**: This DoS attack floods the target system with Modbus/TCP Read Discrete Inputs packets (Function Code 02).
- **modbus/function/readHoldingRegister (DoS)**: This DoS attack floods the target system with Modbus/TCP Read Holding Register packets (Function Code 03).
- **modbus/function/readCoils (DoS)**: This DoS attack floods the target system with Modbus/TCP Read Coils packets (Function Code 01).
- **modbus/function/readInputRegister**: This unauthorised attack sends a Modbus/TCP packet (Function Code 04) used to read the values of specific input registers.
- **modbus/function/writeSingleRegister**: This unauthorised access attack sends a Modbus/TCP packet (Function Code 06) in order to write a value to a specific holding register.
- **modbus/dos/writeSingleCoils**: This DoS attack floods the target system with Modbus/TCP Write Single Register packets (Function Code 06).
- **modbus/function/readDiscreteInput**: This unauthorised access attack sends a Modbus/TCP packet (Function Code 02) to read the status of specific discrete inputs.
- **modbus/scanner/uid**: This reconnaissance attack enumerates which slave IDs are activated.
- **modbus/function/readCoils**: This unauthorised access attack sends a Modbus/TCP packet (Function Code 01) to read the status of specific coils.

- **modbus/function/readHoldingRegister**: This unauthorised access attack sends a Modbus/TCP packet (Function Code 03) to read the values of specific holding registers.

The aforementioned cyberattacks are implemented by Smod, a widely known pen-testing tool related to Modbus [39, 40]. The Modbus Network Flow-Based Anomaly Detection Model adopts the DIDEROT Autoencoder [41], identifying anomalous Modbus/TCP network flows. The DIDEROT autoencoder is analysed in our previous work in [41]. Finally, the last model focuses on the payload of the Modbus/TCP packets, recognising Modbus/TCP anomalous packets based on the Isolation Forest method [42]. Since there are no sufficient intrusion/anomaly detection datasets related to the Modbus/TCP, it is worth mentioning that relevant Modbus/TCP intrusion/anomaly detection datasets were constructed, by implementing Modbus/TCP cyberattacks against a real smart home as well as an emulated SG environment. To this end, the directions provided by A. Gharib et al. [43] were followed. The evaluation analysis related to the Modbus/TCP intrusion/anomaly detection models is analysed in section 4.

3.3.2.2. DNP3 Intrusion/Anomaly Detection Models. The BDAC Analysis Engine encloses two detection models related to DNP3: (a) DNP3 Network Flow-Based Intrusion Detection Model and (b) DNP3 Network Flow-Based Anomaly Detection Model. Both of them rely on DNP3-related network flow statistics. In particular, the DNP3 Network Flow-Based Intrusion Detection Model utilises a Decision Tree Classifier [38], which recognises the following five DNP3-related cyberattacks.

- **Injection**: Since the DNP3 protocol does not include sufficient authorisation mechanisms, this attack injects malicious DNP3 packets in a communication established between a DNP3 outstation and master.
- **Flooding**: This DoS attack floods continuously the target system with DNP3 packets.
- **DNP3 Reconnaissance**: This reconnaissance attack identifies whether the DNP3 protocol is used by the target system or not.
- **Replay**: This attack replays DNP3 packets originating from a legitimate party to the other endpoint.
- **Masquerading**: In this attack, the cyberattacker imitates the behaviour of a legitimate asset, sending the appropriate DNP3 packets.

The DNP3 Network Flow-Based Anomaly Detection Model uses the ABOD method [44, 45], thus identifying anomalous DNP3 network flows. Both models were trained, utilising normal DNP3 network flow statistics coming from a real substation environment as well as from the DNP3 intrusion detection dataset of N.Rodofile et al. [46]. The evaluation analysis of these DNP3 intrusion/anomaly detection models is presented in our previous work in [41].

3.3.2.3. IEC 60870-5-104 Intrusion/Anomaly Detection Models. Three IEC 60870-5-104-related detection models are incorporated into the BDAC Analysis Engine: (a) IEC 60870-5-104 Network Flow-Based Intrusion Detection Model, (b) IEC 60870-5-104 Network Flow-Based Anomaly Detection Model and (c) IEC 60870-5-104 Packet-Based Anomaly Detection Model. The first two rely on IEC 60870-5-104 related network flow statistics specified by the 2404 TCP port. In particular, the IEC 60870-5-104 Network Flow-Based Intrusion Detection Model utilises a Decision Tree Classifier [38], detecting the following cyberattacks.

- **c_ci_na_1_DoS**: This DoS attack floods the target system with c_ci_na_1 IEC 60870-5-104 packets.
- **c_sc_na_1**: This unauthorised access attack injects a c_sc_na_1 IEC 60870-5-104 packet to the target. Since IEC 60870-5-104 does not comprise sufficient authentication and authorisation mechanisms, a potential cyberattacker can perform malevolent IEC 60870- 5-104 commands to manipulate the target system.
- **c_ci_na_1**: This unauthorised access attack injects a c_ci_na_1 IEC 60870-5-104 packet to the target.
- **c_se_na_1**: This unauthorised access attack injects a c_se_na_1 IEC 60870-5-104 packet to the target system.
- **c_sc_na_1_DoS**: This DoS attack floods the target system with c_sc_na_1 IEC 60870-5-104 packets.
- **c_se_na_1_DoS**: This DoS attack floods the target system with c_se_na_1 IEC 60870-5-104 packets.
- **m_sp_na_1_DoS**: This DoS attack floods the target system with m_sp_na_1 IEC 60870-5- 104 packets.

The IEC 60870-5-104 Network Flow-Based Anomaly Detection Model adopts the Isolation Forest method [42], detecting anomalous IEC 60870-5-104 network flows. Finally, the last model focuses on the IEC 60870-5-104 packets' payload information, identifying IEC 60870-5-104 anomalous packets. To this end, it applies the Local Outlier Factor (LOF) method [47, 48]. For the training process, a suitable IEC 60870-5-104 intrusion detection dataset was constructed, utilising an emulated substation environment. For this purpose, the directions of A. Gharib et al. [43] were used. The evaluation results related to the IEC 60870-5-104 detection models are presented in section 4.

3.3.2.4. IEC 61850 (MMS) Anomaly Detection Model. The BDAC Analysis Engine includes a single model related to the IEC 61850 (MMS) protocol. The proposed model is named IEC 61850 (MMS) Network Flow-Based Anomaly Detection Model and relies on outlier/novelty detection and network flow statistics defined by the TCP port 102. In particular, it utilises the Minimum Covariance Determinant (MCD) method [45, 49]. Since there are no sufficient intrusion/anomaly detection datasets related to IEC 61850 (MMS), an IEC 61850 (MMS) anomaly detection dataset was constructed, by combining normal IEC 61850 (MMS) network flows from an emulated substation environment and abnormal IEC 61850 (MMS) network flows generated statistically. The evaluation analysis of the specific model is detailed in section 4.

3.3.2.5. BACnet Intrusion/Anomaly Detection Models. The BDAC Analysis Engine includes two detection models related to BACnet. The first one called BACnet Network Flow-based Intrusion Detection Model utilises the Random Forest method [50], thus detecting three BACnet cyberattacks: (a) fuzzing, (b) flooding and (c) tampering. The second model focuses on the BACnet packets' payload and is named BACnet Packet-Based Anomaly Detection Model. It uses a custom text Convolutional Neural Network (CNN) [51, 52], which detects abnormal BACnet packets. This method is named Payload Text CNN Classifier. Due to the lack of publicly available intrusion/anomaly detection datasets for BACnet, an appropriate dataset was implemented utilising the equipment of a real smart home environment. The evaluation analysis of the aforementioned models and more details about the Payload Text CNN Classifier are included in section 4.

3.3.2.6. MQTT Intrusion/Anomaly Detection Models. Two detection models are integrated into the BDAC Analysis Engine regarding the MQTT protocol: (a) MQTT Network Flow-Based Intrusion Detection Model and (b) MQTT Packet-Based Intrusion Detection Model. On the one hand, the first model applies the Random Forest method [50] with MQTT network flow statistics and detects three kinds of MQTT-related cyberattacks: (a) unauthorised subscribe, (b) large payload DoS attack and (c) connection flooding attack. On the other hand, the second model uses the SPEAR Payload Text CNN [51, 52] with the payload attributes of the MQTT packets in order to recognise the anomalous MQTT packets. For the training process, an appropriate MQTT intrusion detection dataset was constructed, following the directions of [43]. As in the previous cases, the evaluation results of the aforementioned models are documented in section 4.

3.3.2.7. HTTP Intrusion/Anomaly Detection Models. The BDAC Analysis Engine integrates two detection models associated with the HTTP protocol: (a) HTTP Network Flow-Based Intrusion Detection Model, (b) HTTP Network Flow-Based Anomaly Detection Model. The first model adopts a Decision Tree Classifier [38] capable of discriminating the following HTTP-related cyberattacks.

- **DoS**: This DoS attack floods the target system with HTTP packets.
- **SQL-Injection**: This attack aims to exploit vulnerabilities of web applications in order to access unauthorised information.
- **Bruteforce-Web**: This attack attempts to access a password-protected web application by using multiple passwords combinations.
- **XSS**: Cross-Site Scripting (XSS) is a type of injection attack, where malicious scripts are injected into web applications.

The HTTP Network Flow-Based Anomaly Detection Model relies on LOF [47, 48]. Both models mentioned above take as input HTTP network flow statistics specified by the 80 TCP port. For the training process, a combined dataset was utilised, including normal HTTP network flows originating from an emulated substation environment and malicious HTTP network flow statistics of the CSE-CIC-IDS2018 dataset [34]. Section 4 details the evaluation results for both HTTP detection models.

3.3.2.8. SSH Intrusion/Anomaly Detection Models. Two SSH-related detection models are involved in the BDAC Analysis Engine. The first one is named SSH Network Flow-Based Intrusion Detection Model and uses Adaboost [53, 54] to recognise SSH bruteforce attacks. The second model, called SSH Network Flow-Based Anomaly Detection Model applies the MCD method [45, 49] to identify anomalous SSH network flows. Both models take as input SSH network flow statistics. The training process relies on a combined dataset, which includes normal SSH network flows from an emulated substation environment and malicious SSH network flows of the CSE-CIC-IDS2018 dataset [34]. Section 4 details the relevant evaluation results.

3.3.2.9. Operational Data Based Anomaly Detection Models. The BDAC Analysis Engine includes four detection models that analyse operational data (i.e., time series electricity measurements), detecting anomalies related to four SG use cases: (a) hydropower plant, (b) substation, (c) power plant and (d) smart home. In particular, the first model related to the hydropower plant environment adopts a GAN [52], which was presented in our previous work in [29]. Next, the second model (i.e., related to the substation environment) applies LOF. The remaining models related to the anomalies of the power plant and the smart home use also the GAN presented in [29]. For the training process, real data was used for each SG use case. As in the previous cases, the evaluation of the particular models is detailed in section 4.

3.3.2.10. Honeypots-Based Detection Model. The Honeypot-Based Detection Model relies on SG honeypots coming from our previous works in [55] and [36, 37]. In particular, the honeypots' logs are collected by the Honeypot Manager that forwards them to the Honeypots-Based Detection Model. The latter undertakes to normalise and transform them into security events based on the format of Table A.8. The Honeypot Manager is analysed in our previous work in [14].

3.3.3. Training Module

The Training Module is responsible for providing the BDAC Analysis Engine with the various ML/DL based intrusion/anomaly detection models. In particular, the main goal behind this module is to train the initial intrusion/anomaly detection models of the BDAC Analysis Engine and re-train them periodically with more and updated data. The previous intrusion/anomaly detection models of the BDAC Analysis Engine are replaced whether the performance of the new ones is better in terms of the accuracy and the F1 score metrics.

3.3.4. Security Event Extraction Module

The Security Event Extraction Module undertakes to generate normalised security events based on the outcome of the BDAC Analysis Engine intrusion/anomaly detection models. The format of the SPEAR security events is given in Table A.8. The Security Event Extraction Module utilises the information given by the Data Receiving Module concerning (a) the network flow statistics, (b) packet payload information of the SG application-layer protocols, (c) operational data and (d) honeypots logs to fill in the necessary fields of the SPEAR security event format. Moreover, it communicates with DAPS in order to receive more information for the assets related to a security event, such as the asset ID, the asset name and the network ID. Finally, it pushes the BDAC security events to Message Bus. It is noteworthy that based on the security event information, this module can also indicate and form automatic firewall rules that are introduced in the Userdata fields of the security event format (Table A.8). These firewall rules rely on the syntax of the Linux firewall, i.e., iptables [56].

3.4. VIDS: Visual-based Intrusion Detection System

VIDS has been designed to receive, store, present, manipulate and visualise data (security events, network packets, operational data (i.e., time-series electricity measurements) and network assets data) from the other SPEAR SIEM components on a simple and easy-to-use visual environment. Moreover, VIDS correlates the Modbus-related security events of BDAC, thus composing Modbus security alerts based on correlation rules. First, VIDS communicates with the Message Bus, thus consuming and visualising the security events generated only by BDAC and the VIDS itself. The security events of AlienVault OSSIM are correlated and illustrated by AlienVault OSSIM itself. This communication between VIDS and Message Bus relies on Apache Kafka. Moreover, VIDS communicates with DAPS of SPEAR SIEM Basis in order to receive the appropriate data for the visual-based anomaly detection mechanisms. As in the case of BDAC, VIDS receives from DAPS the payload of the SG application-layer protocols, network flow statistics and operational data (i.e., time series electricity measurements). Both Apache Kafka (Streaming Bus) and the Elasticsearch API (Storage Infrastructure) are utilised for the communication between VIDS and the SPEAR SIEM Basis. The role of VIDS is complementary to that of BDAC and AlienVault OSSIM, allowing the system operator or the security administrator to observe potential anomalies through appropriate visualisations. Finally, VIDS communicates with GTM to configure it and visualise its reputation values of each asset (i.e., hardware or virtual devices). This communication is based on a REST API.

By focusing on the visual-based detection mechanisms with operational data (i.e., time series electricity measurements), several ML and DL-based dimensionality reduction methods are adopted to detect anomalies. All of them are available in the VIDS dashboard, thereby giving the user the capability to show different visualisations. It is inherently arduous to visualise the incoming network and operational data in a manner straightforward to understand by humans since in most cases, they comprise a large number of features. The role of dimensionality reduction in this context is to reduce these features in a lower-dimensional space and represent all of them with a single 2D or 3D point in space, which is easy to understand by the system operator. Towards this goal, each ML/DL dimensionality reduction method produces a latent space in the form of a manifold in two or three dimensions. The produced output includes a colour indication at each point, which is automatically adjusted based on the distance from the statistical centre of the expected data. This distance value corresponds to the measured distance from the centroid of normal values in the reduced dimensionality space and indicates how close to normal the observed data is. The methods also produce a covariance matrix, showing the correlation between the recorded features over time, indicating how each parameter influences the rest. The outputs of each algorithm are saved into a PostgreSQL database of VIDS and are used to plot the visualisation diagrams (Fig. 8, Fig. 9, Fig. 10 and Fig. 11).

Fig. 8 presents a line-chart displaying the anomaly score of the operational data (i.e., time series electricity measurements) over time. The red horizontal line represents the threshold of normal values, calculated as the statistical centre of the normal data. The black line represents the distance from this threshold, indicating how close to normal the incoming data is at each time instant. There are two such diagrams, one for the live operational data and one for the historical operational data stored in the VIDS database. In the latter, the user can select a time window (i.e., 3 hours) and scroll through the diagram, observing the anomaly score over this time window.

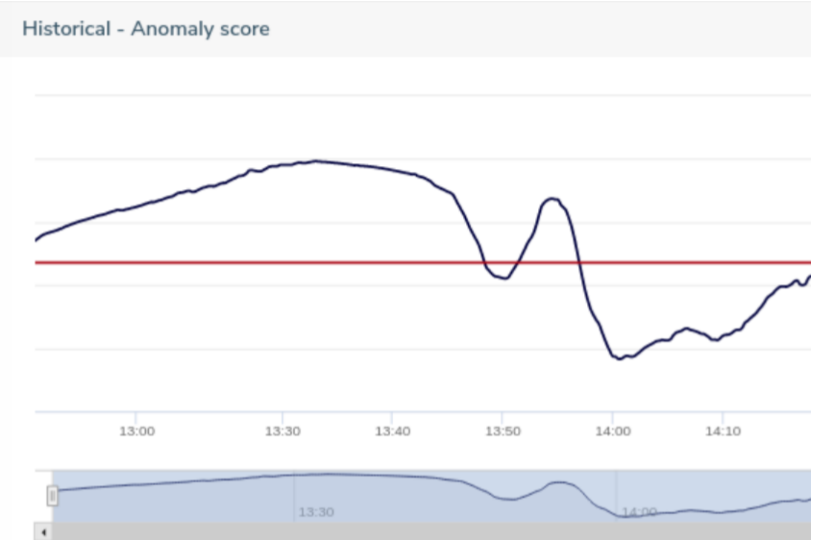


Figure 8: Anomaly score line chart over time. The values below the red line correspond to normal data and the ones above indicate potential anomalies.

Fig. 9 and Fig. 10 depict the reduced dimensionality space of the operational data. The user can choose between representations in either two or three dimensions, with the live and historical data. At each time instant, the live scatter plot displays the network's current status, after executing the ML and DL-based dimensionality reduction methods, using the most recent operational data received from the Storage Infrastructure of the SPEAR SIEM Basis. In the case of the historical data, the scatter plot represents the status of the grid throughout the whole selected date. The visual patterns formed in these diagrams allow the operator to observe the network's status and determine anomalies by looking at the projected points' position and tint. The potential anomalies are showcased by grouped points having a red tint. By rendering these charts, VIDS offers an overview of the network status with respect to anomalies in the operational data and provides a comprehensive visualisation through several methods. The security administrator can deduce whether an anomaly occurs at any given time instant by observing the respective patterns.

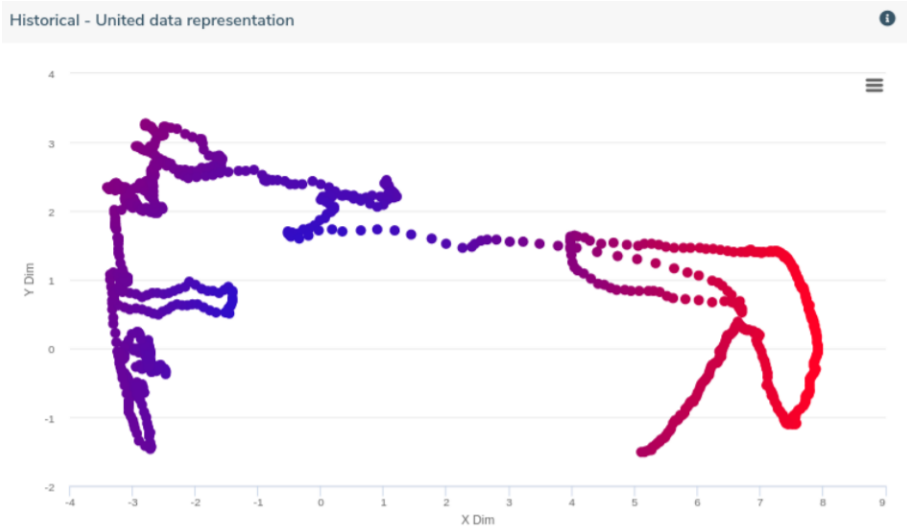


Figure 9: Scatter plot of the 2D data representation of the recorded features. In this case, points having a blue tint and located to the left correspond to normal data, while red points located to the right side indicate potential anomalies. X Dim and Y Dim denote the dimensions after the dimensionality reduction process.

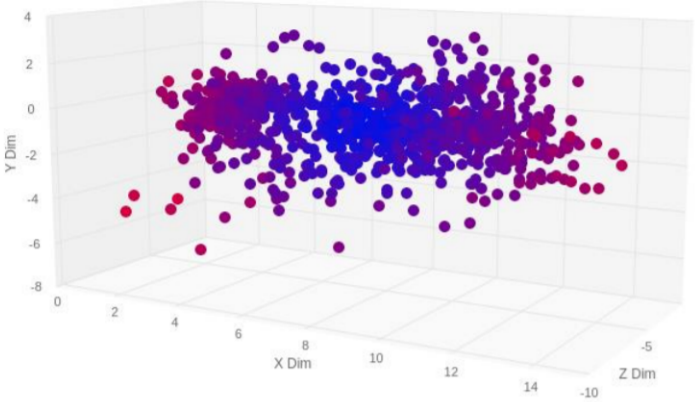


Figure 10: Scatter plot of the 3D data representation of the recorded features. In this case, points having a blue tint, located towards the middle correspond to the normal data, while points with a red tint indicate potential anomalies. X Dim, Y Dim and Z Dim denote the dimensions after the dimensionality reduction process.

Fig. 11 illustrates the correlation among the recorded features of the operational data. A higher line width indicates a more substantial influence between the corresponding features. The user can hover at each line and observe the actual value of the connection. Values close to 0.05 indicate no correlation, while values close to 1 recommend strong relation. The live dependency diagram shows the status corresponding to the most recent operational data at each time instant. Finally, the historical diagram displays the average value throughout the selected date for each connection.

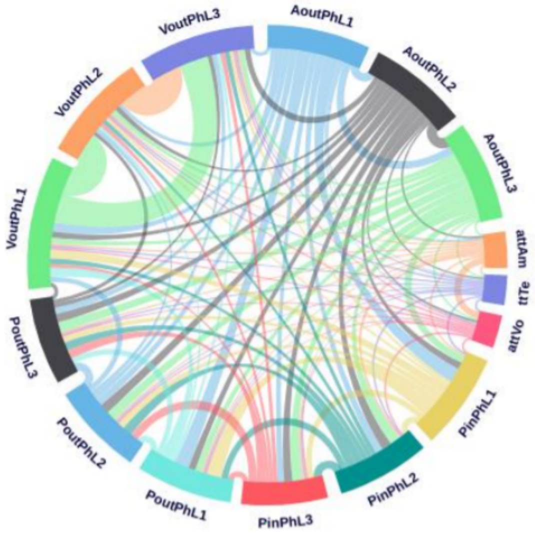
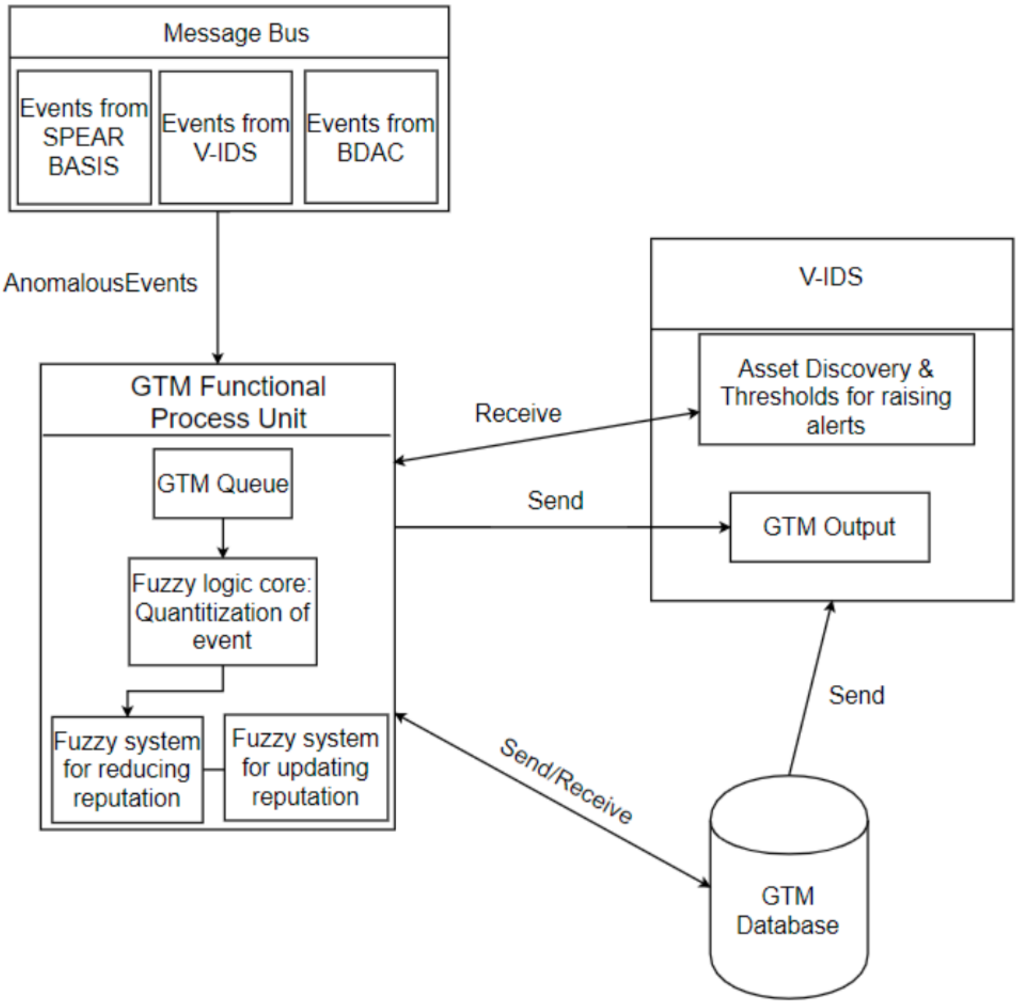


Figure 11: Dependency wheel diagram showing the recorded features correlation. Higher line width indicates a stronger relation between the features.

The VIDS correlation capability relies on correlation rules that focus on the security events generated by the Modbus Network Flow-Based Intrusion Detection Model. However, similar rules can be identified for the other industrial protocols. This kind of correlation aims to identify relationships among the Modbus security events, composing alerts reflecting multi-step attack scenarios related to Modbus. The correlation rules are constructed by combining the information of the security events (Table A.8) as well as additional fields, such as time information (e.g., a sequence of events appearing in a specific period time) or the number of continuous security events. Event Processing Language (EPL) statements are utilised for the syntax of these correlation rules. Table B.9 in Appendix B summarises these rules.

3.5. GTM: Grid Trusted Module

The goal of GTM is to correlate the various security events and calculate a reputation value for each SG asset (hardware or virtual). This kind of correlation intends to reflect how trustworthy, safe and secure each asset is. To this end, GTM communicates with the Message Bus to receive the various security events produced by AlienVault OSSIM, BDAC and VIDS. Fig. 12 shows the architecture of GTM. In particular, since GTM is a backend component, VIDS is utilised for its configuration, defining a specific threshold value for each asset. If an asset's reputation value exceeds the particular threshold, then a GTM alert is generated for the specific asset. This communication between VIDS and GTM is implemented via a REST API. Then, all security events are received from the Message Bus, and the GTM Functional Process Unit undertakes to calculate a reputation value for each asset. These reputation values are sent to the VIDS, which undertakes to visualise them. Finally, the reputation values of GTM are stored into the GTM database as historical data.



Message Bus
Events from SPEAR BASIS
Events from V-IDS
Events from BDAC
AnomalousEvents
GTM Functional Process Unit
GTM Queue
Fuzzy logic core: Quantitization of event
Fuzzy system for reducing reputation
Fuzzy system for updating reputation
Receive
Send
Send/Receive
V-IDS
Asset Discovery & Thresholds for raising alerts
GTM Output
Send
GTM Database

Figure 12: GTM Architecture.

The operation core of GTM is the GTM Functional Process Unit, which consists of four elements: (a) the GTM queue, (b) the Fuzzy Logic Core, (c) the Fuzzy Reputation Reduction System and (d) the Fuzzy Reputation Recovery System. First, GTM receives continually security events stored into the GTM queue, which applies a First In First Out (FIFO) model. Next, the Logic Core undertakes to quantify the severity of each security event based on fuzzy logic rules, considering the asset value, the subcategory, the event risk, the priority and the reliability of the security events based on Table A.8. The Fuzzy

Logic Core utilises the fuzzy theory to map the value of each aforementioned variable into a quantified value without strict rules. Table 1 shows indicative fuzzy logic rules used by the Fuzzy Logic Core. These rules are derived by forming the fuzzy universe. The fuzzy universe is unique and mandatory for each variable used to calculate the quantified value of the security event.

Table 1: Indicative Rules of the Fuzzy Logic Core.

No	**Input**	**Output**
Rule #1	Asset Value: high, Priority: high, Risk: high, Reliability: high, Subcategory: modbus/function/readCoils	Quantified Value: Low
Rule #2	Asset Value: low, Priority: low, Risk: low, Reliability: medium, Subcategory: SQL Injection	Quantified Value: High
Rule #3	Asset Value: high, Priority: high, Risk: high, Reliability: medium, Subcategory: HTTP DoS	Quantified Value: Low
...	...	...
Rule #20	Asset Value: high, Priority: medium, Risk: high, Reliability: medium, Subcategory: DNP3 Reconnaissance	Quantified Value: Low
Rule #21	Asset Value: high, Priority: medium, Risk: medium, Reliability: high, Subcategory: Masquerading	Quantified Value: Low
Rule #22	Asset Value: high, Priority: medium, Risk: low, Reliability: high, Subcategory: Port Scanning	Quantified Value: Low

The purpose of the Fuzzy Reputation Reduction System is to produce the reputation value of any asset related to the corresponding security event. The reputation value of each asset is computed, taking into account the time difference between the previous reputation value and the current security event as well as the outcome of the Fuzzy Logic Core. Table 2 includes indicative fuzzy logic rules used by the Fuzzy Reputation Reduction System in order to calculate the reputation value of each asset.

Table 2: Indicative Rules of the Fuzzy Reputation Reduction System.

No	**Input**	**Output**
Rule #1	Time: Low, Quantified Value: Low	Reputation Value: Low
Rule #2	Time: Low, Quantified Value: Low	Reputation Value: Medium
Rule #3	Time: Low, Quantified Value: High	Reputation Value: Medium
Rule #4	Time: Medium, Quantified Value: Low	Reputation Value: Low
Rule #5	Time: Medium, Quantified Value: Medium	Reputation Value: Medium
Rule #6	Time: Medium, Quantified Value: High	Reputation Value: High
Rule #7	Time: High, Quantified Value: Low	Reputation Value: Low
Rule #8	Time: Medium, Quantified Value: Medium	Reputation Value: High
Rule #9	Time: Medium, Quantified Value: High	Reputation Value: High

Finally, the Fuzzy Reputation Recovery System undertakes to increase the reputation value based on the time difference between the last reduction of an asset's reputation value and the current time. A threshold in the VIDS determines the frequency, which is utilised to check a possible increment of the reputation value. The functionality of the Fuzzy Reputation Recovery System is also based on fuzzy rules. Table 3 shows a sample of them.

Table 3: Indicative Rules of the Fuzzy Reputation Recovery System.

No	**Input**	**Output**
Rule #1	Time: Low, Reputation Value: Low	Reputation Value: Medium
Rule #2	Time: High, Reputation Value: Low	Reputation Value: Medium
Rule #3	Time: Low, Reputation Value: Medium	Reputation Value: Medium
Rule #4	Time: High, Quantified Value: Medium	Reputation Value: High
Rule #5	Time: Low, Quantified Value: High	Reputation Value: High
Rule #6	Time: High, Quantified Value: High	Reputation Value: High

3.6. Message Bus

Message Bus plays the role of a gateway providing a communication system among all SPEAR SIEM components that generate and handle security events. It applies a publish-subscribe pattern based on Apache Kafka. In particular, BDAC and VIDS (via the system operator or the security administrator) produce security events to the appropriate Apache Kafka topic of the Message Bus. In contrast, VIDS and GTM consume them in order to visualise them and compute the assets' reputation value, respectively.

4. Evaluation Analysis

This section focuses on evaluating the detection performance of SPEAR SIEM. First, subsection 4.1 describes the evaluation environment. Next, subsection 4.2 and subsection 4.3 present the datasets and the comparative methods used in the evaluation analysis. Finally, subsection 4.4.1 and subsection 4.4.2 presents the evaluation results of BDAC and VIDS, respectively.

4.1. Evaluation Environment

The detection mechanisms of BDAC and VIDS were implemented and evaluated, utilising real data originating from four SG use cases, namely (a) hydropower plant, (b) substation, (c) power plant and (e) smart home. The first three cases (i.e., hydropower plant, substation and power plant) use logic controllers, such as Programmable Logic Controllers (PLCs) and Remote Terminal (RTUs) that monitor and control the operation of the entire infrastructure and mainly that of industrial devices, such as turbines, transformers and generators. These controllers communicate with a centralised server called Master Terminal Unit (MTU) managed by the system operator through a Human Machine Interface (HMI). In particular, through HMI, the system operator can monitor and handle the operation of PLCs and RTUs, sending the appropriate commands via the corresponding SG application-layer protocols (e.g., Modbus, DNP3 and IEC 61850). Finally, the smart home environment involves smart meters that measure energy consumption and relevant statistics. This information is also stored in an MTU, using the corresponding SG application-layer protocols. The SPEAR Sensors were deployed in each SG infrastructure, using a Switched Port Analyser (SPAN). Therefore, the overall network traffic is directed to the SPEAR sensors. In addition, the operational data of each SG infrastructure is stored in MTU, which transmits them to DAPS.

4.2. Datasets

For each SG application-layer protocol mentioned above in subsection 3.3, appropriate datasets were formed and utilised to train and test the ML and DL models of the BDAC Analysis Engine. These datasets were composed either by creating them from scratch with the emulation of the respective cyberattacks/anomalies or combining existing intrusion datasets with the normal records coming from the aforementioned SG use cases (i.e., hydropower plant, substation, power plant and smart home). New datasets were formed for the Modbus/TCP, IEC 60870-5-104, IEC 61850, BACnet and MQTT. In addition, the CSE-CIC-IDS2018 dataset [34] was used for the HTTP and SSH. For the anomaly detection models of the BDAC Analysis Engine and VIDS using operational data (i.e., time-series electricity measurements), suitable datasets were produced from scratch based on the indications of security and safety experts from each SG infrastructure (i.e., hydropower plant, substation, power plant and smart home). Due to the sensitive nature of these datasets, they cannot be published in the current work.

4.3. Comparative Methods

This subsection is devoted to the comparative methods used to evaluate BDAC and VIDS. In particular, subsection 4.3.1 is focused on the ML and DL comparative methods related to the BDAC Analysis Engine, while subsection 4.3.2 describes the ML and DL dimensionality reduction methods of VIDS.

4.3.1. BDAC Comparative Methods

Multiple ML and DL methods were investigated and evaluated for each detection model of the BDAC Analysis Engine. In particular, regarding the detection models adopting a multiclass classification, the following ML methods were used: Logistic Regression [57], Linear Discriminant Analysis (LDA) [58], Decision Tree Classifier [38], Naive Bayes [59], SVM Linear [60], SVM RBF [60], Random Forest [61], Adaboost [53], Multi-Layer Perceptron (MLP) [62], Quadratic Discriminant Analysis [63], K Nearest Neighbour (KNN) [64]. Moreover, three custom DNNs were also used in our evaluation analysis. The first two called Dense DNN Relu and Dense DNN Tanh are originating from our previous work in [29]. The remaining one called SDAE was implemented during this work.

The SPEAR SDAE is a DNN consisting of consequent encoding layers of individual Denoising Autoencoders (DAEs), which can be considered a type of MLP. In the beginning, the original input data is used to generate higher representation. Afterwards, the output of the first trained DAE's hidden layer is used as the next autoencoder's input to extract higher representations. The training process of the SPEAR SDAE consists of two phases.

The first phase is the unsupervised layer-wise pre-training and the second phase is the supervised fine-tuning phase. During the first phase, each layer is trained separately. For the first phase, the labels are not required since the goal is to extract the feature representations from the input data. Then, after the training of all layers, the fine-tuning phase starts, which is a backpropagation phase, using supervised training algorithms. This greedy layer-wise procedure has been shown to yield significantly better local minima than random initialisation of deep networks, achieving better generalisation on a number of tasks [65]. The SPEAR SDAE was tested to detect possible cyberattacks against MQTT and BACnet based on the corresponding network flow statistics. In particular, during the training phase, it receives as input 83 network flow-related statistic features and the label for each MQTT or BACnet network flow. These features pass through two or three encoder layers depending on the specific architecture of each protocol. Then, the representative features are extracted, passing through a final softmax classification layer with an equal number of nodes as the number of classes.

Regarding the models using outlier/novelty detection mechanisms to identify whether there is an anomaly or not, the following ML methods were evaluated: ABOD [44], Isolation Forest [42], PCA [66], MCD [45, 49] and LOF [47, 48]. Furthermore, two DNNs were also adopted and evaluated. The first one is called DIDEROT Autoencoder and originates from our previous work in [41]. The second one was developed during this work. It relies on text CNN, which is a slight variant of a typical CNN. The difference between them is that in the conventional CNNs, the sizes of filters in a single layer are usually the same. In contrast, in text-CNNs, the filters have a fixed width equal to the embedding size of the input sentences but different heights. The sentences are formed by parsing the SG application-layer payload of each packet and decomposing it into tokens. Each token is usually either a payload field or its value. The Payload Text CNN Classifier consists of 3 layers. The first layer is an embedding layer, which transforms the words of each payload/sentence in word embeddings. Word embeddings are dense vectors representing the projection of the word into a continuous vector space. During the convolution process, a filter w of size hxd is applied in a window of h words of the sentence to extract a new feature. In particular, h represents the height and d denotes the width of the token embeddings that form a sentence. This filter is applied to each possible window generating a feature map. After this procedure, a global max-pooling layer follows, extracting the most important feature of each feature map. Filters of 3 different window sizes (4, 6, 8) are used in the different channels to extract more features by processing 4-grams, 6-grams and 8-grams. Consequently, the features from the global max-pooling layers are concatenated and passed through a dense feature layer and a final output layer.

4.3.2. VIDS Comparative Methods

Concerning the visual-based detection mechanisms of VIDS, four ML dimensionality reduction methods are investigated and compared with each other, including PCA [66], Singular Value Decomposition (SVD) [67], Independent Component Analysis (ICA) [68] and Semi-Random Projection (SRP) [69]. In addition, four DNNs, namely (a) DeepDense Autoencoder, (b) Feed Forward (FF) Autoencoder, (c) Long Short-Term Memory (LSTM)

Autoencoder and (d) Classic Dense Sequence Autoencoder were constructed during this work. Adam is utilised as the optimisation method, the Mean Squared Error (MSE) is used for the loss, while the Rectified Linear Unit (ReLU) and sigmoid are used for the activation functions. Tables-4-7 summarise these DNNs. For each ML/DL dimensionality reduction method, a time window of 30 instances (the most recent operational data of the corresponding evaluation environment) is used as input.

Table 4: Overview of the DeepDense Autoencoder.

Layer (type)	**Output Shape**	**Param #**
input_1(Input Layer)	(None, inputDim)	0
dense_1 (Dense)	(None,128)	57728
dense_2 (Dense)	(None,64)	8256
dense_3 (Dense)	(None,32)	2080
dense_4 (Dense)	(None,lowDim)	99
dense_5 (Dense)	(None,32)	128
dense_6 (Dense)	(None,64)	2112
dense_7 (Dense)	(None,128)	8320
dense_8 (Dense)	(None,inputDim)	58050

Table 5: Overview of the FF Autoencoder.

Layer (type)	**Output Shape**	**Param #**
input_1(Input Layer)	(None, inputDim)	0
dense_1 (Dense)	(None,lowDim)	1353
dense_2 (Dense)	(None,inputDim)	1800

Table 6: Overview of the LSTM Autoencoder.

Layer (type)	**Output Shape**	**Param #**
input_1(Input Layer)	(None, inputDim,1)	0
lstm_1 (LSTM)	(None,lowDim)	60
repeat_vector_1 (ReepatVector)	(None,inputDim,3)	0
lstm_2 (LSTM)	(None,inputDim,1)	20

Table 7: Overview of the Classic Dense Sequence Autoencoder.

Layer (type)	**Output Shape**	**Param #**
dense_1 (Dense)	(None,128)	57728
dense_2 (Dense)	(None,64)	8256
dense_3 (Dense)	(None,lowDim)	195

dense_4 (Dense)	(None,64)	256
dense_5 (Dense)	(None,128)	8320
dense_6 (Dense)	(None,inputDim)	58050

4.4. Evaluation Results

Before proceeding to the analysis of the BDAC and VIDS detection performance, we need to introduce first the necessary background terms. True Positives (TP) define the number of the correct classifications that detected the cyberattacks and anomalies as malicious/anomalous behaviours. Accordingly, True Negatives (TN) denote the number of the correct classifications that recognised the normal behaviour activities as normal. On the other side, False Negatives (FN) denote the number of the wrong classifications that identified malicious activities as normal. Finally, False Positives (FP) define the number of the incorrect classifications that detected the normal activities as malicious or anomalous. Therefore, the following metrics are defined (Equations 1-4).

Accuracy (ACC) (equation (1)) indicates the ratio between the correct classifications and the total number of data samples. ACC can be utilised as an unbiased evaluation metric when the training dataset comprises an equivalent quantity of data samples for all classes. For example, if the training dataset contains 90% data samples characterised as normal and 10% data samples as anomalous, then the ACC can reach 90% by classifying every case as normal.

$$Accuracy(ACC) = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

The False Positive Rate (FPR) (equation (2)) denotes the proportion of normal behaviours recognised as malicious/anomalous. FPR is calculated by dividing FP with the sum of FP and TN.

$$FPR = \frac{FP}{FP+TN} \tag{2}$$

The True Positive Rate (TPR) (equation (3)) determines what proportion of actual malicious/anomalous activities was identified as malicious/anomalous. TPR is focused essentially on FN and is calculated by dividing TP with the sum of FN and TP.

$$TPR = \frac{TP}{TP+FN} \tag{3}$$

Finally, the F1 score (equation (4)) expresses the golden ratio between the TPR and Precision, taking into account both FN and FP. Precision is another evaluation metric, which computes the proportion of those data samples classified as malicious/anomalous.

$$F1 = \frac{2 \times Precision \times TPR}{Precision + TPR} \text{ where } Precision = \frac{TP}{TP+FP} \tag{4}$$

4.4.1. BDAC Evaluation Results

This subsection summarises the evaluation results of the various intrusion and anomaly detection models that compose the BDAC Analysis Engine. The comprehensive ML/DL comparative analysis of the BDAC evaluation results is provided by Tables C.10-C.28 in Appendix C. It is noteworthy that all ML and DL methods were fine-tuned after several experiments. Fig. 13 summarises the detection performance of the BDAC network flow-based intrusion detection models. The Modbus/TCP Network Flow-Based Intrusion Detection Model adopts a decision tree, where $ACC = 0.964$, $TPR = 0.749$, $FPR = 0.019$ and $F1 = 0.749$. Decision trees are efficient ML methods used for both classification and regression problems. Their architecture consists of internal nodes and leaves. The internal nodes and their edges separate the whole space into smaller sub-spaces based on the training features. In contrast, the leaves symbolise the various classes. Consequently, different paths are formed that can be translated into logical rules leading to particular classes. In this paper, we use the Classification and Regression Tree (CART) method with the Information Gain (IG) criterion. More details about the decision trees are given in [38]. The IEC 60870-5-104 Network Flow-Based Intrusion Detection Model adopts also a CART decision tree whose ACC, TPR, FPR and the F1 score reach 0.953, 0.815, 0.026 and 0.815. On the other side, the BACnet and the MQTT Network Flow-Based Intrusion Detection Models apply the SPEAR SDAE method, which is analysed previously in subsection 4.3. In the first case, the ACC, TPR, FPR and the F1 score reach 0.909, 0.991, 0.090 and 0.979, respectively. On the contrary, the efficiency of the MQTT Network Flow-Based Intrusion Detection Model is reflected by the following metrics $ACC = 0.992$, $TPR = 0.984$, $FPR = 0.005$ and $F1 = 0.984$. Finally, both HTTP Network Flow-Based Intrusion Detection Model and SSH Network Flow-Based Intrusion Detection Model use a CART decision tree classifier where their performance is defined by the following metrics, respectively: $ACC = 0.964$, 0.911, 0.026 and 0.911 and $ACC = 0.960$, $TPR = 0.958$, $FPR = 0.038$ and $F1 = 0.955$.

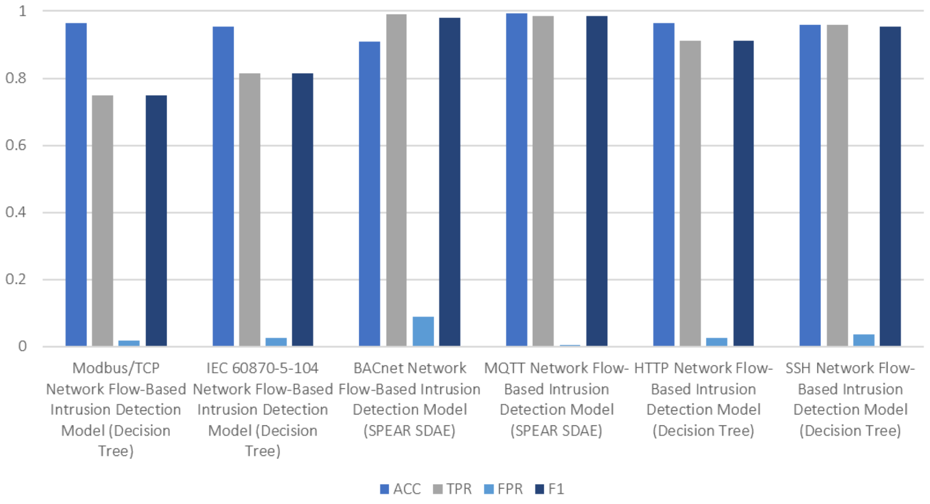


Figure 13: BDAC Network Flow-based Intrusion Detection Models Evaluation Results

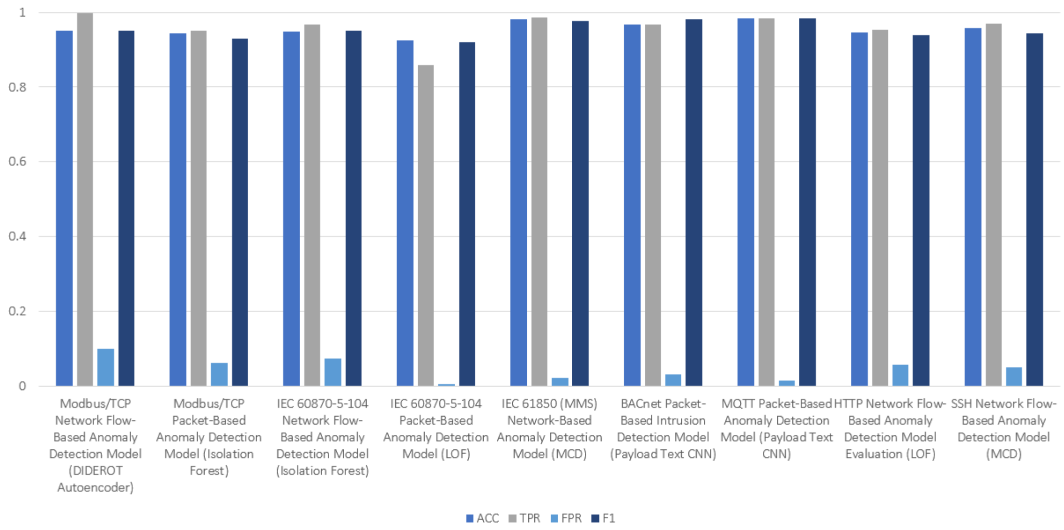


Figure 14: BDAC Network Flow-based & Packet-Based Anomaly Detection Models Evaluation Results

Fig. 14 illustrates the detection performance of those BDAC Analysis Engine models detecting anomalies based on outlier or novelty detection techniques. First, the Modbus Network Flow-Based Anomaly Detection Model utilises the DIDEROT autoencoder, where its detection performance is defined by $ACC = 0.950$, $TPR = 0.999$, $FPR = 0.099$ and $F1 = 0.952$. The DIDEROT autoencoder is described by our previous work in [42]. In particular, it is a DNN composed of six fully connected layers that represent the encoder and decoder, evenly. Both the encoder and decoder map the input data x to an output y. Based on the dimensionality reduction property, the training process intends to reduce the reconstruction error $L(x, y)$, which typically is the Euclidean distance in space X. The anomaly detection process is conducted by calculating and comparing the reconstruction error $L(x, y)$ with a threshold T, which is defined heuristically. In contrast, the Modbus Packet-Based Anomaly Detection Model applies the isolation forest method [42], where ACC, TPR, FPR and the $F1$ score are calculated at 0.943, 0.952, 0.062 and 0.930. The isolation forest method detects outliers or differently anomalies by intentionally ”overfitting” a function memorising each data point. Since the data space is relatively empty around outliers/anomalies, the function requires fewer memorisation steps. To this end, full decision trees are used, calculating the path length between the root and each leaf (data point). The final measure for each data point is the average path length, which is relatively short. Similarly, the IEC 60870-5-104 Network Flow-Based Anomaly Detection Model adopts the isolation forest, where $ACC = 0.948$, $TPR = 0.967$, $FPR = 0.074$, $ACC = 0.952$. However, on the other side, the IEC 60870-5-104 Packet-Based Anomaly Detection Model utilises the LOF method [47]. The evaluation metrics for this model are $ACC = 0.926$, $TPR = 0.859$, $FPR = 0.005$, $ACC = 0.921$. The LOF functionality relies on the local density. An outlier/anomaly is detected by comparing the local density of the point investigated with the local density of its neighbours. The locality is provided by KNN [64] through which the density is estimated by measuring their distance. When the density of the point investigated is significantly lower than its neighbours' density, it is considered an outlier/anomaly. The IEC 61850 (MMS) Network Flow-Based Anomaly Detection Model applies the MCD method [49] with $ACC = 0.981$, $TPR = 0.986$, $FPR = 0.22$ and $F1 = 0.977$. The MCD is a robust estimator of multivariate scatter and location. Its resiliency to the masking effect, makes it efficient to detect outliers/anomalies. M. Hubert and D. Michiel in [49] provide a detailed description about MCD, using simplified examples. Next, both the BACnet Packet-Based Intrusion Detection Model and the MQTT Packet-Based Anomaly Detection Model adopt the Payload Text CNN described earlier in subsection 4.3. The detection performance of the first is reflected by the following metrics $ACC = 0.967$, $TPR = 0.967$, $FPR = 0.032$ and $F1 = 0.982$. Similarly, the performance of the MQTT Packet-Based Anomaly Detection Model is defined by $ACC = 0.985$, $TPR = 0.985$, $FPR = 0.014$ and $F1 = 0.985$. Finally, the HTTP Network Flow-Based Anomaly Detection Model and the SSH Network Flow-Based Anomaly Detection Model use LOF and MCD, respectively. The detection performance of the first is reflected by $ACC = 0.946$, $TPR = 0.954$, $FPR = 0.058$ and $F1 = 0.938$. In contrast, the evaluation metrics of the SSH Network Flow-Based Anomaly Detection Model are $ACC = 0.957$, $TPR = 0.970$, $FPR = 0.050$ and $F1 = 0.944$.

Fig. 15 depicts the detection performance of the BDAC Operational Data-Based Anomaly Detection Models. In particular, the ARIES GAN [29] is applied in the three of the four SG use cases: (a) hydropower plant, (b) power plant and (c) smart home. As mentioned in section 2, the ARIES GAN is discussed in our previous work in [29]. In contrast, in the substation use case, the LOF [47] method is used, where $ACC = 0.873$, $TPR = 0.993$, $FPR = 0.157$ and $F1 = 0.759$. Regarding the ARIES GAN, the evaluation metrics in the hydropower plant use case equal with $ACC = 0.746$, $TPR = 0.978$, $FPR = 0.311$ and $F1 = 0.607$. Similarly, the efficacy of the ARIES GAN in the power plant use case is reflected by $ACC = 0.851$, $TPR = 0.982$, $FPR = 0.188$ and $F1 = 0.755$.

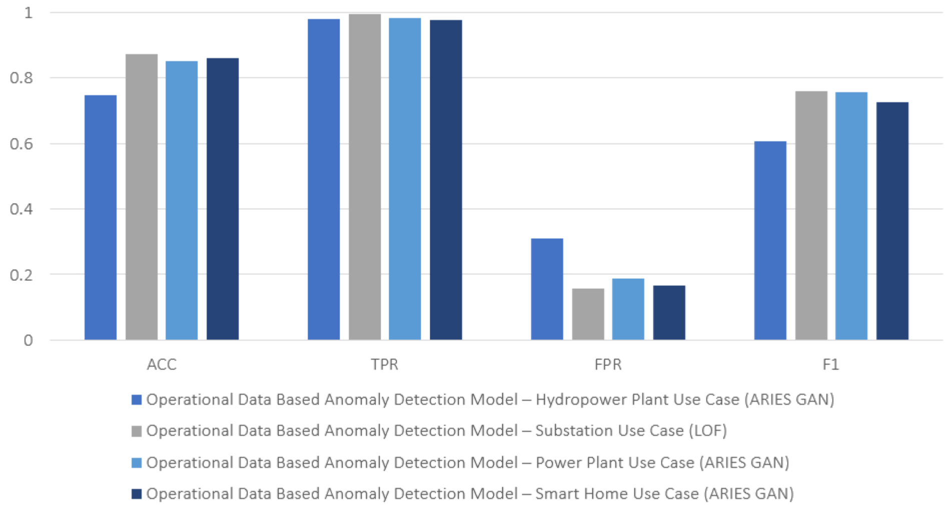


Figure 15: BDAC Operational Data-Based Anomaly Detection Models Evaluation Results

4.4.2. VIDS Evaluation Results

This subsection is devoted to evaluating the detection performance of VIDS. The detailed ML/DL comparative analysis is provided by Tables D.29-D.32 in Appendix D. As illsutrated by Fig. 16, almost in all SG use cases the LSTM-Autoencoder presents the best efficacy in terms of ACC and the F1 score. Only, in the smart home environment, the FF-Autoencoder overcomes the LSTM-Autoencoder. Both LSTM-Autoencoder and FF-Autoencoder are detailed in subsection 4.3.

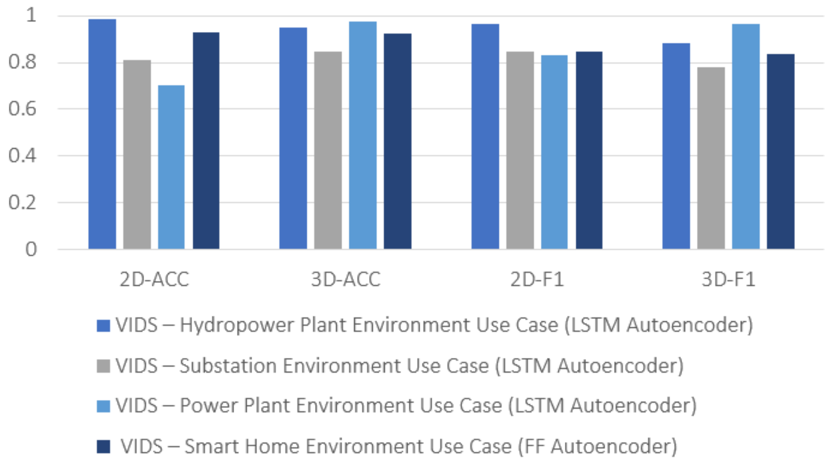


Figure 16: VIDS Evaluation Results

5. Conclusions

Although the modern electrical grid provides several benefits, such as pervasive control and self-healing, it involves crucial cybersecurity risks. In particular, the combination of the insecure SG communication protocols, the IoT security issues and the rapid evolution of cyberattacks and malware can lead to disastrous consequences, such as extensive blackouts and brownouts. The SIEM systems constitute a state-of-the-art cybersecurity technology, which can organise and manage the monitoring, detection and prevention measures.

In this work, we presented the SPEAR SIEM, which focuses on the peculiarities of SG. In particular, SPEAR SIEM is composed of four main components, namely (a) SPEAR SIEM Basis, (b) BDAC, (c) VIDS and (d) GTM. SPEAR SIEM Basis undertakes to monitor the infrastructure, thus providing the necessary data to the other components. Next, BDAC integrates a set of ML/DL-based intrusion and anomaly detection models related to the SG communication protocols and SG operational data (i.e., time-series electricity measurements). Next, VIDS is a parallel detection and correlation mechanism, which relies on visual analytics. Finally, GTM correlates the various security events and computes the reputation value of each SG asset. The evaluation analysis demonstrates the efficiency and applicability of SPEAR SIEM in four SG environments, namely (a) hydropower plant, (b) substation, (c) power plant and (d) smart home.

Our future plans related to this work include the incorporation of more intrusion and anomaly detection models in the BDAC Analysis Engine that will focus on Profinet and EtherCAT. Moreover, appropriate association rules will be investigated in order to correlate security events related to other industrial protocols. To this end the Apriori and Eclat ML methods will be investigated. Finally, appropriate self-healing mechanisms will be examined to be integrated into SPEAR SIEM, taking full advantage of the network automation capabilities offered by the Software-Defined Networking (SDN) technology. In particular, the SDN controller will be able to mitigate the potential malicious flows or re-arrange them, thus ensuring the stability of the SG infrastructure.

6. Acknowledgement

This project has received funding from the European Unions Horizon 2020 Research and Innovation Programme under grant agreement No. 787011 (SPEAR).

[1] H. Farhangi, The path of the smart grid, IEEE power and energy magazine 8 (1) (2009) 18–28.

[2] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, A. Martin, Smart grid metering networks: A survey on security, privacy and open research issues, IEEE Communications Surveys Tutorials 21 (3) (2019) 2886–2927.

[3] S. Kwon, H. Yoo, T. Shon, Ieee 1815.1-based power system security with bidirectional rnn-based network anomalous attack detection for cyber-physical system, IEEE Access 8 (2020) 77572–77586.

[4] S. Tan, D. De, W.-Z. Song, J. Yang, S. K. Das, Survey of security advances in smart grid: A data driven approach, IEEE Communications Surveys & Tutorials 19 (1) (2017) 397–422.

[5] P. Radoglou-Grammatikis, P. Sarigiannidis, I. Giannoulakis, E. Kafetzakis, E. Panaousis, Attacking iec-60870-5-104 scada systems, in: 2019 IEEE World Congress on Services (SERVICES), Vol. 2642-939X, 2019, pp. 41–46.

[6] P. I. Radoglou Grammatikis, P. G. Sarigiannidis, I. D. Moscholios, Securing the internet of things: Challenges, threats and solutions, Internet of Things 5 (2019) 41 – 70. doi:https://doi.org/10.1016/j.iot.2018.11.003.
URL `http://www.sciencedirect.com/science/article/pii/S2542660518301161`

[7] S. S. Hussain, T. S. Ustun, A. Kalam, A review of iec 62351 security mechanisms for iec 61850 message exchanges, IEEE Transactions on Industrial Informatics.

[8] T. S. Ustun, S. S. Hussain, Iec 62351-4 security implementations for iec 61850 mms messages, IEEE Access 8 (2020) 123979–123985.

[9] R. Schlegel, S. Obermeier, J. Schneider, Assessing the security of iec 62351, in: 3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015) 3, 2015, pp. 11–19.

[10] S. Bhatt, P. K. Manadhata, L. Zomlot, The operational role of security information and event management systems, IEEE Security Privacy 12 (5) (2014) 35–41.

[11] P. I. Radoglou-Grammatikis, P. G. Sarigiannidis, Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems, IEEE Access 7 (2019) 46595–46620.

[12] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, V. Venkatakrishnan, Holmes: real-time apt detection through correlation of suspicious information flows, in: 2019 IEEE Symposium on Security and Privacy (SP), IEEE, 2019, pp. 1137–1152.

[13] AlienVault, Alienvault ossim documentation, `https://www.alienvault.com/documentation/` (2020).

[14] P. Radoglou-Grammatikis, P. Sarigiannidis, E. Iturbe, E. Rios, A. Sarigiannidis, O. Nikolis, D. Ioannidis, V. Machamint, M. Tzifas, A. Giannakoulias, M. Angelopoulos, A. Papadopoulos, F. Ramos, Secure and private smart grid: The spear architecture, in: 2020 6th IEEE Conference on Network Softwarization (NetSoft), 2020, pp. 450–456.

[15] S. G. Zarzosa, D2.1 in-depth analysis of siems extensibility, Tech. Rep. 1, DiSIEM Project (2017).

[16] L. Cui, Y. Qu, L. Gao, G. Xie, S. Yu, Detecting false data attacks using machine learning techniques in smart grid: A survey, Journal of Network and Computer Applications (2020) 102808.

[17] S. E. Quincozes, C. Albuquerque, D. Passos, D. Mossé, A survey on intrusion detection and prevention systems in digital substations, Computer Networks 184 (2020) 107679.

[18] M. Z. Gunduz, R. Das, Cyber-security on smart grid: Threats and potential solutions, Computer Networks 169 (2020) 107094.

[19] M. U. Hassan, M. H. Rehmani, J. Chen, Differential privacy techniques for cyber physical systems: A survey, IEEE Communications Surveys Tutorials 22 (1) (2020) 746–789. doi:10.1109/COMST.2019.2944748.

[20] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, J. Lopez, A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services, IEEE Communications Surveys Tutorials 20 (4) (2018) 3453–3495. doi:10.1109/COMST.2018.2855563.

[21] R. Leszczyna, M. R. Wróbel, Evaluation of open source siem for situation awareness platform in the smart grid environment, in: 2015 IEEE World Conference on Factory Communication Systems (WFCS), IEEE, 2015, pp. 1–4.

[22] C. iView, Cyberoam iview centralized logging & reporting for distributed enterprises and mssps, Tech. Rep. 1, Cyberoam iView (2020).

[23] T. Andrejak, Prelude-siem documentation release 4.0, Tech. Rep. 4, CS (2017).

[24] B. Sahay, A. Gupta, Development of software selection criteria for supply chain solutions, Industrial Management & Data Systems.

[25] K. Zhang, S. Luo, Y. Xin, H. Zhu, Y. Chen, Online mining intrusion patterns from ids alerts, Applied Sciences 10 (8) (2020) 2983.

[26] M. Albanese, R. F. Erbacher, S. Jajodia, C. Molinaro, F. Persia, A. Picariello, G. Sperlì, V. Subrahmanian, Recognizing unexplained behavior in network traffic, in: Network Science and Cybersecurity, Springer, 2014, pp. 39–62.

[27] M. Albanese, C. Molinaro, F. Persia, A. Picariello, V. Subrahmanian, Finding" unexplained" activities in video., in: IJCAI, 2011, pp. 1628–1634.

[28] K. Zhang, F. Zhao, S. Luo, Y. Xin, H. Zhu, An intrusion action-based ids alert correlation analysis and prediction framework, IEEE Access 7 (2019) 150540–150551.

[29] P. Radoglou-Grammatikis, P. Sarigiannidis, E. Georgios, P. Emmanouil, Aries: A novel multivariate intrusion detectionsystem for smart grid, Sensors.

[30] G. Efstathopoulos, P. R. Grammatikis, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, K. Stamatakis, M. K. Angelopoulos, S. K. Athanasopoulos, Operational data based intrusion detection system for smart grid, in: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), IEEE, 2019, pp. 1–6.

[31] M. Ali, S. Shiaeles, G. Bendiab, B. Ghita, Malgra: Machine learning and n-gram malware feature extraction and detection system, Electronics 9 (11) (2020) 1777.

[32] M. Ghafouri, M. Au, M. Kassouf, M. Debbabi, C. Assi, J. Yan, Detection and mitigation of cyber attacks on voltage stability monitoring of smart grids, IEEE Transactions on Smart Grid 11 (6) (2020) 5227–5238.

[33] M. Tsoukalos, Using tshark to watch and inspect network traffic, Linux Journal 2015 (254) (2015) 1.

[34] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization., in: ICISSP, 2018, pp. 108–116.

[35] G. F. Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning, Insecure, Sunnyvale, CA, USA, 2009.

[36] I. Siniosoglou, G. Efstathopoulos, D. Pliatsios, I. D. Moscholios, A. Sarigiannidis, G. Sakellari, G. Loukas, P. Sarigiannidis, Neuralpot: An industrial honeypot implementation based on deep neural networks, in: 2020 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2020, pp. 1–7.

[37] P. Diamantoulakis, C. Dalamagkas, P. Radoglou-Grammatikis, P. Sarigiannidis, G. Karagiannidis, Game theoretic honeypot deployment in smart grid, Sensors 20 (15) (2020) 4199.

[38] P. I. Radoglou-Grammatikis, P. G. Sarigiannidis, An anomaly-based intrusion detection system for the smart grid based on cart decision tree, in: 2018 Global Information Infrastructure and Networking Symposium (GIIS), 2018, pp. 1–5.

[39] J. Luswata, P. Zavarsky, B. Swar, D. Zvabva, Analysis of scada security using penetration testing: A case study on modbus tcp protocol, in: 2018 29th Biennial Symposium on Communications (BSC), IEEE, 2018, pp. 1–5.

[40] P. Radoglou-Grammatikis, I. Sinisioglou, T. Liatifis, A. Kourouniadis, K. Rompolos, P. Sarigiannidis, Implementation and detection of modbus cyberattacks, in: 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2020, pp. 1–4.

[41] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, P.-A. Karypidis, A. Sarigiannidis, Diderot: An intrusion detection and prevention system for dnp3-based scada systems, in: Proceedings of the 15th International Conference on Availability, Reliability and Security, ARES '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1–8. doi:10.1145/3407023.3409314.
URL https://doi.org/10.1145/3407023.3409314

[42] S. Hariri, M. C. Kind, R. J. Brunner, Extended isolation forest, arXiv preprint arXiv:1811.02141.

[43] A. Gharib, I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, An evaluation framework for intrusion detection dataset, in: 2016 International Conference on Information Science and Security (ICISS), IEEE, 2016, pp. 1–6.

[44] H.-P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 444–452.

[45] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, arXiv preprint arXiv:1901.01588.

[46] N. R. Rodofile, K. Radke, E. Foo, Framework for scada cyber-attack dataset creation, in: Proceedings of the Australasian Computer Science Week Multiconference, 2017, pp. 1–10.

[47] L. You, Q. Peng, Z. Xiong, D. He, M. Qiu, X. Zhang, Integrating aspect analysis and local outlier factor for intelligent review spam detection, Future Generation Computer Systems 102 (2020) 163–172.

[48] S. Mishra, M. Chawla, A comparative study of local outlier factor algorithms for outliers detection in data streams, in: Emerging Technologies in Data Mining and Information Security, Springer, 2019, pp. 347–356.

[49] M. Hubert, M. Debruyne, Minimum covariance determinant, Wiley interdisciplinary reviews: Computational statistics 2 (1) (2010) 36–43.

[50] G. Apruzzese, M. Andreolini, M. Colajanni, M. Marchetti, Hardening random forest cyber detectors against adversarial attacks, IEEE Transactions on Emerging Topics in Computational Intelligence.

[51] R. U. Khan, X. Zhang, M. Alazab, R. Kumar, An improved convolutional neural network model for intrusion detection in networks, in: 2019 Cybersecurity and Cyberforensics Conference (CCC), IEEE, 2019, pp. 74–77.

[52] D. S. Berman, A. L. Buczak, J. S. Chavis, C. L. Corbett, A survey of deep learning methods for cyber security, Information 10 (4) (2019) 122.

[53] D. Tang, L. Tang, R. Dai, J. Chen, X. Li, J. J. Rodrigues, Mf-adaboost: Ldos attack detection based on multi-features and improved adaboost, Future Generation Computer Systems 106 (2020) 347–359.

[54] A. Subasi, E. Kremic, Comparison of adaboost with multiboosting for phishing website detection, Procedia Computer Science 168 (2020) 272–278.

[55] D. Pliatsios, P. Sarigiannidis, T. Liatifis, K. Rompolos, I. Siniosoglou, A novel and interactive industrial control system honeypot for critical smart grid infrastructure, in: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2019, pp. 1–6.

[56] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Liatifis, T. Apostolakos, S. Oikonomou, An overview of the firewall systems in the smart grid paradigm, in: 2018 Global information infrastructure and networking symposium (GIIS), IEEE, 2018, pp. 1–4.

[57] Y. Wang, A multinomial logistic regression modeling approach for anomaly intrusion detection, Com-

puters & Security 24 (8) (2005) 662–674.

[58] H. Li, L. Zhang, B. Huang, X. Zhou, Cost-sensitive dual-bidirectional linear discriminant analysis, Information Sciences 510 (2020) 283–303.

[59] S. Mukherjee, N. Sharma, Intrusion detection using naive bayes classifier with feature reduction, Procedia Technology 4 (2012) 119–128.

[60] M. Awad, R. Khanna, Support Vector Machines for Classification, Apress, Berkeley, CA, 2015, Ch. 3, pp. 39–66. doi:10.1007/978-1-4302-5990-9˙3.
URL `https://doi.org/10.1007/978-1-4302-5990-9_3`

[61] S. Liu, L. Liu, Y. Fan, L. Zhang, Y. Huang, T. Zhang, J. Cheng, L. Wang, M. Zhang, R. Shi, D. Mao, An integrated scheme for online dynamic security assessment based on partial mutual information and iterated random forest, IEEE Transactions on Smart Grid 11 (4) (2020) 3606–3619.

[62] P. I. Radoglou-Grammatikis, P. G. Sarigiannidis, Flow anomaly based intrusion detection system for android mobile devices, in: 2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2017, pp. 1–4.

[63] B. Ghojogh, M. Crowley, Linear and quadratic discriminant analysis: Tutorial, arXiv preprint arXiv:1906.02590.

[64] H. Yang, S. Liang, J. Ni, H. Li, X. Shen, Secure and efficient knn classification for industrial internet of things, IEEE Internet of Things Journal (2020) 1–1.

[65] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

[66] A. A. Imayakumar, A. Dubey, A. Bose, Anomaly detection for primary distribution system measurements using principal component analysis, in: 2020 IEEE Texas Power and Energy Conference (TPEC), IEEE, 2020, pp. 1–6.

[67] O. Edfors, M. Sandell, J.-J. Van de Beek, S. K. Wilson, P. O. Borjesson, Ofdm channel estimation by singular value decomposition, IEEE Transactions on communications 46 (7) (1998) 931–939.

[68] A. Tharwat, Independent component analysis: An introduction, Applied Computing and Informatics.

[69] R. Zhao, K. Mao, Semi-random projection for dimensionality reduction and extreme learning machine in high-dimensional space, IEEE Computational Intelligence Magazine 10 (3) (2015) 30–41.

Appendix A. SPEAR SIEM Security Event Format

Table A.8 summarises the format of the SPEAR security events.

Table A.8: SPEAR SIEM security event format.

Security Event Field Name	Security Event Field Description
SPEAR Component	Identifier of the SPEAR SIEM component, which generates the security event. Three options are available: AlienVault OSSIM, BDAC and VIDS.
Date	Date and time of the security event.
Sensor	The sensor, which processed the security event.
Device IP	The IP address of the sensor, which processed the security event.
Event Type ID	Identifier assigned by the component, which generates the security event.
Unique Event ID	Unique identifier assigned by the component, which generates the security event.
Protocol	Protocol related to the security event.
Category	Event taxonomy for the security event. In the context of BDAC and VIDS, it is “Cyberattack” or “Anomaly”.
Subcategory	Subcategory of the security event taxonomy type listed under Category. In the context of BDAC and VIDS, it is a specific cyberattack or anomaly.
Data Source Name	Name of the external application or device that produced the security event. In the context of BDAC and VIDS, it related to VIDS itself or the internal modules of BDAC.
Data Source ID	Identifier related to the external application or device which generated the security event. In the context of BDAC and VIDS, it is related to the internal modules of BDAC or VIDS itself.
Product Type	Product type related to the security event.
Additional Info	Uniform Resource Locator (URL) including more details about the security event.
Priority	It reflects the significance of the security event in the range between 0-5.
Reliability	It reflects the detection reliability in the range between 0-10.
Risk	Risk calculation relies on the formula: *Asset Value * Event Reliability * Event Priority / 25*
OTX Indicators	Number of indicators related to an OTX IP reputation or OTX pulse event. In the context of BDAC and VIDS, it is null.
Source/Destination ID	Identifier of the source/destination related to the security event.

Source/Destination IP	IP addresses of source/destination, respectively related to security event.
Source/Destination Hostname	Hostname of source/destination.
Source/Destination MAC Address	Media Access Control (MAC) of source/destination.
Source/Destination Port	Port of source/destination.
Source/Destination Latest Update	The last time when the component, which generated the security event updated the source/destination properties.
Source/Destination Username and Domain	Username and domain related to source/destination.
Source/Destination Asset Value	Asset value of source/destination. It reflects the significance of source/destination.
Source/Destination Location	If the origin of source/destination is known, it reflects the host country.
Source/Destination Context	If the asset belongs to a user-defined group of entities, AlienVault OSSIM shows the relevant contexts. In the context of BDAC and VIDS, it is null.
Source/Destination Asset Groups	When the source/destination belongs to one or more asset groups, this field lists the asset group name or names.
Source/Destination Networks	When the source/destination belongs to one or more networks, this field lists the networks.
Source/Destination Logged Users	A list of users and their information related to source/destination.
Source/Destination OTX IP Reputation	(Yes or No) Whether or not the OTX IP Reputation identifies the IP address as suspicious.
Source/Destination Service	List of services or applications related to the source/destination ports.
Service Port	Port utilised by the service or application.
Service Protocol	Protocol utilised by the service or application.
Raw Log	Raw log details of the security event.
Filename	Name of a file related to the security event.
Username	Usernames related to the security event.
Password	Passwords related to the security event.
Userdata 1-9	User-generated log fields.
Rule Detection	AlienVault OSSIM NIDS rule used to detect the security event. In the context of BDAC and VIDS, BDAC internal modules and VIDS itself are used, respectively.

Appendix B. VIDS Correlation Rules for Modbus

Table B.9 summarises the VIDS correlation rules for Modbus.

Table B.9: VIDS Correlation Rules for Modbus

No	**Description**
Rule #1	If there are X or more consecutive events denoting a modbus/function/ readInputRegister (DoS) attack, then an alert called 'modbus/function/ readInputRegister (DoS)' is raised. X is defined by the user.
Rule #2	If there are X or more consecutive events denoting a modbus/dos /writeSingleRegister attack, then an alert called 'modbus/dos /writeSingleRegister' is raised. X is defined by the user.
Rule #3	If there are X or more consecutive events denoting a modbus/function/ readDiscreteInputs (DoS) attack, then an alert called 'modbus/function/readDiscreteInputs (DoS)' is raised. X is defined by the user.
Rule #4	If there are X or more consecutive events denoting a modbus/ function/readHoldingRegister (DoS) attack, then an alert called 'modbus/function/readHoldingRegister (DoS)' is raised. X is defined by the user.
Rule #5	If there are X or more consecutive events denoting a modbus /function/readCoils (DoS) attack, then an alert called 'modbus/function/readCoils (DoS))' is raised. X is defined by the user.
Rule #6	If there are X or more consecutive events denoting a modbus/dos /writeSingleCoils attack, then an alert called 'modbus/dos /writeSingleCoils' is raised. X is defined by the user.
Rule #7	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/scanner/getfunc, then an alert called 'Modbus Reconnaissance'. X is defined by the user.
Rule #8	If there are X or more consecutive events denoting a modbus/ scanner/getfunc attack, then an alert called 'Modbus Reconnaissance' is raised. X is defined by the user.
Rule #9	If there are X or more consecutive events denoting a modbus/scanner /uid attack, then an alert called 'Modbus Reconnaissance' is raised. X is defined by the user.
Rule #10	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/writeSingleCoils, then an alert called 'modbus/function/writeSingleCoils' is raised. X is defined by the user.
Rule #11	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/writeSingleCoils, then an alert called 'modbus/function/writeSingleCoils' is raised. X is defined by the user.

Rule #12	If there are X or more consecutive events denoting a modbus/function /writeSingleCoils, then an alert called 'modbus/function/ writeSingleCoils' is raised. X is defined by the user.
Rule #13	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/readInputRegister, then an alert called 'modbus/function/readInputRegister' is raised. X is defined by the user.
Rule #14	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/readInputRegister, then an alert called 'modbus/function/readInputRegister' is raised. X is defined by the user.
Rule #15	If there are X or more consecutive events denoting a modbus/function /readInputRegister, then an alert called 'modbus/function/ readInputRegister' is raised. X is defined by the user.
Rule #16	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/writeSingleRegister, then an alert called 'modbus/function/writeSingleRegister' is raised. X is defined by the user.
Rule #17	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/writeSingleRegister, then an alert called 'modbus/function/writeSingleRegister' is raised. X is defined by the user.
Rule #18	If there are X or more consecutive events denoting a modbus/function /writeSingleRegister, then an alert called 'modbus/function /writeSingleRegister' is raised. X is defined by the user.
Rule #19	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/readDiscreteInput, then an alert called 'modbus/function/readDiscreteInput' is raised. X is defined by the user.
Rule #20	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/readDiscreteInput, then an alert called 'modbus/function/readDiscreteInput' is raised. X is defined by the user.
Rule #21	If there are X or more consecutive events denoting a modbus/function /readDiscreteInput, then an alert called 'modbus/function /readDiscreteInput' is raised. X is defined by the user.
Rule #22	If there are X events denoting a modbus/scanner/uid attack and right after X events denoting a modbus/function/readHoldingRegister, then an alert called 'modbus/function/readHoldingRegister' is raised. X is defined by the user.
Rule #23	If there are X events denoting a modbus/scanner/getfunc attack and right after X events denoting a modbus/function/readHoldingRegister, then an alert called 'modbus/function/readHoldingRegister' is raised. X is defined by the user.

Rule #24	If there are X or more consecutive events denoting a modbus/function /readHoldingRegister, then an alert called ‘modbus/function /readHoldingRegister’ is raised. X is defined by the user.

Appendix C. BDAC Evaluation Results - Comprehensive ML/DL Comparative Analysis

The Appendix C presents the ML/DL comparative analysis related to the intrusion and anomaly detection models of the BDAC Analysis Engine. In particular, Tables C.10-C.28 reflect this evaluation process. It is worth noting that all ML and DL methods were fine-tuned after several experiments.

Table C.10: Modbus/TCP Network Flow-Based Intrusion Detection Model Evaluation Results.

Classification Problem	Multi-Class Classification			
Data Type	Network flow statistics (related only to Modbus/TCP network flows specified by the 502 TCP port)			
Features	All features exported by CICFlowMeter [54], excluding FlowID, SrcIP, DstIP and Timestamp			
Total Dataset Size	255000 Modbus flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.943	0.603	0.030	0.603
LDA	0.943	0.604	0.030	0.604
Decision Tree Classifier	0.964	0.749	0.019	0.749
Nave Bayes	0.928	0.497	0.038	0.497
SVM RBF	0.918	0.426	0.044	0.426
SVM Linear	0.921	0.453	0.042	0.453
Random Forest	0.947	0.633	0.028	0.633
MLP	0.938	0.570	0.033	0.570
Adaboost	0.887	0.214	0.060	0.214
Quadratic Discriminant Analysis	0.941	0.593	0.031	0.593
Dense DNN Relu	0.945	0.619	0.029	0.619
Dense DNN Tanh	0.945	0.619	0.029	0.619

Table C.11: Modbus/TCP Network Flow-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type	Network flow statistics (related only to Modbus/TCP network flows specified by the 502 TCP port)			
Features	Flow Duration, TotLen Fwd Pkts, Fwd Pkt Len Mean, Bwd Pkt Len Std, Flow IAT Std, Bwd Pkts/s, Subflow Fwd Byts, Init Fwd Win Byts, Active Mean			
Total Dataset Size	255000 Modbus flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.949	0.999	0.100	0.951

Isolation Forest	0.950	0.999	0.099	0.952
PCA	0.540	0.846	0.567	0.488
MCD	0.948	0.999	0.102	0.950
LOF	0.947	0.999	0.104	0.950
DIDEROT Autoencoder	0.950	0.999	0.099	0.952

Table C.12: Modbus/TCP Packet-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type	Attributes of Modbus/TCP Payload			
Features	TCP-LEN, TRANSACTION-ID, PROTOCOL-ID, UNIT-ID , FCODE, LEN, START-ADDR, BYTE-COUNT			
Total Dataset Size	255000 Modbus packets			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.819	0.800	0.166	0.784
Isolation Forest	0.943	0.952	0.062	0.930
PCA	0.909	0.869	0.062	0.888
MCD	0.905	0.857	0.062	0.878
LOF	0.943	0.952	0.062	0.930
DIDEROT Autoencoder	0.888	0.074	0.898	0.968

Table C.13: IEC 60870-5-104 Network Flow-Based Intrusion Detection Model Evaluation Results.

Classification Problem	Multi-class Classification			
Data Type	Network flow statistics (related only to IEC 60870-5-104 network flows specified by the 2404 TCP port)			
Features	Flow Duration, TotLen Fwd Pkts, Fwd Pkt Len Mean, Bwd Pkt Len Std, Flow IAT Std, Bwd Pkts/s, Subflow Fwd Byts, Init Fwd Win Byts, Active Mean			
Total Dataset Size	100000 IEC 60870-5-104 flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.900	0.602	0.056	0.602
LDA	0.904	0.619	0.054	0.619
Decision Tree Classifier	0.953	0.815	0.026	0.815
Nave Bayes	0.855	0.421	0.082	0.421
SVM RBF	0.853	0.413	0.083	0.413
SVM Linear	0.843	0.375	0.089	0.375
Random Forest	0.918	0.672	0.046	0.672

MLP	0.904	0.619	0.054	0.619
Adaboost	0.843	0.375	0.089	0.375
Quadratic Discriminant Analysis	0.899	0.598	0.057	0.598
Dense DNN Relu	0.909	0.636	0.051	0.636
Dense DNN Tanh	0.916	0.664	0.047	0.664

Table C.14: IEC 60870-5-104 Network Flow-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type				
Features	Flow Duration, TotLen Fwd Pkts, Fwd Pkt Len Mean, Bwd Pkt Len Std, Flow IAT Std, Bwd Pkts/s, Subflow Fwd Byts, Init Fwd Win Byts, Active Mean			
Total Dataset Size	100000 IEC 60870-5-104 flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.937	0.947	0.074	0.942
Isolation Forest	0.948	0.967	0.074	0.952
PCA	0.699	0.892	0.347	0.537
MCD	0.948	0.999	0.102	0.950
LOF	0.953	0.941	0.038	0.941
DIDEROT Autoencoder	0.881	0.852	0.089	0.877

Table C.15: IEC 60870-5-104 Packet-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type	Attributes of IEC 60870-5-104 Payload			
Features	frame_length, testfr_con, testfr_act, stopdt_con, stopdt_act, startdt_con, startdt_act			
Total Dataset Size	100000 IEC 60870-5-104 packets			
Training Dataset Size	75%			
Tesing Dataset Size	25%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.508	0.452	0.440	0.466
Isolation Forest	0.893	0.860	0.074	0.889
PCA	0.535	0.500	0.431	0.513
MCD	0.734	0.594	0.125	0.691
LOF	0.926	0.859	0.005	0.921
DIDEROT Autoencoder	0.748	0.568	0.072	0.692

Table C.16: IEC 61850 (MMS) Network Flow-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type	Network flow statistics (related only to MMS network flows identified by the 102 TCP port)			
Features	Flow Duration, TotLen Fwd Pkts, Fwd Pkt Len Mean, Bwd Pkt Len Std, Flow IAT Std, Bwd Pkts/s, Subflow Fwd Byts, Init Fwd Win Byts, Active Mean			
Total Dataset Size	80000 IEC 61850 (MMS) flows			
Training Dataset Size	75%			
Testing Dataset Size	25%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.973	0.970	0.024	0.966
Isolation Forest	0.977	0.971	0.019	0.971
PCA	0.506	0.524	0.511	0.514
MCD	0.981	0.986	0.022	0.977
LOF	0.954	0.924	0.022	0.945
DIDEROT Autoencoder	0.960	0.982	0.115	0.9743

Table C.17: BACnet Network Flow-Based Intrusion Detection Model Evaluation Results.

Classification Problem	Multi-class Classification			
Data Type	Attributes of BACnet Payload			
Features	All features exported by CICFlowMeter [54], excluding FlowID, SrcIP, SrcPort, DstIP DstPort, Protocol and Timestamp			
Total Dataset Size	100000 BACnet flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.960	0.982	0.115	0.974
Nave Bayes	0.902	0.925	0.115	0.894
KNN	0.934	0.968	0.090	0.928
SVM RBF	0.924	0.952	0.090	0.897
SPEAR SDAE	0.909	0.991	0.090	0.979
Random Forest	0.959	0.969	0.090	0.972

Table C.18: BACnet Packet-Based Intrusion Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection
Data Type	Attributes of BACnet Payload
Features	BACnet payload text is parsed and split into tokens, using the ntlk regular expression tokenizer. The result is a sentence composed of tokens for each packet.
Total Dataset Size	100000 BACnet packets

Training Dataset Size	75%			
Tesing Dataset Size	25%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Multinomial Nave Bayes	0.771	0.661	0.339	0.761
Logistic Regression	0.808	0.850	0.330	0.872
SVM RBF	0.962	0.961	0.032	0.978
Payload Text CNN	0.967	0.967	0.032	

Table C.19: MQTT Network Flow-Based Intrusion Detection Model Evaluation Results.

Classification Problem	Multi-Class Classification			
Data Type	Network flow statistics (related only to MQTT network traffic identified by the 1883/8883 TCP ports)			
Features	All features exported by CICFlowMeter [54], excluding FlowID, SrcIP, SrcPort, DstIP DstPort, Protocol and Timestamp			
Total Dataset Size	90000 MQTT flows			
Training Dataset Size	75%			
Tesing Dataset Size	25%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.939	0.878	0.040	0.863
Nave Bayes	0.869	0.739	0.086	0.761
KNN	0.941	0.950	0.065	0.926
SVM RBF	0.956	0.913	0.028	0.907
Random Forest	0.970	0.967	0.017	0.982
SPEAR SDAE	0.992	0.984	0.005	0.984

Table C.20: MQTT Packet-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type	Attributes of MQTT Pyload			
Features	MQTT payload text is parsed and split into tokens using ntlk regular expression tokenizer. The result is a sentence with tokens for each packet.			
Total Dataset Size	90000 MQTT packets			
Training Dataset Size	75%			
Tesing Dataset Size	25%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Multinomial Nave Bayes	0.728	0.728	0.271	0.667
Logistic Regression	0.890	0.890	0.109	0.880
SVM RBF	0.890	0.890	0.109	0.880
Payload text CNN	0.985	0.985	0.014	0.985

Table C.21: HTTP Network Flow-Based Intrusion Detection Model Evaluation Results.

Classification Problem	Multi-Class Classification			
Data Type	Network flow statistics (related only to HTTP(S) network flows identified by the 80 TCP port)			
Features	All features exported by CICFlowMeter [37],excluding FlowID, SrcIP, DstIP and Timestamp			
Total Dataset Size	150000 HTTP flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.937	0.844	0.038	0.844
LDA	0.946	0.866	0.033	0.866
Decision Tree Classifier	0.964	0.911	0.026	0.911
Nave Bayes	0.878	0.696	0.075	0.696
SVM RBF	0.908	0.770	0.057	0.770
SVM Linear	0.928	0.822	0.044	0.822
Random Forest	0.922	0.807	0.048	0.807
MLP	0.940	0.851	0.037	0.851
Adaboost	0.760	0.400	0.150	0.400
Quadratic Discriminant Analysis	0.911	0.777	0.055	0.777
Dense DNN Relu	0.940	0.851	0.037	0.851
Dense DNN Tanh	0.940	0.851	0.0370	0.851

Table C.22: HTTP Network Flow-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type	Network flow statistics (related only to HTTP(S) network flows identified by the 80 TCP port)			
Features	All features exported by CICFlowMeter [37],excluding FlowID, SrcIP, DstIP and Timestamp			
Total Dataset Size	150000 HTTP flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.577	0.571	0.416	0.558
Isolation Forest	0.833	0.948	0.281	0.850
PCA	0.596	0.592	0.400	0.581
MCD	0.719	0.545	0.106	0.660
LOF	0.946	0.954	0.058	0.938
DIDEROT Autoencoder	0.934	0.927	0.061	0.902

Table C.23: SSH Network Flow-Based Intrusion Detection Model Evaluation Results.

Classification Problem	Multi-Class Classification			
Data Type	Network flow statistics (related only to SSH network flows identified by the 22 TCP port)			
Features	Dst Port, Flow Duration, TotLen Fwd Pkts, Fwd Pkt Len Mean, Bwd Pkt Len Std, Flow IAT Std, Bwd Pkts/s, Subflow Fwd Byts, Init Fwd Win Byts, Active Mean			
Total Dataset Size	10000 SSH flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
Logistic Regression	0.859	0.750	0.058	0.821
LDA	0.945	0.920	0.038	0.928
Decision Tree Classifier	0.960	0.958	0.038	0.955
Nave Bayes	0.823	0.741	0.154	0.640
SVM RBF	0.837	0.660	0.339	0.788
SVM Linear	0.799	0.845	0.307	0.307
Random Forest	0.955	0.903	0.009	0.942
MLP	0.903	0.841	0.010	0.910
Adaboost	0.950	0.890	0.010	0.934
Quadratic Discriminant Analysis	0.500	0.500	0.250	0.666
Dense DNN Relu	0.916	0.985	0.014	0.906
Dense DNN Tanh	0.916	0.836	0.011	0.904

Table C.24: SSH Network Flow-Based Anomaly Detection Model Evaluation Results.

Classification Problem	Outlier/Novelty Detection			
Data Type	Network flow statistics (related only to SSH network flows identified by the 22 TCP port)			
Features	Flow Duration, TotLen Fwd Pkts, Fwd Pkt Len Mean, Bwd Pkt Len Std, Flow IAT Std, Bwd Pkts/s, Subflow Fwd Byts, Init Fwd Win Byts, Active Mean			
Total Dataset Size	10000 SSH flows			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.935	0.870	0.013	0.922
Isolation Forest	0.943	0.901	0.013	0.941
PCA	0.701	0.596	0.247	0.564
MCD	0.957	0.970	0.050	0.944
LOF	0.925	0.913	0.066	0.909
DIDEROT Autoencoder	0.946	0.954	0.058	0.938

Table C.25: Operational Data Based Anomaly Detection Model Hydropower Plant Use Case.

Classification Problem	Outlier/Novelty Detection			
Data Type	Oprational Data - Hydropower Plant Use Case			
Features	'DE', 'power', 'waterlevel', 'NDE', 'nozzles'			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.581	0.993	0.522	0.487
Isolation Forest	0.716	0.948	0.341	0.572
PCA	0.745	0.978	0.312	0.606
MCD	0.733	0.210	0.135	0.240
LOF	0.579	0.996	0.525	0.486
ARIES GAN	0.746	0.978	0.311	0.607

Table C.26: Operational Data Based Anomaly Detection Model Substation Use Case.

Classification Problem	Outlier/Novelty Detection			
Data Type	Operational Data - Substation Use Case			
Features	ACTIVE_POWER_SOE, APPARENT_POWER_SOE, CURRENT_SOE, FRECUENCY_SOE REACTIVE_POWER_SOE, TEMPERATURE_SOE, TRAFOS_POSITION_SOE, VOLTAGE_SOE			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.839	0.995	0.200	0.713
Isolation Forest	0.850	0.951	0.175	0.718
PCA	0.847	0.961	0.181	0.716
MCD	0.822	0.991	0.220	0.691
LOF	0.873	0.993	0.157	0.759
ARIES GAN	0.840	0.961	0.189	0.708

Table C.27: Operational Data Based Anomaly Detection Model Power Plant Use Case.

Classification Problem	Outlier/Novelty Detection
Data Type	Operational Data - Power Plant Use Case

Features	v24_batteries, v60_batteries, generator_speed, gen_motor_voltage, gen_motor_current, exc_motor_voltage, exc_motor_current, incom_cooling_water, gen_status_winding2, gen_outlet_air, exc_set_bearing2, grid_phase_r, grid_phase_s, grid_phase_t, main_mg_nn, exc_mg_nn, overvolt_main_gen, overcur_main_gen, rem_command, com_fault			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.692	0.989	0.397	0.600
Isolation Forest	0.813	0.960	0.231	0.705
PCA	0.851	0.982	0.187	0.755
MCD	0.715	0.299	0.158	0.329
LOF	0.829	0.992	0.220	0.730
ARIES GAN	0.851	0.982	0.188	0.755

Table C.28: Operational Data Based Anomaly Detection Model Smart Home Use Case.

Classification Problem	Outlier/Novelty Detection			
Data Type	Operational Data - Smart Home Use Case			
Features	AoutPhL1, AoutPhL2, AoutPhL3, BattAmp, BattTemp, BattVolt, PinPhL1, PinPhL2, PinPhL3, PoutPhL1, PoutPhL2, PoutPhL3, VoutPhL1, VoutPhL2, VoutPhL3			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**ACC**	**TPR**	**FPR**	**F1**
ABOD	0.649	0.668	0.362	0.597
Isolation Forest	0.769	0.976	0.279	0.615
PCA	0.859	0.976	0.167	0.724
MCD	0.729	0.992	0.332	0.581
LOF	0.690	0.735	0.344	0.676
ARIES GAN	0.859	0.976	0.167	0.725

Appendix D. VIDS Evaluation Results - Comprehensive ML/DL Comparative Analysis

The Appendix D shows the ML/DL comparative analysis related to the intrusion and anomaly detection models of VIDS. In particular, Tables D.29-D.32 reflect this evaluation process. It is worth noting that all ML and DL methods were fine-tuned after several experiments.

Table D.29: VIDS Evaluation Results Hydropower Plant Use Case.

Classification Problem	Visual-based Anomaly Detection			
Data Type	Oprational Data - Hydropower Plant Use Case			
Features	’DE’, ’power’, ’waterlevel’, ’NDE’, ’nozzles’			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**2D-ACC**	**3D-ACC**	**2D-F1**	**3D-F1**
PCA	0.8626	0.3568	0.7618	0.4205
SVD	0.8912	0.3881	0.7954	0.4172
ICA	0.8357	0.6738	0.7291	0.5635
SRP	0.7806	0.2344	0.6442	0.3797
Deep Dense Autoencoder	0.7354	0.4207	0.0782	0.4089
FF Autoencoder	0.2344	0.2344	0.4780	0.3797
LSTM Autoencoder	0.9830	0.9500	0.9637	0.8819
Classic Dense Sequence Autoencoder	0.2344	0.3548	0.3797	0.4111

Table D.30: VIDS Evaluation Results Substation Use Case.

Classification Problem	Visual-based Anomaly Detection			
Data Type	Oprational Data - Substation Use Case			
Features	ACTIVE_POWER_SOE, APPARENT_POWER_SOE, CURRENT_SOE, FRECUENCY_SOE REACTIVE_POWER_SOE, TEMPERATURE_SOE, TRAFOS_POSITION_SOE, VOLTAGE_SOE			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**2D-ACC**	**3D-ACC**	**2D-F1**	**3D-F1**
PCA	0.6210	0.2754	0.5637	0.4318
SVD	0.7184	0.3999	0.6361	0.1597
ICA	0.7615	0.5376	0.6573	0.0647
SRP	0.8053	0.7316	0.7260	0.2431
Deep Dense Autoencoder	0.7573	0.2754	0.6654	0.4318
FF Autoencoder	0.6412	0.2754	0.5575	0.4318
LSTM Autoencoder	0.8122	0.8491	0.8443	0.7815

Classic Dense Sequence Autoencoder	0.7643	0.2754	0.5285	0.4318

Table D.31: VIDS Evaluation Results Power Plant Use Case.

Classification Problem	Visual-based Anomaly Detection			
Data Type	Oprational Data - Power Plant Use Case			
Features	v24_batteries, v60_batteries, generator_speed, gen_motor_voltage, gen_motor_current, exc_motor_voltage, exc_motor_current, incom_cooling_water, gen_status_winding2, gen_outlet_air, exc_set_bearing2, grid_phase_r, grid_phase_s, grid_phase_t, main_mg_nn, exc_mg_nn, overvolt_main_gen, overcur_main_gen, rem_command, com_fault			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**2D-ACC**	**3D-ACC**	**2D-F1**	**3D-F1**
PCA	0.8571	0.6466	0.7879	0.0371
SVD	0.8187	0.8701	0.7494	0.7873
ICA	0.8037	0.7007	0.7274	0.6396
SRP	0.8126	0.2973	0.7360	0.4583
Deep Dense Autoencoder	0.7969	0.6116	0.7084	0.5723
FF Autoencoder	0.7861	0.9143	0.7257	0.8657
LSTM Autoencoder	0.7027	0.9776	0.8298	0.9631
Classic Dense Sequence Autoencoder	0.7782	0.9085	0.7009	0.8501

Table D.32: VIDS Evaluation Results Smart Home Use Case.

Classification Problem	Visual-based Anomaly Detection			
Data Type	Oprational Data - Power Plant Use Case			
Features	AoutPhL1, AoutPhL2, AoutPhL3, BattAmp, BattTemp, BattVolt, PinPhL1, PinPhL2, PinPhL3, PoutPhL1, PoutPhL2, PoutPhL3, VoutPhL1, VoutPhL2, VoutPhL3			
Total Dataset Size	10000 time-series			
Training Dataset Size	70%			
Tesing Dataset Size	30%			
ML/DL Method	**2D-ACC**	**3D-ACC**	**2D-F1**	**3D-F1**
PCA	**0.9220**	0.9229	0.8351	0.8286
SVD	0.910	0.940	0.105	0.881
ICA	0.8814	0.8551	0.7799	0.7341
SRP	0.8822	0.7178	0.7818	0.5801
Deep Dense Autoencoder	0.9212	0.8864	0.8318	0.7528

FF Autoencoder	0.9280	0.9254	0.8468	0.8358
LSTM Autoencoder	0.7627	0.6890	0.6577	0.5954
Classic Dense Sequence Autoencoder	0.9017	0.8881	0.7943	0.7740