A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments

Ilias Siniosoglou†, Panagiotis Radoglou-Grammatikis†, Georgios Efstathopoulos‡, Panagiotis Fouliras§ and Panagiotis Sarigiannidis†

***Abstract*—The interconnected and heterogeneous nature of the next-generation Electrical Grid (EG), widely known as Smart Grid (SG), bring severe cybersecurity and privacy risks that can also raise domino effects against other Critical Infrastructures (CIs). In this paper, we present an Intrusion Detection System (IDS) specially designed for the SG environments that use Modbus/Transmission Control Protocol (TCP) and Distributed Network Protocol 3 (DNP3) protocols. The proposed IDS called $MENSA$ (anoMaly dEtection aNd claSsificAtion) adopts a novel Autoencoder-Generative Adversarial Network (GAN) architecture for (a) detecting operational anomalies and (b) classifying Modbus/TCP and DNP3 cyberattacks. In particular, $MENSA$ combines the aforementioned Deep Neural Networks (DNNs) in a common architecture, taking into account the adversarial loss and the reconstruction difference. The proposed IDS is validated in four real SG evaluation environments, namely (a) SG lab, (b) substation, (c) hydropower plant and (d) power plant, solving successfully an outlier detection (i.e., anomaly detection) problem as well as a challenging multiclass classification problem consisting of 14 classes (13 Modbus/TCP cyberattacks and normal instances). Furthermore, $MENSA$ can discriminate five cyberattacks against DNP3. The evaluation results demonstrate the efficiency of $MENSA$ compared to other Machine Learning (ML) and Deep Learning (DL) methods in terms of Accuracy, False Positive Rate (FPR), True Positive Rate (TPR) and the F1 score.**

***Index Terms*—Anomaly Detection, Auto-encoder, Cybersecurity, Generative Adversarial Network, Deep Learning, Machine Learning, Modbus, Smart Grid,**

I. Introduction

The rapid advance of the Industrial Internet of Things (IIoT) leads the conventional Electrical Grid (EG) into a new digital paradigm called Smart Grid (SG), providing significant benefits, such as better utilisation of the existing resources, pervasive control and self-healing. According to [1], the SG will compose the biggest Internet of Things (IoT) application. However, the evolution of the smart technologies introduces severe cybersecurity issues due to (a) the necessary presence of insecure, legacy systems, such as Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) [2], (b) the vulnerability nature of Transmission Control Protocol/Internet Protocol (TCP/IP) [3] and (c) the new attack surface introduced by the smart technologies [4].

Denial of Service (DoS), unauthorised access and False Data Injection (FDI) compose expected attack vectors targeting the SG with disastrous consequences. The first one target the availability of the involved systems, while the other ones exploit the vulnerabilities of the industrial protocols in order to compromise the confidentiality, integrity and authenticity of the exchanged information. A characteristic Advanced Persistent Threat (APT) [5] was the BlackEnergy3 [6] in 2015 against a Ukrainian substation, resulting in the power outage for more than 225,000 people. Moreover, the Crashoverride APT in 2016 caused another blackout in Ukraine [6]. Other devastating APTs against Critical Infrastructures (CIs) are Stuxnet, Flame, Duqu [7] and TRITON [8]. Also, in 2014 and 2017, the Dragonfly and Dragonfly 2.0 APTs targeted multiple energy companies [2].

Both industry and academia have provided valuable countermeasures [9]–[13]. In particular, IEC 62351 [14], [15] specifies a set of guidelines in order to enhance the security of ICS/SCADA. Furthermore, based on the aforementioned remarks, the timely, accurate and consistent intrusion detection is necessary. In particular, signature-based Intrusion Detection Systems (IDS), such as `Snort` and `Suricata` can recognise a plethora of known intrusions. Moreover, anomaly-based IDS adopting statistical analysis, Machine Learning (ML) and Deep Learning (DL) methods can detect zero-day attacks and unknown anomalies. However, despite the benefits of the aforementioned solutions, they are characterised by essential limitations [16]. First, in many CIs, such as the SG, the adoption of the IEC 62351 is challenging, especially for the adjustments that need to be taken place in real-time. On the other side, the signature-based IDS can detect only known cyberattack patterns and include only a limited set of signature rules related to industrial communication protocols like Modbus, Distributed Network Protocol 3 (DNP3) and IEC 61850 [2]. Finally, the anomaly-based IDS suffer from a high

*This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 787011 (SPEAR).

† I. Siniosoglou, P. Radoglou-Grammatikis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece - E-Mail: {isiniosoglou, pradoglou, psarigiannidis}@uowm.gr

‡ G. Efstathopoulos is with the 0infinity Limited, Imperial Offices, London, UK, E6 2JG - E-Mail: george@0infinity.net

§ P. Fouliras is with the Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece - E-Mail: pfoul@uom.edu.gr

number of False Positives (FP).

In this paper, we provide an anomaly detection model capable of: (a) detecting anomalies and (b) classifying anomalies into particular cyberattack types. The anomaly detection refers to the process of identifying whether an action is malicious or not. On the other side, the anomaly classification categorises the malicious activities into particular cyberattack types. The proposed model called $MENSA$ (anoMaly dEtection aNd claSsificAtion) combines simultaneously two Deep Neural Networks (DNNs): (a) autoencoder and (b) Generative Adversarial Network (GAN). We validated the efficiency of $MENSA$ with three types of datasets: (a) Modbus/TCP network flows, (b) DNP3 network flows and (c) operational data (i.e., time-series electricity measurements). The datasets related to Modbus/TCP and the operational data are originating from four SG environments: (a) SG lab, (b) substation, (c) hydropower plant and (d) power plant. The DNP3 cyberattacks are related only to the substation environment. Consequently, the contributions of this paper are summarised in the following sentences.

- **Providing a DL-based anomaly detection and classification model called MENSA**. $MENSA$ can detect in parallel both anomalies and particular cyberattacks with high performance in terms of Accuracy, True Positive Rate (TPR), False Positive Rate (FPR) and the F1 score. In particular, the average Accuracy, TPR, FPR and F1 are calculated at 0.947, 0.812, 0.036 and 0.7942, respectively. Compared to the existing anomaly-based IDS [16], $MENSA$ addresses efficiently the FP.
- **Detecting a plethora of Modbus/TCP and DNP3 cyberattacks**: $MENSA$ is able to solve a difficult classification problem by detecting and discriminating efficiently 14 Modbus/TCP-related cyberattacks. Moreover, it can recognise five DNP3 cyberattacks. The $MENSA$ detection capability relies on TCP/IP network flow statistics. Therefore, the $MENSA$ detection efficiency demonstrates also its scalability since similar statistics can be used for detecting cyberattacks against any protocol at the application layer.
- **Detecting anomalies upon operational data**: $MENSA$ can detect anomalies upon various operational data (i.e., electricity measurements) coming from different SG environments.
- **Validating MENSA with real data originating from four use cases**: The efficiency of $MENSA$ was validated using network traffic data and operational data originating from four SG evaluation environments: (a) SG lab, (b) substation, (c) hydropower plant and (d) power plant.
- **Evaluating a plethora of ML/DL methods**: Various ML/DL models were evaluated and compared with each other in terms of Accuracy, TPR, FPR and the F1 score. $MENSA$ DL models provide the best performance.

The rest of this paper is organised as follows. Section II discusses previous relevant works. Section III provides the necessary background. Finally, section IV analyses the $MENSA$ architecture, while section V describes how $MENSA$ is implemented in a SG environment. Finally, section VII concludes this paper.

II. Related Work

Several papers have investigated the IoT and SG security issues. Some remarkable cases are listed in [16]–[36]. In particular, in our previous work in [16], we present a comprehensive study related to the SG intrusion detection solutions. After introducing the necessary background related to the architectural ingredients of the SG, 37 cases are analysed, taking into account the architecture schema, the detection method and their efficiency. Accordingly, in [18] R. Mitchel and I. Chen provide a survey related to the intrusion detection techniques for Cyber-Physical Systems (CPS). Similarly, after giving the necessary information regarding the CPS and intrusion detection methods, R. Mitchel and I. Chen study a plethora of specially designed IDS for the CPS. In [22], S. Rakas et al. examine 26 IDS cases related to SCADA systems. The authors define first an evaluation methodology, which considers the IDS performance, test environment, implementation tools, detection techniques and protocols. Next, after explaining the factors affecting the design and development of the SCADA IDS, they briefly discuss 26 SCADA IDS cases, thereby identifying research gaps and directions for future research work. In parallel, multiple survey papers have studied DL techniques for detecting and classifying anomalies. Characteristic examples are provided in [37]–[39]. Therefore, taking into account the aforementioned points, subsequently, we discuss some specific IDS cases that use DL techniques for detecting intrusions against the SG and SCADA systems. Each paragraph focuses on a dedicated case. Finally, we highlight how our work is differentiated.

In [40], R. Shire et al. provide a malware intrusion detection system for IoT environments, utilising a Convolutional Neural Network (CNN). The proposed IDS consists of three main steps. Fist, a network sniffer undertakes to capture the overall network traffic. For this purpose, a socket Python library is adopted. Next, the `Binvis` tool [41] is used to convert the stored network traffic (i.e., pcap file) into an image. In particular, the Hilbert space-filling curve clustering algorithm [42] is used to extract the images. The specific algorithm overcomes other solutions in maintaining the locality among the objects in multi-dimensional spaces, thus generating a more suitable image imprint. Finally, the image is inserted into a CNN, which undertakes to identify the corresponding malware. The CNN is constructed, utilising `Tensorflow` and more precisely the `MobileNet` module. The performance analysis demonstrates the effectiveness of the proposed IDS.

In [43], Y. He et al. present a DL-based detection method, which is capable of recognising FDI attacks against SCADA systems for stealing energy. In particular, the proposed method is composed of two main detection schemes: (a) State Vector Estimator (SVE) and (b) Deep-Learning Based Identification (DLBI). SVE assesses the real-time measurements by computing the $l^2 - norm$ and comparing it with a particular

threshold value t, which is defined experimentally. If the calculation result is higher than t, then an alarm is reported. Otherwise, the DLBI is activated for evaluating further the real-time measurements. DLBI constitutes a Deep Belief Network (DBN) called Conditional DBN (CDBN), which utilises a Conditional Gaussian Bernoulli Restrictive Boltzmann Machine (CGBRBM) in order to identify the appropriate features. The resiliency of the proposed method is demonstrated based on four simulated cases, utilising an IEEE 118-bus power test system and an IEEE 300-bus system. Moreover, the efficiency of the proposed method is validated by comparing its detection results with the outcomes of two ML solutions: (a) Artificial Neural Network (ANN) and (b) Support Vector Machine (SVM).

In [44], M. Saharkhizan et al. provide an intrusion detection mechanism for the Modbus IoT environments, which aggregates an ensemble of multiple Long-Short-Term-Memory (LSTM) networks. LSTM is a fundamental type of Recurrent Neural networks (RNNs) that can learn the log-term pattern of the training data. The proposed mechanism utilises the dataset of I. Fazao et al. [45] that consists of four cyberattacks-categories, namely (a) Man In The Middle (MITM) attacks, (b) Ping Distributed DoS (DDoS) attacks, (c) TCP SYN DoS attacks and (d) Modbus query flood attacks. Moreover, the authors use the `CICFlowMeter` to generate the corresponding bidirectional network flows. Finally, the output of six LSTM networks is aggregated with the help of a decision tree in order to classify the exported network flows into the categories mentioned above. Based on the evaluation results, the accuracy of the proposed mechanism reaches 99%.

In [46], H. Yang et al. present a network IDS for the DNP3 SCADA systems. The proposed IDS relies on a CNN, which consists of five convolutional layers that are followed by the Rectified Linear Unit (ReLu) to increase the non-linearity of the feature maps. Next, the max-pooling function is applied in order to increase the spatial invariance. The input of CNN is a two-dimensional matrix with an rxD size where r denotes a time window and D the total size of the DNP3 packets' attributes. The time window r is equal to the number of the DNP3 packets transmitted within a second. On the other side, D is equal to 25, i.e., there are 25 DNP3 network packets' attributes originating from the (a) link layer, (b) network layer, (c) transport layer and (d) the application layer. The proposed IDS solves a difficult classification problem consisting of multiple attacks-categories, namely (a) Address Resolution Protocol (ARP) poisoning attacks, (b) TCP SYN Flood attacks, (c) TCP RST attacks, (d) User Datagram Protocol (UDP) flood attacks, (e) DNP3 application transmission attacks, (f) outstation DFC flag attacks, (g) function reset attacks, (h) pseudo-transport layer sequence modification attacks, (i) fragmented message interruption attacks, (j) data-link layer length overflow attacks, (k) configuration capture attacks, (l) outstation data reset attacks, (m) clear object attacks, (n) outstation write without reading, (o) address alteration attacks, (p) unavailable function attacks, (q) dual single-packet attacks and (r) dual multiple-packet attacks. Based on the evaluation results, the overall accuracy of the proposed CNN reaches 99.38%.

In [47], the authors present an Intrusion Prevention System (IPS) focused on the DNP3 cyberattacks. The architecture of the proposed IPS is composed of three modules (a) Data Monitoring Module, (b) DIDEROT Analysis Engine and (c) Response Module. The Data Monitoring Module undertakes to monitor and capture the DNP3 network traffic, extracting the respective network. Then, the DIDEROT Analysis Engine applies a decision tree and an autoencoder in order to recognise potential DNP3 cyberattacks and anomalies, respectively. The decision tree focuses on a classification problem, which is composed of five cyberattacks, namely (a) injection, (b) flooding, (c) DNP3 reconnaissance, (d) replay and (e) masquerading. On the other side, the autoencoder solves an anomaly detection problem, which tries to identify DNP3 anomalies. Finally, the Response Module informs the Software-Defined-Networking (SDN) controller to disrupt the malicious DNP3 flows by transmitting the necessary OpenFlow commands to the SDN Switches. Based on the evaluation results, the F1 score of the proposed decision tree and the DIDEROT autoencoder reach 0.991 and 0.953, respectively.

In our previous work in [48], we provide an anomaly-based IDS called $ARIES$ (smArt gRid Intrusion dEtection System), which secures the SG communications. The architecture of the proposed IDS consists of three modules, namely (a) Data Collection Module, (b) $ARIES$ Analysis Engine and (c) Response Module. The Data Collection module sniffs the overall network traffic, producing the analogous bidirectional network flow statistics. These statistics are analysed by the $ARIES$ Analysis Engine, thus detecting successfully relevant cyberattacks and anomalies. Finally, the Response Module informs the system operator about potential cyberattacks. The $ARIES$ Analysis Engine is composed of three detection layers, namely (a) Network-flow Based detection, (b) Packet-based detection and (c) Operational Data based detection. The first layer is responsible for recognising specific cyber-attacks and anomalies by processing network flow statistics. In particular, it can detect (a) DoS cyberattacks, (b) Secure Shell (SSH) brute-force attacks, (c) File Transfer Protocol (FTP) brute-force attacks, (d) port-scanning cyberattacks and (e) bots. To this end, a decision tree classifier is applied. The second layer focuses on Modbus/TCP anomalies by processing Modbus/TCP packets' attributes via the Isolated Forest algorithm. Finally, the third layer analyses operational data (i.e., time-series electricity measurements) via a GAN called $ARIESGAN$. The evaluation analysis demonstrates the efficiency of all $ARIES$ detection layers. In particular, the F1 score of the first detection layer reaches 0.982, while the F1 score of the second and third layer reaches 0.751 and 0.853, respectively.

Undoubtedly, the works analysed earlier provide valuable insights and methodologies concerning the intrusion detection in CIs. DL is an emerging technology, which can contribute significantly to the defence against the rapid evolution of the cyberthreats and malware. In particular, the lack of labelled data renders DL techniques an ideal solution for constructing

effective security applications since they can identify the appropriate features autonomously. Nevertheless, it is noteworthy that most of the previous works have not been validated with real SG environments and data. Furthermore, apart from [46], [47], most of them either do not consider the SCADA protocols that constitute the root of the most anomalies/intrusions in CIs or cover them partially (i.e., they recognise only a few relevant attacks). Therefore, based on the aforementioned remarks, this paper extends our previous work in [48] by enhancing $ARIESGAN$ and introducing an Autoencoder-GAN architecture with novel minimisation functions, taking into account both the adversarial error and the reconstruction difference. In particular, the proposed Autoencoder-GAN architecture was validated in four real SG evaluation environments that use the Modbus/TCP and DNP3 protocols. Our previous work in [48] could detect only Modbus/TCP anomalies. In contrast, this paper examines and detects a plethora of Modbus/TCP cyberattacks that can be performed by `Smod` [49], a widely known penetration-testing tool related to Modbus.

III. Background

This section provides the necessary background regarding (a) Modbus, (b) DNP3, (c) Autoencoders and (d) GANs. In particular, after describing the core architecture of the Modbus and DNP3 protocols, we specify which Modbus/TCP and DNP3 attacks can be successfully recognised by $MENSA$. Next, the functionality of the Autoencoder and GAN DNNs is provided so that the reader can normally proceed to the unified Autoencoder-GAN architecture described in the following sections. More detailed information about Autoencoders and GAN is provided in [37]–[39].

A. Modbus/TCP and DNP3 Threat Identification

Modbus is an industrial communication protocol adopted widely by SCADA systems in the energy sector due to its simplicity, easy deployment and open specifications. In particular, the general Modbus frame is called Application Data Unit (ADU), which in turn consists of (a) the Protocol Data Unit (PDU), (b) Addressing and (c) Error Checking. PDU encloses the primary information of the Modbus packets, including the function code and the respective data [2]. Each function code defines a different functionality. The addressing and error checking functionalities rely on the Modbus version (i.e., (a) Modbus/Remote Terminal Unit (RTU) or (b) Modbus/TCP). In the Modbus/RTU version, the master and each slave are characterised by unique IDs, while the error checking is achieved through Cyclic Redundancy Check (CRC). On the other hand, in the Modbus/TCP version, the Slave ID field is replaced by the Modbus Application Protocol (MBAP) header, which in turn includes (a) the Transaction Identifier, (b) the Protocol Identifier, (c) Length and (d) Unit Identifier. The protocol identifier is always equal to zero for the current Modbus services, while other values are reserved for potential extensions. Length indicates the size of the remaining field, including Unit ID, Function Code and Data. The Unit ID is utilised for serial connecting to a Modbus device, which does not use the Modbus/TCP version. Finally, the error checking functionality was replaced by the corresponding mechanisms of TCP/IP.

DNP3 is a reliable protocol applied largely in the CIs in the US. In the SG, DNP3 is used to transfer messages between master devices and outstations. It supports several topologies, comprising (a) point-to-point, where an outstation and one master communicate with each other, (b) multiple-drop, where several masters and outstations interact with each other and (c) hierarchical interface, where an entity can operate with both roles. DNP3 includes three layers: (a) link layer, (b) transport layer and (c) application layer. The link-layer offers addressing services, multiplexing, data fragmentation, error checking and link control. On the other side, the transport layer is used as in the case of the Open Systems Interconnection (OSI) model, and it is represented with one byte utilised for fragmenting the DNP3 packets. Finally, the application layer defines a set of functional commands utilised for managing and controlling the SG entities, such as RTUs, Programmable Logic Controllers (PLCs), Intelligent Electronic Devices (IEDs) and smart meters. Apart from the DNP3 serial line communication, DNP3 can be used over TCP/IP, wherein the aforementioned DNP3 layers are incorporated into the application layer of TCP/IP.

Both Modbus and DNP3 are characterised by severe security issues since they were not constructed having cybersecurity in mind [2]. In our previous work in [49] we have identified the Modbus/TCP cyberattacks based on `Smod`. Similarly, in [50], N. Rodofile et al. discuss possible cyberattacks against DNP3. Based on these works, Table I and Table II enumerate the Modbus/TCP and DNP3 cyberattacks that $MENSA$ can classify, respectively.

B. Autoencoder and GAN

A GAN [51], [52] relies on two sub-neural networks, the Generator G and the Discriminator D. The Generator G takes random noise data and generates data similar to the real data. On the other hand, the Discriminator D inputs a data sample and tries to classify it as real or fake. The GAN aims to push and train both sub-networks that rival each other so that the Generator G can produce data that the Discriminator D cannot distinguish from the real ones. Equation (1) shows the relation between G and D.

$$\min_G \max_D V(G,D) = \min_G \max_D \mathbb{E}_{x\sim p_{data}}[log(D(x))] + \mathbb{E}_{z\sim p_z}[log(1-D(G(z)))] \quad (1)$$

G accumulates noise z from space Z mapping it to the space X from which D inputs x. ($p_{data}(x)$ and $(p_z(z))$ denote the probability distribution of spaces X and Z, respectively.

The autoencoders are DNNs that learn to imitate the input data by compressing and inflating it into a multilayer pipeline. In particular, an autoencoder consists of two sub-networks, the Encoder and the Decoder. The Encoder sub-network compresses the input data of space X to a manifold F. In contrast, the decoder sub-network inflates the data of manifold F to a sample P, where $P \sim X$. The goal of the

TABLE I: Modbus/TCP Cyberattacks

Modbus/TCP Cyberattack	Description
modbus/dos/writeSingleCoils	This DoS Modbus/TCP cyberattack uses Modbus/TCP packets with the function code 05 to change the value of a single coil to ON or OFF.
modbus/dos/writeSingleRegister	This DoS Modbus/TCP attack transmits continuously Modbus/TCP packets with the function code 06 to the target system. The goal of the cyberattacker is to affect the availability of the target.
modbus/function/readCoils	This Modbus/TCP unauthorised access cyberattack accesses the content of a single coil. To this end, a Modbus/TCP packet with the function code 01 is utilised.
modbus/function/readCoils (DoS)	This Modbus/TCP cyberattack is another DoS attack, which exploits the function code 01. The attacker sends continuously to the target system a plethora of Modbus/TCP packets with the function code 01 that read the status of a single coil.
modbus/function/readDiscreteInput	This Modbus/TCP unauthorised access cyberattack violates the confidentiality of a Modbus/TCP device by reading the content of multiple discrete inputs. It uses Modbus/TCP packets with the function code 02.
modbus/function/readDiscreteInputs (DoS)	This DoS Modbus/TCP cyberattacks sends a plethora of Modbus/TCP packets with the function code 02.
modbus/function/readHoldingRegister	It constitutes the most usual unauthorised access attack against Modbus/TCP targeting the content of a holding register via a Modbus/TCP packet with the function code 03.
modbus/function/readHoldingRegister (DoS)	This Modbus/TCP cyberattack also targets the availability of a Modbus/TCP device by sending multiple Modbus/TCP packets with the function code 03. This function code is used to read the content of a holding register.
modbus/function/readInputRegister	This unauthorised access Modbus/TCP cyberattack aims to violate the confidentiality of a Modbus/TCP input register by reading its content.
modbus/function/readInputRegister (DoS)	This Modbus/TCP attack sends continuously a plethora of Modbus/TCP packets with the function code 04 (Modbus Read Input Register packet) to the target system, thus aiming to corrupt its availability.
modbus/function/writeSingleCoils	This unauthorised access Modbus/TCP attack takes full advantage of the lack of authentication and authorisation mechanisms by changing the status of single coil either to ON or OFF through a Modbus/TCP packet with the function code 05.
modbus/function/writeSingleRegister	This unauthorised access Modbus/TCP cyberattack targets both the confidentiality and integrity of a Modbus/TCP single register by sending a Modbus/TCP packet with the function code 06, thus changing its content.
modbus/scanner/getfunc	This reconnaissance Modbus/TCP attack enumerates all Modbus/TCP function codes supported by the target system.
modbus/scanner/uid	This Modbus/TCP reconnaissance cyberattack enumerates the slave IDs supported by the target system.

TABLE II: DNP3 Cyberattacks

DNP3 Cyberattack	Description
DNP3 Injection	This cyberattack takes full advantage that DNP3 does not include any authentication and authorisation mechanism. It injects malicious DNP3 packets between the communication of a DNP3 master and DNP3 outstation.
DNP3 Flooding	This cyberattack floods continually the target system with DNP3 messages.
DNP3 Reconnaissance	This cyberattack diagnoses whether the DNP3 protocol is used by the target system.
Replay	This cyberattack replays the DNP3 messages originating from a legitimated entity.
Masquerading	It imitates the behaviour of a legitimate DNP3 entity.

autoencoder architecture is to help the network through the training process, thus producing samples p that are similar to the given real data r. After the training process, the network inputs new data similar to the training data. Equation (2) shows the data pipeline of the autoencoder architecture.

$$r, p : \underset{r,p}{argmin} \left\| X - (p \circ r)X \right\|^2 \\ r : X \to F,\ p : F \to P \tag{2}$$

IV. MENSA Architecture

$MENSA$ combines the DNNs mentioned above to compose a unified DNN architecture for (a) anomaly detection and (b) anomaly classification purposes. This union is accomplished by encapsulating the autoencoder architecture into the structure of the GAN network. The Generator takes the form of the Decoder, while the Discriminator takes the structure of the Encoder. In this schema, the Generator-Decoder takes an input of a noise sample $N \times M$, where N is the number of the noise points in a sample and M is the number of the input samples. Next, the Generator-Decoder inflates those samples to produce samples that imitate the desired data. The Discriminator-Encoder compresses the Generator-Decoder's output into a single point, which is the validity label of the sample. This function is used to discriminate the real and fake samples. An intermediate model is exported after the training process from the Discriminator-Encoder sub-network. This model is part of the Discriminator-Encoder and it is utilised for the anomaly detection procedure. It comprises the input layer up to the latent layer before the output of the network. In particular, it is used to reduce the input dimension into a specified latent space. Two samples pass through the intermediate model: (a) the real data sample and (b) the generated sample. At this point, the Generator-Decoder has learned to generate close to real data that imitates the normal samples. To calculate the anomaly score for the real sample, the Adversarial Loss function is utilised. The Adversarial Loss is the difference between the generated and the real sample. Since the Generator-Decoder has learned to produce normal samples, the greater the Adversarial Loss, the higher the probability of the real sample being abnormal. The equation below describes the Adversarial Loss.

$$AdvL(d_r, d_p) = \|d_r - d_p\| \tag{3}$$

where $AdvL(x)$ is the adversarial loss score, d_r and d_p are the prediction of the latent model in the real and the generated sample, respectively. On the other side, regarding the anomaly classification purpose, a second, lightweight implementation of the combined Autoencoder-GAN architecture is adopted. Both Autoencoder-GAN architectures for anomaly detection and anomaly classification are analysed in the following subsections.

A. MENSA Autoencoder-GAN for Anomaly Detection

In this case, the combined $MENSA$ Autoencoder-GAN works as an anomaly detector. It is trained only with a set of normal samples and can discriminate outliers in a dataset containing both normal and anomalous samples. The structure of the entire network can be separated into three components, (a) the input layer, (b) the Generator-Decoder and (c) the Discriminator-Encoder. Fig. 1 depicts the $MENSA$ Autoencoder-GAN network for anomaly detection.

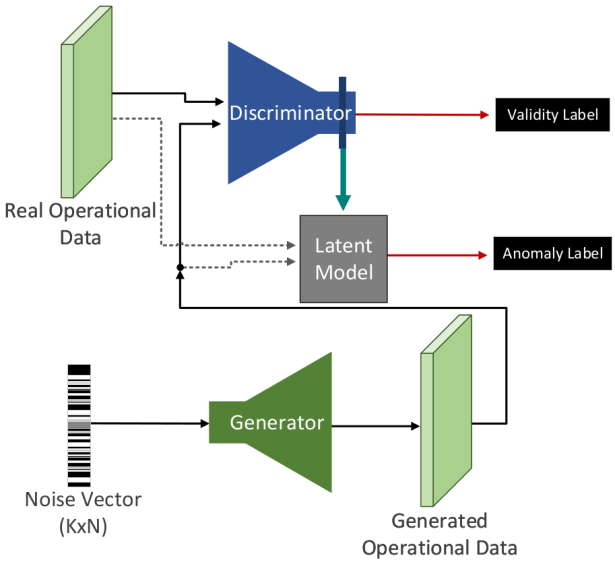

Fig. 1: $MENSA$ Autoencoder-GAN for anomaly detection

Input Layer for Anomaly Detection: The input layer represents the input of the proposed DNN. It takes a noise vector of size N generated based on the uniform distribution with mean μ and standard deviation σ.

Generator-Decoder for Anomaly Detection: The Generator-Decoder is in charge of inflating a random noise input vector of size $z = 10$ to a size M, where M is the number of features, while the generated data imitates the real one. It is trained to produce normal samples. The Generator-Decoder's structure consists of thirteen layers, an input layer, an output $Tanh$ layer and a sequence of $Dense$, $ReLU$, $LeakyReLU$, $Batch$, $Normalization$ and $Dropout$ layers.

$$tanh(x) = 2s(2x) - 1, tanh \to [-1, 1] \tag{4}$$

where equation (4) describes the $Tanh$ function. $tanh(x)$ is the output of the $tanh$ function, $s(x)$ is the sigmoid function (6) and x is the input vector.

An explanatory illustration of the Generator-Decoder's structure is shown in Fig. 2. This network is compiled with the Binary Cross-Entropy function (equation 5) and the RMSprop optimizer with a learning rate parameter of $lr = 0.0002$. The Binary Cross-Entropy function is defined as follows. N is the number of samples given, while y is the label. $p(y_i)$ is the probability of the sample being a match to the label sample when $1-p(y_i)$ presents the inverse of that probability. Finally, H represents the result of the Binary Cross-Entropy loss in a given point.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1-y_i) \cdot log(1-p(y_i)) \quad (5)$$

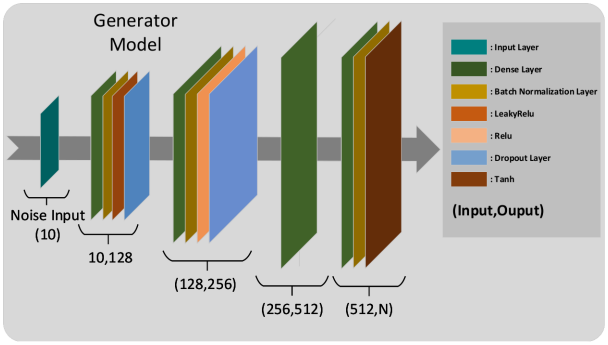

Fig. 2: Generator-decoder structure for anomaly detection

Discriminator-Encoder for Anomaly Detection: The role of the Discriminator-Encoder is to distinguish the real and the generated data samples (i.e., the samples generated by the Generator-Decoder). It takes a vector of M features representing a data instance sample. Next, it compresses the data through a multi-layer pipeline into a single point representing the validity layer (i.e., the binary classification of the sample being real or fake). The Discriminator-Encoder is trained alongside the Generator-Decoder, receiving both real and generated samples, each with a ground truth label. The ground truth labels given as input to the Discriminator-Encoder are represented by $tl \rightarrow 1$ for the Generator-Decoder's output, while $fl \rightarrow 0$ represents the real sample. In the training process, the Discriminator-Encoder's training ability is deactivated when the Generator-Decoder is trained. From the Discriminator-Encoder, the intermediate model is extracted. This network is also compiled with the Binary Cross-Entropy function (equation 5) and the RMSprop optimizer with a learning rate parameter of 0.0002. Thirteen layers compose the Discriminator-Encoder: an input layer, an output Sigmoid layer (equation 6) and a sequence of Dense, ReLU, Leaky ReLU, Batch Normalization and Dropout layers. Fig. 3 illustrates the Discriminator-Encoder's structure. It is noteworthy that the Discriminator-Encoder operates also as an encoder. This means that it reduces the dimension of the input sample from its original dimension to a manifold of size 1, indicating the validity of the sample. The extracted latent model is an intermediate model, describing the fist n layers of D before the output sequence. This n^{th} layer outputs a reduced manifold of size k, which makes the detection easier and faster than comparing the original samples. There is no standard way to determine the n^{th} latent layer. Usually, n^{th} is defined experimentally, identifying the best accuracy/information-loss trade-off.

$$s(x) = \frac{1}{1+e^{-x}}, s \rightarrow [0,1] \quad (6)$$

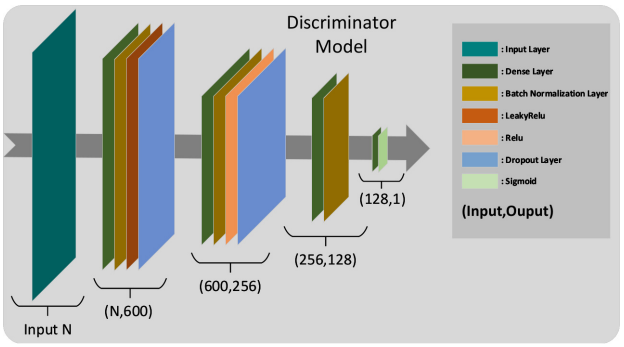

Fig. 3: Discriminator-encoder structure for anomaly detection

Based on the aforementioned remarks, the $MENSA$ anomaly detection process uses the following steps. Given a real sample g_r, first, the Generator-Decoder generates a sample g_p using random noise data. Subsequently, both g_r and g_p are given to the latent model, which in turn outputs the reduced samples d_r and d_p of size k. Next, d_r and d_p are given to formula (3). In order to detect the anomaly, a threshold $t \rightarrow [0,1]$, is leveraged. Finally, if the $AdvL$ outcome is greater than t, then an anomaly is detected.

B. MENSA Autoencoder-GAN for Anomaly Classification

The $MENSA$ Autoencoder-GAN for anomaly classification is derived by the previous $MENSA$ Autoencoder-GAN for anomaly detection. This implementation combines both the process of anomaly detection and anomaly classification into a single DNN. In particular, it produces three ground-truth label points, (a) one for the validity of the sample, (b) one for the anomaly approximation and (c) one describing the anomaly class of the sample. This architecture can also be separated into three parts, (a) the Input layer, (b) the Generator-Decoder and (c) the Discriminator-Encoder. The structure of this DNN is depicted in Fig. 4. The main difference with the previous $MENSA$ Autoencoder-GAN for anomaly detection is that this network is designed to handle multiclass data with fewer features. In contrast, the $MENSA$ Autoencoder-GAN for anomaly detection is designed to handle one class and data with a large number of features.

Input Layer for Anomaly Classification: The input layer takes a noise vector input of size N and a vector containing the classes of the sample. The elements of the random noise vector follow a normal distribution with $\mu = 0$ and $\sigma = 1$. The label vector with a dimension of $[1$ x $C]$, is a zero vector with 1 in the position of the class. C denotes the number of classes that exist in the given dataset. The class of the sample is represented by c_p, which is derived by the following formula.

$$c_p = argmax(V_{label}) \quad (7)$$

where V_{label} is the label vector.

Generator-Decoder for Anomaly Classification: The Generator-Decoder is a modified version of the Generator-Decoder used in Autoencoder-GAN for anomaly detection. In this case, the Generator-Decoder inputs the two vectors

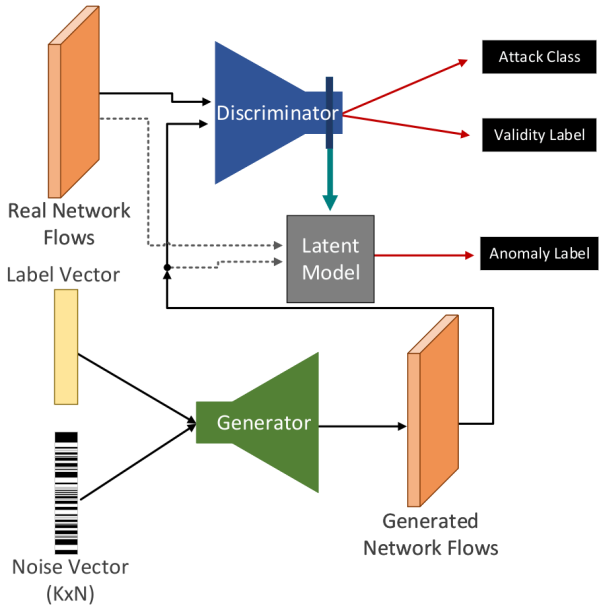

Fig. 4: $MENSA$ Autoencoder-GAN for anomaly classification

explained in the input layer and concatenates them in order to pass through the Generator-Decoder's structure. The Generator-Decoder's structure is illustrated in Fig. 5. It consists of nine layers, an input layer, an output $Relu$ layer and a sequence of $Dense$ and $ReLU$ layers. This network is compiled with the Categorical Cross-Entropy function (equation 8) and the Adadelta optimizer [53]. During the training process, the Generator-Decoder learns to reproduce the data representing each class in the dataset using a label vector. This means that it produces a sample of a certain class, which is introduced as a label vector. The output of this module is a vector of size M, where M is the number of features of the sample.

$$L_{cc}(r,p) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(r_{ij} * log(p_{ij})) \tag{8}$$

The above equation denotes the Categorical Cross-Entropy loss function used to compile the Generator-Decoder. $L_{cc}(y,p)$ is the Categorical Cross-Entropy output, r is the real sample, and p is the generated sample.

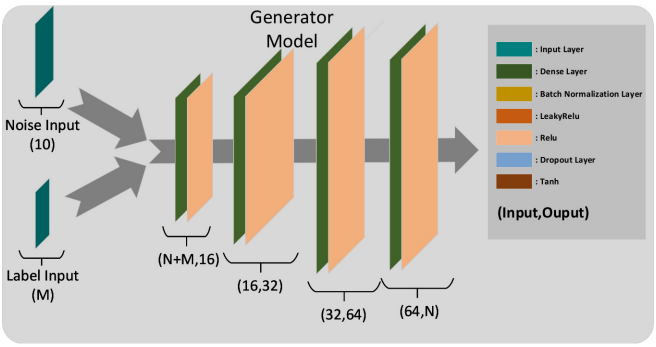

Fig. 5: Generator-Decoder structure for anomaly classification

Discriminator-Encoder for Anomaly Classification: The Discriminator-Encoder takes an input vector of M features, representing a data sample. Since the proposed architecture produces not only the validity approximation but also the anomaly classification of the introduced sample, the output of the Discriminator-Encoder includes two parts. The first part is the validity label of the given sample, distinguishing the sample as real or fake. The second part is a label vector that denotes the multiclass classification of the sample based on the classes given in the dataset. This vector of size C contains the numbers predicted by the Discriminator-Encoder in the range between $[0, 1]$, using the Softmax activation function (equation 9),

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j,n} e^{z_j}} \tag{9}$$

where $softmax(z)_i$ is the output of the layer, n is the dimension of the encoded input vector, z_i denotes the input score and z_j describes each individual score of the encoded input vector.

The class of the sample is the position of the highest value in that vector, as described by equation 7. As in the case of $MENSA$ Autoencoder-GAN for anomaly detection, the Discriminator-Encoder is trained alongside the Generator-Decoder, receiving both real and generated samples, each with a ground truth label and a label vector. The ground truth labels given as input to the Discriminator-Encoder are $tl \rightarrow 1$ for the Generator-Decoder's output and $fl \rightarrow 0$ for the real sample. In the case of the label vectors, for the real sample, the corresponding label vector is given as input to the Discriminator-Encoder, while for the fake or predicted sample, a vector with a random label is given. As previously, the Discriminator-Encoder's training ability is deactivated when the Generator-Decoder is trained. The Discriminator-Encoder is compiled with the Binary Cross-Entropy (equation 5) for the validity. For the classification procedure, the Categorical Cross-Entropy (equation 8) and the Adadelta optimizer [53] are used.

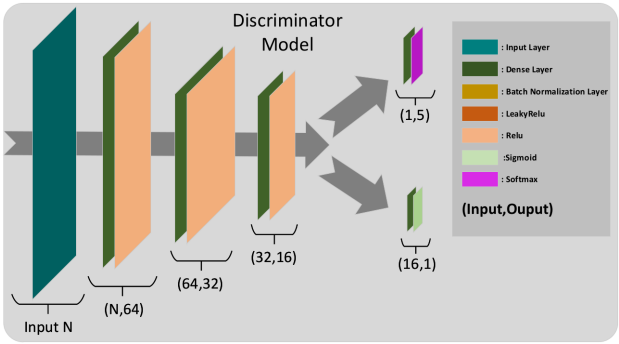

Fig. 6: Discriminator-Encoder structure for anomaly classification

Therefore, to solve the anomaly classification problem the $MENSAAutoencoder - GANforAnomalyDetection$ is extended a step further. Since classification is a multi-class problem, the comparison between the input sample with a

randomly generated sample is not adequate. To overcome this issue, a new conditional architecture is defined. A conditional GAN can generate samples for each class. By asking the Generator-Decoder to generate samples from all the available classes and applying the above methodology used for the anomaly detection, $MENSA$ produces C outputs. By applying the $AdvL$ for each combination of samples d_r^c and d_p^c, where $c \in C$, and choosing the c with the lowest loss, we result in the best approximation of the class of the given sample. Fortifying the effort to optimize the classification process, an additional utility has been added to the $MENSA$ Autoencoder-GAN for anomaly classification. In particular, apart from predicting the validity of the input samples, the Discriminator-Encoder is also designed to predict the classes. Thus, it overtakes the role of a classifier. During the training, the Discriminator-Encoder optimises both the validation and classification processes.

V. MENSA Implementation Capabilities

The SG comprises multiple environments and infrastructures related to the energy generation, transmission and distribution. Therefore, a reliable IDS for the entire SG ecosystem should be able to be adapted appropriately based on the corresponding conditions. These conditions can be expressed sufficiently by the communication protocols and the operational data (i.e., time series electricity measurements) used and exchanged respectively by the components of each SG infrastructure. Furthermore, an essential safety requirement for an IDS in an SG environment is to consider the computing resources of the SG components. In general, the cybersecurity and privacy solutions should not affect and burden the functionality of the SG components [16]. Finally, an IDS solution should act timely and reliably, detecting the possible anomalies and intrusions [16].

Based on the aforementioned remarks, Fig. 7 depicts how $MENSA$ is implemented in an SG environment. $MENSA$ is running on a dedicated computing system without deploying software sensors or services in the SG environment. Thus, it does not affect the computing resources and the normal operation of the SG equipment. In particular, the implementation of $MENSA$ follows five steps: (a) network traffic sniffing, (b) operational data collection, (c) network flow extraction statistics, (d) MENSA anomaly detection and classification and (e) notification. The first step is responsible for capturing the entire network traffic through a Switched Port Analyser (SPAN). To this end, `Tshark` is adopted. `Tshark` can be configured to monitor and sniff the overall network traffic per a specific time threshold, which is defined based on the network characteristics of each SG environment. Next, the appropriate operational data (i.e., time series electricity measurements) is received. This kind of data is received per a specific threshold time through a REpresentational State Transfer (REST) Application Programming Interface (API) with a centralised server called Master Terminal Unit (MTU). MTU is a common ingredient of the SCADA systems, collecting measurements and statistics from the SG equipment, such as PLCs, RTUs and IEDs. Next, the network flow statistics are produced from the network traffic data received from the first step. For this purpose, `CICFlowMeter` is utilised. `CICFlowMeter` is a TCP/IP network flow generator that extracts bidirectional network flow statistics on a predefined flow timeout [54]. Subsequently, the MENSA anomaly detection and classification is applied, as analysed in section IV. The anomaly detection is applied to the operational data, while the anomaly classification is used to discriminate particular cyberattacks based on the network flow statistics. Finally, the last step includes the user notification based on the outcome of the previous step.

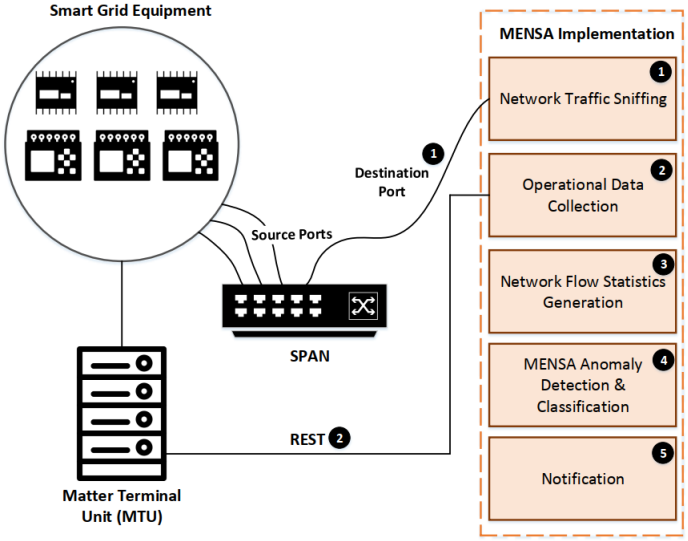

Fig. 7: MENSA Implementation

VI. Evaluation Analysis

This section is devoted to the $MENSA$ evaluation analysis. First, the SG evaluation environments and the operation of $MENSA$ in the prediction phase are described. Next, the datasets and the comparative methods follow. Finally, after introducing the necessary definitions and the evaluation metrics, the $MENSA$ evaluation results are presented in a comparative study with other ML and DL methods.

A. Evaluation Environments

$MENSA$ was evaluated and validated in four real SG evaluation environments coming from the SPEAR project [55], namely (a) SG lab, (b) distribution substation, (c) hydropower plant and (d) power plant. Table III summarises them, showing what communication protocols are supported for each case. Moreover, each of the aforementioned SG environments generates different operational data. Each SG environment possesses its own SCADA system, which monitors and controls the automation procedures. In particular, they are characterised by the presence of appropriate RTUs that manage the operation of industrial elements, such as generators, turbines and transformers. The RTUs communicate with the MTU, utilising the Modbus/TCP protocol. For each SG environment,

TABLE III: MENSA Evaluation: Smart Grid Environment, Protocols and Operational Data

SG Environment	Modbus/TCP	DNP3	Operational Data
SG Lab	✓		✓(SG Lab operational data)
Distribution Substation	✓	✓	✓(Distribution Substation operational data)
Hydropower Plant	✓		✓(Hydropower Plant operational data)
Power Plant	✓		✓(Power Plant operational data)

the Modbus/TCP protocol is utilised in a different way (i.e., different Modbus/TCP function codes). In the substation environment, there are also some IEDs that communicate with the MTU via DNP3. Through a Human Machine Interface (HMI) installed on MTU, the system operator can transmit the necessary commands to the RTUs. Moreover, based on SPAN, $MENSA$ receives the entire Modbus/TCP network traffic and performs the $MENSA$ Autoencoder-GAN for anomaly classification in order to discriminate the Modbus/TCP and DNP3 cyberattacks. In addition, each SG environment generates and stores in MTU respective operational data (i.e., time-series electricity measurements) that is inserted in the $MENSA$ Autoencoder-GAN for anomaly detection. This operational data is received by $MENSA$ through a REST API. $MENSA$ is running in a separate computing system with an Intel(R) Core(TM) i7-8550U CPU - 1.80GHz, 16GB Random Access Memory (RAM) and Ubuntu 20.04.2.0 LTS (Focal Fossa). This machine is also used to extract the evaluation results. Consequently, $MENSA$ is evaluated in several different ways. First, $MENSA$ is evaluated against four SG environments (SG Lab, Distribution Substation, Hydropower Plant, and Power Plant) using the Modbus/TCP protocol in a different way (i.e., different Modbus/TCP function codes). Second, $MESA$ is evaluated in a Distribution Substation under both protocols, i.e., Modbus/TCP and DNP3. Finally, $MENSA$ is evaluated with respect to different operational data per SG environment.

B. Datasets and Comparative Methods

Appropriate datasets were constructed in order to evaluate both $MENSA$ Autoencoder-GAN for anomaly detection and $MENSA$ Autoencoder-GAN for anomaly classification. In the first case, statistically created anomalous samples were injected manually in the database of MTU, thus creating a dataset composed of normal and anomalous time-series electricity measurements for each SG environment mentioned earlier. This data is different for each SG environment. During the pre-processing step, the data is formatted utilising a sliding window of 30 instances and is normalized in the range of $[0, 1]$. On the other side, regarding the validation of $MENSA$ for anomaly classification, the Modbus/TCP cyberattacks of Table I were emulated in a safe manner, utilising Smod [49]. Regarding the DNP3 cyberattacks, the intrusion detection dataset of N.Rodofile et al. [50] was combined with normal DNP3 network flows of the substation environment. Thus, datasets consisting of normal and malicious Modbus/TCP and DNP3 network flows were generated. `CICFlowMeter` was used to extract the Modbus/TCP and DNP3 network flows from the network packet capturing files (i.e., pcap files). Both datasets were labelled since in the first case, the anomalous instance were known, while in the second, the malicious IPs were known. Furthermore, multiple ML and DL methods were adopted in each case in order to compare and evaluate the performance of $MENSA$. In particular, for the anomaly detection, the following ML and DL methods were used: (a) Angle-Based Outlier Detection (ABOD) [56] [57], (b) Isolation Forest (Iforest) [58], (c) Principal Component Analysis (PCA) [59], (d) Minimum Covariance Determinant (MCD) [60], (e) Local Outlier Factor (LOF) [61], (f) DIDEROT Autoencoder [47], (g) ARIES GAN [48] and BlackBox IDS [62]. Similarly, for the anomaly classification, the subsequent methods were utilised: (a) Logistic Regression [63], (b) Linear Discriminant Analysis (LDA) [64], (c) Decision Tree Classifier [65], (d) Gaussian Naive Bayes (Gaussian NB) [66], (e) Support Vector Machine (SVM), (f) Random Forest [67], (g) Multilayer Perceptron (MLP) [68], (h) Adaptive Boosting (AdaBoost) [69], (i) Quadratic Discriminant Analysis [70], (j) Dense DNN ReLU [48] and (k) Dense DNN Tanh [48]. The DIDEROT Autoencoder and the ARIES GAN, Dense DNN Relu and Dense DNN Tanh originate from our previous works in [47] and [48], respectively. It is worth mentioning that the ARIES GAN [48] and the BlackBox IDS [62] constitute advanced, custom DNNs for anomaly detection and anomaly classification, respectively. Finally, for the anomaly classification, `Suricata` was also used with the Quickdraw ICS IDS signatures [71]. `Suricata` is a widely known network IDS, which can detect malicious packets [71]. In order to compare the efficacy of `Suricata` with $MENSA$, we correlated the packets-related alerts extracted by `Suricata` with the corresponding malicious flows.

C. Evaluation Results

Before explaining the evaluation results of $MENSA$, we have to introduce the necessary terms and determine the appropriate evaluation metrics. TP denotes the number of the classifications that recognise correctly an anomaly or a cyberattack as an intrusion. Accordingly, TN implies the amount of the correct classifications that recognise the normal instances as normal. On the other side, FP denotes the number of the incorrect classifications that categorise the normal instances as intrusions. Finally, FN signifies the wrong classifications that classify the anomalous or malicious instances as normal. Therefore, based on these values, the following metrics (equations 10- 13) are defined.

Accuracy (ACC) (equation (10)) expresses the proportion of

the correct classifications and the overall instances. It is a fair evaluation metric when the training dataset consists of an equal number of instances for all categories.

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

The False Positive Rate (FPR) (equation (11)) represents the symmetry of the normal instances that were detected as anomalous/malicious. FPR is determined by dividing FP with the sum of TN and FP.

$$FPR = \frac{FP}{FP + TN} \quad (11)$$

The True Positive Rate (TPR) (equation (12)) defines what ratio of the original anomalous or intrusion instances were recognised as anomalous/intrusions. TPR is computed by dividing TP with the sum of TP and FN.

$$TPR = \frac{TP}{TP + FN} \quad (12)$$

Finally, the F1 score (equation (13)) denotes the golden ratio of $Precision$ and TPR, considering both FP and FN.

$$F1 = \frac{2 \times Precision \times TPR}{Precision + TPR} \text{ where } Precision = \frac{TP}{TP + FP} \quad (13)$$

Table IV presents the $MENSA$ evaluation results for detecting operational anomalies in the first evaluation environment (i.e., the SG lab). $MENSA$ achieves the best performance where $ACC = 0.9647$, $TPR = 0.9418$, $FPR = 0.0282$ and $F1 = 0.9257$. On the other side, MCD presents the worst evaluation results where $ACC = 0.7151$, $TPR = 0.2994$, $FPR = 0.1584$ and $F1 = 0.329$. Accordingly, Table V shows the evaluation results for detecting anomalies in the substation environment. In this case, LOF achieves the best performance, where $ACC = 0.8732$, $TPR = 0.9938$, $FPR = 0.15716$ and $F1 = 0.7591$. In contrast, the ACC, TPR, FPR and the $F1$ $Score$ of $MENSA$ reach 0.8810, 0.7163, 0.0775 and 0.7076, respectively. Table VI reflects the evaluation results for recognising anomalies in the hydropower plant environment. In this case, the best performance is carried out by $MENSA$ where $ACC = 0.8835$, $TPR = 0.8715$, $FPR = 0.1134$ and $F1 = 0.7498$. On the contrary, the lowest performance is accomplished by MCD where $ACC = 0.7337$, $TPR = 0.2103$, $FPR = 0.1351$ and $F1 = 0.2403$.

In a similar manner, Table VII reflects the evaluation results of $MENSA$ for distinguishing the Modbus/TCP cyberattacks emulated in the SG lab. Based on the comparative results, $MENSA$ overcomes the other ML and DL solutions since its ACC, TPR, FPR and the $F1$ $Score$ reach 0.964, 0.7307, 0.0192 and 0.7307. On the other side, the lowest performance is accomplished by AdaBoost, where $ACC = 0.9111$, $TPR = 0.3333$, $FPR = 0.0476$ and $F1 = 0.3333$. Accordingly, Table VIII includes the evaluation results for discriminating the Modbus/TCP cyberattacks in the substation environment. Again, $MENSA$ achieves the best performance, where $ACC = 0.9655$,$TPR = 0.7591$, $FPR = 0.0185$ and $F1 = 0.7591$. Moreover, as previously, AdaBoost shows the worst performance, where $ACC = 0.9183$,$TPR = 0.4281$, $FPR = 0.0439$ and $F1 = 0.4281$. In the same SG environment, Table XI shows the efficiency of $MENSA$ against the DNP3 cyberattacks. $MENSA$ exceeds the performance of the other solutions, while the lowest efficiency is accomplished by Quadratic Discriminant Analysis. Table IX presents the evaluation results related to classifying the Modbus/TCP cyberattacks of Table I in the hydropower plant environment. Similarly, $MENSA$ achieves the best outcome, where $ACC = 0.9668$,$TPR = 0.7679$, $FPR = 0.0178$ and $F1 = 0.7679$. In this case, Adaboost achieves even lower evaluation results compared to the previous environments, where $ACC = 0.8877$,$TPR = 0.2142$, $FPR = 0.0604$ and $F1 = 0.2142$. Finally, Table X illustrates the evaluation results of $MENSA$ for discriminating the Modbus/TCP cyberattacks in the power plant environment. Also, in this case, $MENSA$ accomplishes the best outcome where ACC, TPRm, FPR and the $F1$ $Score$ reach 0.9646, 0.7349, 0.0189 and 0.7349. On the other hand, AdaBoost shows again the worst results,

TABLE IV: MENSA evaluation results for detecting operational anomalies in the first SG environment - SG Lab.

Model	ACC	TPR	FPR	F1
ABOD	0.692	0.989	0.397	0.600
Iforest	0.813	0.960	0.231	0.705
PCA	0.851	0.982	0.187	0.755
LOF	0.829	0.992	0.220	0.730
MCD	0.715	0.299	0.158	0.329
DIDEROT Autoencoder	0.851	0.982	0.188	0.755
ARIES GAN	0.930	0.875	0.053	0.853
MENSA	**0.964**	**0.941**	**0.028**	**0.925**

TABLE V: MENSA evaluation results for detecting operational anomalies in the second SG environment - substation.

Model	ACC	TPR	FPR	F1
ABOD	0.839	0.995	0.200	0.713
Iforest	0.850	0.951	0.175	0.718
PCA	0.847	0.961	0.181	0.716
LOF	**0.873**	**0.993**	**0.1571**	**0.759**
MCD	0.822	0.991	0.220	0.691
DIDEROT Autoencoder	0.840	0.961	0.189	0.708
ARIES GAN	0.834	0.653	0.120	0.613
MENSA	0.881	0.716	0.077	0.707

TABLE VI: MENSA evaluation results for detecting operational anomalies in the third SG environment - hydropower plant.

Model	ACC	TPR	FPR	F1
ABOD	0.581	0.993	0.522	0.487
Iforest	0.716	0.948	0.341	0.572
PCA	0.745	0.978	0.312	0.606
LOF	0.579	0.996	0.525	0.486
MCD	0.733	0.210	0.135	0.240
DIDEROT Autoencoder	0.746	0.978	0.311	0.607
ARIES GAN	0.817	0.966	0.219	0.680
MENSA	**0.883**	**0.871**	**0.113**	**0.749**

TABLE VII: MENSA evaluation results for classifying Modbus/TCP cyberattacks in the first SG environment - SG Lab.

Model	ACC	TPR	FPR	F1
Logistic Regression	0.945	0.588	0.029	0.588
LDA	0.937	0.529	0.033	0.529
Decision Tree Classifier	0.960	0.706	0.020	0.706
Gaussian NB	0.941	0.563	0.031	0.563
SVM RBF	0.931	0.485	0.036	0.485
SVM Linear	0.932	0.494	0.036	0.494
Random Forest	0.945	0.588	0.029	0.588
MLP	0.941	0.559	0.031	0.559
AdaBoost	0.911	0.333	0.047	0.333
Quadratic Discriminant Analysis	0.937	0.528	0.033	0.528
Dense DNN ReLU	0.943	0.578	0.030	0.578
Dense DNN Tanh	0.940	0.552	0.031	0.552
BlackBox IDS	0.948	0.612	0.027	0.601
Suricata	0.839	0.664	0.000	0.798
MENSA	**0.964**	**0.730**	**0.019**	**0.730**

TABLE VIII: MENSA evaluation results for classifying Modbus/TCP cyberattacks in the second SG environment - substation.

Model	ACC	TPR	FPR	F1
Logistic Regression	0.944	0.614	0.029	0.614
LDA	0.944	0.608	0.030	0.608
Decision Tree Classifier	0.964	0.749	0.019	0.749
Gaussian NB	0.938	0.566	0.033	0.566
SVM RBF	0.931	0.518	0.037	0.518
SVM Linear	0.930	0.511	0.037	0.511
Random Forest	0.947	0.631	0.028	0.631
MLP	0.940	0.584	0.031	0.584
AdaBoost	0.918	0.428	0.043	0.428
Quadratic Discriminant Analysis	0.944	0.613	0.029	0.613
Dense DNN ReLU	0.945	0.619	0.029	0.619
Dense DNN Tanh	0.944	0.611	0.029	0.611
BlackBox IDS	0.948	0.948	0.027	0.633
Suricata	0.839	0.664	0.000	0.798
MENSA	**0.965**	**0.759**	**0.018**	**0.759**

TABLE IX: MENSA evaluation results for classifying Modbus/TCP cyberattacks in the third SG environment - hydropower plant

Model	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
Gaussian NB	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
BlackBox IDS	0.948	0.641	0.029	0.630
Suricata	0.839	0.664	0.000	0.798
MENSA	**0.966**	**0.767**	**0.017**	**0.767**

where $ACC = 0.9111$,$TPR = 0.333$, $FPR = 0.0476$ and $F1 = 0.3333$.

Even though the data samples per case are morphologically similar, they differ in various ways, such as the features, the values magnitude and the sparsity. Thus, it is impracticable

TABLE X: MENSA evaluation results for classifying Modbus/TCP cyberattacks in the fourth SG environment - power plant

Model	ACC	TPR	FPR	F1
Logistic Regression	0.946	0.597	0.028	0.597
LDA	0.939	0.548	0.032	0.548
Decision Tree Classifier	0.960	0.703	0.021	0.703
Gaussian NB	0.942	0.565	0.031	0.565
SVM RBF	0.929	0.468	0.037	0.468
SVM Linear	0.933	0.502	0.035	0.502
Random Forest	0.949	0.623	0.026	0.623
MLP	0.941	0.562	0.031	0.562
AdaBoost	0.911	0.333	0.047	0.333
Quadratic Discriminant Analysis	0.937	0.529	0.033	0.529
Dense DNN ReLU	0.948	0.616	0.027	0.616
Dense DNN Tanh	0.940	0.551	0.032	0.551
BlackBox IDS	0.940	0.611	0.027	0.604
Suricata	0.839	0.664	0.000	0.798
MENSA	**0.964**	**0.734**	**0.018**	**0.734**

TABLE XI: MENSA evaluation results for classifying DNP3 cyberattacks in the second SG environment - substation

Model	ACC	TPR	FPR	F1
Logistic Regression	0.907	0.722	0.055	0.722
LDA	0.896	0.688	0.062	0.688
Decision Tree Classifier	0.977	0.991	0.001	0.991
Gaussian NB	0.910	0.731	0.053	0.731
SVM RBF	0.864	0.592	0.081	0.592
SVM Linear	0.893	0.680	0.063	0.680
Random Forest	0.931	0.733	0.053	0.733
MLP	0.911	0.733	0.053	0.733
AdaBoost	0.798	0.396	0.120	0.396
Quadratic Discriminant Analysis	0.722	0.166	0.166	0.166
Dense DNN ReLU	0.941	0.823	0.035	0.823
Dense DNN Tanh	0.932	0.797	0.040	0.797
BlackBox IDS	0.965	0.896	0.021	0.895
Suricata	0.795	0.636	0.000	0.777
MENSA	**0.994**	**0.983**	**0.003**	**0.983**

to formulate a model using standard hyperparameters per case. In contrast, each case is optimised experimentally. To evaluate the $MENSA$ performance in terms of the various hyperparameters, two evaluation metrics are utilised: (a) the F1 score variation per threshold t and (b) the F1 score saturation curve per iteration. Both cases aim to maximise the F1 score. In Fig. 8, the behaviour of the F1 score is depicted for four different experiments. As illustrated, the F1 score changes exponentially after a value of t. In particular, this value describes the optimal threshold leading to the most efficient discrimination between the normal and the abnormal instances. After this value, the F1 score saturates completely. Subsequently, Fig. 9 shows how the F1 score is improved based on the number of epochs. For the first iterations, the F1 score increases exponentially. Next, it saturates slowly for the rest of the training process. When the curve starts to flatten, the training is stopped to avoid overfitting and memorisation. Thus, the best checkpoint is selected. Finally, regarding the batch size, the larger the number of features, the higher the batch scaling in $2^a, a \in \mathbb{N}^*$, while the learning rate of each optimizer is kept in the scale $1/1000$.

Therefore, according to Tables IV- X, almost in all SG

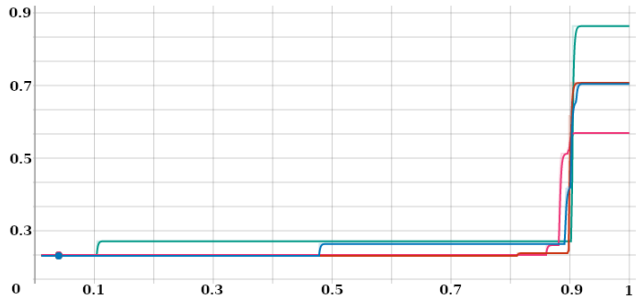

Fig. 8: F1-Score variation through the change of t

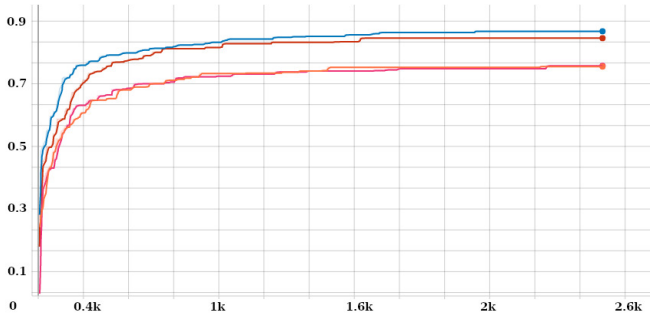

Fig. 9: F1-Score Saturation Curve

environments $MENSA$ achieves the best performance either for detecting operational anomalies or discriminating the Modbus/TCP and DNP3 cyberattacks. In general, a high TPR and low FPR can be observed. This is due to $MENSA$ dynamic deep threshold discovery. Usually, ML and DL classifiers use a threshold to provide the optimal outcome. Since $MENSA$ is designed to be adaptable for each SG environment and type of data, $MENSA$ dynamically calculates the appropriate threshold during the training process, thus achieving the best detection results. To this end, a brute force approach is utilised. It is noteworthy that $MENSA$ overcomes other advanced DL solutions, such as ARIES GAN [48] and BlackBox IDS [62] for anomaly detection and anomaly classification, respectively. Moreover, $MENSA$ exceeds the efficiency of `Suricata` since the existing Quickdraw ICS IDS signatures [71] do not cover all possible intrusions related to the Modbus/TCP and DNP3 payloads. In addition, the TCP/IP network flow statistics generated by `CICFlowMeter` render $MENSA$ a scalable solution for detecting and classifying anomalies for other application-layer protocols, such as IEC 60870-5-104, Message Queuing Telemetry Transport (MQTT) and IEC 61850 Manufacturing Message Specification (MMS). Finally, the successful anomaly detection against different kinds of operational data demonstrates the $MENSA$ scalability.

VII. Conclusion

The next generation EG, commonly called SG, creates significant advantages and challenges in society. On the one side, valuable services are already provided, such as the two-way power flow and self-monitoring, but on the other side, new cybersecurity concerns are generated. It is worth mentioning that the interconnected nature of the SG ecosystem also affects the safety status of other CIs. Therefore, the presence of novel intrusion and anomaly detection mechanisms and eliminating FP and FN are necessary. The ML and DL solutions compose valuable mechanisms capable of detecting zero-day attacks.

In this paper, we implemented an anomaly detection and classification model capable of detecting 13 Modbus/TCP cyberattacks, 5 DNP3 cyberattacks and potential anomalies related to operational data (i.e., time-series electricity measurements). The proposed model called $MENSA$ combines two DNNs: (a) Autoencoder and (b) GAN in a prototype architecture, which applies a novel minimisation function, taking into account (a) the adversarial error and (b) the reconstruction difference. The efficiency of MENSA was validated and evaluated in four SG evaluation environments: (a) SG lab, (b) substation, (c) hydropower plant and (d) power plant. To this end, other ML and DL methods were also adopted.

Our future plans in this field include the implementation of other DL models in order to detect cyberattacks against other ICS/SCADA protocols, such as Profinet and EtherCat. Moreover, sufficient association rules will be examined to correlate the outcome of these DL models with each other. Finally, optimisation solutions mitigating sufficiently such cyberattacks in CIs will be investigated.

VIII. Acknowledgment

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 787011 (SPEAR).

References

[1] S. Tan, D. De, W.-Z. Song, J. Yang, and S. K. Das, "Survey of security advances in smart grid: A data driven approach," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 397–422, 2017.

[2] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A survey on scada systems: Secure protocols, incidents, threats and tactics," *IEEE Communications Surveys & Tutorials*, 2020.

[3] M. De Vivo, G. O. de Vivo, R. Koeneke, and G. Isern, "Internet vulnerabilities related to tcp/ip and t/tcp," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 1, pp. 81–85, 1999.

[4] R.-G. Panagiotis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the internet of things: Challenges, threats and solutions," *Internet of Things*, vol. 5, pp. 41–70, 2019.

[5] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[6] S. Kwon, H. Yoo, and T. Shon, "Ieee 1815.1-based power system security with bidirectional rnn-based network anomalous attack detection for cyber-physical system," *IEEE Access*, vol. 8, pp. 77 572–77 586, 2020.

[7] B. Bencsáth, G. Pék, L. Buttyán, and M. Felegyhazi, "The cousins of stuxnet: Duqu, flame, and gauss," *Future Internet*, vol. 4, no. 4, pp. 971–1003, 2012.

[8] A. Di Pinto, Y. Dragoni, and A. Carcano, "Triton: The first ics cyber attack on safety instrument systems," in *Proc. Black Hat USA*, 2018, pp. 1–26.

[9] J. Sakhnini, H. Karimipour, A. Dehghantanha, R. M. Parizi, and G. Srivastava, "Security aspects of internet of things aided smart grids: A bibliometric survey," *Internet of things*, p. 100111, 2019.

[10] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, "Smart grid metering networks: A survey on security, privacy and open research issues," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.

[11] A. Ghosal and M. Conti, "Key management systems for smart grid advanced metering infrastructure: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2831–2848, 2019.

[12] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, 2020.

[13] R. Leszczyna, "Cybersecurity and privacy in standards for smart grids–a comprehensive survey," *Computer Standards & Interfaces*, vol. 56, pp. 62–73, 2018.

[14] S. M. S. Hussain, T. S. Ustun, and A. Kalam, "A review of iec 62351 security mechanisms for iec 61850 message exchanges," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5643–5654, 2020.

[15] T. S. Ustun and S. S. Hussain, "Iec 62351-4 security implementations for iec 61850 mms messages," *IEEE Access*, 2020.

[16] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems," *IEEE Access*, vol. 7, pp. 46 595–46 620, 2019.

[17] S. V. B. Rakas, M. D. Stojanovi, and J. D. Markovi-Petrovi, "A review of research work on network-based scada intrusion detection systems," *IEEE Access*, vol. 8, pp. 93 083–93 108, 2020.

[18] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 1–29, 2014.

[19] S. Ghosh and S. Sampalli, "A survey of security in scada networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135 812–135 831, 2019.

[20] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Liatifis, T. Apostolakos, and S. Oikonomou, "An overview of the firewall systems in the smart grid paradigm," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, 2018, pp. 1–4.

[21] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A survey on scada systems: Secure protocols, incidents, threats and tactics," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020.

[22] S. V. B. Rakas, M. D. Stojanović, and J. D. Marković-Petrović, "A review of research work on network-based scada intrusion detection systems," *IEEE Access*, vol. 8, pp. 93 083–93 108, 2020.

[23] G. Efstathopoulos, P. Radoglou-Grammatikis, P. Sarigiannidis, V. Argyriou, A. Sarigiannidis, K. Stamatakis, M. K. Angelopoulos, and 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)*, 2019, pp. 1–6.

[24] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "Attack graph-based moving target defense in software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1653–1668, 2020.

[25] I. Hafeez, M. Antikainen, A. Y. Ding, and S. Tarkoma, "Iot-keeper: Detecting malicious iot network activity using online traffic analysis at the edge," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 45–59, 2020.

[26] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 924–935, 2019.

[27] T. V. Phan, T. G. Nguyen, N.-N. Dao, T. T. Huong, N. H. Thanh, and T. Bauschert, "Deepguard: Efficient anomaly detection in sdn with fine-grained traffic flow monitoring," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1349–1362, 2020.

[28] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del Rincon, and D. Siracusa, "Lucid: A practical, lightweight deep learning solution for ddos attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876–889, 2020.

[29] C. Yinka-Banjo and O.-A. Ugot, "A review of generative adversarial networks and its application in cybersecurity," *Artificial Intelligence Review*, pp. 1–16, 2019.

[30] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Malware detection using deep transferred generative adversarial networks," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 556–564.

[31] A. Arora and Shantanu, "A review on application of gans in cybersecurity domain," *IETE Technical Review*, pp. 1–9, 2020.

[32] P. Dixit and S. Silakari, "Deep learning algorithms for cybersecurity applications: A technological and status review," *Computer Science Review*, vol. 39, p. 100317, 2021.

[33] S.-C. Li, Y. Huang, B.-C. Tai, and C.-T. Lin, "Using data mining methods to detect simulated intrusions on a modbus network," in *2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2)*. IEEE, 2017, pp. 143–148.

[34] W. Yusheng, F. Kefeng, L. Yingxu, L. Zenghui, Z. Ruikang, Y. Xiangzhen, and L. Lin, "Intrusion detection of industrial control system based on modbus tcp protocol," in *2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)*. IEEE, 2017, pp. 156–162.

[35] I. N. Fovino, A. Carcano, T. D. L. Murel, A. Trombetta, and M. Masera, "Modbus/dnp3 state-based intrusion detection system," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 2010, pp. 729–736.

[36] T. Morris, R. Vaughn, and Y. Dandass, "A retrofit network intrusion detection system for modbus rtu and ascii industrial control systems," in *2012 45th Hawaii International Conference on System Sciences*. IEEE, 2012, pp. 2338–2345.

[37] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, 2019.

[38] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, 2019.

[39] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[40] R. Shire, S. Shiaeles, K. Bendiab, B. Ghita, and N. Kolokotronis, "Malware squid: a novel iot malware traffic analysis framework using convolutional neural network and binary visualisation," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, 2019, pp. 65–76.

[41] binvis.io, "binvis.io," 2018. [Online]. Available: https://binvis.io/

[42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[43] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, 2017.

[44] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting iot cyber attacks using network traffic," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8852–8859, 2020.

[45] P. Simoes, "Denial of service attacks: Detecting the frailties of machine learning algorithms in the classification process," in *Critical Information Infrastructures Security: 13th International Conference, CRITIS 2018, Kaunas, Lithuania, September 24-26, 2018, Revised Selected Papers*, vol. 11260. Springer, 2019, p. 230.

[46] H. Yang, L. Cheng, and M. C. Chuah, "Deep-learning-based network intrusion detection for scada systems," in *2019 IEEE Conference on Communications and Network Security (CNS)*, 2019, pp. 1–7.

[47] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, P.-A. Karypidis, and 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*, 2020, pp. 1–8.

[48] P. Radoglou Grammatikis, P. Sarigiannidis, G. Efstathopoulos, and E. Panaousis, "Aries: A novel multivariate intrusion detection system for smart grid," *Sensors*, vol. 20, no. 18, p. 5305, 2020.

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

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

[51] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, 10 2017.

[52] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial nets and its variants work: An overview of gan," *ACM Computing Surveys*, vol. 52, 11 2017.

[53] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *ArXiv*, vol. abs/1212.5701, 2012.

[54] P. Radoglou-Grammatikis, P. Sarigiannidis, A. Sarigiannidis, D. Margounakis, A. Tsiakalos, and G. Efstathopoulos, "An anomaly detection mechanism for iec 60870-5-104," in *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 2020, pp. 1–4.

[55] 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, and F. Ramos, "Secure and private smart grid: The spear architecture," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, 2020, pp. 450–456.

[56] N. Pham and R. Pagh, "A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2012.

[57] X. Li, J. C. Lv, and D. Cheng, "Angle-based outlier detection algorithm with more stable relationships," in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1*, H. Handa, H. Ishibuchi, Y.-S. Ong, and K. C. Tan, Eds. Cham: Springer International Publishing, 2015, pp. 433–446.

[58] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[59] M. ling Shyu, S. ching Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03*, 2003, pp. 172–179.

[60] P. Rousseeuw and K. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, pp. 212–223, 08 1999.

[61] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in *2007 IEEE Symposium on Computational Intelligence and Data Mining*, 2007, pp. 504–515.

[62] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," *arXiv preprint arXiv:1809.02077*, 2018.

[63] Y. Wang, "A multinomial logistic regression modeling approach for anomaly intrusion detection," *Computers and Security*, vol. 24, pp. 662–674, 11 2005.

[64] K. Kim, H. Choi, C. Moon, and C.-W. Mun, "Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions," *Current Applied Physics - CURR APPL PHYS*, vol. 11, pp. 740–745, 05 2011.

[65] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 2451–2455.

[66] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, p. 516, Oct. 2006. [Online]. Available: https://doi.org/10.1145/1163593.1163596

[67] Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1, 2017, pp. 635–638.

[68] P. I. Radoglou-Grammatikis and 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.

[69] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection." *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 38, pp. 577–583, 04 2008.

[70] S. Naseer, D. Y. Saleem, S. Khalid, M. Khawar, J. Han, M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, pp. 1–1, 08 2018.

[71] K. Wong, C. Dillabaugh, N. Seddigh, and B. Nandy, "Enhancing suricata intrusion detection system for cyber security in scada networks," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–5.

Ilias Siniosoglou received his Diploma degree (5 years) from the Dept. of Electrical and Computer Eng., University of Western Macedonia, Greece, in 2020. He is now a Ph.D.student in the same department. His main area of research is Deep and Federated Learning on Next Generation IoT platforms, primarily focusing on optimization, deployment and scalability methodologies. Currently, he is working as a research associate at the University of Western Macedonia in national and European funded research projects, including (a) H2020-DS-SC7-2017 (DS-07-2017), SPEAR: Secure and PrivatE smArt gRid, (b) H2020-SU-DS-2018 (SU-DS04-2018), SDN-microSENSE: SDN-microgrid reSilient Electrical eNergy SystEm, (c) MARS: sMart fArming with dRoneS (Competitiveness, Entrepreneurship, and Innovation) and (d) H2020-ICT-2020-1 (ICT-56-2020) TERMINET: nexT gEneRation sMart INterconnectEd ioT.

Panagiotis Radoglou-Grammatikis received the Diploma degree (MEng, 5 years) from the Dept. of Informatics and Telecommunications Eng. (now Dept. of Electrical and Computer Eng.), Faculty of Eng., University of Western Macedonia, Greece, in 2016. He is now a PhD candidate in the same department. His main research interests are in the area of cybersecurity and mainly focus on intrusion detection, vulnerability research and applied cryptography. He has published 18 research papers in international scientific journals, conferences and book chapters, including IEEE Access, Computer Networks (ELSEVIER), Internet of Things (ELSEVIER) and Sensors (MDPI). Moreover, he received the Best Paper award in 2019 IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (IEEE CAMAD). He has served as a reviewer for several scientific journals and possesses working experience as a security engineer and software developer. Currently, he is working as a research associate at the University of Western Macedonia in national and European funded research projects, including (a) H2020-DS-SC7-2017 (DS-07-2017), SPEAR: Secure and PrivatE smArt gRid, (b) H2020-SU-DS-2018 (SU-DS04-2018), SDN-microSENSE: SDN-microgrid reSilient Electrical eNergy SystEm, (c) MARS: sMart fArming with dRoneS (Competitiveness, Entrepreneurship, and Innovation), (d) H2020-ICT-2020-1 (ICT-56-2020) TERMINET: nexT gEneRation sMart INterconnectEd ioT and (e) H2020-LC-SC3-EE-2020-1 (LC-SC3-EC-4-2020) EVIDENT: bEhaVioral Insgihts anD Effective eNergy policy acTions. Finally, he is a member of the IEEE and the Technical Chamber of Greece.

Georgios Efstathopoulos studied in National Technical University of Athens, where he received the Diploma of Electrical and Computer Engineer with distinction. He received his PhD degree under the supervision of Professor A. Manikas in the Communications and Signal Processing Group, Department of Electrical and Electronic Engineering, Imperial College London. Also, he worked as software developer and quantitative analyst at the Investment Bank sector. He has been working as a quantitative analyst in the nancial sector for the last 9 years. Over the last 3 years, Georgios has been actively involved in a number of data analytics, machine learning and AI projects in various industries, which includes autonomous vehicles, nance, smart grid, insurance and healthcare sectors.

Dr. Panayotis Fouliras is Assistant Professor at the Department of Applied Informatics at the University of Macedonia, Thessaloniki, Greece. He obtained his B.Sc. in Physics (Aristotle University of Thessaloniki, Greece), M.Sc. and Ph.D in Computer Science from University of London, UK (QMW). His research interests span computer networks, QoS, multimedia and system evaluation methods.

Prof. Panagiotis Sarigiannidis is an Associate Professor in the Department of Electrical and Computer Engineering in the University of Western Macedonia, Kozani, Greece since 2016. He received the B.Sc. and Ph.D. degrees in computer science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2007, respectively. He has published over 180 papers in international journals, conferences and book chapters, including IEEE Communications Surveys and Tutorials, IEEE Transactions on Communications, IEEE Internet of Things, IEEE Transactions on Broadcasting, IEEE Systems Journal, IEEE Wireless Communications Magazine, IEEE/OSA Journal of Lightwave Technology, IEEE Access, and Computer Networks. He has been involved in several national, European and international projects. He is currently the project coordinator of three H2020 projects, namely a) H2020-DS-SC7-2017 (DS-07-2017), SPEAR: Secure and PrivatE smArt gRid, b) H2020-LC-SC3-EE-2020-1 (LC-SC3-EC-4-2020), EVIDENT: bEhaVioral Insgihts anD Effective eNergy policy acTions, and c) H2020-ICT-2020-1 (ICT-56-2020), TERMINET: nexT gEneRation sMart INterconnectEd ioT, while he coordinates the Operational Program MARS: sMart fArming with dRoneS (Competitiveness, Entrepreneurship, and Innovation) and the Erasmus+ KA2 ARRANGE-ICT: SmartROOT: Smart faRming innOvatiOn Training. He also serves as a principal investigator in the H2020-SU-DS-2018 (SU-DS04-2018), SDN-microSENSE: SDN-microgrid reSilient Electrical eNergy SystEm and in three Erasmus+ KA2: a) ARRANGE-ICT: pArtneRship foR AddressiNG mEgatrends in ICT, b) JAUNTY: Joint undergAduate coUrses for smart eNergy managemenT sYstems, and c) STRONG: advanced firST RespONders trainiG (Cooperation for Innovation and the Exchange of Good Practices). His research interests include telecommunication networks, internet of things and network security. He is an IEEE member and participates in the Editorial Boards of various journals, including International Journal of Communication Systems and EURASIP Journal on Wireless Communications and Networking.