ERENO: A Framework for Generating Realistic IEC–61850 Intrusion Detection Datasets for Smart Grids

Silvio Ereno Quincozes, Célio Albuquerque, Diego Passos, and Daniel Mossé

***Abstract*—Connected and digital electricity substations based on IEC–61850 standards enable novel applications. On the other hand, such connectivity also creates an extended attack surface. Therefore, Intrusion Detection Systems (IDSs) have become an essential component of safeguarding substations from malicious activities. However, in contrast to traditional information technology systems, there is a serious lack of realistic data for training, testing, and evaluating IDSs in smart grid scenarios. Many existing substation IDSs rely on datasets from other contexts or on proprietary datasets that do not allow reproducibility, validation, or performance comparison with competing algorithms. To address this issue, we propose the Efficacious Reproducer Engine for Network Operations (ERENO) synthetic traffic generation framework based on the IEC–61850 standard specifications. As an additional contribution, and as a proof-of-concept, we create and make available a suite of realistic IEC–61850 datasets that model 8 use cases, namely traffic for seven common attacks and one for normal network traffic. Based on those datasets, we further evaluate how enriched features combining raw data from the substation can significantly improve intrusion detection performance. Our results suggest that it can improve F1-Score up to 47.22% for masquerade attacks.**

***Index Terms*—Smart grid, cybersecurity, intrusion detection, datasets, substation, dataset generation.**

I. Introduction

DIGITAL substations based on IEC–61850 play a critical role in the electrical power grid, since they are responsible for splitting, transforming, and combining energy flows. With the advent of smart grids, the power grid infrastructure was integrated with communication networks and computing. This has brought many possibilities for novel applications such as automated data acquisition as well as remote control and monitoring of electrical infrastructure, services, and components [1]. However, despite these advantages, such integration poses numerous security challenges for smart grids [2], [3], [4].

Intrusion Detection Systems (IDSs) serve to identify malicious activities and mitigate the attacker's actions. In fact, IDSs are already widely deployed in traditional information technology (IT) systems for this purpose. However, since IEC–61850 introduced new protocols, such as Generic Object Oriented Substation Event (GOOSE) and Sampled Values (SV), specialized attacks targeting them generate different traffic and attack patterns. Attacks may include replay attacks [5], message injection [6], masquerade attacks [7], and DoS (Denial of Service) attacks [8]. Thus, IEC–61850-based IDSs require new signatures to train, test, validate, and evaluate their performance.

Unfortunately, as IDS research for substations is still at an early stage, there is a dearth of datasets, and many existing IDS proposals rely on data collected from scenarios other than IEC–61850-based substations [9], [10]. Some testbeds have been developed to enable data capturing in similar scenarios, such as the *Geek Lounge Lab* [11] in Argentina and the *Electric Power Intelligent Control* (EPIC) testbed [12] from the Singapore University of Technology and Design. Unfortunately, they do not cover GOOSE and SV protocols, which are prevalent for automated protection and control in digital substations used in smart grids [13].

In fact, the lack of realistic data is an important challenge that should be addressed to enable the advancement of more robust IDSs for modern connected substations. The few existing IDS proposals [4], [5], [14], [15], [16] for IEC–61850 protocols that include GOOSE and SV are typically assessed by using private data and, thus, it is not possible to globally evaluate nor validate the IDS proposals. Thus, most of the current proposals cannot have their performance reproduced or compared with each other.

In this work, we present the Efficacious Reproducer Engine for Network Operations (ERENO) dataset generation. The ERENO framework is derived from a thorough analysis of current attacks, normal, and faulty (e.g., any abnormal electric current, such as a short circuit) conditions, and allows for connection with their industry-standard traffic simulators. A very important contribution of the ERENO Framework, as well as the resulting generated dataset, is the possibility of developing an IDS to cross reference two IEC–61850 protocols: GOOSE and SV.

Manuscript received 12 November 2022; revised 15 September 2023; accepted 22 November 2023. Date of publication 4 December 2023; date of current version 11 July 2024. This work was supported in part by CAPES, CNPq, FAPERJ, and CGI/FAPESP. *(Corresponding author: Silvio Ereno Quincozes.)*

Silvio Ereno Quincozes is with Faculdade de Computação, Universidade Federal de Uberlândia, Monte Carmelo 38304-402, Brazil (e-mail: silvioquincozes@unipampa.edu.br).

Célio Albuquerque is with Computer Science Department, Universidade Federal Fluminense, Niterói 24220-900, Brazil (e-mail: celio@ic.uff.br).

Diego Passos is with Computer Science Department, Universidade Federal Fluminense, Niterói 24220-900, Brazil, and also with ISEL, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, 1549-020 Lisboa, Portugal (e-mail: dpassos@ic.uff.br).

Daniel Mossé is with the Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: mosse@pitt.edu).

Digital Object Identifier 10.1109/TDSC.2023.3336857

Our main contributions are summarized as follows:

- We present a study of the current attack scenarios targeting IEC–61850 systems (Section II-C);
- We propose the ERENO framework to generate realistic GOOSE and SV traffic features (Section IV);
- We model a real electrical transmission line to simulate normal and faulty electrical signals that serve as input to the ERENO framework (Section V-A);
- We implement 8 use cases (including 7 attack classes) on the ERENO framework, with which we generate the ERENO–IEC–61850 dataset suite with 69 features[1] (Sections V-B and V-C);
- We analyze which features can be used for detecting each attack class (Section VI-C);
- We analyze how *feature enrichment* can improve intrusion detection by using an IDS implementation based on machine learning (Section VI-D);

The remaining of this work is organized as follows. In Section II, we present a brief background of the IEC–61850 standard and attack models. We discuss the gaps of related works in Section III and propose the ERENO framework to address them in Section IV. As a proof of concept, we generate the ERENO–IEC–61850 dataset – details are given in Section V. The proposed datasets are analyzed in Section VI. Finally, in Section VII, we present our final remarks.

II. IEC–61850 Overview and Attack Models

The IEC–61850 standard [17] has the following goals: 1) interoperability, 2) long-term stability, and 3) simplified configuration through the Substation Configuration Language (SCL) and remote commands. Additionally, the standard provides a comprehensive data model to ensure compatibility across multiple power system device manufacturers [18]. Aside from the data format and interoperability aspects, the IEC–61850 standard specifies the possible physical topologies (e.g., ring topology, redundant LANs), network protocols (SV, GOOSE, MMS), and object modeling [17], [19].

Note that the IEC–61850 standard defines other protocols, but in this work, we focus on the SV and GOOSE protocols since they are those involved in substation protection functions and are most prone to attacks. For a more in-depth description of the standard, please refer to [20].

A. SV Protocol

The SV (Sampled Values) protocol is defined by the IEC–61850–9–2 [21] standard to enable digitized current and voltage samples to be transferred to *Intelligent Electronic Devices* (IEDs) using the Ethernet protocol through the Publish/Subscribe paradigm, where publisher devices send multicast messages to subscriber devices such as control and/or protection IEDs. Such measurements are collected through analog signals from electrical equipment and converted to digital signals by Merging Units (MUs) and transmitted to subscriber devices.

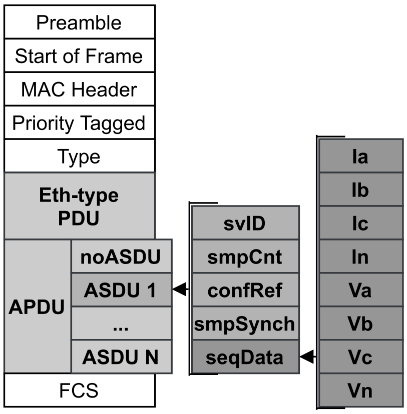

Fig. 1. SV Ethernet frame structure.

Protection applications consume SV messages to detect faults based on their protection schemes [4]. Therefore, to ensure a fast response time, SV messages are periodically sent at a high transmission rate both for protection and measurement purposes. Specifically, for protection applications, the standard defines the rate of 80 samples per cycle. Each cycle is 16.67 ms for substations operating at 60 Hz or 20 ms for substations operating at 50 Hz. For measurement applications that require more accurate data, the recommended rate is 256 samples per cycle [4], [22]. Voltage and current measurements are put into the *Application Service Data Unit* (ASDU) field of SV messages. Thus, the number of samples per cycle is proportional to the number of ASDUs transmitted in each SV message. Typically, one ASDU is sent in SV protection messages and eight ASDUs are sent in SV measurement messages [23]. Applications other than protection may contain multiple *Application Protocol Data Units* (APDUs) within the Sequence of ASDU fields. In Fig. 1, the Ethernet frame and the internal structure of the APDU of an SV message are illustrated. All shaded items are part of the SV message, the remaining items are generic Ethernet fields.

As shown in Fig. 1, each ASDU carries four current measures (I_a, I_b, I_c, I_n) and four voltage measurements (V_a, V_b, V_c, V_n) in the *Sequence of Data* (seqData) ASDU field, referring to the four electrical phases (A, B, C, and Neutral). These fields are the most sensitive target for attackers because they carry the domain-related physical data (e.g., fake measurement injection or data manipulation).

B. GOOSE Protocol

The GOOSE protocol enables IEDs to exchange messages to report substation events, such as status change notifications, alarms, and control commands. Events from various components are exchanged by GOOSE messages, including temperature alarms, circuit-breaker status, disconnector switch interlocking, etc. These data are put into a field named *GOOSE datSet* and transmitted using the publish/subscribe paradigm to a set of subscriber IEDs. Each IED may subscribe to specific topics related to its domain, such as control, protection, or measurement. Fig. 2 shows the GOOSE Ethernet frame structure.

[1] available publicly at https://github.com/sequincozes/ereno

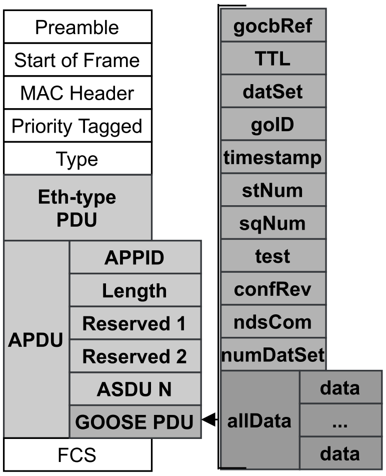

Fig. 2. GOOSE Ethernet frame structure.

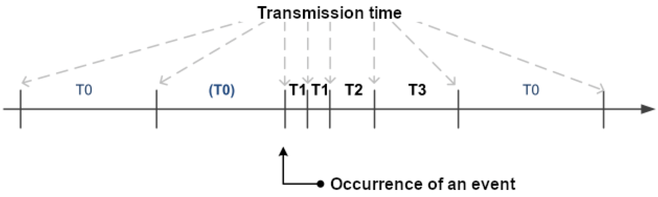

Fig. 3. GOOSE transmission intervals. Adapted from [8].

In stable situations, in which no new events occur (i.e., no changes are detected in the GOOSE dataset values), a GOOSE message is transmitted at a fixed T_0 interval, with an increased sequence number (*SqNum*). Once an event occurs, the *SqNum* field is set to zero, the status number (*StNum*) is increased and a new GOOSE message is sent immediately. This message is retransmitted at increasingly larger intervals, starting with the shorter transmission interval (T_1), which is used twice as a separator for the three first messages, and at increasing intervals for every retransmission thereafter (T_2, T_3, etc.), until reaching the original stable interval (T_0). This process is illustrated in Fig. 3. Although the precise values for the increments are not defined, exponentially growing intervals are typically adopted [8].

Therefore, by considering these standard behaviors, there are different features that may be analyzed by an IDS to distinguish between legitimate and malicious activities. For example, the GOOSE timestamp may reveal messages delayed by a DoS attack, since, in normal conditions, the interval between two received messages should not exceed T_0. SqNum and StNum are relevant features because they are potential indicators of fake message injection or message replay attacks [5], [7], [8].

C. Attack Models

This section describes the attack models used to generate malicious traffic. As our primary focus is on the protection function of IEDs, we focus on both GOOSE and SV traffic to reproduce the following attacks (see Section V):

Replay Attack captures and retransmits a previously sent GOOSE message – either immediately after the message is captured or after a longer delay. Replay attacks may be launched at multiple moments and they do not modify the original message content [1], [5].

Message Injection Attack builds and transmits fake and potentially malicious GOOSE messages into the network. In this model, attackers may perform either random modifications (i.e., without observing their consistency with the IEC–61850 standards) or standard-aware modifications (i.e., compliant with the IEC–61850 standards) [1], [24].

Masquerade Attack is a specialization of injection attacks [7]. Attackers learn from past GOOSE messages' content and mimic legitimate behavior by injecting fake messages that are harder to be detected than traditional injection attacks. This includes intelligently (not randomly) modifying message fields such as *StNum* and *SqNum* and injecting fake malicious state change events to cause the same behavior shift as a legitimate message [1], [7].

High-Status Number Attack is a DoS attack in which GOOSE messages are captured and sent with a higher `StNum` than the range of legitimate messages. The effect on the subscriber devices is to discard many subsequent legitimate GOOSE frames since their status number will appear to be outdated (lower than the fake `StNum`) [1], [24].

High-Rate Flooding Attack is another variant of DoS attacks, in which the attacker floods the multicast channel by sending multiple fake messages in short intervals (ahead of the normal traffic). Each fake message will increase StNum by 1, as expected at the subscriber devices. The result will be similar to the previous variant, except for their StNum and StNum which are consistent to the previous message and the potential of using the interval between two messages to help the detection of such attacks [1], [24].

Note that, as the industrial field evolves, novel vulnerabilities and threats can be explored by other attack models [25].

III. Related Work

The most popular available datasets for intrusion detection are designed considering traditional computer network traffic. They include the KDDCup99 [26], NSL-KDD [27], UNSW-NB15 [28], and CICIDS2017 [29]. Substation networks and systems use industrial protocols, such as GOOSE and SV. In this regard, a workflow for creating a synthesized intrusion dataset containing GOOSE traffic is defined by Biswas [13]. The authors propose the use of SCL files to get data models from IEDs and rely on power system simulators to generate substation traffic. An important remark is that the used tools (e.g., PowerWorld or DIgSILENT PowerFactory) are designed for the industry and are expensive, especially for research. In contrast, in this work, we propose an open-source tool to enable both researchers and industry professionals to set up the IED settings according to the desired scenario (more details will be given in the following sections). Moreover, the authors simply explore a few signatures of simple injection attacks by modifying the GOOSE *StNum*, *SqNum*, and a boolean value in the GOOSE data fields. In contrast, in our work we propose several novel features, as will be further discussed. Therefore, works in the literature are often based on generic traditional datasets or private datasets – as they do not make the data publicly available, their results are not reproducible. Below, we discuss those two approaches used for evaluating IEC–61850 IDS proposals in the literature.

A. Public Datasets

There are a group of works that propose IDSs for substations but assess them with generic/traditional traffic. An IDS to detect anomalous behavior in IEC–61850 networks is proposed by [9]. However, although they argue the network is IEC–61850-based, the studied attacks are generated through tools that target traditional protocols such as HTTP, FTP, and telnet, and not IEC–61850 protocols. Similarly, [10] used the KDD Cup 99' dataset with traditional traffic only to assess an IDS designed for IEC–61850 networks.

On the other hand, *Geek Lounge Lab* [11] and EPIC Lab [12] built testbeds for assessing industrial networks. Nevertheless, they did not consider scenarios of automated protection and control using GOOSE and SV protocols.

Therefore, the proposed IDSs proposed could not be assessed with the aforementioned resources in the face of the typical IEC–61850 attacks because there is no data about attacks involving these protocols.

B. Private Datasets

In [5], the authors use the classic IEEE 39-bus system [30] to simulate attacks for assessing a rule-based IDS. These rules aim to detect inconsistent messages according to predefined parameters. This work is extended in [16] to enable the detection of application-level attacks, considering IED logs as input data to detect malicious activities such as wrong password attempts and file system modifications. The authors cover a large range of attacks. In [6], a study suggests a method of anomaly detection for Manufacturing Message Specification (MMS) and GOOSE protocols. The one-class Support Vector Machine (SVM) algorithm is used to perform intrusion detection. However, the authors do not present any intrusion detection evaluation – only normal behavior is analyzed in their experiments. Therefore, their results are inconclusive and can not demonstrate the IDSs' ability to detect attacker behavior.

In [31], a photovoltaic inverter is used to compose a testbed for the evaluation of IDSs. Injection attacks are simulated through the injection of manipulated messages, containing false measurements. The analysis is based on an open-source IDS named Sucirata. However, they do not explore multiple attack classes, nor the use of novel features to correlate information between different protocols.

In [32], the authors present testing concepts and methods to assess MUs in accordance with protection application requirements. Thus, one goal of the authors is to check the SV messages' integrity. However, no advanced intrusion detection techniques nor features are used.

In [33], the authors simulate a smart digital substation to assess their proposed IDS. A number of rules were proposed to detect inconsistent behavior of SV, GOOSE, and MMS attacks in the generated traffic. The authors use the ITACA network analyzer tool and apply specification rules to detect intrusions. An extended version of [33] is presented in [14]. Both works depend on the proper rules specification, which requires expert domain knowledge.

In [4], a collaborative IDS is proposed to detect intrusions that may target multiple IEDs. The authors simulated a three-phase fault by using the Real-Time Digital Simulator (RTDS), which simulates MUs and circuit breakers. Besides RTDS, other elements were put together into a testbed to generate realistic traffic. Then, injection attacks were launched by modifying and retransmitting fabricated GOOSE packets to the network. They explore only one attack class (i.e., injection attacks).

In [34], the authors propose a behavior-based IDS. The proposed solution is evaluated through a testbed. The network traffic analyzed is based on the transmission of one week of captured network traffic from a digital substation. In addition, cyber-attacks were replicated. The authors claim that the lack of openly available intrusion detection datasets is a significant challenge. Nevertheless, they do not make their own data available.

In [35], the authors highlight the vulnerability of digital technology integrated into electrical substations, which are exposed to cyber attacks that can cause power outages. Based on the assumption that sufficient datasets are not readily available to train and evaluate the machine learning models, they propose using synthetic data generation with power grid domain knowledge to train robust machine learning models against different types of attacks. The results based on a single attack class (i.e., malicious command injection) show that ML-based attack detection can be significantly faster and more accurate than traditional, physics-based approaches. However, they did not consider generating traffic based on the IEC–61850 standards. Finally, there are no data available mentioned in their manuscript.

Hegazi et al. [36] described the modeling and generation of GOOSE communication traffic. However, since the security application was out of scope, the attacker model was not considered in their modeling.

Lopes et al. [37] proposed a GOOSE traffic generator for the evaluation and testing of security mechanisms. While their approach is similar to ours, the authors did not discuss realistic substation models or threat models. Thus, the practical use of the tool requires further design and understanding of the substation systems.

In the same vein, Blair et al. [38] proposed an open-source platform for rapid prototyping of GOOSEs and SV traffic without discussing the threat models. In contrast, we provide a framework to easily generate attack-free and attack traces for benchmarking security mechanisms developed by the research community.

The dataset is about an attacker who maliciously opens switches to create an invalid grid configuration state. The attacker strategically attacks circuit breakers to cut off power supply to certain sections of the network.

C. Discussion

Table I summarizes the related work, the used approaches, data sources, and evaluation methods. Unfortunately, the data

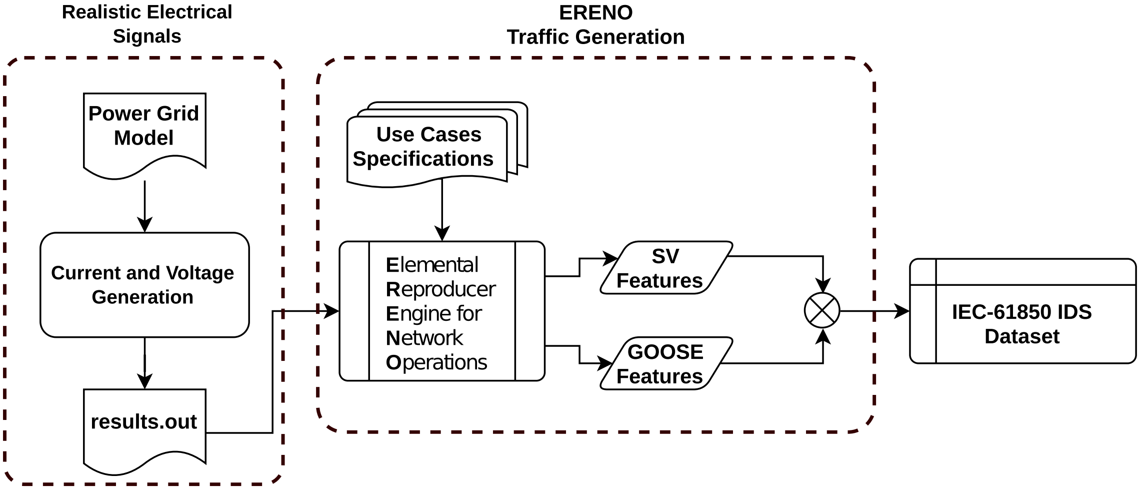

Fig. 4. Overview of the ERENO framework.

TABLE I
DATA USED BY THE CURRENT IDSS APPROACHES (ADAPTED FROM OUR PREVIOUS SURVEY [1])

Ref.	Year	Attacks	Protocols	Availability
[9]	2010	2	Generic	Public
[5]	2014	3	GOOSE, SV	Private
[16]	2014	5	GOOSE, SV	Private
[34]	2015	8	GOOSE, MMS	Private
[6]	2015	0	GOOSE, MMS	Private
[33]	2016	4	GOOSE, SV, MMS	Private
[31]	2016	1	MMS	Private
[14]	2017	2	GOOSE, SV, MMS	Private
[32]	2017	1	SV	Private
[10]	2019	5	Generic	Public
[13]	2019	4	Generic	Private
[4]	2019	1	GOOSE, SV	Private
[35]	2022	1	GOOSE, SV	Private
ERENO	**2023**	7	**GOOSE, SV**	**Public**

(or the framework used to generate the data) used for evaluating most of the aforementioned IDS proposals were not made available. Therefore, future works can not reproduce the results nor compare them with other proposals in the literature. Some IDS proposals are evaluated using public datasets to overcome these issues; however, these datasets are built upon traditional network traffic and typically do not include substation protocols, especially the IEC–61850 protocols.

Most existing IEC–61850 IDS proposals that include GOOSE/SV use data collected from private testbeds and are typically designed to detect specific attacks. Thus, considering multiples attack models and public datasets to feed dynamic techniques (e.g., machine learning) are key aspects for assessing IEC–61850 IDSs and ensuring their efficiency [39].

Another important point is that the literature does not consider intrusions in scenarios that combine the behavior from normal situations and electrical faults over the IEC–61850 networks. That behavior should also be considered to provide more realistic datasets.

Therefore, in the next sections, we present the ERENO Framework to address the gaps found.

IV. THE ERENO FRAMEWORK

To solve the lack of realistic intrusion detection datasets for digital substations, we propose the ERENO framework. The main goal of our framework is to reproduce the real behaviors of both legitimate and malicious users of digital substation networks. Since the digital substations are a Cyber-Physical System (CPS) application, we cover the substation data which comes from both cyber and physical domains. An overview of the ERENO framework is depicted in Fig. 4.

A. SV Features

To make realistic SV features, the physical data (e.g., current and voltage) used as a payload for the SV messages are acquired from an existing source. Such a source in ERENO may include simulated electrical signals or real data collected from the substation domain. This enables ERENO to be implemented according to the targeting scenario the user wishes to reproduce, which includes the setup of topology and corresponding devices. Besides, the SV data may contain both normal operation behaviors and electrical faults.

Based on the input electrical data, we propose novel features through an enrichment process. These methods are inspired by the substations' protection functions from IEDs. In particular, we employ the *Sum of the Trapezoid Area* (TrapAreaSum) and the *Root Mean Square* (RMS) methods for deriving novel SV features.

Our proposed SV features include:

- v_{phase}: the voltage value from a given device in a specific phase;
- i_{phase}: the electric current value from a given device in a specific phase;
- $TrapAreaSum_{phase}$: the Sum of the Trapezoid Area from the sinusoidal electrical signals (i_{phase} or v_{phase});
- RMS_{phase}: the Root Mean Square from the sinusoidal electrical signals (i_{phase} or v_{phase});

Note that each phase (*A, B, C, or N*) may have a different feature for each item above. Besides, these features may be collected from multiple sources, each generating an individual feature (e.g., two or more devices or substations may produce them). Finally, note that $TrapAreaSum_{phase}$ and RMS_{phase} may be computed for both current and voltage for any of the aforementioned sources. A complete list of SV features is given in our proof of concept, in Section V.

B. GOOSE Features

For generating realistic GOOSE features, it is necessary to consistently model the behavior of the devices that belong to the targeting scenario in which the ERENO framework is implemented. Note that ERENO is extensible and can support the design of multiple IEDs, according to the desired system topology. In a hypothetical system, where only legitimate actions occur, a realistic model would include both normal operation and faulty scenarios (i.e., containing reports of electrical fault events). However, in a typical vulnerable scenario (which is the most realistic) it is also necessary to predict and model the attacker's actions.

To reproduce the legitimate behavior of devices that communicate through the GOOSE protocol, a targeting application (e.g., protection application) should be considered and the transmitted messages (e.g., GOOSE messages reporting electrical faults) by the devices involved (e.g., protection IEDs) should be generated. As the expected behavior is well-defined through formal standards (i.e., IEC–61850-8-1 [40]), realistic reproduction of that behavior can be reached by following the protocol's specifications.

On the other hand, predicting the attackers' behavior is a more challenging task and cannot be generalized by following a single specification. Therefore, ERENO was designed to support the setting of custom attack use cases. Thus, based on any state-of-art intrusions, one can plug the attacker scripts into the ERENO framework.

Based on the resulting data acquired from legitimate and malicious GOOSE behavior, several features can be derived. Each field from the GOOSE protocol can be extracted and considered as a basic GOOSE feature. The basic GOOSE features listed below are the most likely to have inconsistent values when attacks occur, thus they are the most representative to IDSs:

- *stNum:* the status number of a GOOSE message;
- *sqNum:* the sequence number of a GOOSE message;
- *cbStatus:* the reported circuit-breaker status.

Besides the basic features, we propose novel enriched GOOSE features based on simple mathematical operations that aggregate information from different messages and reveal novel information (e.g., how much the sequence or status number was incremented from the latest message). A complete list of GOOSE features, comprising both basic and enriched ones, is given in our proof of concept, in Section V.

V. Proof of Concept: ERENO Dataset

As a proof of concept, we modeled a real electrical power grid using the PSCAD simulation tool [41]. To reproduce realistic data, our model includes both normal and faulty transmission line scenarios from the power grid. Based on these scenarios, we (i) extracted electrical measurements (left side of Fig. 4) to feed the ERENO framework and support the SV features generation, and (ii) reproduced the corresponding GOOSE behavior to report the fault events (center of Fig. 4). The SV and GOOSE feature generation are detailed in Sections V-A and V-B, respectively.

A. SV Feature Generation

The ERENO Framework takes the Power Grid Model output (`results.out` in Fig. 4) as a reference to build the substation's normal and faulty traffic. Note that faulty traffic is also benign traffic because it represents natural electrical faults rather than an attacker's action. The modeled power grid used in ERENO is based on a real transmission line in the Brazilian electrical power system which interconnects two substations, namely Serra da Mesa in the State of Goiás and Samambaia in the Federal District (see Fig. 5). The Power Systems Computer-Aided Design (PSCAD) [41] simulates electromagnetic transients to perform modeling and analysis of power systems, including steady-state and transient scenarios, as well as electrical faults. This tool also enables generating multiple fault profiles through parameter sweeps. These parameters include the fault location, resistance, and type.

The simulated transmission line has three phases, A, B, and C, and is 249 km long, with 12 segments of 20.75 km each. We created several fault scenarios to generate a large number of faulty and normal behavior data instances. Eleven fault types are applied along the transmission line in twelve different locations. These fault types include single-phase faults (i.e., AG, BG, and CG, where "G" denotes the ground), phase-to-phase faults and three-phase faults (i.e., AB, BC, CA, ABC, and ABCG). In addition, each fault case considers three different fault resistance values (i.e., 10 Ω, 50 Ω, 100 Ω). The used parameters are based on the contemporary literature [42].

Using this methodology, the ensuing non-attack samples are obtained by considering combinations of all parameter values, namely 12 fault locations, 3 fault resistances, and 11 fault types, or $12 \times 3 \times 11 = 396$ scenarios. Each simulation run lasts 1 s, of which 900 ms represent normal conditions and 100 ms represents a programmed fault starting at the fixed timestamp of 500 ms and lasting 100 ms. This fault duration is the maximum fault duration, based on the requirements specified by the ONS (*Operador Nacional do Sistema Elétrico*, the Brazilian electrical system operations organization) [43]. Since the IEC–61850-9-LE specifies 80 messages per cycle for protection applications, a 60 Hz system will require 4,800 messages per second. Therefore, to generate an equivalent (approximated) number of samples on PSCAD, we set the `plot step` parameter on PSCAD to 208 (i.e., one sample is collected every 208 μs).

The analog current and voltage values from a substation are extracted by the simulated Voltage Transformer (VT) and Current Transformer (CT) illustrated in Fig. 5, digitized by our traffic generator, and converted to SV features. These features are in compliance with the values from the SV message specified in the IEC–61850-9-LE standard. A complete list of the generated SV features is shown in Table II. Basic features (left column

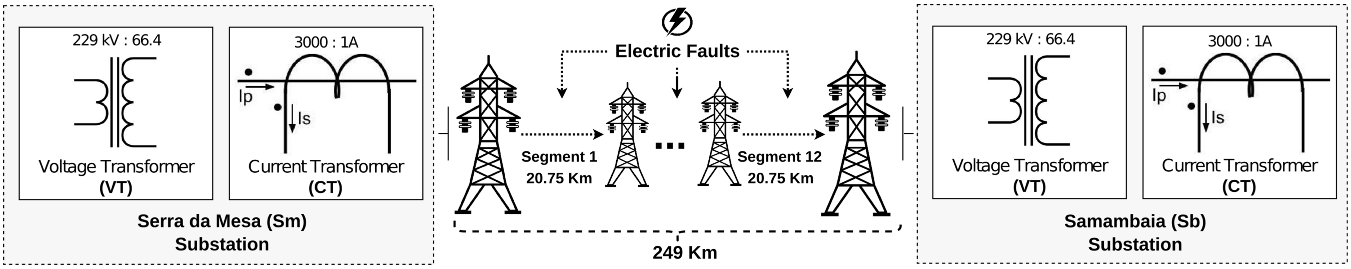

Fig. 5. Simulated electric power grid model.

TABLE II
SV Features Generated by ERENO

Basic Features {index}	Description	Enriched Features {index}	
		RMS Value	Trapezoidal Area Sum
time {1}	The SV timestamp.	-	-
isbA {2}	Current (i) from Samambaia (sb) at Phase A.	isbARmsValue {14}	isbATrapAreaSum {26}
isbB {3}	Current from Samambaia substation at Phase B.	isbBRmsValue {15}	isbBTrapAreaSum {27}
isbC {4}	Current from Samambaia at Phase C.	isbCRmsValue {16}	isbCTrapAreaSum {28}
ismA {5}	Current from Serra da Mesa (sm) at Phase A.	ismARmsValue {17}	ismATrapAreaSum {29}
ismB {6}	Current from Serra da Mesa at Phase B.	ismBRmsValue {18}	ismBTrapAreaSum {30}
ismC {7}	Current from Serra da Mesa at Phase C.	ismCRmsValue {19}	ismCTrapAreaSum {31}
vsbA {8}	Voltage (v) from Samambaia at Phase A.	vsbARmsValue {20}	vsbATrapAreaSum {32}
vsbB {9}	Voltage from Samambaia at Phase B.	vsbBRmsValue {21}	vsbBTrapAreaSum {33}
vsbC {10}	Voltage from Samambaia at Phase C.	vsbCRmsValue {22}	vsbCTrapAreaSum {34}
vsmA {11}	Voltage from Serra da Mesa at Phase A.	vsmARmsValue {23}	vsmATrapAreaSum {35}
vsmB {12}	Voltage from Serra da Mesa at Phase B.	vsmBRmsValue {24}	vsmBTrapAreaSum {36}
vsmC {13}	Voltage from Serra da Mesa at Phase C.	vsmCRmsValue {25}	vsmCTrapAreaSum {37}

of Table II), with indexes 1 to 13, are simple current (i) and voltage (v) measurements regarding each phase (*A, B, and C*) that come from either Samambaia (sb) or Serra da Mesa (sm) substations. Thus, the **i** measurement of Phase **A** from **sb** substation is represented as **isbA**, and so on. *Enriched* Features, with indexes 14 to 37, are computed through statistical methods to combine multiple readings of the same feature on different timestamps and generate more representative features. These features are inspired by the rules used by specification-based IDSs from the literature (see Table I). We explore two methods to derive novel features: *Trapezoidal Area Sum* (TrapAreaSum) [44] (for features with index 14 to 25) and *Root Mean Square* (RmsValue) [45] (for features with index 26 to 37). These methods are commonly used to detect faults in a power transmission line. Besides the generated features for this proof of concept, other features may be extracted.

Therefore, the process of choosing the GOOSE and SV features to include in our dataset involved leveraging domain expert knowledge and conducting a thorough literature review of specification-based IDSs.

TABLE III
Basic GOOSE Features

Features {Index}	Value
t {38}	The timestamp of the last state change;
gTimestamp {39}	The GOOSE timestamp;
sqNum {40}	The GOOSE sequence number;
stNum {41}	The GOOSE status number;
cbStatus {42}	Circuit-breaker status on GOOSE;
frameLen {43}	The GOOSE ethernet frame length;
ethDst {44}	The GOOSE ethernet destination address;
ethSrc {45}	The GOOSE ethernet frame source address;
ethType {46}	The GOOSE ethernet frame type;
gooseTTL {47}	The time allowed to live;
gooseAppid {48}	The GOOSE application ID;
gooseLen {49}	The GOOSE frame length;
TPID {50}	The tag priority ID;
gocbRef {51}	The GOOSE control block reference;
datSet {52}	The IED dataset path;
goID {53}	The GOOSE flow ID;
test {54}	The test flag;
confRev {55}	The configuration revision;
ndsCom {56}	The "Needs Commissioning" parameter;
numDatSetEntries {57}	The number of entries on the datSet;
APDUSize {58}	The Application Data Unit (APDU) size;
protocol {59}	The used protocol (expected: GOOSE).

B. GOOSE Feature Generation

In this section, we describe a normal-traffic Use Case (UC00) in the modeled substation and 7 malicious use cases (UC01 to UC07). These last 7 use cases are based on the attack models presented in Section II-C. The generated GOOSE features are shown in Tables III (basic) and IV (enriched).

1) Normal Use Case (UC00): Based on the modeled normal operation with fault events, the ERENO Framework reproduces the expected GOOSE messages which report them. In substation protection applications, a GOOSE message may be related to the operation of circuit-breakers (i.e., opening or closing the circuit to isolate or energize the equipment). In this proof of concept, we focus on GOOSE messages coming from the Serra da Mesa (Sm) substation of Fig. 5. In our benign scenario (i.e., where no attacks occur), the GOOSE messages are transmitted every T_0 *milisseconds* with the boolean value that represents the circuit-breaker status set to a false, meaning a "closed" circuit-breaker. When an event occurs (e.g., a fault), a GOOSE

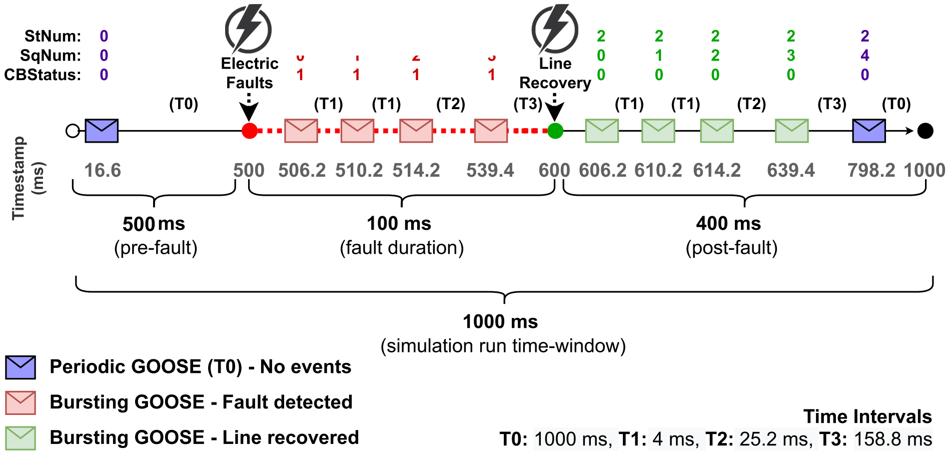

Fig. 6. Generated GOOSE messages per simulation run for UC00 (when no attacks are launched).

TABLE IV
ENRICHED GOOSE FEATURES

Features {Index}	Value
stDiff {60}	StNum{i} - StNum{i-1};
sqDiff {61}	SqNum{i} - SqNum{i-1};
gooseLengthDiff {62}	gooseLength{i} - gooseLength{i-1};
cbStatusDiff {63}	cbStatus{i} - cbStatus{i-1};
apduSizeDiff {64}	apduSize{i} - apduSize{i-1};
frameLengthDiff {65}	frameLength{i} - frameLength {i-1};
timestampDiff {66}	gTimestamp{i} - gTimestamp{i-1};
tDiff {67}	t{i} - t{i-1};
timeFromLastChange {68}	gTimestamp{i} - t{i};
delay {69}	gTimestamp{i} - time{i}.

message is sent immediately and retransmitted according to the behavior described in Section II-B. In real scenarios, both T_1 and T_0 are defined in the Substation Configuration Description (SCD) file through the `MinTime` and `MaxTime` tags. In our simulations, we assume these values as 4 ms and 1,000 ms, respectively, as in [46], [47]. We also assume that electrical fault events occur in the middle (at t=500 ms) of each 1-second simulation run. Despite the focus on investigating 1-second time intervals with an inherent electric fault occurring at the midpoint, our methodology is capable of producing extensive datasets of varying magnitudes and fault occurrences at any desired time instance. Remarkably, this study successfully generated a comprehensive dataset. The dataset comprises 5,914,230 records of combined SV message and GOOSE messages, representing samples collected at random instances within one-second windows, totaling around 20 minutes of traffic.

It is important to note that electric fault events will trigger a GOOSE message only after the IEDs' protection algorithm detects it (i.e., through processing the corresponding SV messages). We refer to this processing time as T_{IED}. We assume $T_{IED} = 6.20\tilde{}ms$, based on the time reported in [48]. Since we assumed *MaxTime* as 1,000 ms, no more than one GOOSE message per 1-second simulation run is expected under normal conditions (i.e., if no event occurs).

Fig. 6 depicts the transmitted messages for each 1-second simulation run. The first GOOSE message is transmitted at the beginning of each simulation run, at timestamp 16.6 ms to correspond to the end of the first SV cycle. As we simulated faults at 500 ms, the second message is transmitted after T_{IED} (i.e., at 506.20 ms), a third message is transmitted after T_1 (i.e., at 510.20 ms), and a fourth message is also transmitted after another T_1 (i.e., at 514.20 ms). Assuming an Exponential Backoff function[2] for message transmission, two more messages would be sent (i.e., at 539.4 ms and 698.2 ms) before the message interval reaches *T0* = 1,000 ms and returns to regular intervals. However, in our scenario, the fault is recovered at 600 ms (i.e., before the message programmed to time 698.2 ms). In this case, the previous burst of messages is canceled and a new GOOSE message burst reporting a status change to *normal operation* is sent immediately after this event is detected (i.e., at 606.2 ms). The bursting mode is restarted with the first message sent after T_1 (at 610.2 ms), the second message transmitted after T_1 again (i.e., at 614.2), the third message sent after T_2 (i.e., at 639.4 ms), and the last message sent after T_3 (i.e., at 798.2 ms) when the bursting mode finishes and regular messages would be sent every *T0*.

2) Random Replay Attacks (UC01): This use case is based on the capture of a previously sent message and retransmission at multiple random simulation moments (i.e., 1000 replay messages per second with random timestamps). Consequently, these replay attacks may have different impacts depending on the message content and context in which they are retransmitted, varying from power outages, equipment damages, or, in the worst case, offering risk to human life. Note that this model considers that the attacker does not modify the original message content, as described in Section II-C.

3) Inverse Replay Attacks (UC02): We propose a variation of UC01, named Inverse Replay Attacks: instead of performing random retransmissions, the attacker attempts to cause more damage by choosing to replay messages with a different status

[2] The Exponential Backoff function used is available online at http://backoffcalculator.com.

than the current one. This can also be seen as a special case of UC01.

To run this attack, the attacker captures previously sent GOOSE messages. Then, it monitors the network to identify specific events and attempts to report a fake status by injecting an old message with a different event. The malicious message may be retransmitted during, before, or after a fault event. Similarly to UC01, these attacks are launched at multiple simulation moments (i.e., 1000 replay messages per run with random timestamps but always in a planned time-window corresponding to the inverse circuit-breaker status). For example, if a fault event occurs, the attacker sends (an older) GOOSE message (i.e., with outdated/inverse circuit-breaker status) to the subscribers. The result is an undesired operation to close the circuit breaker, rather than opening it as expected in this fault scenario. In practice, this may damage equipment or jeopardize human life (e.g., by reestablishing a transmission line during maintenance). To cause the inverse result (i.e., improperly opening the circuit breaker), the attacker sends the malicious message during normal conditions.

4) Masquerade Attacks Towards Outage (UC03): The attacker transmits fabricated messages that mimic fault events under normal situations. This includes imitating the retransmission period between two messages (see T1, T2, and T3 after the occurrence of an event in Fig. 3), as well as *StNum* and *SqNum* field values, to cause the same behavior as the correspondent legitimate messages. As result, Masquerade GOOSE messages cannot be distinguished from legitimate messages by processing only their content.

In particular, the UC03 was set up to fabricate fake GOOSE events to report a fault during a normal line operation, aiming to cause an outage. Thus, whereas the SV payload corresponds to a normal line operation, the GOOSE messages carry a tampered control block status that reports a fault. For each 1-second simulation run, masquerade attacks are sent at random moments of normal operation.

5) Masquerade Attacks Towards Equipment Damage (UC04): This use case aims to reestablish the transmission line during an anomalous condition such as line maintenance. Thus, it sends malicious GOOSE messages reporting a normal status operation during a fault. The attacker transmits fabricated messages that also mimic the GOOSE bursts (see T1, T2, and T3 after the occurrence of an event in Fig. 3), *StNum* and *SqNum* field values to cause the same behavior as the correspondent legitimate messages.

This use case sends fake messages at random times within the time window between 500 ms and 600 ms of each simulation run (i.e., when a fault occurs). We set up ERENO to generate 1,000 messages per second.

6) Random Message Injection (UC05): This implementation of the Injection Attack assumes that the attacker is able to fabricate and transmit fake messages with either random modifications, without observing consistency with the IEC–61850 standards, and smart modifications that do not violate the IEC–61850 standards).

In this attack, an attacker just injects new fabricated messages without previously observing the network traffic to learn about its patterns. Thus, to send messages that do not violate the IEC–61850 standard, an attacker uses only the standards' specifications, not network traffic.

7) High Status Number Attack (UC06): This use case explores the setting of high – and inconsistent – values to `StNum`. To implement that behavior, GOOSE messages are captured and sent with their `StNum` set to be higher than the range of legitimate messages. In particular, we assume legitimate messages with a `StNum` lower than 10,000 and put random values (from 10,000 to 100,000) in this field. The current `StNum` range could be checked by the attacker through packet sniffing. Otherwise, an attacker can send messages with a very high `StNum` to attempt to guess a sufficiently large value.

The expected behavior on the subscriber devices is to discard any legitimate GOOSE frames since their status number will appear to be outdated (lower than expected).

8) High-Rate Flooding Attack (UC07): This use case explores DoS attacks, in which an attacker floods the multicast channel by sending multiple fake messages. We sent 1,000 messages per second. Each fake message will increase by one the status number expected at the subscriber devices. Instead of causing a DoS due to resource exhaustion, the UC07 deliberate change of the `StNum` field, which can cause the IED to discard legitimate messages and potentially disrupt the operation of the power grid.

C. Dataset and Source-Code

We made the ERENO Framework an open-source project at GitHub.[3] Additionally, we made the ERENO–IEC–61850 datasets publicly available at Kaggle.[4]

ERENO was set up to generate a suite containing 7 datasets, each covering one of the proposed attack use cases – the legitimate signatures (UC00) are present in all of them. The ERENO Framework is primarily designed to reproduce both malicious and benign substation traffic, thereby generating features that are suitable and ready to use for machine learning applications. While the generation of other output formats, such as Packet CAPture (PCAP) files, is conceptually supported by the framework, this functionality was not incorporated in our proof-of-concept implementation. Each dataset is in the Attribute-Relation File Format (ARFF) format and has 69 features (columns) and different numbers of attack and normal samples (rows). Information about each dataset is presented in Table V.

We made available already-separated training and testing data for each dataset. They have the same number of samples. Note that as an alternative to training and testing with separate datasets, one can join them and perform k-fold cross-validation. We do not recommend this because having samples within a small time window can introduce bias to the machine learning model. All results presented in this work can be reproduced using the available dataset and source code.

[3] [Online]. Available: https://github.com/sequincozes/iec61850generator

[4] [Online]. Available: https://www.kaggle.com/datasets/sequincozes/ereno-iec61850-ids

TABLE V
ATTACK AND SAMPLES DISTRIBUTION

Dataset	Samples	Attacks	Attacks transmission timestamp
UC01	410,397	39,000 (9.5%)	Randomly, either during a fault or normal operation.
UC02	557,076	30,319 (5.4%)	Randomly within an inverse cb-Status time window.
UC03	388,597	17,200 (4,4%)	Randomly, during a fault time window.
UC04	388,817	17,420 (4,4%)	Randomly, except during a fault time window.
UC05	410,397	39,000 (9,5%)	Completely random timestamps.
UC06	410,397	39,000 (9,5%)	Completely random timestamps.
UC07	389,967	18,570 (4,7%)	Completely random timestamps.

TABLE VI
FEATURE ENRICHMENT RESULTS FOR UC01

	Accuracy	Precision	Recall	F1-Score
GOOSE	97.44%	78.82%	99.95%	88.14%
GOOSE & SV	93.75%	60.32%	99.96%	75.23%
GOOSE & SV++	92.56%	56.08%	99.88%	71.83%
GOOSE++ & SV++	**100%**	**100%**	**99.99%**	**99.99%**

The bold values represent the higher values for each metric.

VI. ERENO DATASET ASSESSMENT

This section analyzes three main aspects of the generated datasets through the ERENO Framework. Our methodology is presented in Section VI-A. Then, in Section VI-B, we analyze how feature enrichment can improve the detection for each attack use case. In Section VI-C, we perform an analysis of the best features selected by the J48 classifier algorithm to compose its decision trees for each attack use case. Finally, in Section VI-D, we assess the impact of feature enrichment on decision tree generation.

A. Methodology

J48 is one of the most popular machine learning algorithms. It builds decision trees based on the Iterative Dichotomiser 3 algorithm [49]. We chose J48 for all analysis in this work because of its popularity and because it generates a single decision tree for each experiment, which enables us to *assess the features used to make decisions* through the generated trees. As J48 employs a pruning method, it generates trees with different features according to the attack use case experimented with. It is important to observe that our goal is not assessing the J48 performance itself. Rather, we aim at checking if the generated features can enable J48 to build consistent decision trees to make good decisions (e.g., without any bias).

Nevertheless, we can only assess the features added to datasets by checking the resulting performance of the IDS itself; for that, we computed four main metrics for J48-based IDS, used by most machine learning works: accuracy, F1-Score, recall, and precision. More details about these metrics can be found in our survey [1].

B. Feature Enrichment

In this section, we assess how feature enrichment can contribute to the IDSs' detection performance. We start by taking the basic GOOSE features (shown in Table III) as a baseline and then we perform incremental GOOSE and SV feature enrichment. In particular, we consider the following variations of the ERENO-IEC-61850 dataset suite:

- `GOOSE`: only the basic GOOSE features, presented in Section V-B. These are the features with indexes 38 to 59 in the ERENO-IEC-61850 dataset suite.
- `GOOSE & SV`: the basic GOOSE features (described above) and basic SV features presented in Section V-A. The SV features are those with indexes 1 to 13 in the ERENO-IEC-61850 dataset suite.
- `GOOSE & SV++`: the basic GOOSE and basic SV features (as in the previous item) with the addition of the enriched SV features presented in Section V-A. The enriched SV features are those with indexes 14 to 37 in the ERENO-IEC-61850 dataset suite.
- `GOOSE++ & SV++`: the basic and enriched features from both GOOSE and SV protocols (i.e., the features used for this experiment are all the 69 features).

Below we discuss the IDS results for each attack use case.

1) Random Replay Attacks (UC01): The detailed results for classifying UC01 are shown in Table VI. J48 reached an accuracy of 97.44% and a recall of 99.95% for detecting Random Replay attacks (UC01) with the 22 basic GOOSE features. However, its precision was only 78.82%, which leads to an F1-Score of 88.14%.

Neither the `GOOSE & SV` nor `GOOSE & SV++` features were able to enhance the IDS performance – they resulted in 75.23% and 71.83% F1-Score, respectively. These results show that having more features is not necessarily better (i.e., since the additional features may induce wrong decisions when they are irrelevant).

On the other hand, an interesting improvement was observed with the `GOOSE++ & SV++` features, leading the F1-Score to 99.99%. We observed a significant improvement in the precision metric by these features because they are more representative to detect injection attacks.

2) Inverse Replay Attacks (UC02): For detecting Inverse Replay attacks (UC02), J48 reached an accuracy of 99.33% and a precision of 98.27% with the `GOOSE` features. Its recall was lower (89.18%), which results in an F1-Score of 93.5%.

F1-Score using `GOOSE & SV` features yielded a slightly worse performance. However, `GOOSE & SV++` and `GOOSE++ & SV++` features were able to improve the results because the enrichment process can reveal inconsistency in old messages when they are retransmitted. In particular, `GOOSE++ & SV++` lead the F1-Score to 99.39%, a gain of 5.89%. See detailed results in Table VII.

3) Masquerade Attacks: Outage (UC03): Masquerade attacks towards Outage (UC03) were detected with a very low F1-Score by J48 when using the basic `GOOSE` features. The poor precision (43.87%) and low recall (69.17%) led J48 to reach the lowest F1-Score (52.24%) when compared to all other studied attacks. These poor results are already expected

TABLE VII
FEATURE ENRICHMENT RESULTS FOR UC02

	Accuracy	Precision	Recall	F1-Score
GOOSE	99.33%	98.27%	89.18%	93.50%
GOOSE & SV	99.33%	99.93%	87.77%	93.46%
GOOSE & SV++	99.81%	100%	96.52%	98.23%
GOOSE++ & SV++	**99.93%**	**100%**	**98.80%**	**99.39%**

The bold values represent the higher values for each metric.

TABLE VIII
FEATURE ENRICHMENT RESULTS FOR UC03

	Accuracy	Precision	Recall	F1-Score
GOOSE	94.78%	43.87%	64.55%	52.24%
GOOSE & SV	95.68%	51.01%	60.31%	55.27%
GOOSE & SV++	97.06%	65.52%	70.65%	67.99%
GOOSE++ & SV++	**99.95%**	**99.70%**	**99.22%**	**99.46%**

The bold values represent the higher values for each metric.

since masquerade attacks are designed to confuse the IDSs by mimicking the legitimate behavior and the complexity level to detect it is high [7].

The only way to overcome the challenge imposed by masquerade attacks is to combine features from both GOOSE and SV and enrich them. The simple correlation of these protocols presented by GOOSE & SV features, however, does not yield significant improvement – F1-Score improved by only 3.03%. This happens because basic SV features are not very representative for masquerade attacks. However, when SV and GOOSE features are enriched, they gain information value and allow the detection of masquerade attacks with high accuracy (99.95%), precision (99.7%), recall (99.22%), and F1-Score (99.46%). The detailed results are shown in Table VIII.

4) Masquerade Attacks: Equipment Damage (UC04): In this use case, J48 also reached a low performance when using the GOOSE features to detect Masquerade Attacks towards Equipment Damage (UC04). The low precision (56.16%) and recall (80.1%) led to an F1-Score of 65.96%. The reason is the same discussed for UC03, as UC04 follows a similar logic: masquerade attacks are designed to confuse the IDSs by mimicking legitimate behavior.

Just combining the GOOSE & SV features is not enough: such a combination results in an F1-Score of only 79.22%. Thus, similarly to the UC03, the only way to overcome the challenge imposed by masquerade attacks is not only to combine but also to enrich features. However, in contrast to UC03, simply enriching the SV features was enough to improve the results across the board for UC04.

Note that the attacker's goal in UC04 is to mimic a normal condition in the transmission line during a fault. Therefore, as in UC03, the enrichment of SV features (SV++) for this attack is more relevant than the enrichment of GOOSE (GOOSE++) because SV++ can provide information about the actual transmission line status (i.e., if there is an electrical fault occurring or not). Interestingly, GOOSE++ & SV++ had a slightly lower performance (99.83% F1-Score). This shows that, for UC04, the GOOSE enrichment is not necessary (nor recommended), as it may confuse the classifier. Using feature selection methods after the feature enrichment can solve this issue [50], [51]. See detailed results in Table IX.

TABLE IX
FEATURE ENRICHMENT RESULTS FOR UC04

	Accuracy	Precision	Recall	F1-Score
GOOSE	96.30%	56.06%	80.10%	65.96%
GOOSE & SV	97.67%	66.02%	99.02%	79.22%
GOOSE & SV++	**99.99%**	**99.94%**	**99.88%**	**99.91%**
GOOSE++ & SV++	99.98%	99.87%	99.79%	99.83%

The bold values represent the higher values for each metric.

TABLE X
FEATURE ENRICHMENT RESULTS FOR UC05 AND UC07

	Accuracy	Precision	Recall	F1-Score
GOOSE	100%	100%	100%	100%
GOOSE & SV	100%	100%	100%	100%
GOOSE & SV++	100%	100%	100%	100%
GOOSE++ & SV++	100%	100%	100%	100%

TABLE XI
FEATURE ENRICHMENT RESULTS FOR UC06

	Accuracy	Precision	Recall	F1-Score
GOOSE	98.84%	89.12%	100%	94.25%
GOOSE & SV	98.84%	89.12%	100%	94.25%
GOOSE & SV++	98.84%	89.12%	100%	94.25%
GOOSE++ & SV++	98.84%	89.12%	100%	94.25%

In light of the potential bias in the *timestampDiff* feature, which stems from a fixed interval between the initial GOOSE message and the simulated electrical faults during each simulation run in benign electrical fault scenarios (UC00), one potential improvement for future work could be to introduce randomness in the occurrence of electrical faults. Despite the fact that the *timestampDiff* feature is one of the 12 features utilized by the J48 algorithm in UC04, an isolated proof-check experiment showed that its removal only led to a minor decrease of 0.03% in the F1-Score. This implies that the decisions made by the J48 algorithm neither rely solely on the *timestampDiff* feature, nor are they significantly impacted by its presence.

5) Random Message Injection (UC05), High-Status Number (UC06), and High-Rate Flooding (UC07) Attacks: In contrast to the previously discussed attacks, the Random Message Injection (UC05) and High-Rate Flooding (UC07) attacks are easily detected by J48 even when using only the GOOSE features. They reached 100% on all metrics, as shown in Table X.

For detecting High-Status Number Attack (UC06), J48 reached a perfect recall (100%) and a reasonably good accuracy (98.84%). Nevertheless, its precision was only 89.12% and, consequently, its F1-Score was impaired, as shown in Table XI. None of the variations of the dataset with enriched features could provide the classifier with more information to improve the results reached by J48.

All results remain consistent across different feature subsets when detecting UC05, UC06, and UC07. This is because the J48 decision tree algorithm bases its decisions on features already present in the smaller feature subset (i.e., GOOSE). Consequently, including more features for these specific attacks neither enhances nor biases the detection performance.

C. Best Features for Each Attack Use Case

In this section, we consider the complete GOOSE++ & SV++ feature dataset, comprising the 69 features proposed in this work (i.e., covering both basic and enriched ones) to assess how well they are used for detecting each attack class.

UC01: The decision tree model generated by J48 has the timestampDiff feature at the root. This feature enables J48 to detect replay messages sent in a too-short period of time from the previous one. However, since some replay messages are sent with a significant delay from the original message, other features are considered, such as timeFromLastChange and frameLen. The former can detect replay attacks transmitted in a period of time longer than *MaxTime* (i.e., 1000 ms in our scenario). The latter can also give important information as anomalies in frame length can indicate message tampering or injection of malicious content. The combination of the aforementioned features enables the J48 decision tree to reach a very high F1-Score (99.99%). The precision and recall are 100% and 99.99%, respectively.

UC02: J48 used sqDiff at the root of the decision tree. This feature enables J48 to decide whether and by how much the SqNum differs from the last GOOSE message. According to Section II-B, any changes on the devices' physical status reported by GOOSE should reset the SqNum and trigger the GOOSE burst mode. Otherwise, SqNum will keep increasing at regular periods. Therefore, monitoring the sequence number can help detect message replay attacks, where an attacker retransmits an earlier message to disrupt the system. The timeFromLastChange and timestampDiff are also considered in the decision tree. Monitoring the time elapsed since the last status change can reveal abnormal behavior or frequent fluctuations, possibly signifying a malicious intrusion. Similarly, the time elapsed between two messages should follow the retransmission times defined in the IEC–61850, otherwise, it may be a strong indicator of an attack.

UC03 and UC04: The decision trees built by J48 to detect these more elaborate attacks are more complex and require a deeper analysis (more features need to be considered). A very important contribution of the ERENO Framework, as well as the resulting generated dataset, is the possibility of developing an IDS to cross reference two IEC–61850 protocols: GOOSE and SV. Because of that, even very sophisticated GOOSE attacks can be identified based on the correlation between the physical/electrical measures and the current circuit-breaker status reported by IEDs. Whereas features derived from the current and voltage can be extracted from the SV messages payload, the physical circuit-breaker status is reported by the cbStatus field in the GOOSE messages. Any malicious attempt of sending a fake cbStatus may cause an inconsistency (e.g., a fault being reported by GOOSE while stable current and voltage measures are transmitted on the SV messages payload).

The J48 algorithm considered several features to build a decision tree for Masquerade Attacks aiming to cause outages by reporting fake electrical faults (UC03). An outage may happen when an actuator IED processes a modified GOOSE and responds by opening the circuit breaker during normal operation. The root node of the decision tree for this attack is cbStatus which represents the (possibly) fake status imposed by the masquerade attack. Whenever cbStatus represents a closed circuit-breaker (i.e., cbStatus $= 0$, which happens 335,908 times in the dataset), J48 concludes that any UC03 attack is not happening. Otherwise, other features are analyzed to check the consistency with the *cbStatus* value. As already expected, most of these features are derived from SV messages. These features are representative to check if a fault is occurring (i.e., they are expected to have anomalous values during a fault). Thus, when there are no faults, J48 concludes that a fault event is forged by the masquerade attack. The delay feature is also used to improve the classification performance, as it can reveal messages are received too fast (i.e., additional fake messages among the legitimate ones). The Masquerade Attacks reporting fake normal situations (UC04) have similar logic to UC03, but a very different decision tree model was generated by J48. As in the UC03 decision tree model, the most predominant features are those related to the electrical signals carried by SV and to the cbStatus. However, the root feature was the vsbARmsValue. Other electrical signals were used close to the root (e.g., isbARmsValue, ismC, and isbA). Additionally, the final decision also considers timestampDiff and cbStatus. They can determine, respectively, whether a suspicious message has an anomalous delay from the previous legitimate message and whether it has the circuit-breaker status indicating an electrical fault. Therefore, the decision tree model generated by J48 also corresponds to the expected attacker behavior mapping for this attack.

UC05: These attacks are simpler to detect than others due to the naive attacker behavior and lack of information about the targeted environment, which produces messages with inconsistent features. Therefore, there are multiple features that are inconsistent with the expected domain features. One of them is the confRev (i.e., a configuration parameter field defined by the substation environment), used at the root of the J48 decision tree model for UC05. Whereas some messages have a consistent value for the GOOSE confRev field, a large part of the attacks is detected by using the gooseTimeAllowedtoLive range. Note that when deploying an IDS, the legitimate value for confRev would change over time. Surprisingly, these two features are enough to enable 99.99% accuracy, 100% precision, 99.99% recall, and 99.99% F1-Score.

UC06: The decision tree built by J48 is very simple. Only the StNum feature is used to check whether it is too high or not. In fact, this assumption would be enough if all the future StNum are known and the correct StNum range is mapped. In a real scenario, this assumption can lead to a too simple and inefficient decision tree model. That was the case in our realistic experimentation scenario. By analyzing all training instances, J48 built a decision tree in which messages with StNum higher than 4954 are considered attacks.

TABLE XII
RESULTS FOR INDIVIDUAL ATTACK CLASSES (ALL 69 FEATURES)

Use Case	Accuracy	Precision	Recall	F1-Score
UC01	100%	100%	99.99%	99.99%
UC02	99.93%	100%	98.80%	99.39%
UC03	99.95%	99.70%	99.22%	99.46%
UC04	99.98%	99.87%	99.79%	99.83%
UC05	100%	100%	100%	100%
UC06	98.84%	89.12%	100%	94.25%
UC07	100%	100%	100%	100%

UC07: Finally, detecting this attack is also easy because the attacker sends too many messages in a very short period of time. Such behavior results in a clearly anomalous `timestampDiff`. In a few cases, when the *stNum* is excessively increased, the `frameLen` may help the decision because the message is longer than the legitimate ones.

We summarize all aforementioned results for individual use case analyses (when using all 69 features, basic and enriched) in Table XII. Based on these results we can conclude that the proposed features generated by us through the ERENO Framework are very representative to detect most of the attack use cases.

D. The Impact of Feature Enrichment on J48 Trees

In this section we analyze how each subset of features (i.e., `GOOSE`, `GOOSE&SV`, `GOOSE++&SV`, and `GOOSE++&SV++`) impacts on the decision trees generated by the J48 algorithm.

To demonstrate the impact of feature enrichment on the decision trees generated by the J48 algorithm, we choose to assess the most complex attack use case: the *Masquerade Attacks towards Outage (UC03)*.

The generated decision trees for `GOOSE`, `GOOSE & SV`, `GOOSE & SV++`, and `GOOSE++ & SV++` are represented by Fig. 7(a), (b), (c), and (d), respectively. The nodes (ellipses) represent features and arrows represent the decision taken according to their values compared to a threshold. The leaves (red or green boxes) represent the output classes (attack or normal, respectively).

Fig. 7(a) and (b) reveal that `GOOSE & SV` features generate a more complex tree than just `GOOSE` features. Most of the features selected in the `GOOSE & SV` case are those extracted from the GOOSE protocol (64%). As shown in Table VIII, the combination of both protocols improved the F1-Score from 52.24% to 55.27%, compared with `GOOSE` features alone.

Fig. 7(c) shows that the enriched SV features *SV++* were the most commonly selected by J48 (44.2%) to compose the decision tree when `GOOSE & SV++` was used. Note that although we use the `SV++` notation to refer to both SV and enriched SV features in the manuscript, in Fig. 7 these features are distinguished by SV (basic SV) and SV++ (enriched SV). The results of this decision tree reveal that feature enrichment gives a significant improvement (As discussed in Section VI and shown in Table VIII, these features improved the F1-Score from 52.24% to 67.99%).

Finally, Fig. 7(d) shows an interesting result: when all features are available for J48, the resulting tree was actually smaller.

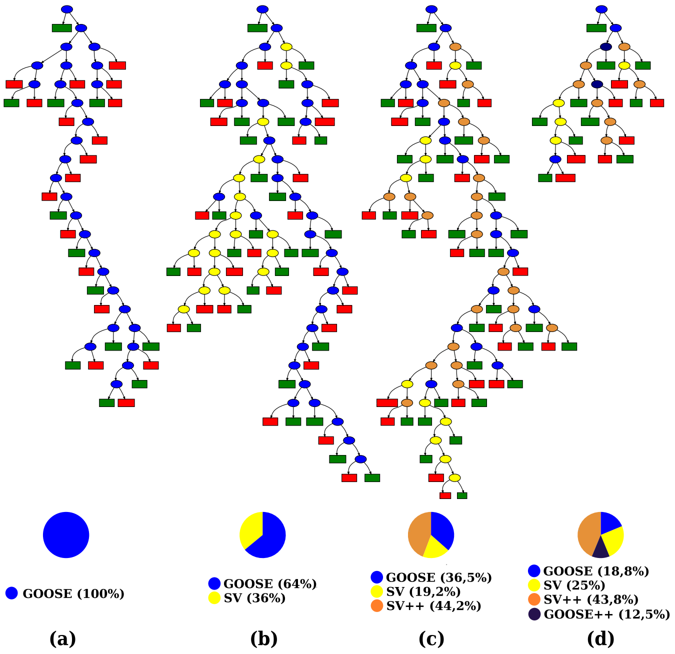

Fig. 7. Masquerade Attacks - Outage (UC03).

This happened because this case has rich information to more easily describe the attacker's behavior. Moreover, this smaller tree resulted in the best performance in terms of F1-Score: according to Table VIII, this tree was responsible for J48 reaching a 99.46% F1-Score. For this attack profile, the enriched SV features compose 43.8% of the decision tree, whereas 12.5% are enriched GOOSE features. Thus, 56.3% of features come from the enrichment process.

VII. Conclusion

The IEC–61850 standard plays a critical role in the modern electrical power grid, and existing and future security issues should be carefully addressed. Deploying IDSs is a way to identify malicious activities and mitigate the attacker's actions. The training and testing of IDSs designed for substations are challenging in part due to the lack of public IEC–61850-based intrusion datasets.

We proposed the ERENO traffic generation framework and generated a suite of public and realistic IEC–61850 datasets, named ERENO–IEC–61850. ERENO currently generates datasets with 69 traffic features comprising attributes extracted from GOOSE and SV protocols in addition to newly generated features that explore the inter-protocol correlation and sequential messages consistency. It supports the generation of both benign and attack traffic, including the typical behavior without faults and with the presence of faults on the power line. We validated our proposed framework through the J48 decision tree algorithm using 7 attack classes and benign traffic. The results show that the enriched features can enable the J48 classifier to classify these attacks with high accuracy, precision, recall, and F1-Score metrics.

The ERENO framework is modular and open-source, and it is easy to use as a basis for future research on IDS. While ERENO is designed to support any scenario, in the proof-of-concept presented in this paper we focused on the scenario of an attacker attempting to disrupt the system by introducing fake GOOSE messages. This scenario allowed us to explore a range of attacks and possibilities, but we encourage future works using our tool to perform further research on other customized scenarios.

In future works, we plan to extend ERENO with more use cases to cover novel attacks. Additionally, we plan to deploy feature selection methods and real-time processing techniques to process more complex traffic generated by ERENO, such as detecting multiple attack classes simultaneously. We also plan to implement more attacks (e.g., stealthy injection) and other features, such as the support for MMS protocol for the generation of PCAP files as an alternative output to the dataset files. Such generation could also be done by consuming SCL files to self-configure the ERENO parameters. Finally, we plan to implement, assess, and compare existing IDSs proposals by using the ERENO–IEC–61850 realistic datasets.

References

[1] S. E. Quincozes, C. Albuquerque, D. Passos, and D. Mossé, "A survey on intrusion detection and prevention systems in digital substations," *Comput. Netw.*, vol. 184, 2021, Art. no. 107679.

[2] M. Popovic, M. Mohiuddin, D.-C. Tomozei, and J.-Y. Le Boudec, "iPRP–The parallel redundancy protocol for IP networks: Protocol design and operation," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1842–1854, Oct. 2016.

[3] A. Elgargouri and M. Elmusrati, "Analysis of cyber-attacks on IEC 61850 networks," in *Proc. 11th Int. Conf. Appl. Inf. Commun. Technol.*, 2017, pp. 1–4.

[4] J. Hong and C. Liu, "Intelligent electronic devices with collaborative intrusion detection systems," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 271–281, Jan. 2019.

[5] J. Hong, C. Liu, and M. Govindarasu, "Detection of cyber intrusions using network-based multicast messages for substation automation," in *Proc. Innov. Smart Grid Technol.*, 2014, pp. 1–5.

[6] H. Yoo and T. Shon, "Novel approach for detecting network anomalies for substation automation based on IEC 61850," *Multimedia Tools Appl.*, vol. 74, no. 1, pp. 303–318, 2015.

[7] T. S. Ustun, S. M. Farooq, and S. S. Hussain, "A novel approach for mitigation of replay and masquerade attacks in smartgrids using IEC 61850 standard," *IEEE Access*, vol. 7, pp. 156044–156053, 2019.

[8] J. Hoyos, M. Dehus, and T. X. Brown, "Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure," in *Proc. IEEE Globecom Workshops*, 2012, pp. 1508–1513.

[9] U. K. Premaratne, J. Samarabandu, T. S. Sidhu, R. Beresh, and J.-C. Tan, "An intrusion detection system for IEC61850 automated substations," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2376–2383, Oct. 2010.

[10] Q. Yang, W. Hao, L. Ge, W. Ruan, and F. Chi, "FARIMA model-based communication traffic anomaly detection in intelligent electric power substations," *IET Cyber-Physical Syst.: Theory Appl.*, vol. 4, no. 1, pp. 22–29, 2019.

[11] G. Lounge, "Capture files from 4SICS geek lounge," 2019. Accessed Jan. 29, 2022. [Online]. Available: https://www.netresec.com/?page=PCAP4SICS

[12] S. Adepu, N. K. Kandasamy, and A. Mathur, "EPIC: An electric power testbed for research and training in cyber physical systems security," in *Computer Security*, Berlin, Germany: Springer, 2018, pp. 37–52.

[13] P. P. Biswas, H. C. Tan, Q. Zhu, Y. Li, D. Mashima, and B. Chen, "A synthesized dataset for cybersecurity study of IEC 61850 based substation," in *Proc. IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids*, 2019, pp. 1–7.

[14] Y. Yang, H.-Q. Xu, L. Gao, Y.-B. Yuan, K. McLaughlin, and S. Sezer, "Multidimensional intrusion detection system for IEC 61850-based SCADA networks," *IEEE Trans. Power Del.*, vol. 32, no. 2, pp. 1068–1078, Apr. 2017.

[15] M. Kabir-Querrec, S. Mocanu, J.-M. Thiriet, and E. Savary, "Power utility automation cybersecurity: IEC 61850 specification of an intrusion detection function," in *Proc. 25th Eur. Saf. Rel. Conf.*, CRC, 2015, pp. 1–8.

[16] J. Hong, C.-C. Liu, and M. Govindarasu, "Integrated anomaly detection for cyber security of the substations," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1643–1653, Jul. 2014.

[17] I. E. Commission, *Communication Networks and Systems in Substations - ALL PARTS*. London, U.K.: IET, 2003.

[18] R. E. Mackiewicz, "Overview of IEC 61850 and benefits," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, 2006, pp. 8–pp.

[19] J. O'Raw, D. M. Laverty, and D. J. Morrow, "IEC 61850 substation configuration language as a basis for automated security and SDN configuration," in *Proc. Power Energy Soc. Gen. Meeting*, 2017, pp. 1–5.

[20] M. A. Aftab, S. S. Hussain, I. Ali, and T. S. Ustun, "IEC 61850 based substation automation system: A survey," *Int. J. Elect. Power Energy Syst.*, vol. 120, 2020, Art. no. 106008.

[21] International Electrotechnical Commission, *IEC 61850–9-2 Communication networks and systems in substations–Part 9-2: Specific Communication Service Mapping (SCSM) – Sampled values over ISO/IEC 8802-3*, 1st ed. London, U.K.: IET, 2004.

[22] S. Kariyawasam, A. D. Rajapakse, and N. Perera, "Investigation of using IEC 61850-sampled values for implementing a transient-based protection scheme for series-compensated transmission lines," *IEEE Trans. Power Del.*, vol. 33, no. 1, pp. 93–101, Feb. 2018.

[23] E. Solomin, D. Topolsky, and N. Topolsky, "Arrangement of data exchange between adaptive digital current and voltage transformer and SCADA-system under IEC 61850 standard," *Procedia Eng.*, vol. 129, pp. 207–212, 2015.

[24] M. T. A. Rashid, S. Yussof, Y. Yusoff, and R. Ismail, "A review of security attacks on IEC61850 substation automation system network," in *Proc. 6th Int. Conf. Inf. Technol. Multimedia*, 2014, pp. 5–10.

[25] V. Borgiani, P. Moratori, J. F. Kazienko, E. R. Tubino, and S. E. Quincozes, "Toward a distributed approach for detection and mitigation of denial-of-service attacks within industrial Internet of Things," *IEEE Internet of Things J.*, vol. 8, no. 6, pp. 4569–4578, Mar. 2021.

[26] S. Stolfo et al., "KDD cup 1999," Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Oct. 2007.

[27] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, 2009, pp. 1–6.

[28] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf.*, 2015, pp. 1–6.

[29] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.

[30] T. Athay, R. Podmore, and S. Virmani, "A practical method for the direct analysis of transient stability," *IEEE Trans. Power App. Syst.*, vol. PAS-98, no. 2, pp. 573–584, Mar. 1979.

[31] B. Kang, K. McLaughlin, and S. Sezer, "Towards a stateful analysis framework for smart grid network intrusion detection," in *Proc. 4th Int. Symp. ICS SCADA Cyber Secur. Res.*, 2016, pp. 124–131.

[32] Q. Yang, D. Keckalo, D. Dolezilek, and E. Cenzon, "Testing IEC 61850 merging units," in *Proc. 44th Annu. Western Protective Relay Conf.*, Washington, DC, USA, 2017, pp. 17–19.

[33] Y. Yang, K. McLaughlin, L. Gao, S. Sezer, Y. Yuan, and Y. Gong, "Intrusion detection system for IEC 61850 based smart substations," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2016, pp. 1–5.

[34] Y. Kwon, H. K. Kim, Y. H. Lim, and J. I. Lim, "A behavior-based intrusion detection technique for smart grid infrastructure," in *Proc. IEEE Eindhoven PowerTech*, 2015, pp. 1–6.

[35] J. W. Teo, S. Gunawan, P. P. Biswas, and D. Mashima, "Evaluating synthetic datasets for training machine learning models to detect malicious commands," in *Proc. IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids*, 2022, pp. 315–321.

[36] O. Hegazi, E. Hammad, A. Farraj, and D. Kundur, "IEC-61850 GOOSE traffic modeling and generation," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2017, pp. 1100–1104.

[37] Y. Lopes, D. C. Muchaluat-Saade, N. C. Fernandes, and M. Z. Fortes, "Geese: A traffic generator for performance and security evaluation of IEC 61850 networks," in *Proc. IEEE 24th Int. Symp. Ind. Electron.*, 2015, pp. 687–692.

[38] S. M. Blair, F. Coffele, C. D. Booth, and G. M. Burt, "An open platform for rapid-prototyping protection and control schemes with IEC 61850," *IEEE Trans. Power Del.*, vol. 28, no. 2, pp. 1103–1110, Apr. 2013.

[39] S. E. Quincozes, C. Raniery, R. C. Nunes, C. Albuquerque, D. Passos, and D. Mosse, "A counselors-based intrusion detection architecture," in *Proc. 9th Latin Amer. Netw. Operations Manage. Symp.*, 2019, pp. 1–8.

[40] I. E. Commission, *Communication Networks and Systems in Substations - Part 8–1: Specific Communication Service Mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*, London, U.K.: IET, 2003.

[41] M. H. I. Ltd., “The world’s most advanced tool for power systems EMT simulations,” 2019. [Online]. Available: https://www.pscad.com/software/pscad/overview

[42] P. H. Pinheiro, M. Helena, A. Colombini, B. França, and M. Fortes, “Detailed modelling and analysis of digital MHO distance relay with single-pole operation,” *Acta Polytechnica*, vol. 61, pp. 537–551, Aug. 2021.

[43] O. Operador Nacional do Sistema Elétrico, “Submódulo 2.11 requisitos mínimos para os sistemas de proteção, de registro de perturbações e de teleproteção,” 2021. Accessed: 2021. [Online]. Available: http://apps08.ons.org.br/ONS.Sintegre.Proxy/ecmprsite/ecmfragmentsdocuments/Subm%C3%B3dulo%202.11-RQ_2020.12.pdf

[44] J. A. C. Weideman, “Numerical integration of periodic functions: A few examples,” *Amer. Math. Monthly*, vol. 109, no. 1, pp. 21–36, 2002.

[45] H. Satoh, K. Yamashita, K. Shirasaki, and Y. Kitauchi, “Root-mean square model of three-phase photovoltaic inverter for unbalanced fault,” *IEEE Open Access J. Power Energy*, vol. 7, pp. 501–513, 2020.

[46] H. Hadeli, R. Schierholz, M. Braendle, and C. Tuduce, “Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration,” in *Proc. IEEE Conf. Emerg. Technol. Factory Automat.*, 2009, pp. 1–8.

[47] H. Hadeli, R. Schierholz, M. Braendle, and C. Tuduce, “Generating configuration for missing traffic detector and security measures in industrial control systems based on the system description files,” in *Proc. IEEE Conf. Technol. Homeland Secur.*, 2009, pp. 503–510.

[48] C. Fernandes, S. Borkar, and J. Gohil, “Testing of GOOSE protocol of IEC61850 standard in protection IED,” *Int. J. Comput. Appl.*, vol. 93, no. 16, pp. 30–35, 2014.

[49] M. Slocum, “Decision making using ID3 algorithm,” *Insight: River Academic J.*, vol. 8, no. 2, pp. 1–12, 2012.

[50] S. E. Quincozes, D. Mossé, D. Passos, C. Albuquerque, L. S. Ochi, and V. F. dos Santos, “On the performance of GRASP-based feature selection for CPS intrusion detection,” *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 1, pp. 614–626, Mar. 2022.

[51] S. E. Quincozes, D. Passos, C. Albuquerque, L. S. Ochi, and D. Mossé, “GRASP-based feature selection for intrusion detection in CPS perception layer,” in *Proc. 4th Conf. Cloud Internet Things*, 2020, pp. 41–48.

Silvio Ereno Quincozes received the BSc degree in software engineering from the Federal University of Pampa, in 2015, the MSc degree in computer science from the Federal University of Santa Maria, in 2018, and the DSc degree in Computing from the Fluminense Federal University with a sandwich period with the University of Pittsburgh, in 2022. His research topics are related to intrusion detection systems, information security, computer networks, smart grids, electrical substation systems, Internet of Things, data mining, and software engineering. He is a professor with the Federal University of Pampa and with the Graduate Program in Computing (PPGCO) with the Uberlândia Federal University. He received the award for the best doctoral thesis with Thesis and Dissertation Contest in Information Security and Computational Systems, in 2022.

Célio Albuquerque received the BS and MS degrees in electrical and electronics engineering from Universidade Federal do Rio de Janeiro, Brazil, in 1993 and 1995, and the MS and PhD degrees in information and computer science from the University of California at Irvine, in 1997 and 2000, respectively. From 2000 to 2003, he served as the networking architect for Magis Networks, designing high-speed wireless medium access control. Since 2004 he has been a professor with the Computer Science Department of Universidade Federal Fluminense, Brazil. His research interests include security, wireless networks, and multimedia distribution.

Diego Passos received the BSc, MSc, and DSc degrees in computer science from Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil, in 2007, 2009, and 2013, respectively. From 2013 to 2014, he worked as a postdoctoral fellow researcher with the Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil. From 2014 to 2021, he was a professor with the Computer Science Department of UFF. He is currently a professor with the Electronics, Telecommunications, and Computers Engineering Department of the Instituto Superior de Engenharia de Lisboa (ISEL). His research interests include multi-hop wireless networks, network coding, and wireless routing.

Daniel Mossé received the BS degree in mathematics from the University of Brasilia, Brazil, in 1985, and the MS and PhD degrees in computer science from the University of Maryland, College Park, in 1990 and 1993, respectively. He has been a professor with the University of Pittsburgh since 1992, including six years as department chair, and has co-founded HiberSense, a startup company in the area of Smart Homes. He has been involved in the design and implementation of a couple of distributed, real-time operating systems. His main research interest is in the allocation of resources (computing, energy, and network resources) in the realm of sustainable computing, computing for sustainability, and real-time, with the main concerns being power management, security, and fault tolerance. He bridges the gap between the operating systems and networking research fields, between practice and theory. For the last 20 years, most of his systems research has focused on power and energy management, and for the last decade on how to increase diversity and how promote reproducible research in computing.