

UNIVERSIDADE FEDERAL FLUMINENSE

SILVIO ERENO QUINCOZES

**ERENO: AN EXTENSIBLE TOOL FOR  
GENERATING REALISTIC IEC-61850  
INTRUSION DETECTION DATASETS**

NITERÓI

2022

SILVIO ERENO QUINCOZES

**ERENO: AN EXTENSIBLE TOOL FOR  
GENERATING REALISTIC IEC-61850  
INTRUSION DETECTION DATASETS**

Ph.D. thesis presented to the Graduate Program in Computing at the Fluminense Federal University as a requirement for obtaining a Doctoral Degree in Computing. Concentration area: Computer Science.

Advisor:

CÉLIO ALBUQUERQUE

Co-advisor:

DIEGO PASSOS

NITERÓI

2022

Ficha catalográfica automática - SDC/BEE  
Gerada com informações fornecidas pelo autor

Q7e Quincozes, Silvio Ereno  
ERENO: AN EXTENSIBLE TOOL FOR GENERATING REALISTIC IEC--  
61850 INTRUSION DETECTION DATASETS / Silvio Ereno Quincozes ;  
Célio Vinícius Neves de Albuquerque, orientador ; Diego  
Passos, coorientador. Niterói, 2022.  
122 f. : il.

Tese (doutorado)-Universidade Federal Fluminense, Niterói,  
2022.

DOI: <http://dx.doi.org/10.22409/PGC.2022.d.02690526077>

1. Rede de Computadores. 2. Segurança da Informação. 3.  
Produção intelectual. I. Albuquerque, Célio Vinícius Neves  
de, orientador. II. Passos, Diego, coorientador. III.  
Universidade Federal Fluminense. Instituto de Computação.  
IV. Título.

CDD -

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

Figure 1: Ficha catalográfica

# SILVIO ERENO QUINCOZES

## ERENO: AN EXTENSIBLE TOOL FOR GENERATING REALISTIC IEC-61850 INTRUSION DETECTION DATASETS

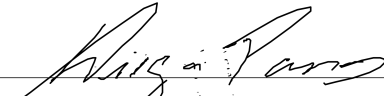
Ph.D. thesis presented to the Graduate  
Program in Computing at the Fluminense  
Federal University as a requirement for ob-  
taining a Doctoral Degree in Computing.  
Concentration area: Computer Science

Approved in February 2022.

### EXAMINATION BOARD



Prof. Dr. Célio Albuquerque - Advisor, UFF



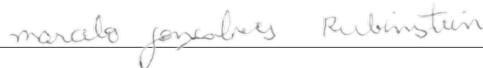
Prof. Dr. Diego Passos - Co-advisor, UFF




Prof. Dr. Daniel Mossé, University of Pittsburgh



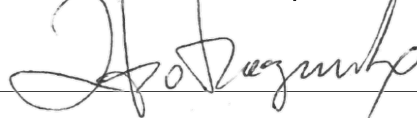
Prof. Dr. Igor Moraes, UFF



Prof. Dr. Marcelo Rubinstein, UERJ



Prof. Dr. Yona Lopes, UFF



Prof. Dr. Juliano Kazienko, UFSM

Niterói

2022

*The only place success comes before work is in the dictionary.*

*(Vince Lombardi)*

# Acknowledgment

First of all, I thank God for being my main guide throughout this journey. Then, I thank the family members who were present, giving me their support in every way. Thank you for your understanding during the times I was absent from family activities to continue the work here. I also express my thanks to my advisors for their wise counsel. Additionally, I thank the founding institution that provide resources to support my research: The *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) supported me at the whole Ph.D (both in Brazil and in the United States sandwich period mission); Transmissora Aliança de Energia Elétrica (TAESA) and Agência Nacional de Energia Elétrica (ANEEL) supported me in part-time under the project identified as *Projeto 0053: Teleproteção em IEC-61850*. Finally, I would like to thank the colleagues who in some way contributed to my journey.

# Resumo

As subestações elétricas digitais são um elemento chave para fornecer uma base confiável para as redes elétricas inteligentes do futuro, conhecidas como *Smart Grids*. As funções automatizadas dentro de tais subestações pressupõem conectividade crescente entre Dispositivos Eletrônicos Inteligentes, do inglês, *Intelligent Electronic Devices* (IEDs). O padrão IEC-61850 propõe um conjunto de protocolos de comunicação para definir como os IEDs se comunicam. Nossos estudos mostram que existem vulnerabilidades que podem comprometer tais protocolos, causando mau funcionamento dos equipamentos elétricos. Portanto, os Sistemas de Detecção de Intrusões (IDSs) se tornaram um componente essencial para proteger as subestações de atividades maliciosas. Embora as técnicas de detecção de intrusão sejam comumente estudadas em redes e sistemas convencionais, apenas alguns estudos abordam esse problema, considerando os requisitos, limitações e protocolos de comunicação específicos das subestações IEC-61850. Nossos estudos revelam que a falta de dados realistas para treinamento, teste e avaliação de IDSs em cenários industriais realistas é considerada um grande desafio. Como consequência, o desenvolvimento de IDSs atualmente é limitado pelos conjuntos de dados disponíveis. Esta tese visa a uma proposta de um *framework* para apoiar uma solução robusta para detecção e prevenção de intrusões em subestações IEC-61850. Nossa principal contribuição está no desenvolvimento do *Efficacious Reproducer Engine for Network Operations* (ERENO). ERENO é um *framework* de código aberto para gerar conjuntos de dados IEC-61850 com *features* (atributos) representativas — extraídas de protocolos de comunicação de subestação em nível de rede e do domínio elétrico — para detectar diferentes tipos de intrusões. Como uma contribuição adicional e como uma prova de conceito, apresentamos um conjunto de 8 conjuntos de dados IEC-61850 realistas, que modelam 7 casos de uso de ataques e 1 para tráfego de rede normal. Ainda, apresentamos uma nova taxonomia dos aspectos de IDSs baseados em IEC-61850. Demonstramos que o ERENO é capaz de gerar atributos representativos para o serem processados por algoritmos de aprendizado de máquina. Combinando-se extração e seleção de atributos, ganhos significativos foram observados, incluindo-se a detecção de ataques não detectados adequadamente na literatura. No melhor caso, melhorou-se a F1-Score do algoritmo J48 de 52,24% para 99,46%.

---

**Palavras-chave:** Sistema de Detecção de Intrusões, Subestações Digitais, Dispositivos Eletrônicos Inteligentes, Norma IEC-61850, GOOSE, SV, MMS.



# Abstract

Digital electric grid substations are a key element to create reliable future smart grids. The IEC-61850 standards proposes a set of communication protocols to define how Intelligent Electronic Devices (IEDs) can communicate. Our studies show that there is a range of vulnerabilities that may compromise the IEC-61850 communication protocols and cause improper functioning of the physical power system. Therefore, detecting and preventing cyber intrusions play a vital role, and Intrusion Detection Systems (IDSs) have become an essential component of safeguarding substations from malicious activities. Even though intrusion detection techniques are commonly studied in conventional networks and systems, only a few studies address this issue considering the IEC-61850 substations requirements, limitations, and specific communication protocols. Furthermore, our studies reveal that the lack of realistic data for training, testing, and evaluating IDSs in realistic industrial scenarios is considered a major challenge. As consequence, the development of IDSs is currently limited by the datasets available. This thesis aims at building a framework to support a robust solution for detecting and preventing intrusions on IEC-61850 substations. Our main contribution is in the development of the Efficacious Reproducer Engine for Network Operations (ERENO). ERENO is an open-source framework for generating IEC-61850 datasets with representative features — extracted both from substation communication protocols and the electric domain — for detecting different types of intrusions. As an additional contribution and as a proof-of-concept, we present a suite of realistic IEC-61850 datasets that model 8 use cases, namely traffic for 7 common attacks and 1 for normal network traffic. Finally, we also present a novel taxonomy for the IEC-61850-based IDSs aspects. Our results show that our traffic generation solution with attack signatures is able to generate representative features to be processed by machine learning algorithms. With the combination of feature extraction and feature selection, significant gains were observed, including the detection of attacks that are not properly detected by the existing techniques in the literature. For more challenging attacks, we present F1-Score gains for the J48 classifier that took its performance from 52.24% to 99.46%.

**Keywords:** Intrusion Detection Systems, IDS, Digital Electric Grid Substations, Intelligent Electronic Devices, IEC-61850 Standard, GOOSE, SV.

# List of Figures

1	Ficha catalográfica . . . . .	
2	IEC–61850 Typical Architecture. . . . .	20
3	Ethernet frames of SV messages. . . . .	21
4	Ethernet frame of GOOSE messages. . . . .	22
5	GOOSE messages transmission. Extracted from ( <a href="#">HOYOS; DEHUS; BROWN, 2012</a> ). . . . .	23
6	Proposed taxonomy for intrusion detection aspects on IEC–61850 digital substations. . . . .	42
7	The proposed solution architecture. . . . .	58
8	Proposed ERENO Extensible Tool for generation of realistic IEC–61850-based datasets. . . . .	59
9	Simulated electric power grid model. . . . .	60
10	Balanced three-phase readings ( <a href="#">OMAKAZI, 2022</a> ). . . . .	68
11	Unbalanced three-phase readings ( <a href="#">OMAKAZI, 2022</a> ). . . . .	68
12	GRASP-based FS process. . . . .	73
13	Random Replay Attacks (UC01) . . . . .	85
14	Inverse Replay Attacks (UC02) . . . . .	86
15	Masquerade Attacks - Outage (UC03) . . . . .	87
16	Masquerade Attacks - Equipment Damage (UC04) . . . . .	88
17	Random Message Injection attacks (UC05) . . . . .	88
18	High Status Number attacks (UC06) . . . . .	89
19	Poisoned High-Rate Attack (UC07) . . . . .	90

# List of Tables

1	Summary of threats on IEC-61850 digital substation networks. . . . .	36
2	Summary of IDSs evaluation techniques (ordered by date) . . . . .	47
3	Specification rules assessment results. . . . .	52
4	Generated GOOSE messages per simulation run. . . . .	63
5	Field values for the generated GOOSE messages per simulation run. . . . .	63
6	The chosen parameters for our methodology. . . . .	79
7	Detection performance of J48 for Random Replay attacks with different input features. . . . .	80
8	Inverse Replay attacks feature selection and enrichment assessment. . . . .	81
9	Masquerade (outage) attacks feature selection and enrichment assessment. . . . .	81
10	Masquerade (equipment damage) attacks feature selection and enrichment assessment. . . . .	82
11	Random Message Injection attacks feature selection and enrichment assessment. . . . .	83
12	High-Status Number attacks feature selection and enrichment assessment. . . . .	83
13	High-Rate Flooding attacks feature selection and enrichment assessment. . . . .	84
14	Results for each individual attack class . . . . .	90
15	The impact of feature enrichment on multi-class detection. . . . .	92
16	The resulting confusion matrix for J48 when using only the GOOSE features. . . . .	92
17	The confusion matrix for J48 when using only the GOOSE & SV features. . . . .	93
18	The confusion matrix for J48 when using only the GOOSE & SV++ features. . . . .	94
19	The confusion matrix for J48 when using all features (GOOSE++ & SV++). . . . .	95

# List of Abbreviations and Acronyms

**AMI** Advanced Metering Infrastructure

**APDU** Application Protocol Data Unit

**ARP** Address Resolution Protocol

**ASDU** Application Service Data Units

**BF** Bit-Flip

**DoS** Denial of Service

**ERENO** Efficacious Reproducer Engine for Network Operations

**GOOSE** Generic Oriented Object Substation Events

**GR** Gain Ratio

**GRASP** Greedy Randomized Adaptive Search Procedure

**IDS** Intrusion Detection System

**IED** Intelligent Electronic Device

**IG** Information Gain

**IWSS** Incremental Wrapper-based subset Selection

**IWSSR** IWSS with replacement

**MAC** Media Access Control

**MITM** Man-In-The-Middle

**MMS** Manufacturing Message Specification

**PLC** Programmable Logic Controller

**PTP** Precision Time Protocol

**RCL** Restricted Candidate List

**RVND** Random Variable Neighborhood Descent

**SCADA** Supervisory Control and Data Acquisition

**SV** Sampled Values

**USA** United States of America

**VND** Variable Neighborhood Descent

# Contents

<b>1</b>	<b>Introduction</b>	<b>12</b>
1.1	Motivation . . . . .	13
1.2	Research Problem . . . . .	15
1.3	Goals . . . . .	16
1.3.1	General Goal . . . . .	16
1.3.2	Specific Goals . . . . .	16
1.4	Organization . . . . .	17
<b>2</b>	<b>Background</b>	<b>18</b>
2.1	The IEC-61850 Standard . . . . .	18
2.1.1	Physical Topology . . . . .	18
2.1.2	Network Communication Protocols . . . . .	19
2.1.2.1	SV Protocol . . . . .	19
2.1.2.2	GOOSE Protocol . . . . .	21
2.2	Feature Selection for IDSs . . . . .	23
<b>3</b>	<b>Related Work</b>	<b>27</b>
3.1	Intrusion Detection and Datasets . . . . .	27
3.1.1	Generic Traffic . . . . .	27
3.1.2	Self-generated Traffic . . . . .	28
3.2	Feature Selection Methods: State of the Art . . . . .	29
3.2.1	Filter-based Feature Selection . . . . .	29

---

3.2.2	Metaheuristics as Wrapping Methods for Feature Selection . . . . .	31
3.3	Discussion . . . . .	33
<b>4</b>	<b>Research Problem</b>	<b>34</b>
4.1	Security and Threats . . . . .	34
4.1.1	Attacks to IEC-61850 Multicast Protocols . . . . .	35
4.1.1.1	Replay Attack . . . . .	35
4.1.1.2	Message Injection . . . . .	37
4.1.1.3	Masquerade Attack . . . . .	37
4.1.1.4	Poisoning Attack . . . . .	38
4.1.2	Other Inter- and Intra-Substation Threats . . . . .	39
4.1.3	IEC-62351 Standard . . . . .	40
4.2	Study of the Digital Substations IDS . . . . .	41
4.2.1	Design Aspects . . . . .	41
4.2.1.1	Architecture . . . . .	42
4.2.1.2	Detection Approach . . . . .	43
4.2.1.3	Analysis . . . . .	44
4.2.1.4	Action . . . . .	44
4.2.2	Deployment Aspects . . . . .	45
4.2.2.1	Data Sources and Detection Range . . . . .	45
4.2.2.2	Evaluation . . . . .	46
4.2.2.3	Metrics . . . . .	47
4.3	Specification Rules Assessment . . . . .	48
4.3.1	Attacker Assumptions . . . . .	48
4.3.2	Specifications Rules . . . . .	49
4.4	Discussion on Digital Substations IDS . . . . .	56



---

<b>5 Hypothesis and Proposed Solution</b>	<b>57</b>
5.1 Hypothesis . . . . .	57
5.2 Data Generation: ERENO Extensible Tool . . . . .	58
5.2.1 Power Grid Simulations . . . . .	60
5.2.2 SV Traffic Generation . . . . .	61
5.2.3 GOOSE Traffic Generation . . . . .	61
5.2.4 Malicious Traffic through Attack Use Cases . . . . .	63
5.2.4.1 Random Replay Attacks (UC01) . . . . .	64
5.2.4.2 Inverse Replay Attacks (UC02) . . . . .	64
5.2.4.3 Masquerade Attacks towards Outage (UC03) . . . . .	64
5.2.4.4 Masquerade Attacks towards Equipment Damage (UC04) . . . . .	65
5.2.4.5 Message Injection (UC05) . . . . .	65
5.2.4.6 High Status Number Attack (UC06) . . . . .	66
5.2.4.7 High-Rate Flooding Attack (UC07) . . . . .	66
5.3 Feature Extraction and Enrichment . . . . .	66
5.3.1 SV Features . . . . .	67
5.3.2 Enriched SV Features . . . . .	68
5.3.3 Basic GOOSE Features . . . . .	70
5.3.4 Enriched GOOSE Features . . . . .	71
5.4 Feature Selection: GRASP-FS . . . . .	72
5.4.1 Construction Phase . . . . .	74
5.4.2 Local Search Phase . . . . .	75
<b>6 Experiments, Results and Discussion</b>	<b>78</b>
6.1 Feature Processing: Selection and Enrichment . . . . .	78
6.1.1 Feature Selection . . . . .	78
6.1.2 Feature Enrichment . . . . .	79

---

6.1.3	Results for Feature Processing . . . . .	80
6.1.3.1	Random Replay attacks (UC01) . . . . .	80
6.1.3.2	Inverse Replay Attacks (UC02) . . . . .	80
6.1.3.3	Masquerade Attacks towards Outage (UC03) . . . . .	81
6.1.3.4	Masquerade Attacks towards Equipment Damage (UC04) . . . . .	82
6.1.3.5	Random Message Injection (UC05) . . . . .	82
6.1.3.6	High-Status Number Attack (UC06) . . . . .	83
6.1.3.7	High-Rate Flooding Attack (UC07) . . . . .	84
6.2	Dataset Features Assessment . . . . .	84
6.3	Multi-class . . . . .	90
6.3.1	Analyzing the Confusion Matrices . . . . .	91
6.4	Results Discussions and Lessons Learned . . . . .	94
<b>7</b>	<b>Conclusion</b> . . . . .	<b>97</b>
7.1	Contributions . . . . .	98
7.2	Publications . . . . .	98
7.2.1	Production of Direct Results . . . . .	99
7.2.2	Production on Correlated Themes . . . . .	99
7.3	Open Issues and Future Works . . . . .	102
	<b>REFERENCES</b> . . . . .	<b>105</b>

# 1 Introduction

Smart Grids are intended to integrate communication technologies with the traditional power grid system ([RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019a](#)). The integration of the power grid infrastructure with communication networks has brought many possibilities for digital substations. It was responsible for enabling the development of novel applications such as automated data acquisition, remote control and monitoring of electrical infrastructure, services, and components ([QUINCOZES; ALBUQUERQUE, et al., 2021](#); [FAISAL et al., 2014](#); [HONG; LIU, 2019](#)).

As a result of the growing number of devices connected to Smart Grids, new communication protocols have emerged in recent years. In particular, in the context of automated substation communication, the IEC–61850 ([COMMISSION, 2019](#)) standard defines important globally established standards for substation automation, revolutionizing the way substations are configured and maintained ([HONG; LIU, C.-C.; GOVINDARASU, 2014](#)). Thus, communication between devices from different manufacturers is standardized through protocols with a well-defined structure. Some of the benefits include the potential to reduce errors and misconfigurations.

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. However, our studies show that there is a range of vulnerabilities that can compromise IEC–61850 communication protocols and cause substation physical elements to malfunction. Therefore, despite the IEC–61850 advantages, the integration poses numerous security challenges for industrial systems. From the perspective of information security, it exposes digital substation networks to various threats, such as the injection of improper messages (replay of previously sent messages or new fabricated messages) into the network. Such malicious practices are real threats that can lead to catastrophic damage ([POPOVIC et al., 2016](#); [ELGARGOURI; ELMUSRATI, 2017](#); [HONG; LIU, 2019](#)). Recently, hundreds of attacks have caused power outages, affecting hundreds of thousands of people in countries like the United States of America (USA) and Ukraine ([HONG; LIU, C.](#);

GOVINDARASU, 2014).

Besides the communication improvements resulting from the adoption of well-defined protocols, there are particular factors that favor the execution of attacks and make their detection process difficult. One of these factors is the resource limitations that Intelligent Electronic Device (IED) typically have – they limit the implementation of traditional security mechanisms, such as complex encryption algorithms (YANG; MCLAUGHLIN, et al., 2016; RASHID et al., 2014; HOYOS; DEHUS; BROWN, 2012).

Therefore, studies to understand the threat vector of these scenarios and the implementation of adequate means for intrusion detection are fundamental. The adoption of Intrusion Detection Systems (IDSs) is crucial to protect any device deployed in automated substations. Such systems must be able to detect traditional threats, inherited from information technology and network protocols, and specific attacks directed at intelligent substations, resulting from the exploitation of new protocols and adopted devices.

However, research on IDSs for automated substations, such as those based on IEC-61850, is still at a preliminary stage. Although intrusion detection techniques are commonly studied in conventional networks and systems, only a few studies address this problem, considering the requirements, limitations, and specific communication protocols of IEC-61850 substations. Our studies reveal that the lack of realistic data for training, testing, and evaluating IDSs in realistic industrial settings is considered a major challenge. The scarce realistic data available for assessing novel IDS proposals, as well as the lack of enriched features for enabling an accurate intrusion detection are key issues on this field. As a consequence, the development of IDSs is currently limited by the available datasets.

## 1.1 Motivation

Cyberattacks on communication systems in digital substations are not only considered one of the main threats by researchers in the field (HONG; LIU, 2019) but also present real threats that have already been witnessed by people from many countries around the world (KANG; MCLAUGHLIN; SEZER, 2016; PATEL, 2017; RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019a).

Until the beginning of 2000, Supervisory Control and Data Acquisition (SCADA) system networks were assumed to be electronically isolated from the rest of the networks and, therefore, the industry’s focus was on the physical security of the network (PATEL, 2017). In 2010, *malware Stuxnet* attacked Iran’s nuclear program (KANG; MCLAUGH-

LIN; SEZER, 2016; PATEL, 2017; RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019a). *Stuxnet* specifically targeted Iranian Programmable Logic Controller (PLC) and caused fast-spinning centrifuges to separate. According to Patel *et al.* (PATEL, 2017), this was one of the main incidents caused by cyber attacks that prompted a perception of the urgent need to provide security in the communication network of SCADA systems, including digital electrical substations.

In addition, USA's computer systems were compromised by more than 150 cyber attacks between 2010 and 2014. Between 2011 and 2014, utilities reported 362 instances of attacks that caused blackouts or other power outages – among of these attacks, 14 were cybernetic and the rest were physical (HONG; LIU, 2019; RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019a). In 2016, an alert was issued about coordinated cyber attacks on 35 Ukrainian substations. As a result, more than 225,000 people were left without electricity (HONG; LIU, 2019).

Information security vulnerabilities are continually growing. The US National Vulnerabilities Database (NVD) (NIST, 2021) registered a growth from 6,447 vulnerabilities in 2016 to 20,138 vulnerabilities in 2021.

Both industry and academia are concerned about information security in electrical substations. In fact, this is one of the main concerns about smart grids. The provision of security and robustness in this domain is limited due to the computing capacity of the equipment, which does not support the mechanisms used to protect traditional networks. Therefore, the employment of detecting measures to enable responding to attacks is crucial to leverage the potential of the secure integration of computing and communications protocols and devices.

Finally, the use of IDSs will play a vital role in ensuring the correct functioning of electrical substations. Such systems must be capable of blocking improper actions, as well as alerting network operators about possible attacks in progress, in order to enable decision-making and strengthening of security policies and mechanisms. Such systems can be based either on the implementation of new *software* applications, or on the use of modern *hardware* platforms that can extend the processing capabilities with a low impact on the infrastructure cost.

## 1.2 Research Problem

Deploying IDSs serves to identify malicious activities and mitigate the attacker 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 Substation Events (GOOSE) and Sampled Values (SV), specialized attacks targeting them imply different traffic and attack patterns. The specialized attacks are typically based on traditional attacks but they explore particularities from the IEC-61850 communication protocols. Attacks may include replay attacks (HONG; LIU, C.; GOVINDARASU, 2014), message injection (YOO; SHON, 2015), masquerade attacks (USTUN; FAROOQ; HUSSAIN, 2019), and DoS attacks (HOYOS; DEHUS; BROWN, 2012).

As consequence, IEC-61850-based IDSs require attack signatures from multiple attack classes to enable a robust detection based on the knowledge extracted from the traffic patterns (*e.g.*, the machine learning-based IDSs need data to perform training, testing, and assessment of their performance; the rule-based IDSs need data for extracting detection rules; and, the anomaly-based IDSs need data to assess their ability to identify anomalies).

The investigated research problem in this thesis focuses on the lack of available datasets for intrusion detection in communication systems and networks in the context of electrical substations. This is because, to enable a robust classification model for machine learning attack classifiers in digital substations, signatures should be built by considering realistic data, including electrical samples from SV and proper response of GOOSE protocols.

Furthermore, it is clear that, to represent the realistic traffic behavior inside of a substation or of a transmission line between two or more substations, both normal operation and transmission line faulty scenarios must be considered. In both situations, a variety of updated attack classes, as well as legitimate activities, must be considered.

Building a robust IDS requires not only the availability of data that includes realistic attack scenarios but also the identification of which features are more representative for detecting each attack class. Therefore, extraction, enrichment, and selection of features are also necessary to robust intrusion detection.

## 1.3 Goals

Aiming at robustness and resilience in smart power grids, specifically in IEC–61850 digital substations, our key goals are to support the training, evaluation, and testing of IDSs. In the following, the general objective and the specific objectives are elucidated.

### 1.3.1 General Goal

Our main goal is to create an extensible tool for generating realistic and representative IEC–61850 datasets with different types of intrusions for training, testing, and evaluating IDSs. Furthermore, we aim at proposing a novel feature selection method based on metaheuristics to perform feature selection on the generated dataset.

### 1.3.2 Specific Goals

Our specific goals are the following:

- The study of the current attack scenarios targeting IEC–61850 systems and the state-of-art IDSs solutions;
- The proposal of a novel taxonomy for the IEC–61850-based IDSs aspects;
- The reproduction of normal and faulty scenarios through the simulation based on the modeling of a real transmission line between two substations to generate and log realistic electrical samples;
- An extensible tool for generating realistic GOOSE and SV traffic features, based on realistic electrical signals, with support to state-of-art attacks targeting the GOOSE protocol (SV and Manufacturing Message Specification (MMS) attacks are beyond the scope of this work);
- A set of 8 public IEC–61850 datasets to support the training, testing, and evaluating of IDSs;
- Algorithms and methods to extract, enrich and select representative features from electrical domain and computer networks to maximize the results of IDSs.

## 1.4 Organization

The remainder of this document is organized as follows. In Chapter 2, we present the theoretical fundamentals for the IEC-61850 Standard and Feature Selection for IDS. In Chapter 3, we present related works that address Intrusion Detection methods and datasets for substations and feature selection methods – both filter and wrapping. In Chapter 4, we investigate the research problem, covering attacks to IEC-61850 protocols and the state-of-art IDSs aspects. Then, in Chapter 5, we present our hypothesis and the proposed solution that includes feature generation and selection. In Chapter 6, we present our experiments, results, and discussion. Finally, in Chapter 7 we present our conclusions, pointing our contributions, resulting publications, and future work.



# 2 Background

## 2.1 The IEC–61850 Standard

Legacy substation protocols defined communication conventions between hard-wired electrical devices and monitoring/control components (MACKIEWICZ, 2006). In contrast, the IEC–61850 standard (COMMISSION, 2019) was defined with the following goals: (i) interoperability; (ii) long term stability; and (iii) simplified configuration. Besides the structure of transmitted data and interoperability aspects, the IEC–61850 standard specifies the physical topology (*e.g.*, ring topology, redundant LANs), network protocols, and object modeling (COMMISSION, 2019; O’RAW; LAVERTY; MORROW, 2017).

### 2.1.1 Physical Topology

A typical substation infrastructure is composed of three levels: *station*, *bay*, and *process* (or *field*). Each of those levels is illustrated in Figure 2 and contains devices with different capabilities. Two communication channels are used in these levels, allowing both horizontal (*i.e.*, between devices of the same level) and vertical (*i.e.*, between devices of different levels) communication (AHMED et al., 2019; KABIR-QUERREC et al., 2015; EL HARIRI et al., 2017).

The *station level* provides the interface for humans managing the substation, and includes monitoring systems, engineering workstations, SCADA systems, and the *Remote Terminal Unit* (RTU). The RTU includes remote access, opening an entry point for remote attackers. Thus, it is imperative to employ security mechanisms to deal with potential threats (AHMED et al., 2019).

The *bay level* consists of an intermediate level where automatic functions with real-time requirements are performed without the need for human intervention. It includes IEDs for control and metering, protection, and time synchronization (HAHN; SUN; LIU, 2016). These devices are connected to both the *substation* and *process busses* that link,

respectively, the station and bay levels and the bay and field levels, enabling station level devices to perform operations, such as reading and writing from bay level IEDs (AHMED et al., 2019).

The lowest level, the *process level*, includes both conventional and non-conventional switchyard equipment<sup>1</sup> from the electrical domain. Because conventional equipment supports only dedicated wired data, additional elements may be employed to act as an interface between the cyber and physical domains, such as *Merging Units* (MU) and Actuators (*e.g.*, Intelligent Terminals).

In particular, MUs play a significant role in digital substations. They provide synchronized phase, voltage, and current measurements collected from primary conventional equipment. These devices use the process bus to communicate, thus increasing the data availability and reducing wiring, as Ethernet replaces hard-wired connections. Note that this level carries extremely sensitive and time-critical applications (IEC, 2004) and thus cybersecurity is one of the most crucial challenges, especially considering the limited processing power of MUs that turn simple security measures, such as data encryption, impracticable (EL HARIRI et al., 2017). Future power grid systems are expected to have non-conventional equipment (*e.g.*, modern switchgear) capable of supporting communication protocols without depending on intermediate sensors and actuators (COMMISSION, 2019). A typical infrastructure topology involving both conventional and non-conventional devices is illustrated in Figure 2.

## 2.1.2 Network Communication Protocols

### 2.1.2.1 SV Protocol

The SV (Sampled Values) protocol is defined by the IEC–61850–9–2 (IEC, 2004) standard to enable digitized current and voltage samples to be transmitted to IED using the *Ethernet* protocol through the Publish/Subscribe paradigm, where publisher devices send multi-cast messages to subscriber devices (*i.e.*, control and/or protection IEDs). Note that unicast messages are also supported. Such measurements are collected through analog signals from electrical equipment and converted to digital signals by Merging Units (MUs) and transmitted to subscriber devices.

SV messages are consumed for different SV applications, such as the protection applica-

---

<sup>1</sup>Switchyard refers to an enclosed area of a power system containing the switching equipment used in the transmission of electricity.

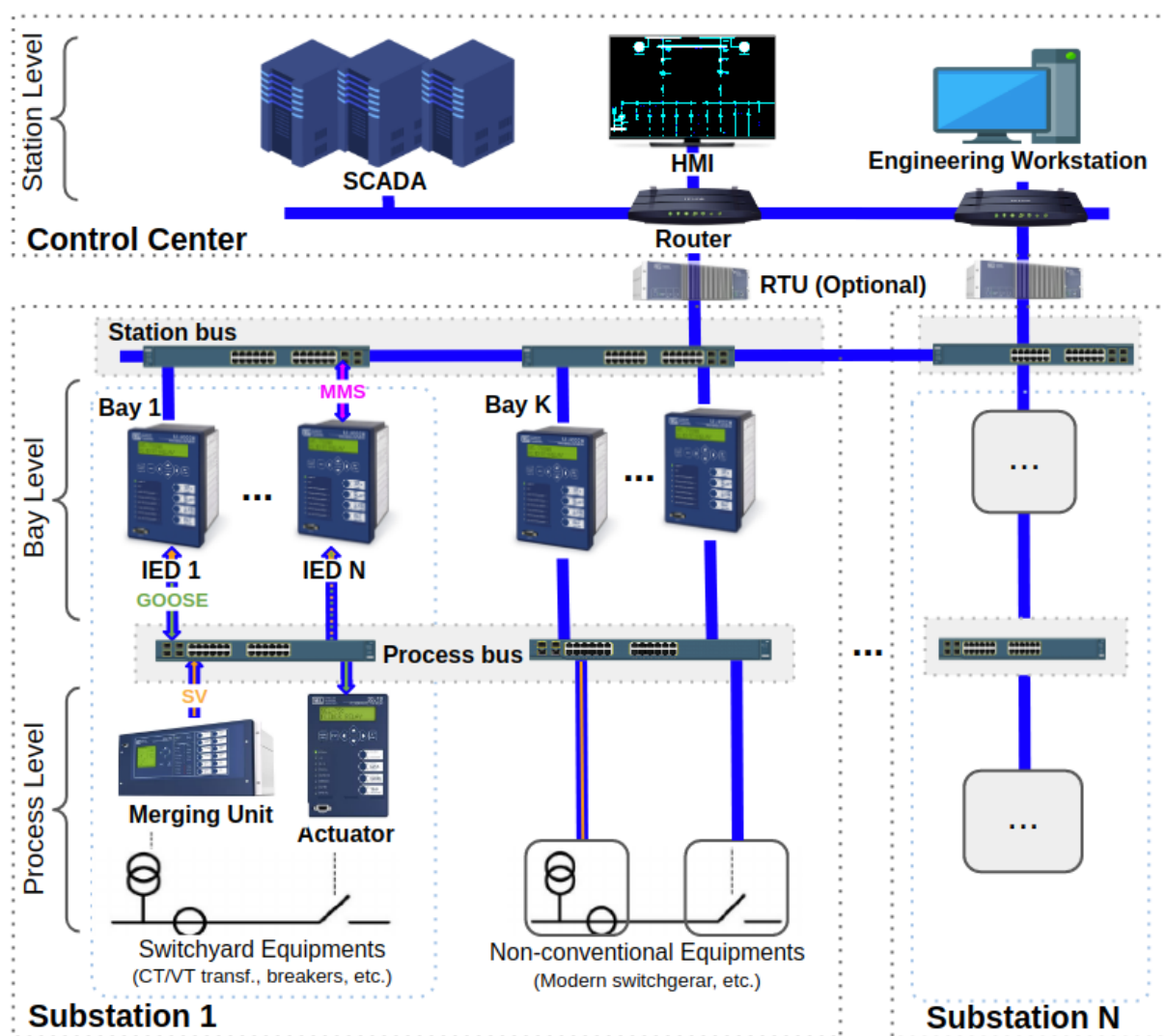


Figure 2: IEC-61850 Typical Architecture.

tions. These applications process the SV payload to detect faults based on their protection schemes (HONG; LIU, 2019). Therefore, to ensure a fast response-time, SV messages should be 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<sup>2</sup>, while measurement applications, which require more accurate data, the rate is 256 samples per cycle (HONG; LIU, 2019; KARIYAWASAM; RAJAPAKSE; PERERA, 2017). The voltage and current measurements are put into the Application Service Data Units (ASDU) field. Thus, the number of samples per cycle is proportional to the number of ASDUs transmitted in each SV message: eight ASDUs are sent in measurement messages and one ASDU is sent into protection messages (SOLOMIN; TOPOLSKY; TOPOLSKY, 2015).

<sup>2</sup>A cycle represents 16.6ms for substations operating at 60Hz or 20ms for substations operating at 50Hz.

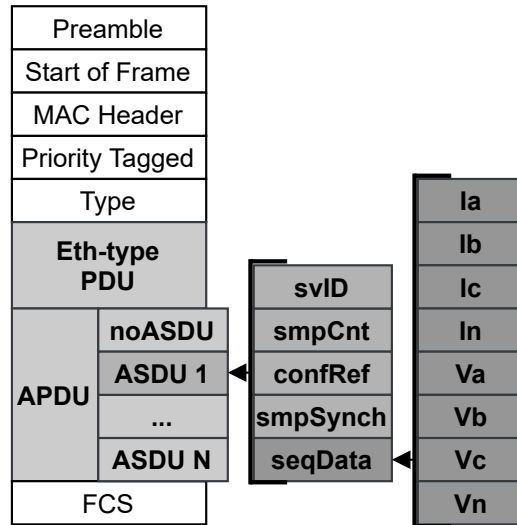


Figure 3: Ethernet frames of SV messages.

In Figure 3, the Ethernet frame structure and the internal structure of the Application Protocol Data Unit (APDU), from an SV message are illustrated. All shaded items are part of the SV message, the remaining are generic ethernet fields. Except for protection applications, in which there is a single ASDU, each transmitted message may contain multiple ASDUs within the Sequence of ASDU field – each with the same structure as illustrated in ASDU 1.

As shown in Figure 3, each ASDU carries four current measures (*i.e.*, top rows  $I_a, I_b, I_c, I_n$ ) and four voltage measurements (*i.e.*, bottom rows  $V_a, V_b, V_c, V_n$ ) in the *Sequence of Data* (seqData) ASDU field, referring to the four electrical phases (A, B, C, and Neutral). Each of them may be a potential target for attackers (*e.g.*, fake measurement injection or data manipulation).

### 2.1.2.2 GOOSE Protocol

The GOOSE protocol enables the IEDs to exchange messages to report substation events, status changes notification, alarms, and control commands. Events of different components are exchanged by GOOSE messages, including temperature alarms, circuit-breaker status, disconnecter switch interlocking, etc. These data are put into a field named *GOOSE datSet*, inside the GOOSE PDU, and transmitted by 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. Figure 4 shows the GOOSE Ethernet frame structure.

In stable situations, in which no events occur (*i.e.*, no changes are detected in the

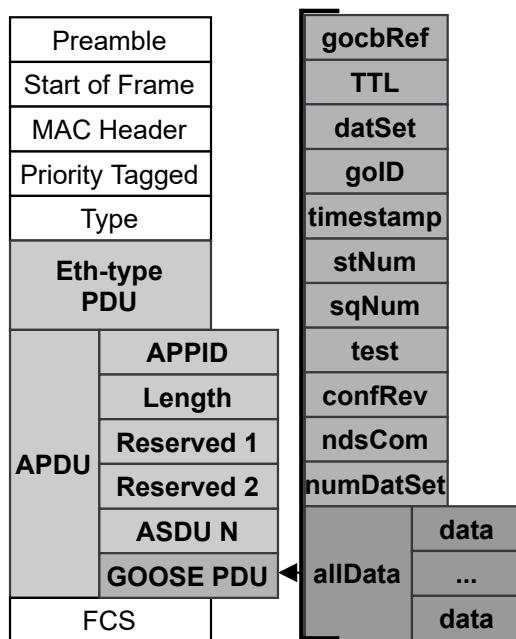


Figure 4: Ethernet frame of GOOSE messages.

GOOSE dataset values), a GOOSE message is transmitted at a fixed  $T_0$  period, with an increased sequence number ( $sqNum$ ); this ensures that the component is alive and it does not have different values to transmit. Once an event occurs, the  $sqNum$  field is set to zero and the status number ( $stNum$ ) is increased and a new GOOSE message is sent immediately. This message is retransmitted at increasingly larger intervals, each with an increased ( $sqNum$ ), starting with the shorter transmission interval ( $T_1$ ), which is used as interval between the two first messages, and increasing at every retransmission ( $T_2$ ,  $T_3$ , etc.), until reaching the original interval ( $T_0$ ), as illustrated in Figure 5. Although this increasing method is not standardized, exponential backoffs are typically adopted (HOYOS; DEHUS; BROWN, 2012).

Therefore, by considering these standardized behaviors, there are different features that may be analyzed by an IDS to distinguish legitimate and malicious activities. The GOOSE timestamp may reveal a DoS attack, since in normal conditions it is expected that the interval between two received messages should not exceed  $T_0$ . The  $sqNum$  and  $stNum$  are relevant features because they are examples of potential indicators of fake message injection or message replay attacks (HOYOS; DEHUS; BROWN, 2012; HONG; LIU, C.; GOVINDARASU, 2014; USTUN; FAROOQ; HUSSAIN, 2019) – although other fields may also be important, according to each attack type.

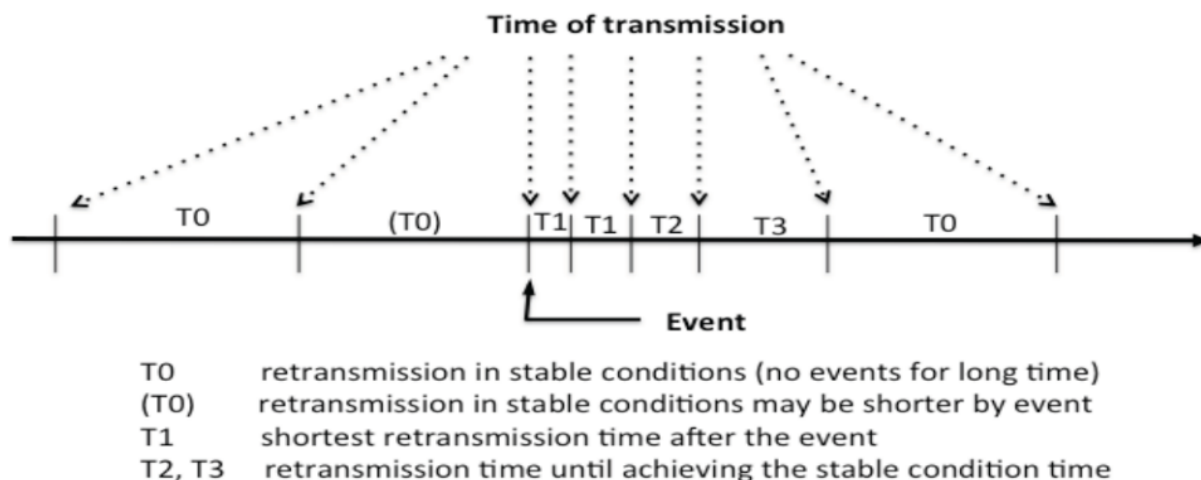


Figure 5: GOOSE messages transmission. Extracted from (HOYOS; DEHUS; BROWN, 2012).

## 2.2 Feature Selection for IDSs

A feature is an individual measurable property of the process being observed (CHANDRASHEKAR; SAHIN, 2014). In intrusion detection, features usually represent data collected from the physical level (*e.g.*, measurements of current and voltage electrical signals, for example), the network level (*e.g.*, values contained in the fields of network communication protocols) or even from the application level (*e.g.*, user activity logs or process metrics on a device).

Given the vast amount of features that can exist in each of the different domains of computer systems, many of them are not representative to describe specific attack patterns. Therefore, it is important to assess and extract only the representative features for each situation to allow accurate predictions. In digital electrical substation networks, there may be different processes that represent potential targets for intrusions. One of the main attackers' targets is the substations' protection (or teleprotection) system, which is responsible for protecting expensive equipment from damage caused by overloads in case of electrical faults. The protection system is also responsible for isolating segments, in case of natural faults, caused by weather conditions, for example. Thus, as it is a critical process, attackers can strategically target attacks on the protection system. In IEC-61850 networks, GOOSE and SV protocols can provide rich information for machine learning algorithms. Thus, these algorithms can process their representative features to detect suspicious activities targeting such applications.

Using a set of features, machine learning algorithms developed to perform data classi-

fication can be trained to distinguish between malicious and benign behaviors. In recent years, in machine learning or pattern recognition applications, the domain of features has expanded from tens to hundreds of features. In particular, the intrusion detection procedure consists of analyzing data relevant to identifying threats and detect malicious activities. Therefore, one of the fundamental steps of any IDS is to define which features will be analyzed (KALEEM; FERENS, 2017). This process is called Feature Selection. According to G. Chandrashekar and F. Sahin (CHANDRASHEKAR; SAHIN, 2014), the focus of feature selection is to improve the prediction results as it selects a subset of input variables that can efficiently describe the input data while reducing the effects of noise or irrelevant variables.

We can also define Feature Selection as the process that defines the input data for an entire intrusion detection analysis workflow. Thus, for an IDS to achieve good results, the proper selection of the most relevant features is crucial. The Feature Selection process must not only select the representative information but also discard redundant features. In particular, the data collected from an electrical substation may contain hundreds of features – many of which can be highly correlated with other features. When two or more features are perfectly correlated, only one of them is sufficient to describe the desired pattern. The highly correlated features do not provide extra information about the classes and therefore serve as noise for the predictor. This means that the total information content can be obtained from fewer unique features that contain maximum discrimination information about classes. Therefore, by eliminating the dependent features, more information may be given with less data being processed. Consequently, it can lead to an improvement in intrusion detection performance — both in terms of time and other detection metrics.

Furthermore, there are features that may not be correlated with the classes of the analyzed data. These features serve as pure noise and can introduce predictor bias and reduce classification performance. This can happen when there is a lack of information about the process under study. By applying feature selection techniques, we can gain some insights into the process and improve the calculation requirement and forecast accuracy.

To perform feature selection, it is necessary to use some criteria for measuring the relevance of each feature. Once a feature selection criterion is selected, a procedure must be developed to find the useful and efficient subset of features. Directly evaluating all subsets of features for a given data is NP-hard, and becomes unwieldy as the number of features increases. Therefore, a suboptimal procedure must be used, which can remove



redundant data in an efficient way.

Besides improving the performance of the detector algorithms, the selection of attributes reduces the consumption of computational resources, thus contributing to the completion of analyses in a timely manner for the application of countermeasures (GANAPATHY et al., 2013).

There are three main approaches to perform feature selection:

- *Embedded Feature Selection*: Some data mining algorithms (e.g., REP Tree and J48) discard irrelevant attributes implicitly. This implicit discarding is characterized as *embedded* feature selection, as it is embedded in the implementation of the classifier method. The main approach used by these algorithms is to incorporate the feature selection as part of their training process (QUINCOZES; KAZIENKO; COPETTI, 2018).
- *Filtering Feature Selection*: There are, however, many algorithms that do not perform any technique with the purpose of eliminating noise or redundancy in the analyzed attributes. Thus, a lightweight way to perform a feature selection is the use of *Filter* methods, which are independent of machine learning algorithms and determine the importance of each attribute uniquely based on statistical methods such as the entropy (LI et al., 2017). However, as this method does not interact with the detection algorithm itself, the detection performance may be affected.
- *Wrapping Feature Selection*: To overcome the lower detection performance presented by the *Filter*-based methods, the same machine learning algorithm that will be used to process the features to make decisions can be used to assess the candidate features, during the feature selection process, and give insights about which of them can improve its detection performance. This way, those features that better improve the chosen intrusion metric will compose the output feature subset (QUINCOZES; KAZIENKO; COPETTI, 2018). The main drawback of *wrapping-based* methods is the longer time needed to process multiple subsets with different feature combinations.

In summary, large data sets make it difficult to perform feature selection. Thus, even for those algorithms that already include feature selection in their implementation, a pre-selection of the input information to give a reduced set of features whenever possible may reduce the machine learning processing complexity. There are multiple techniques that



can be applied to feature selection ([AGRAWAL et al., 2021](#); [COSTA; LIMA; BARBOSA, 2021](#)).

As each method has its advantages and drawbacks, in the next Chapter, in Section [3.2](#), we present the state-of-art methods for feature selection, both for *Filter* (Section [3.2.1](#)) and *Wrapper* (Section [3.2.2](#)) methods. In this thesis, filter and wrapping algorithms are combined, as discussed in Section [5.4](#).

# 3 Related Work

## 3.1 Intrusion Detection and Datasets

The most popular datasets for intrusion detection are designed considering the traditional computer networks traffic. They include the KDDCup99 (STOLFO et al., 1999), NSL-KDD (TAVALLAEE et al., 2009), UNSW-NB15 (MOUSTAFA; SLAY, 2015), and CICIDS2017 (SHARAFALDIN; LASHKARI; GHORBANI, 2018). 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 in (BISWAS et al., 2019). However, the authors of (BISWAS et al., 2019) explore a few signatures of simple injection attack by modifying the GOOSE *StNum*, *SqNum*, and a boolean value in the GOOSE dataset. Therefore, the current works in the literature are often based on generic traditional datasets or self-generated traffic. Below, we discuss how the current IDS proposals are assessed in IEC-61850 scenarios.

### 3.1.1 Generic Traffic

In (PREMARATNE et al., 2010) authors deploy an IDS to detect anomalous behavior into an IEC-61850 network. Nevertheless, although they argue the network is IEC-61850-based, the attacks experimented are generated through tools that target traditional protocols such as HTTP, FTP, and telnet.

Other works (YANG; HAO, et al., 2019) use the KDD Cup 99' dataset to assess an IDS for IEC-61850, which not only contains just attacks targeting a traditional network, but is also outdated.

*Geek Lounge* (LOUNGE, 2019) and EPIC (ADEPU; KANDASAMY; MATHUR, 2018) labs built testbeds for assessing industrial networks, however they did not consider the scenario of automated protection and control in modernized substations. Consequently, GOOSE and SV protocols are not used and no data is available about them.

### 3.1.2 Self-generated Traffic

In (HONG; LIU, C.; GOVINDARASU, 2014), the authors use the well known IEEE 39-bus system (CUPELLI; CARDET; MONTI, 2012) 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 (HONG; LIU, C.-C.; GOVINDARASU, 2014) 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, but, unfortunately, the data used by them is not made publicly available. Thus, as mentioned above, the results are not reproducible and impossible to compare with.

In (KWON et al., 2015), the authors propose a behavior-based IDS. To evaluate it, they prepared one week of network traffic of a digital substation as dump file and replicated cyber attacks against a substation LAN, simulated through a testbed. In fact, the authors claim that the lack of openly available intrusion detection datasets is a significant challenge. Nevertheless, they do not make their data publicly available.

In (YOO; SHON, 2015), a study suggests a method of anomaly detection for Manufacturing Message Specification (MMS) and GOOSE protocols. However, the authors did not consider any attack traffic. The one-class Support Vector Machine (SVM) algorithm is used to detect normal behavior, but the authors argue that, to evaluate the ability to detect attacker behavior, a simulation should be developed.

In (YANG; MCLAUGHLIN, et al., 2016), the authors simulate a smart 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. However, their assessment is not reproducible since the data used is not publicly available.

In (KANG; MCLAUGHLIN; SEZER, 2016), 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. Despite the analysis being based on a open-source IDS named Sucirata, the resulting dataset generated from their testbed is not made available.

In (YANG; KECKALO, et al., 2017), the authors present novel testing concepts and methods to assess MUs accordance with protection application requirements. These devices are responsible for converting analogical data into digital SV messages. Thus, one goal of the authors is to check the SV messages integrity. However, data are not made

available.

In (YANG; MCLAUGHLIN, et al., 2016), an IEC-61850-based IDS is implemented to analyse GOOSE, MMS, and SV traffic. The authors use the ITACA network analyzer tool and specification rules to detect intrusions. That work is further extended (YANG; XU, et al., 2016), but in both versions no traffic or application logs is made available.

In (HONG; LIU, 2019), 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 are put together into a testbed to generate realistic traffic. Then, injection attacks were simulated by modifying and retransmitting fabricated GOOSE packets to the network. Unfortunately, the data used for evaluating the proposed IDS is not made publicly available.

Besides the aforementioned literature, there also commercial tools such as Dragon IDS (SANS, 2002) and Continuous Threat Detection (CTD) that can be used for performing intrusion detection. Dragon IDS has three components: (i) Dragon Sensor, that is a network IDS that monitors network packets. (ii) Dragon Squire, that is a host-based IDS that monitors key system files – it can also receive security information from routers and firewalls via SYSLOG or SNMP; and, (iii) Dragon Server, that manages data from all of the Dragon Sensor and Dragon Squire engines. Continuous Threat Detection (CTD) proposed by Claroty monitors the network for early indicators of attack (e.g., DNS scans and failed login attempts), behavioral anomalies (e.g., atypical communication between devices), activities defined by custom rules, and signatures defined by Snort and YARA tools.

## 3.2 Feature Selection Methods: State of the Art

In addition to the data source definition, feature selection methods to find sub-optimal subsets (*i.e.*, near to the optimal solution) and increase the intrusion detection performance are necessary.

### 3.2.1 Filter-based Feature Selection

One of the simplest feature selection method is the Information Gain (IG) (YANG; CHUANG; YANG, et al., 2010), obtained through the calculation of entropy. Equa-

tion 3.1 shows this calculation, where  $p_i$  is the probability that an instance in the set  $C$  belongs to a given class and  $m$  is the number of available features.

$$Info(C) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (3.1)$$

The IG is proportional to the entropy reduction, therefore, the features with lower entropy tend to be selected for presenting a greater information gain. Decision tree induction algorithms use this technique to discard unnecessary information that could negatively influence the results.

The Chi-square method (LANCASTER; SENETA, 1969), aims to quantitatively evaluate the association between features of two categories. The basic idea of this method is to establish two hypotheses. The hypothesis  $H_0$  says that there is no association between the analyzed feature and the class. On the other hand,  $H_1$  is to say that there is an association, that is, the feature has a dependency on the class.

$$x_2(f_i, y_j) = \frac{N(TZ - YX)_2}{(T + X)(T + Z)(X + Z)(Y + Z)} \quad (3.2)$$

Equation 3.2 illustrates the chi-square test, where  $T$  is the frequency of feature ( $f_i$ ) and class label  $y_j$  in the dataset,  $X$  is the frequency of  $f_i$  appearing without  $y_j$ ,  $Y$  is the frequency of  $y_j$  appearing without  $f_i$ ,  $Z$  is the frequency of neither  $y_j$  nor  $a_i$  appearing together in the dataset,  $N$  is the total number of records (THASEEN; KUMAR; AHMAD, 2019).

The correlation-based feature selection approach, *Correlation-based Feature Selection* (CFS) (HALL, 1999), consists of constructing correlation matrices *Feature*  $\times$  *Feature* and *Feature*  $\times$  *Class*. From these matrices, the weight of each set of features is calculated using the Equation 3.3, where the merit of a set  $S$ , which contains  $k$  features, is calculated based on the average of the correlation between feature-class  $r_{fc}$  (i.e., represented by  $\bar{r}_{fc}$ ) and the mean between feature-feature  $r_{ff}$ .

$$Merit(S) = \frac{k \times \bar{r}_{fc}}{\sqrt{k + k(k - 1)r_{ff}}} \quad (3.3)$$

Based on this equation, the ratio between the predictive capacity and the degree of redundancy of each set is possible. Initially, the CFS method has an empty set. Then, it search features using the *best-first-search* heuristic (they use a stopping criteria of 5

consecutive sets without merit improvements). Finally, the set with the greatest merit should be selected.

There are feature selection approaches that consist of implementing machine learning mechanisms through Artificial Neural Network (ANN). This type of methodology is aimed at pruning agents that carry redundant or unimportant information for classification. In this context, different algorithms in particular can be implemented, such as cognitive models (KALEEM; FERENS, 2017) or learning algorithms based on feature weighting (AMINANTO et al., 2017), where the feature weights represent their importance.

### 3.2.2 Metaheuristics as Wrapping Methods for Feature Selection

The Greedy Randomized Adaptive Search Procedure (GRASP) (QUINCOZES; PASSOS, et al., 2020) is an iterative multi-start metaheuristic proposed originally to achieve an approximate solution for graph planarization problems. It has since been generalized to solve other combinatorial optimization problems in different domains (YUSTA, 2009). The general GRASP strategy can be carried in two iterative phases: *construction* and *local search*. These steps are repeated until a stop criterion is reached (*e.g.*, the maximum number of total iterations). For each iteration, a different random seed solution is generated. The construction phase relies on this seed to generate a Restricted Candidate List (RCL) and choose from it a randomized greedy solution. The RCL consists of a predefined number of candidates to compose a feasible solution. Typically, this initial greedy solution can be improved by a local search phase, that is, the greedy solution is the starting point for a neighborhood exploration.

Yusta (YUSTA, 2009) demonstrates that GRASP can outperform Sequential Forward Floating Selection (SFFS), Tabu Search, Genetic and Memetics algorithms in generating the best feature subset to classify samples from different databases (Spambase, Waveform, Ionosphere, Vehicle, Wincosin, and German). Esseghir and Amir (ESSEGHIR, 2010) also apply GRASP for feature selection considering some of those datasets (Ionosphere and SpamBase) and others (Sonar, Audiology, and Arrhythmia). Whereas the former work used the K-Nearest Neighbors (KNN) classifier for wrapping evaluations, the latter explored Artificial Neural Networks — neither compares the used classifier with alternative algorithms.

Bermejo et al. (BERMEJO; GAMEZ; PUERTA, 2011) use GRASP to deal with large and high-dimensional datasets with focus on reducing the number of wrapping evaluations (*i.e.*, those that rely on machine learning algorithms) by using the Incremental Wrapper

Subset Selection (IWSS) algorithm in the construction phase. The main drawback of this approach is the possibility of excluding important features due to a premature stop caused by the current solution exceeding the solution cardinality threshold.

Moshki *et al.* (MOSHKI; KABIRI; MOHEBALHOJEH, 2015) also use GRASP to deal with large and high-dimensional datasets. They combined GRASP with an extended version of Simulated Annealing as a local search procedure to introduce new parameters that allow the implicit weighting between accuracy and execution time. However, this method is limited to the implicit trade-off control between these parameters. It is still unable to control it explicitly. Besides, using accuracy as the target metric is not suitable for unbalanced datasets because it gives the same weight for all classes (*e.g.*, attacks and benign).

Diez *et al.* (DIEZ-PASTOR; GARCIA-OSORIO, *et al.*, 2011) propose GRASP Forest (G-Forest) for constructing ensembles of decision trees. This method uses concepts of GRASP for both feature selection and choosing splitting points at each tree node. For selecting features for each level of the tree, G-Forest assembles an RCL composed of all features with an *Information Gain* above a certain threshold. It then proceeds to choose features randomly from this RCL. Once a feature is chosen, G-Forest computes the IG of each possible split point for that feature and forms a new RCL following the same strategy (*i.e.*, using those features with an IG above a certain threshold). The split point is then chosen randomly from this list. This procedure is repeated until the desired number of trees is created.

In a later work, Diez *et al.* (DIEZ-PASTOR; GARCIA-OSORIO; RODRIGUEZ, 2014) extend their proposed G-Forest to create an Annealed Randomness Forest (GAR-Forest). The key idea is to introduce a parameter that controls the randomness during the solution construction phase, which ranges from an entirely random procedure to a totally greedy one. The authors consider 62 datasets (including Waveform, Ionosphere, and Vehicle). However, none of those are related to the cyber security domain. Kanakarajan and Muniyasamy (KANAKARAJAN; MUNIASAMY, 2016) apply GAR-Forest to detect DoS, Probe, R2L and U2R attacks in traditional networks. The reported results reveal that F1-Score and accuracy are both slightly over 85%, which means that performance should be improved.

In our previous work (QUINCOZES; PASSOS, *et al.*, 2020), we experimented with an adapted version GRASP for FS, namely GRASP-FS. We used GRASP-FS to select three feature subsets for building expert IDSs for the flooding, blackhole, and grayhole

attacks (each IDS was specialized in one attack class). The results reveal that the selected features outperform those selected by traditional filter-based approaches to detect attacks targeting the CPS perception layer.

### 3.3 Discussion

To the best of our knowledge, most of the existing proposals report results (from proprietary datasets) that can not be reproduced, compared, nor assessed under different parameters or scenarios. Most existing IEC-61850 IDS proposals that include GOOSE and SV use data collected from private testbeds rather than using public datasets. Besides, this is a particularly complex approach since it requires more effort to reach a consistent scenario and to ensure the result's correctness according to the real-world substations traffic.

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.

Therefore, the proposed traffic generation tool addresses an important challenge since it enables the generation of realistic data for training, testing, validating, and evaluating IDSs for digital substations. We implemented the most common attack models that target the IEC-61850 communication network and reproduced them through our traffic generation tool (see Figure 8). A real electrical power grid is modeled using PSCAD to extract electrical measurements of normal and fault scenarios and feed our extensible tool with realistic data.

Besides the feature generation, it is very important to define which of the generated features are relevant for each class of attack. Therefore, feature selection methods can be adopted. Clearly, the use of the GRASP metaheuristic has been frequently considered in the literature for feature selection. However, this approach is still seldom studied in the domain of intrusion detection. Thus, before this thesis, it was still unclear how much GRASP could contribute to the state-of-the-art machine learning-based IDSs.



# 4 Research Problem

## 4.1 Security and Threats

The easier communication with IEDs achieved with the IEC-61850 standard also enables the easier remote manipulation of electrical equipment (*e.g.*, circuit breakers), making substations more vulnerable to a number of cyber threats (USTUN; FAROOQ; HUSSAIN, 2019; HONG; LIU, 2019). There are many potential cyber vulnerabilities within the networks and devices of digital substations that can degrade confidentiality, availability, and data integrity (HAHN; SUN; LIU, 2016). However, cyber-security features are not included in IEC-61850 (RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019b).

Therefore, understanding these potential vulnerabilities is crucial for designing suitable security countermeasures and intrusion detection mechanisms to protect the substation. Once an attacker gains access to the substation networks, the physical protection of the substation is no longer sufficient for protecting the infrastructure from potential harm, thus allowing attackers to cause catastrophic damage (RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019b). This section covers IEC-61850 specific threats (*i.e.*, replay, message injection, masquerade, and poisoning attacks) and traditional threats (*i.e.*, Man-In-The-Middle (MITM), impersonation, password crack, traditional DoS attacks, and Packet sniffer) that may affect digital substations. Furthermore, the IEC-62351 standard for securing substation devices is analyzed.

We assume attackers can access the IEC-61850 network: it may happen through an infected flash drive, remote access (for maintenance), or from a malicious ex-employer, for example, or by any other way.

### 4.1.1 Attacks to IEC-61850 Multicast Protocols

Since the IEC-61850 multicast protocols (GOOSE and SV<sup>1</sup>) are assumed to run within the substations' local network isolated from the Internet, attackers need to gain access to an intranet interface to capture, spoof, modify or retransmit malicious messages. Attackers may gain physical access to protective IEDs or explore alternatives, such as placing malware in device software (*i.e.*, through update patches) or first infecting other devices connected to the network (*e.g.*, technician's computers). Regardless of the method, from this point on we assume the attacker has the ability to analyze, spoof, inject, and transmit malicious frames containing IEC-61850 multicast messages (HONG; LIU, C.; GOVINDARASU, 2014). In Figure 2, these attacks could be launched by an attacker connected to the process bus — that is where the GOOSE messages are transmitted.

According to Hong et al. (HONG; LIU, C.; GOVINDARASU, 2014), there are 9 main potential ways for an attacker to exploit vulnerabilities to cause damage and disrupt the power system components: (i) compromising the user-interface; (ii) interrupting the time synchronization process; (iii) compromising the station level communication bus; (iv) gaining access to bay level devices; (v) changing protective device settings; (vi) capturing and modifying GOOSE messages; (vii) compromising the process level communication bus; (viii) placing forged values in SV messages; and (ix) compromising the firewall to gain access to the substation network. From these entry points, attackers may perform different attack variations, such as message relay, injection, and poisoning to cause Denial of Service (DoS). These attacks are summarized in Table 1 and detailed below.

#### 4.1.1.1 Replay Attack

This attack model is based on the resending of a previously sent message. The attacker captures and replays the message without modifying its content. Such retransmission may occur immediately after the message is captured or after a longer delay.

Checking timestamp and sequence numbers is useful for detecting this malicious behavior (HONG; LIU, C.; GOVINDARASU, 2014). In the IEC-61850 architecture presented in Figure 2, a replay attack could be launched by an attacker connected to either the process or station bus.

Existing tools may be used to perform this kind of attack. The TCPReplay (TURNER, 2005) can read a variety of packet capture (pcap) files and use them as input to perform

---

<sup>1</sup>SV also has a unicast mode.

Table 1: Summary of threats on IEC-61850 digital substation networks.

Prot.	Attack Class	Description	Countermeasure
GOOSE	Naive Injection (HOYOS; DEHUS; BROWN, 2012)	Fabricated messages are transmitted ( <i>e.g.</i> , commands).	IEC-61850 standard consistency checking.
SV	Naive Injection (HOYOS; DEHUS; BROWN, 2012)	Fabricated messages are transmitted ( <i>e.g.</i> , measures).	IEC-61850 standard consistency checking.
GOOSE	IEC-61850 Injection (HOYOS; DEHUS; BROWN, 2012)	Fabricated IEC-61850 compliant commands are transmitted.	Context attributes consistency checking.
SV	IEC-61850 Injection (HOYOS; DEHUS; BROWN, 2012)	Fabricated IEC-61850 compliant messages with fake measures.	Multiple sources measurements correlating.
GOOSE and SV	Replay (HONG; LIU, C.; GOVINDARASU, 2014; RASHID et al., 2014)	Previous messages are retransmitted.	Attributes consistency checking.
GOOSE	Masquerade (USTUN; FAROOQ; HUSSAIN, 2019)	Messages that mimic real behavior are transmitted.	Attributes consistency and correlation checking.
GOOSE	Poisoning (KUSH et al., 2014)	The <i>StNum</i> is excessively increased.	Attributes consistency checking.
GOOSE and SV	Modification (RASHID et al., 2014)	Specific attributes are adulterated.	Attributes consistency checking.
GOOSE and SV	Flooding (RASHID et al., 2014; KUSH et al., 2014)	Many messages are transmitted at high frequency.	Message statistics checking.

message replays. In (NOCE et al., 2017), a GOOSE traffic generator was developed. It was used for capturing and injecting malicious GOOSE messages to the network. Thus, it was shown that an attacker can explore these functions to harm a targeted system. Replay attacks may be especially harmful if the attacker chooses the proper opportunity to mislead the system during a critical operation. For example, imagine a scenario in which a message containing a “circuit breaker close” command is captured by an attacker. If this message is retransmitted (replayed) during an electrical fault or line maintenance, the circuit breaker may be improperly re-closed, causing severe damage.

### 4.1.1.2 Message Injection

Message injection attacks build and transmit false and potentially malicious messages into the network. This attack can be launched in one of two ways. In the simplest form of message injection (*i.e.*, Naive Injection), a message created with random values in its fields without observing its consistency with the rules of the IEC-61850 standard (*i.e.*, it may contain invalid field values). Clearly, a syntax-based IDS that works by simply checking message syntax can detect these fake-injected messages. Also, the combination of multiple syntax rules to generate more complex rules is a potential way to increase the accuracy in the detection of message injection attacks.

The second way is to create new messages or modify captured messages that comply with the IEC-61850 standard syntax rules (*i.e.*, IEC-61850 Injection). Note that in contrast to reply attacks, injection/modification attacks are assumed to send a fabricated or modified message instead of simply retransmitting a past message. A method for exploiting the GOOSE protocol semantic to launch fake data injection attacks is presented in (HOYOS; DEHUS; BROWN, 2012), where legitimate messages are captured and their source and destination Media Access Control (MAC) addresses are spoofed by using legitimate addresses used to impersonate benign devices. Additionally, the payload data of the messages (*i.e.*, a boolean parameter) is adulterated to cause malicious actions in the target. Attackers have to send fake messages in the gap between two legitimate messages to avoid their behavior being detected by a context-oriented IDS which checks the consistency between messages.

### 4.1.1.3 Masquerade Attack

This attack model is a specialization of the injection attack, with a particular improvement: after old messages are captured, they are adulterated to mimic a legitimate behavior. In particular, masquerade attacks have an additional step between the capture and transmission phases: they need to get fresh (and valid) values for `SqNum` and `StNum`. Attackers learn from observing the content of past messages' to mimic their behavior. This improvement based on the analysis of the previous messages makes it more difficult to distinguish fake messages from legitimate ones (USTUN; FAROOQ; HUSSAIN, 2019).

Accordingly, both syntax-based (*i.e.*, that considers individual messages' syntax) and anomaly-based IDSs (*i.e.*, that considers the traffic behavior) are expected to fail to detect it. Instead, an IDS based on more sophisticated techniques, such as machine learning,

analyzing multiple sources of information from various protocols may be promising for such a challenging scenario. In Figure 2, this attack could be launched by an attacker connected either to the process or station buses.

One example of a masquerade attack (USTUN; FAROOQ; HUSSAIN, 2019) is an implementation that changes three specific GOOSE message fields. The main field used to cause damage is the `state` field, which is part of the `GOOSE DataSet`. This field describes the state of the circuit breaker as open (*e.g.*, under a fault) or closed (*e.g.*, under normal conditions). By changing this value, an attacker may cause the undesired operation of a circuit breaker. The attacker may also change additional fields, such as `StNum` and `SqNum`, to make the detection of this malicious action more difficult. Finally, the frequency in which false messages are transmitted is gradually changed to mimic the typical (and legitimate) bursty message transmission behavior observed in state changes. As a consequence, attackers can perform malicious operations such as opening a circuit breaker when it should be closed. This is worse still if an attacker closes a circuit breaker improperly (*e.g.*, during line maintenance), where human life may be in danger.

#### 4.1.1.4 Poisoning Attack

The main goal of poisoning attacks (KUSH et al., 2014) is to harm the communication between publisher and subscriber devices by preventing the subscriber from processing subsequent legitimate GOOSE messages or forcing subscribers to process fabricated GOOSE messages. The consequences of this attack include both DoS and improper operation of the devices. Four poisoning attack variations are proposed in (KUSH et al., 2014), as described below. In Figure 2, all variations of this attack targeting GOOSE messages could be launched by an attacker connected to either the process or station buses.

- *High-Status Number Attack* consists of capturing a GOOSE message and sending a new spoofed message with a higher `StNum` than that of the legitimate messages. The subscriber devices would thereafter discard any subsequent legitimate GOOSE messages with a lower `StNum` than the poisoned number.
- *High-Rate Flooding Attack* is a variant of DoS attacks, in which 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 one expected at the subscriber devices. Therefore, the result will be similar to the previous variant, except by the increased difficulty of distinguishing the legitimate messages from the

flooded fake messages.

- *Flooding Attack* sending multiple fake messages on the multicast channel. Each fake message will increase the status number expected at the subscriber devices, appearing legitimate, and increasing the difficulty of detection. Furthermore, the legitimate messages may be delayed due to contention in the network or devices during the flooding. This attack may be better detected by an anomaly-based IDS since it considers the overall behavior instead of analyzing messages in isolation.
- *Semantic Attack* consists of observing the legitimate traffic for learning/predicting the message content and spoofing realistic messages, increasing the status number every new fake message, at a high rate. Thus, legitimate messages are discarded since their `StNum` are lower than the recently received fake messages. Note that the expected effect is similar to High-Status Number Attack, however, in semantic attacks the `StNum` is increased by multiple messages instead of only one. Thus, whereas High-Status Number Attack may be detected by checking the anomalous `StNum` increasing between two consecutive messages, Semantic attack may be better detected by behavior analysis through an anomaly-based IDS.

### 4.1.2 Other Inter- and Intra-Substation Threats

In addition to the attacks on the process and station buses, the attacks discussed below could be launched by an attacker connected to the control center network in Figure 2. Although these attacks are part of our discussion, we concentrate our scope only on the IEC-61850 multicast protocols (*i.e.*, GOOSE and SV).

A known issue since 1985 (MORRIS, 1985) in the TCP protocol is the possibility of MITM attacks by impersonating a legitimate host (*i.e.*, using its IP address). A defense against this type of attack (BELLOVIN, 1996; GONT; BELLOVIN, 2012) improves the TCP protocol resistance to this vulnerability. However, an attacker that can observe the initial messages for a connection may still be able to launch MITM by impersonating that connection (GONT; BELLOVIN, 2012).

Therefore, the MMS communication protocol over TCP/IP is also vulnerable. Experiments exploiting MITM attacks (KANG; MAYNARD, et al., 2015) demonstrated the feasibility of causing physical effects on the electrical devices via malicious manipulation of IED parameters (*i.e.*, data attribute *MaxWLim*) through injected MMS commands (*i.e.*, write requests). Aside from TCP, other protocols in the MMS stack may present

vulnerabilities, such as TPKT and COTP (KIM; JO; SHON, 2016).

Other auxiliary protocols used in digital substations may suffer from three types of attacks (RASHID et al., 2014; PREMARATNE et al., 2010):

- Password Crack: may target FTP, telnet, or HTTP;
- DoS attacks: high-rate data generated, for example via a PING tool;
- Packet sniffer<sup>2</sup>: targets the Address Resolution Protocol (ARP) protocol.

Part of these attacks may be avoided by blocking such protocols when they are no longer used (*e.g.*, FTP may only be used by devices at the commissioning step, and prohibited thereafter), reducing the risk of exploitation. On the other hand, some protocols (*e.g.*, MMS and Precision Time Protocol (PTP)) may not be blocked, given their vital roles in the proper function of substation devices. In (MOUSSA; DEBBABI; ASSI, 2016), authors describe two approaches to perform delay attacks that desynchronize the clocks of slave nodes and, then, delay the PTP synchronization messages: (i) adding a device to the network (called a delay box) that aims at delaying the synchronization messages; and (ii) retransmitting messages with a modified timestamp. In particular, the first approach requires physical access to the substation to insert such new device, whereas the second one requires compromising — through a malware installation (either by physical or remote access) — a device named grandmaster clock, which is responsible for disseminating the updated timestamp to other devices. Both approaches target the functionality of all the devices in the network, since their proper function relies on precise time synchronization, rather than a particular IED.

### 4.1.3 IEC–62351 Standard

Except for IEC 61850–90-5, which only focuses on cybersecurity of Routable-GOOSE and SV (R-Goose and R-SV), the IEC–61850 standard does not specify security features to address these cyber-security vulnerabilities (USTUN; FAROOQ; HUSSAIN, 2019). Thus, the IEC–62351 standard was published to specify security measures, such as cryptographic, for IEC–61850 applications.

Regarding the MMS, all TCP T-Profile implementations that claim conformance to IEC–62351-4 (C, 2010) shall support TLS (Transport Layer Security) to provide authen-

---

<sup>2</sup>Although sniffing may not be always an attack (*i.e.*, when it is not aimed to steal data and sensitive information), it can also be used as a first step to acquire information and perform further attacks.



tication, confidentiality and data integrity. However, it is important to note that this standard also specifies that such implementations shall permit TLS to be disabled (*i.e.*, it does not ensure that TLS will be used). OSI T-Profiles is outside the scope of the IEC-62351-4 specification.

Because GOOSE and SV have strict timing requirements, IEC-62351 proposes the use of lightweight algorithms. However, IEC-62351 only recommends the adoption of techniques that may provide message integrity and node authentication. Note that TLS uses symmetric cipher after establishing the secure session, which can be performed quickly by secure and dedicated hardware. However, according to IEC-62351-6 (C, 2010), for applications using GOOSE and SV and requiring 4 ms response times, multicast configurations and low CPU overhead, encryption is not recommended. Instead, the SV and GOOSE messages are supposed to be restricted to a logical substation LAN.

IEC-62351 does not have full solutions geared toward mitigating certain attacks, including Masquerade attack (USTUN; FAROOQ; HUSSAIN, 2019). Besides, IEC-62351 is not yet complete: it requires more evaluation to address other security aspects (*e.g.*, key management evaluation) (STROBEL; WIEDERMANN; ECKERT, 2016).

## 4.2 Study of the Digital Substations IDS

The field of intrusion detection can be studied under different aspects. There are multiple design choices that can be used for each of these aspects. Thus, a novel taxonomy is presented in this section, as summarized in Figure 6.

We classify the existing IDSs considering both design and deployment aspects (they are separated by a vertical dashed line in Figure 6). Digital substations differ from traditional information technology systems in many perspectives, such as detection time requirements, specific hardware implementation and protocols, and other specific characteristics of the IEC-61850 standard. Each of them is addressed in the following subsections.

### 4.2.1 Design Aspects

We subdivided the design aspects of an IDS into four main parts: architecture, approach, type of analysis (*i.e.*, online or offline), and actions (*i.e.*, detection only or prevention). These aspects and their subcategories are listed to the right of the dashed vertical line in



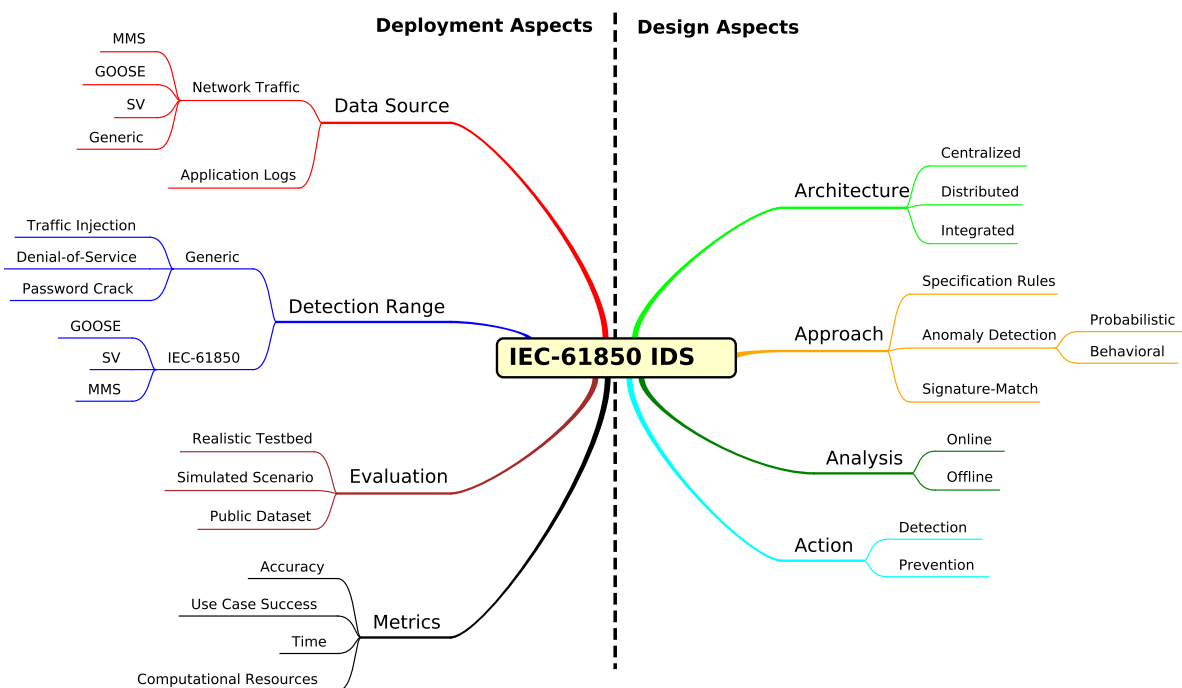


Figure 6: Proposed taxonomy for intrusion detection aspects on IEC–61850 digital substations.

Figure 6.

### 4.2.1.1 Architecture

In terms of architecture, an IDS may be classified as centralized, distributed, or embedded. The centralized is the most common architecture since it requires only one additional network element. The goal is to capture and analyze data application logs or network packets in a central IDS component. The main problem of a centralized IDS is that it has a single point of failure, as well as a potential bottleneck. As such, it may compromise the services’ availability. It may be prohibitive for an IDS to detect and prevent intrusions timely for substation time-critical applications. Despite that, most of the current IEC–61850 IDS proposals are designed considering a centralized architecture (PREMARATNE et al., 2010; HONG; LIU, C.; GOVINDARASU, 2014; KWON et al., 2015; YOO; SHON, 2015; YANG; MCLAUGHLIN, et al., 2016; KANG; MCLAUGHLIN; SEZER, 2016; YANG; XU, et al., 2016; KIM; PARK, 2018; YANG; HAO, et al., 2019).

Embedded IDSs aim at integrating the IDS functionality into substation devices (*i.e.*, IEDs) (PREMARATNE et al., 2010) (KABIR-QUERREC et al., 2015) (HONG; LIU, 2019). The main drawbacks of this approach are the new hardware design requirements and the internal computational overhead. On the other hand, since intrusions are de-

tected at the target device, this approach may detect and block malicious behavior of compromised devices before the attack has an effect (*e.g.*, malicious messages are discarded instead of processed).

The distributed approach avoids the aforementioned problems — if one of the IDSs fails, its load should be sent to other IDSs with available resources. A distributed mechanism (MACWAN *et al.*, 2016) was proposed to detect data injection attacks collaboratively in digital substations based on the IEC–61850 standard. Similarly, there are other proposals (HONG; LIU, C.-C.; GOVINDARASU, 2014; HONG; LIU, 2019) involving a distributed architecture to exchange information among IDSs regarding attack attempts. Their main idea is to introduce specification-based IDSs modules inside protective IEDs and Merging Units. Therefore, every GOOSE or SV message is analyzed for security before being processed for its functionality. These internal modules communicate with each other to share the detected intrusions.

#### 4.2.1.2 Detection Approach

In another categorization, according to Bostani and Sheikhan (BOSTANI; SHEIKHAN, 2017), an IDS can be categorized into three groups based on its detection approach: *signature-based*, *anomaly-based*, or *specification-based*. The same categorization is also employed in more specific Smart Grid scenarios, such as the Advanced Metering Infrastructure (AMI) (TONG *et al.*, 2016).

Signature-based methods are characterized by containing a database with samples that represent known attack profiles, while anomaly- and specification-based methods attempt to profile the legitimate or “normal” behavior of network traffic or adjacent systems (BOSTANI; SHEIKHAN, 2017). In particular, specification-based IDSs model a desirable behavior of a system through its functionalities and security policy (TONG *et al.*, 2016). However, unlike anomaly-based methods, specification-based methods are hard to design and generalize for various protocols (*i.e.*, different specification rules would be necessary for GOOSE, SV, MMS, and other protocols in the substation network) (TONG *et al.*, 2016; BOSTANI; SHEIKHAN, 2017).

Currently, in the context of IEC–61850 digital substations, most IDSs are specification-based ones (RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019a; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; KWON *et al.*, 2015; YANG; MCLAUGHLIN, *et al.*, 2016; KABIR-QUERREC *et al.*, 2015; YANG; XU, *et al.*, 2016; KIM; PARK, 2018; HONG; LIU, 2019), while some IDSs are based on anomalies

or are hybrid (*i.e.*, combination between multiple approaches) (PREMARATNE et al., 2010; YOO; SHON, 2015; YANG; HAO, et al., 2019). Among the methods categorized in (BOSTANI; SHEIKHAN, 2017), the signature-based approach is still the least explored (KANG; MCLAUGHLIN; SEZER, 2016).

### 4.2.1.3 Analysis

To the best of our knowledge, there are no proposals that employ real-time analysis to detect intrusions on IEC-61850 digital substations within a specified time requirement (deadline). On the other hand, there are preliminary studies (BIFET et al., 2010; FAISAL et al., 2014; NIXON; SEDKY; HASSAN, 2019) that address this issue for other Smart Grid domains, and these may be adapted for digital substations in the future.

The MOA (Massive Online Analysis) library (BIFET et al., 2010) can be used for performing intrusion detection. It was used to detect intrusions in devices from different AMI (TONG et al., 2016) layers: smart meters, data concentrators, and control centers (FAISAL et al., 2014). In contrast to the substation devices, AMI applications are closer to the end-user. Thus, these devices are also part of the power grid infrastructure but at a higher level.

Recently, online intrusion detection using MOA to detect traditional attacks was explored (NIXON; SEDKY; HASSAN, 2019). However, from our analysis of the literature, there is a dearth of studies employing such algorithms for detecting intrusions in substation networks.

### 4.2.1.4 Action

Another design aspect that may be considered to build an efficient IDS is the ability to execute actions to block detected attacks. In this context, the IDPSs (Intrusion Detection and Prevention Systems) have the capability of responding to the attack to block the intruder right after the intrusion detection.

The capacity for preventing an attack is closely related to real-time detection since the latter enables a quick response. Current IDSs designed for digital substations are typically focused only on detecting, and do not prevent attacks. Some existing efforts include the proposal of IDSs embedded into IEDs (HONG; LIU, 2019; KABIR-QUERREC et al., 2015), and traffic blocking in network switches (HA et al., 2016). These may be potential alternatives to mitigate malicious activities before they cause undesirable effects, such

as operating improperly an electrical equipment. Currently, there are only a few works proposing IDPSs and this research field needs to be better explored. Thus, it is an open challenge in the field of intrusion detection.

Note that real-time IDS and IDPS are not synonymous. While the former refers to the use of real-time processing tools for detecting intrusions, the latter refers to the response/mitigation to a detected intrusion.

## 4.2.2 Deployment Aspects

To deploy an IDS in a digital substation, it is important to define the range of attacks that will be considered and which data sources will be analyzed to detect them. Similarly, the proper evaluation methods and metrics should be considered. A summary of the deployment aspects and their subcategories are listed at the left of the dashed vertical line in Figure 6.

### 4.2.2.1 Data Sources and Detection Range

As discussed in Section 4.1, there are multiple entry points in substation networks where attacks may take place. Therefore, each possible attack should be considered and different data may be required to address each of them, as follows. Detecting traditional attacks that may target the station level, such as control centers, requires analyzing different data sources (*i.e.*, FTP unauthorized access logs, TCP and UDP traffic statistics) from those used for detecting attacks to the process level, such as fake measurement injection.

Besides the different devices, network segments, and protocols involved in the execution of each attack, particular features may be relevant to represent specific attackers' behavior. These features may include parameters from both network and application layers, ranging from specific field values (*e.g.*, `StNum`, `SqNum`, source IP address) to counters (*e.g.*, the number of transmitted bytes, the number of active connections, rate of packets per second).

Suppose that an IDS is deployed to detect GOOSE Poisoning, GOOSE Injection, and SV Flooding attacks. The key features for this IDS to analyze should include:

- `StNum`: it is typically excessively increased in GOOSE Poisoning attacks;
- `SqNum`: combined with `StNum`, this field may reveal modification of important parameters from the GOOSE dataset, such as circuit breaker state;

- **Timestamp:** this field allows computing the message transmission frequency, used to detect SV Flooding, where many messages are transmitted in a short time.

In fact, it is hard to predict manually all features related to each attack type. This procedure would require complete expert domain knowledge. This is a major issue of specification-based IDSs. To match the available features from the data source, IDSs can employ an automatic feature selection method (CHANDRASHEKAR; SAHIN, 2014). As described in Section 2.2, feature selection methods can be classified as filters, wrappers, and embedded algorithms. Filtering algorithms rely mainly on statistical methods to evaluate individual features, wrapper algorithms employ machine learning for evaluating different feature sets and identify which feature increases their accuracy. Typically, wrapping is slower but more accurate than filtering (GANAPATHY et al., 2013).

#### 4.2.2.2 Evaluation

Once the attack detection range and data sources are defined, proper evaluation strategies must be chosen to assess the IDSs. There are three main methods for evaluating an IDS. The first one is through realistic<sup>3</sup> testbeds, where physical equipment is used to generate data. The second one consists of generating a synthetic dataset by capturing real or simulated data from the substation network and injecting attack samples. The third way is adopting existing labeled dataset containing normal and attack samples – these datasets may be built through one of the previously presented ways and shared among the community. The point is that data are usually not generated by the IDS developer/deployer.

There are well-known datasets containing generic traffic which can be used for evaluating IDSs targeting traditional network protocols (YANG; HAO, et al., 2019). Unfortunately, to the best of our knowledge, there are no IEC-61850-based public datasets available, probably because of the proprietary or sensitive nature of the data. Therefore, acquiring real (or even realistic) traffic represents a big challenge.

In particular, Yoo and Shon (YOO; SHON, 2015) reported experiments based on a real digital substation, where GOOSE and MMS traffic is used to evaluate a specification-based IDS. However, most IDS proposals in the literature are evaluated in small test-beds and/or using simulation tools, as shown in Table 2.

In terms of simulation tools, there is a specific hardware and simulation software

---

<sup>3</sup>We assume as a realistic (not real) scenario, a model of the real substation environment.

named Real-Time Digital Simulator (RTDS). Although it has already been used for evaluating IDSs (HONG; LIU, 2019), it is a professional tool and typically too expensive for wide use in academic research. Additionally, it can not reproduce attacks.

Table 2: Summary of IDSs evaluation techniques (ordered by date)

Ref.	Year	Approach	Data Source	Evaluation
(PREMARATNE et al., 2010)	2010	Anomaly	Generic	Testbed
(HONG; LIU, C.; GOVINDARASU, 2014)	2014	Specification	GOOSE and SV	Testbed
(HONG; LIU, C.-C.; GOVINDARASU, 2014)	2014	Specification	GOOSE and SV	Testbed
(KWON et al., 2015)	2015	Specification	GOOSE and MMS	Testbed
(YOO; SHON, 2015)	2015	Anomaly	GOOSE and MMS	Real Subs.
(YANG; MCLAUGHLIN, et al., 2016)	2016	Specification	GOOSE, SV, and MMS.	Testbed
(KANG; MCLAUGHLIN; SEZER, 2016)	2016	Signatures	MMS	Testbed
(YANG; XU, et al., 2016)	2017	Specification	GOOSE, SV, and MMS.	Testbed
(KIM; PARK, 2018)	2018	Specification	GOOSE and SV	Prototype
(YANG; HAO, et al., 2019)	2019	Anomaly	Generic	Dataset
(HONG; LIU, 2019)	2019	Specification	GOOSE and SV	Simulation

### 4.2.2.3 Metrics

Finally, it is necessary to choose the proper metrics to assess the IDS in light of the expected goals and requirements. From the basic indicators True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), different metrics can be derived. Important metrics include *Accuracy* (*i.e.*, the fraction of correct IDS classifications with respect to the total number of samples analyzed), *Recall* (*i.e.*, how many attacks are detected of the universe of attack samples), and *Precision* (*i.e.*, how many attack classifications are in fact attacks instead of false positives (TATBUL et al., 2018)). Finally, F1-Score is a commonly used metric in machine learning. It is computed using TP, FP, and FN. This metric can be derived from Precision and Recall. As TNs are not a factor in F1-Score, this metric is immune to biases introduced by a large imbalance toward normal instances (something common in security datasets, given the small amount

of attacks in comparison with the large number of non-attack data), contrary to what happens with accuracy. Formally, these metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.3)$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.4)$$

### 4.3 Specification Rules Assessment

According to Table 2, all works that consider both the IEC-61850 multicast protocols (*i.e.*, GOOSE and SV) employ specification-based IDSs — except for (YOO; SHON, 2015), that employs an anomaly-based IDS (RADOGLOU GRAMMATIKIS; SARIGIANNIDIS, 2019a; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; KWON et al., 2015; YANG; MCLAUGHLIN, et al., 2016; KABIR-QUERREC et al., 2015; YANG; XU, et al., 2016; KIM; PARK, 2018; HONG; LIU, 2019). Therefore, in this section, we focus on specification-based IDSs to assess their efficiency. In the following, we present a compilation and analysis of such rules employed by the state-of-the-art specification-based IDSs. Note that at this point we do run practical experiments, but an analytical exercise to check the ability for each rule under each attack behavior. In the following subsections, we define the attacker assumptions and detail the specifications rules.

#### 4.3.1 Attacker Assumptions

This section describes the scenarios considered for the specification rules assessment. This study aims at understanding the real efficiency of the specification-based IDSs. The attackers' behaviors are modeled after four different attacks presented in Section 4.1.1, namely Replay, Injection, Masquerade, and DoS attacks.

Replay and Masquerade attacks assume that the attacker can listen, capture, and

retransmit past messages into the network. As discussed in Section 4.1.1, we expect specification-based IDSs to have more difficulty in detecting Masquerade attacks, due to the careful analysis of the traffic standard carried out by the attacker before sending the masquerade messages.

DoS attacks are assumed to abide by the expected syntax of individual messages, but assumed to violate behavioral rules (*e.g.*, increasing load through message flooding).

Note that each attack is assumed to be performed individually, separately from each other. Thus, one attack does not affect the detection performance for other attack models.

### 4.3.2 Specifications Rules

Through a systematic review of the literature, we have compiled the following list of specification rules. In addition, rules from the same author were grouped and evaluated in rule sets (RSet).

- #R1) GOOSE messages must have MAC address starting with `01-0c-cd-01` (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R2) GOOSE messages must have the TPID field with value `0x8100` (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R3) GOOSE messages must have the `ethertype` field equal to `0x88B8` (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R4) GOOSE messages must have `TimeAllowedToLive` equal to double of the value of `MaxTime` (*e.g.*, 5000ms) (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R5) GOOSE messages must have the `APPID` field formatted as a 4-byte hexadecimal (*e.g.*, 0000-3FFF) (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R6) Consecutive GOOSE messages must have consistent values for fields `gocbRef`, `timeAllowedToLive`, `datSet`, `goID`, `t`, `StNum`, `SqNum`, `test`, `confRev`, `ndsCom` and `numDatSetEntries` (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; YANG; XU, et al., 2016);
- #R7) GOOSE messages must have the `APPID` field matching the last two octets of the destination multicast address (YANG; XU, et al., 2016);



- #R8) The IED control block name must be consistent with the value of the `goID` field (*i.e.*, the `LD/LN` value in the `gocoRef` field must match the `datSet` field from the GOOSE APDU) (YANG; XU, et al., 2016);
- #R9) The size of frames containing GOOSE messages should be equal to 8 *bytes* + *APDU size*, and *APDU size* should be less than 1492 *bytes* (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R10) The `SqNum` in GOOSE messages should be set to zero whenever the value of the `StNum` changes (w.r.t the previous message) (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; YANG; XU, et al., 2016);
- #R11) The number of messages captured in an interval must not exceed a predefined threshold (20% above the expected maximum) (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; KWON et al., 2015);
- #R12) The number of messages captured in an interval must not be equal to zero (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; KWON et al., 2015);
- #R13) The transmitter's timestamp should not be higher than the receiver's timestamp (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014);
- #R14) The transmitter's timestamp from GOOSE messages should not be more than 4 *ms* apart from the receiver's timestamp (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014);
- #R15) The *Recency* metric, represented by the last GOOSE message's arrival, must respect a minimum and a maximum threshold (KWON et al., 2015);
- #R16) The *Frequency* metric, represented by the average number of received GOOSE messages, must respect a minimum and a maximum predefined threshold (KWON et al., 2015);
- #R17) The *Monetary* metric, represented by the total number of received GOOSE messages, must be within a predefined threshold (KWON et al., 2015). The difference from rule #R11) is that this rule considers only received GOOSE messages;

- #R18) Only messages with specific source *port*, *IP* and *MAC* addresses are allowed (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R19) Only MMS, COTP, TPKT, and SNTP protocols are allowed on the station level network and only the GOOSE, SV, and IEEE 1588 protocols are allowed on the process level network (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016). Thus, attackers exploiting other protocols are detected by this rule;
- #R20) There must be consistency between the *GOOSE switch-in* messages (*e.g.*, breaker opening) and the value of the report sent by the MMS protocol (*i.e.*, MMS signal report) (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016);
- #R21) The number of bytes that travel per second must not exceed a predefined threshold (KWON et al., 2015; YANG; XU, et al., 2016);
- #R22) The number of packets that travel per second must not exceed a predefined threshold (KWON et al., 2015; YANG; XU, et al., 2016);
- #R23) The length of the packet (specified in the packet header) must not exceed a predefined threshold (YANG; XU, et al., 2016);
- #R24) The total size of the packet must not exceed a predefined threshold (YANG; XU, et al., 2016).

If one or more specification rules are not satisfied, it is assumed that an anomaly has occurred. This anomaly may be either a misbehavior (*e.g.*, as a consequence of high load or the improper function of some software or device) or an intentional violation caused by the malicious action of an attacker. For this analysis, we focus on the latter.

Each rule is assessed by its ability to detect five different attacks. Specifically, we classified each rule's detection capabilities into four levels:

- **Detect**: the rule always detects all possible variations of the attack.
- **HProb**: there is a high probability that the rule detects most variations of the attack (*i.e.*, it only fails to detect the attack under very specific and unlikely circumstances).
- **Part**: the rule is partially successful, that it, it detects some of the attack variations, or under certain parameters (*e.g.*, a specific field should have a value in a known range). However, there are cases in which the same kind of attack is not detected.

Table 3: Specification rules assessment results.

Rules	Attacks				
	Replay	Naive Injection	IEC-61850 Injection	Masquerade	DoS
#R1	Fail	HProb	Fail	Fail	Fail
#R2	Fail	HProb	Fail	Fail	Fail
#R3	Fail	HProb	Fail	Fail	Fail
#R4	Fail	HProb	HProb	Fail	Fail
#R5	Fail	HProb	Fail	Fail	Fail
#R6	Detect	HProb	HProb	Fail	Fail
#R7	Fail	HProb	Fail	Fail	Fail
#R8	Fail	HProb	HProb	Fail	Fail
#R9	Fail	HProb	Fail	Fail	Fail
#R10	Detect	HProb	Part	Fail	Fail
#R11	Fail	Fail	Fail	Fail	Detect
#R12	Fail	Fail	Fail	Fail	Part
#R13	Fail	Part	Part	Fail	Part
#R14	Part	HProb	HProb	Fail	Fail
#R15	Part	Fail	Part	Fail	Detect
#R16	Fail	Fail	Fail	Fail	Detect
#R17	Fail	Fail	Fail	Fail	Detect
#R18	Fail	HProb	Part	Fail	Fail
#R19	Part	Part	Part	Fail	Fail
#R20	Fail	Fail	Fail	Fail	Fail
#R21	Fail	Fail	Fail	Fail	Detect
#R22	Fail	Fail	Fail	Fail	Detect
#R23	Fail	Part	Fail	Fail	Part
#R24	Fail	Part	Fail	Fail	Part
#RSet1	Part	HProb	HProb	Fail	Fail
#RSet2	Detect	HProb	HProb	Fail	Detect
#RSet3	Detect	HProb	HProb	Fail	Detect
#RSet4	Detect	HProb	HProb	Fail	Detect
#RSet5	Detect	HProb	HProb	Fail	Detect
#RSet6	Part	Fail	Part	Fail	Detect
All	Detect	HProb	HProb	Fail	Detect

**Rule Set References (RSet):**#RSet1: ([YANG; MCLAUGHLIN, et al., 2016](#))#RSet2: ([HONG; LIU, 2019](#))#RSet3: ([HONG; LIU, C.; GOVINDARASU, 2014](#))#RSet4: ([HONG; LIU, C.-C.; GOVINDARASU, 2014](#))#RSet5: ([YANG; XU, et al., 2016](#))#RSet6: ([KWON et al., 2015](#))

- **Fail**: the rule always fails to detect the attack.

When deployed in digital substations, specification-based IDSs are typically configured based on specialized domain knowledge. Most of the specifications rules are defined by considering the consistency between message field values and the specifications established by the IEC-61850 standard.

Our first conclusion is that most of these rules are not able to detect *Replay attacks* since the values of malicious messages are the same as the legitimate ones. However, as shown in the second column of Table 3, rules #R6 and #R10 detect Replay attacks because these rules consider the consistency between multiple consecutive messages instead of considering only the parameters from a single message. In particular, the attributes `StNum` and `SqNum` allow them to detect messages out of context. Rule #R14 detects replay attacks only if there is a delay of at least 4 *ms* between the retransmitted and the legitimate message, which might not always be the case. Similarly, rule #R15 may detect replay attacks exceeding a minimum or a maximum predefined time interval since the last received message. Also, #R19 only works if an IEC-61850 message is transmitted in an unauthorized communication bus (*e.g.*, MMS in the process bus (YANG; MCLAUGHLIN, et al., 2016; YANG; XU, et al., 2016)).

Regarding Naive Injection attacks, Table 3 shows that rules #R1 to #R10 may, individually, detect IEC-61850 standard violations on particular message fields. Whereas each stand-alone rule (*e.g.*, #R6) is limited to detect Naive Injection attacks only when specific fields are violated, we assume that this attack model has a high probability of containing multiple inconsistent fields – as it is not aware of the IEC-61850 standard. For example, `StNum` (a 32-bit integer (IEC, I. E. C., 2003)) may have 4,294,967,295 possible values, thus a naive injection attacker has a very small chance to correctly guess the proper value. Similarly, rules #R14 and #R18 are likely to detect these attacks based on the message context, but they fail if the `SqNum` of the fake message is eventually set to zero or if the transmitter’s timestamp is within the 4 ms from the receiver’s timestamp. There are other rules (*i.e.*, #R13, #R14, #R19, #R23, and #R24) with limited potential to detect naive injection attacks since such rules are based on parameters that may eventually be inconsistent with their specifications. Our conclusion is that even stand-alone rules have a high probability of detecting Naive Injection attacks, which can be still more easily detected when considering the combination of multiple specification rules (see bottom of Table 3).

IEC-61850 Injection attacks assume attackers have the knowledge to send syntac-

tically correct fake messages that match the IEC-61850 standard (but not necessarily behavioral consistent); in this case, only 4 rules have a high probability of detecting such attacks. Both domain-based (*e.g.*, #R4 and #R8) and context-based (*e.g.*, #R6 and #R14) rules consider fields not known by attackers without access to the network traffic pattern and to the substation parameters. In particular, to bypass #R14, an attacker would need timing synchronism with the target devices to have their messages being delivered in valid periods. Thus, these fields are not expected to be properly forged by this attack model. Rule #R10 detects a single syntactically correct message if the attacker does not set the SqNum field to zero after a malicious StNum change. Moreover, rule #R18 detects some *IEC-61850 Injection* attacks if the used port, IP, and MAC addresses refer to an unauthorized device. The efficacy of rules #R13 and #R15 is limited to cases in which the fake message has an invalid timestamp or is transmitted in an excessively short interval, respectively.

It is worth noting that rule #R13 is able to detect in part both Naive and IEC-61850 injection attacks, that is, only when the fake timestamp is higher than the local time at the receiver. Both replay and masquerade attacks are not detectable by this rule because replay attacks do not change the timestamp (*i.e.*, it will be lower than the local time at the receiver whenever the clocks are synchronized) and masquerade attacks have sufficient knowledge to insert a valid timestamp. Most DoS variations are not detectable by rule #R13 since such attacks focus on resource overload. However, DoS attacks that operate by leading the system to an invalid status such as poisoning attacks can be detected. In this case, this rule may be useful to detect messages with an incorrect timestamp.

The masquerade attack manages to circumvent all specification rules included in state-of-the-art solutions that have both syntax and behavioral consistency. The attacker profile indicates an advanced knowledge about the operation of the substation – potentially obtained through a historical analysis of the messages transmitted in the network. Therefore, it is important to note that further improvements to deal with this particular attack are necessary. In the next chapter we provide a solution for such problem.

Rules #R11, #R12 and #R16 are suitable to detect *flooding* or other generic DoS attacks because they consider message counters capable of detecting anomalous behavior. In particular, #R12 works only in an advanced stage of DoS, where no legitimate messages are being delivered. Similarly, rules #R15, #R17, #R21, and #R22 allow the detection of anomalies in transmission time such as those caused by DoS attacks. Therefore, even though these rules may detect intrusion attempts, they are not effective in distinguishing

malicious and legitimate messages. Finally, rules  $\#R13$ ,  $\#R23$ , and  $\#R24$  are limited in detecting DoS attacks because these rules are based on single malformed messages.

A complete and more accurate detection can be achieved by the composition of rules. This may enable IDSs to detect and distinguish the different attack variations as well as to measure the attackers' expertise. Also, the combination of rules may reveal possible correlations between events. For instance,  $\#R18$  may reveal a malicious IP connected to the network, while  $\#R19$  detects a malicious attempt to generate unauthorized traffic, and  $\#R12$  reports a compromised state causing the system to be unavailable. However, despite the potential of complex rules to detect more attacks, even if building an IDS by using the combination of current state-of-art rules is still unable to detect all attacks.

Existing specification-based IDSs employ different combinations of the aforementioned rules. In (YANG; MCLAUGHLIN, et al., 2016) and (KWON et al., 2015), only a part of replay attacks are detected even after combining multiple rules. Also, they fail to detect masquerade attacks. Rules used in (YANG; MCLAUGHLIN, et al., 2016) present a high probability to detect Naive and IEC-61850 injection, but fail to detect DoS. On the other hand, the rules used in (KWON et al., 2015) enable DoS detection but fail to detect Naive Injection and detect only part of IEC-61850 injection attacks. The combinations of rules presented in (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; YANG; XU, et al., 2016) provide a similar capability to detect all attacks: they detect replay and DoS attacks, and present a high probability of detecting Naive and IEC-61850 injection attacks. Finally, the last row of Table 3 shows that, even combining all rules, both Naive and IEC-61850 attacks are not always detected, although they provide a high detection probability. Masquerade attacks, on the other hand, still cannot be detected. A potential solution may be the extraction of more representative features by considering both GOOSE and SV traffic.

Although the computation of the metrics discussed in Section 4.2.2 requires numerical indicators (*i.e.*, TP, TN, FP, and FN), it is possible to qualitatively estimate the expected level of *recall* to specification-based IDSs in detecting the five attacks analyzed in this section based on Table 3. In summary, masquerade attacks would present the lowest *recall* due to their high number of false negatives. Similarly, replay attacks may present a low *recall* by IDSs that do not employ rules  $\#R6$  and  $\#R10$ . On the other hand, Naive Injection and DoS attacks are expected to have a higher recall since there are more rules to detect them.

Since specification-based IDSs are designed by modeling specific malicious actions, it is

reasonable to expect a high *precision* to all rules (*i.e.*, a low number of false positives) even if they have poor recall. Estimating the *accuracy* without the knowledge of the number of samples analyzed would be not possible, because it may be affected by imbalanced datasets (*i.e.*, disproportional number of attacks and legitimate samples).

## 4.4 Discussion on Digital Substations IDS

To the best of our knowledge, this is the first work to present an in-depth survey on IDS aspects for digital substations based on the IEC–61850 standards. We covered intrusion detection approaches, data sources, architectures, evaluation methods, and metrics, and compared 6 existing proposals. Moreover, we assessed 24 specification rules for detecting five different attack types (RADOGLU GRAMMATIKIS; SARIGIANNIDIS, 2019a; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; KWON et al., 2015; YANG; MCLAUGHLIN, et al., 2016; KABIR-QUERREC et al., 2015; YANG; XU, et al., 2016; KIM; PARK, 2018; HONG; LIU, 2019).

Among the assessed aspects we identified the lack of data available for training, testing, and evaluating the current IDSs as the main factor to be studied in the field of IEC–61850 intrusion detection. Our evaluation also shows that further advancement is necessary for state-of-the-art IDSs to deal with the current and mainly the novel attacks that may be launched (*i.e.*, new rules would need to be built manually). In particular, specification-based IDSs have limited attack detection capabilities through their static specification rules (HONG; LIU, 2019; HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; YANG; XU, et al., 2016).

A potential way to handle such challenges is to employ machine learning algorithms. Nevertheless, it would require proper datasets with representative information about multiple types of attacks. In Chapter 5 we formulate a hypothesis and present a solution for the generation of realistic and updated intrusion datasets.

# 5 Hypothesis and Proposed Solution

## 5.1 Hypothesis

Based on an in-depth study on the field of the digital substations' IDS, as presented in Chapter 4, we identified a weakness in the literature in terms of evaluating datasets to support the use of dynamic methods such as machine learning algorithms.

Several existing works employ static rules to detect malicious behavior and perform intrusion detection. On the other hand, IDSs that employ machine learning have their focus on generic attacks, that is, that are not specialized for substations networks.

The lack of attack signatures and public data availability is a key factor that imposes serious challenges to the current state-of-art IDS. Although our study shows that specification-based IDSs are frequently used, other types of IDSs also require data for being assessed. Therefore, the aforementioned conclusions provided evidence to support and formulate the hypothesis of this thesis:

- **The scarcity of dynamic attack datasets based on IEC-61850 limits the development potential of IDSs. By providing realistic datasets and selecting their representative features to describe a range of attacks, it is possible to design more robust IDSs.**

To assess that hypothesis, we implement the workflow shown in Figure 7. According to this workflow, the main scope of this work is on the proposal and assessment of a novel extensible tool named Efficacious Reproducer Engine for Network Operations (ERENO).

ERENO is tailored to reproduce the behavior of the GOOSE and SV protocols, making it possible to model a range of attacks. As shown in Figure 7 ERENO takes the following inputs: (i) **Electric samples**, generated either by simulation tools or real devices and (ii) a set of **Attack Use Cases** that describes the attacker behavior when performing malicious activities on the network. More details about the ERENO, power



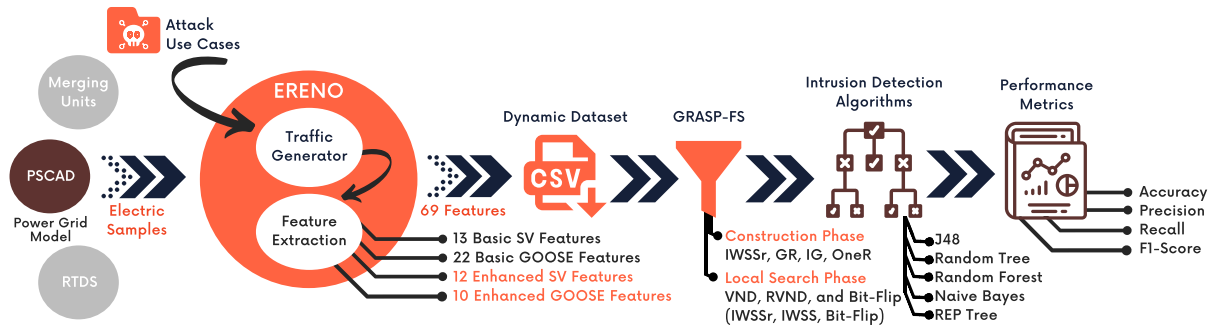


Figure 7: The proposed solution architecture.

grid simulations, substation traffic generation (GOOSE and SV protocols), and malicious traffic generation (attacks targeting the GOOSE protocol) are given in Section 5.2.

The second component embedded into ERENO (at the bottom of ERENO in Figure 7) is the *Feature Extraction*. This process performs operations on the generated traffic data by *Traffic Generator* (at the top of ERENO in Figure 7). From the generated substation traffic, the *Feature Extraction* component extracts basic features (13 from SV and 22 from GOOSE). Additionally, it generates novel enriched features (24 from SV and 10 from GOOSE) to improve the detection performance when they are processed by IDSs.

More details about the *Feature Extraction* process are given in Section 5.3.

As an additional component to improve the detection performance when processing the 69 features of the generated *dynamic datasets*, we propose a novel implementation of the Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic, named GRASP for Feature Selection (GRASP-FS), which is adapted to select features through a hybrid approach that combines filter methods (for selecting good features *candidates*) and wrapping methods (for assessing reduced subsets composed by the previously selected *candidates*).

More details about GRASP-FS are given in the Section 5.4.

Finally, at the right part of the Figure 7, the workflow shows the use of *Intrusion Detection Algorithms* to assess both the resulting features from the generated dataset under different metrics. More details about the assessment of ERENO are given in Chapter 6.

## 5.2 Data Generation: ERENO Extensible Tool

The ERENO Tool is built under 5 pillars: (i) realistic data generation; (ii) representative features generation; (iii) extensible to cover multiple attacks; (iv) modular to separate

each responsibility in an individual replaceable components, and (v) open-source code to allow the academia community to use and improve it.

ERENO can generate realistic data for SV and GOOSE protocols because it reproduces the behavior defined in the IEC-61850 standards based on electrical samples (*i.e.*, current and voltage) input from any source. The produced traffic may contain events from the transmission line with both normal operation and electric fault scenarios.

ERENO is easily extensible as it supports the implementation of customized attack use cases. This way, novel attacks can be modeled and incorporated into ERENO at any time. Thus, ERENO is able to provide realistic and updated attack datasets for training, testing, and assessing IDSs. The overall process is shown in Figure 8 and described below.

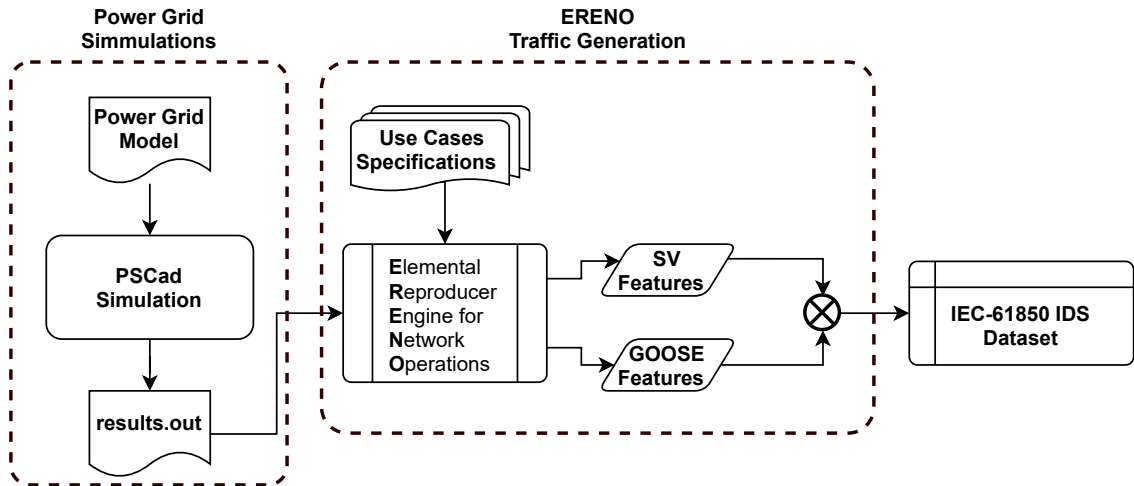


Figure 8: Proposed ERENO Extensible Tool for generation of realistic IEC-61850-based datasets.

Therefore, ERENO takes electrical samples as inputs from any source such as a power grid modeling tool (see Section 5.2.1) in a column-oriented format (*e.g.*, the `results.out` output of PSCad, show in Figure 8) as a reference to build SV (see Section 5.2.2) and GOOSE (see Section 5.2.3) traffic representation of the substation and transmission line normal operation without electric faults and with the occurrence of electric faults; this electric fault traffic is also benign traffic (*i.e.*, not an intrusion) because it represents a natural electrical fault rather than an attacker action. Therefore, we introduce the *Goose Event Manager* component (it is part of the *Traffic Generator* in Figure 7) to support attacking use case specifications (see Section 5.2.4).

### 5.2.1 Power Grid Simulations

To generate data regarding the substation and transmission line operation in a realistic way, we modeled a real transmission line in the Brazilian electrical power system. This transmission line interconnects two substations (Serra da Mesa in the State of Goias, and Samambaia in the Federal District).

In the proposed implementation, we used the Power Systems Computer-Aided Design (PSCAD ([LTD, 2019](#))) tool to reproduce the modeled transmission line. PSCAD is a simulation tool that simulates electromagnetic transients, used for modeling and analysis of power systems, including steady-state and transient scenarios, as well as electrical faults (*e.g.*, any abnormal electric current, such as a short circuit). Thus, this tool enables the modeling of power system components and electrical faults analysis through parameter sweep, such as fault location, resistance and type.

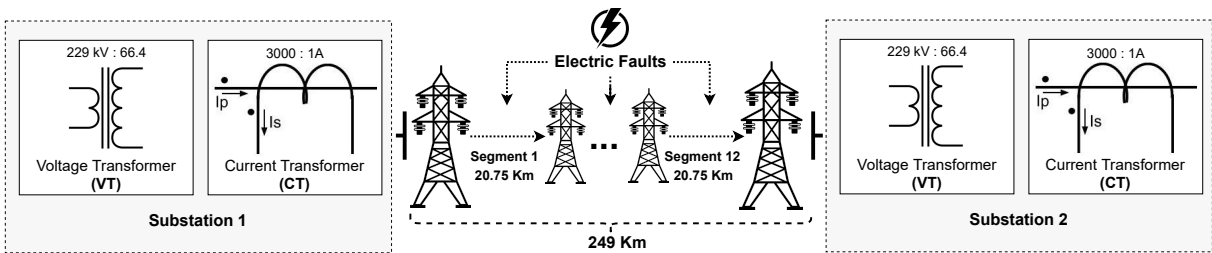


Figure 9: Simulated electric power grid model.

The simulated transmission line has three phases (A, B, C—and G for ground) and is 249 km long, with 12 segments of 20.75 km each. To generate a large number of fault and normal behavior data instances, we created several fault scenarios. Eleven fault types are applied along the transmission line in 12 different locations. These fault types include single-phase faults (*i.e.*, AG, BG, and CG), 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  $\Omega$ , 50  $\Omega$ , 100  $\Omega$ ). These parameters are based on ([PINHEIRO et al., 2021](#)).

Using this methodology, the ensuing dataset is obtained by considering combinations of all parameter values, namely 12 fault locations, 3 fault resistance and 11 fault types, or  $12 \times 3 \times 11 = 396$  scenarios. Each simulation run lasts 1 second, of which 900 ms represent normal conditions and 100ms represent a programmed fault at the fixed timestamp of 500ms (lasting 100ms until time 600ms). This fault duration (*i.e.*, 100ms) is set as the maximum fault duration allowed by the ONS (Brazilian electrical system operations organization) (Operador Nacional do Sistema Elétrico - ONS) ([OPERADOR NACIONAL](#)

DO SISTEMA ELÉTRICO, 2021).

### 5.2.2 SV Traffic Generation

The `plot step` parameter on PSCAD can be adjusted to set the time interval between each electrical signals reading (samples). 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 (*i.e.*,  $60 \text{ Hz} \times 80 \text{ messages} = 4,800 \text{ messages/s}$ ).

Therefore, to generate an equivalent (approximated) number of samples on PSCAD, we set the `plot step` parameter on PSCAD to  $\frac{1}{4,800} \approx \mathbf{208} \mu\text{s}$ .

Once the electric values are available (*i.e.*, current and voltage), the ERENO traffic generator can be implemented. The analog current and voltage values from a substation are measured by the simulated Voltage Transformer (VT) and Current Transformer (CT), illustrated in Figure 9, digitized by our traffic generator, and converted to SV messages (in compliance with the IEC–61850-9-LE standard).

### 5.2.3 GOOSE Traffic Generation

The main goal of GOOSE messages is to enable the communication of substation events, as described in Section 2.1.2.2. Since our study assumes a scenario where electric faults may occur, a GOOSE traffic generator was implemented to simulate the publication of events related to the operation of circuit-breakers as typically done by protection applications. The generated GOOSE features are shown in Section 5.3.

The ERENO generator produces GOOSE messages to simulate what would be sent by a physical IED. Thus, these messages are based on the current circuit-breaker status. Also, we assume that the ERENO is generating GOOSE messages from Substation 1 of Figure 9. During the first 500 *ms* of each simulation run, GOOSE messages are transmitted every  $T_0$  *milliseconds* with the boolean value that represents the circuit-breaker status set to a false, that means a “closed” circuit-breaker. When an event occurs (*e.g.*, a fault), a GOOSE message is sent immediately and retransmitted according to the behavior described in Section 2.1.2.2. In real scenarios, both  $T_1$  and  $T_0$  are defined in the Substation Configuration Description (SCD) file through `MinTime` and `MaxTime` tags. In our simulation, we assume these values as 4 *ms* and 1,000 *ms*, respectively, as in (HADELI, H. et al., 2009; HADELI et al., 2009). Thus, the assumed SCD tags are defined as follows:

1.  $\langle \text{MinTime unit}=\text{"ms"} \rangle 4 \langle /\text{MinTime} \rangle$
2.  $\langle \text{MaxTime unit}=\text{"ms"} \rangle 1000 \langle /\text{MaxTime} \rangle$

To reproduce a realistic behavior of a physical IED, we made the following assumptions to drive our GOOSE traffic generation model:

- *Assumption 1:* We assume that the SV messages are transmitted immediately at the begin of each simulation run; However, the first GOOSE message is transmitted only after the end of the first SV cycle. That means that the expected timestamp to the first GOOSE message is  $16.6\text{ ms}$ . This timestamp is computed assuming a  $60\text{ Hz}$  system (*i.e.*,  $60\text{ cycles} / 1\text{ second}$ ).
- *Assumption 2:* We assume that once an electric fault occurs, it will trigger a GOOSE message reporting such event only after an hypothetical IEDs' protection algorithm detect it (*i.e.*, through processing the corresponding SV messages). Therefore, we refer to this processing time as  $T_{IED}$ . We assume  $T_{IED} = 6.20\text{ ms}$ , based on the time reported in (FERNANDES; BORKAR; GOHIL, 2014).
- *Assumption 3:* Since we assumed *MaxTime* as  $1,000\text{ ms}$ , GOOSE messages are transmitted every 1 second, under normal conditions (*i.e.*, if no event occurs). However, as we simulated faults at the  $500\text{ ms}$  of each simulation run, novel GOOSE messages are sent starting at  $506.2\text{ ms}$  (*i.e.*,  $500\text{ ms} + T_{IED}$ ) and following the *bursting* mode.
- *Assumption 4:* We assume an Exponential Backoff function<sup>1</sup> to define the intervals for message retransmission during the *GOOSE bursting* mode. Thus, after the first GOOSE message (at  $506.2\text{ ms}$ ) reporting the fault event (occurred at  $500\text{ ms}$ ), other messages are transmitted with the same status but with an increasing sequence number (as shown in Table 5).
- *Assumption 5:* As the *GOOSE bursting* would have a duration of  $192\text{ ms}$  before reaching T0, according our assumed backoff, the 5th message would be sent at  $698.2\text{ms}$ . However, the fault duration is  $100\text{ ms}$ . Thus, instead of the expected message at  $698,2\text{ms}$ , another burst is triggered at  $606.2\text{ ms}$  to report the line reestablishment. The description of each message and its respective interval is shown in Table 4. The values of *StNum*, *SqNum*, and the *cbStatus* are shown in Table 5.

---

<sup>1</sup>The Exponential Backoff function used is available online at <http://backoffcalculator.com/>.

Table 4: Generated GOOSE messages per simulation run.

Time	Description	Interval	Next message
16.6 ms	Periodic GOOSE message	T0	1000 <i>ms</i>
500 ms	Fault occurrence		
506.2 ms	GOOSE status change	T1	4 <i>ms</i>
510.2 ms	GOOSE status change	T1	4 <i>ms</i>
514.2 ms	GOOSE status change	T2	25.2 <i>ms</i>
539.4 ms	GOOSE status change	T3	158.8 <i>ms</i>
600 ms	Line recovery		
606.2 ms	GOOSE status change	T1	4 <i>ms</i>
610.2 ms	GOOSE status change	T1	4 <i>ms</i>
614.2 ms	GOOSE status change	T2	25.2 <i>ms</i>
639.4 ms	GOOSE status change	T3	158,8 <i>ms</i>
798.2 ms	Periodic GOOSE message	T0	1000 <i>ms</i>

Table 5: Field values for the generated GOOSE messages per simulation run.

Time	StNum	SqNum	cbStatus
16 <i>ms</i>	0	0	0
506.2 <i>ms</i>	1	0	1
510.2 <i>ms</i>	1	1	1
514.2 <i>ms</i>	1	2	1
539.4 <i>ms</i>	1	3	1
606.2 <i>ms</i>	2	0	0
610.2 <i>ms</i>	2	1	0
614.2 <i>ms</i>	2	2	0
639.4 <i>ms</i>	2	3	0
798.2 <i>ms</i>	2	4	0

#### 5.2.4 Malicious Traffic through Attack Use Cases

In this section we describe the attack use cases that generate malicious traffic. Specifically, we exploit GOOSE protocol vulnerabilities to reproduce attacks from 7 attack use cases based on those attacks studied in Section 4.1.1. The assumptions and specific parameters for each attack use case are described below — attack use cases are responsible for attacker behavior modeling. The attacks described below were implemented and made available (both the dataset<sup>2</sup> and the source-code<sup>3</sup>).

<sup>2</sup>ERENO IEC-61850 Intrusion Dataset: <https://abre.ai/ereno-dataset>

<sup>3</sup>ERENO Traffic Generator Source-code: <https://github.com/sequincozes/iec61850generator>

#### 5.2.4.1 Random Replay Attacks (UC01)

This use case is based on the capture and retransmission of a previously sent message at multiple random simulation moments. Consequently, these replay attacks may have different impacts depending on the message content and context in which it is retransmitted, varying from outages to equipment damage or, in worst case, offering risk to human life. Note that replay attacks assume that the original message content is not changed.

#### 5.2.4.2 Inverse Replay Attacks (UC02)

This use case is a variation of UC01. The main difference is that instead of performing retransmissions with random status, the attacker attempts to cause more damage by choosing messages with a status different to the expected to be transmitted at that given moment.

To run this attack, the attacker captures previously sent GOOSE messages; then, it monitors the network to identify specific events (*e.g.*, electric faults or electric line reestablishment) 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.

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 the undesired closing of the circuit-breaker, rather than remaining open as expected in this fault scenario. The practical effects may be the damage to the electrical equipment or, in the worst case, offering risk to human life (*e.g.*, by reestablishing a transmission line during equipment maintenance). To cause the inverse result (*i.e.*, improperly opening the circuit-breaker) the attacker sends the malicious message under normal conditions. Each message is sent at a random moment. If the attack action is done during a fault situation, one message captured during a normal situation is transmitted. Otherwise, one message captured during a fault situation is transmitted. This attack model also assumes that the original message content is not modified (*i.e.*, the incorrect status comes from an old message with its integrity preserved, including the older *sqNum* and *stNum*).

#### 5.2.4.3 Masquerade Attacks towards Outage (UC03)

Masquerade messages are built with advanced strategies to difficult distinguishing them from legitimate messages. Thus, for this use case, the attacker transmits fabricated mes-

sages that mimic fault events under normal situations. This includes reproducing the retransmission period between two messages (see Figure 5), as well as *StNum* and *SqNum* field values to cause the same behavior shift as the correspondent legitimate messages.

This use case was set up to suggest the existence of a fault through GOOSE messages while the SV measurements are normal during random periods of 100 *ms* for each 1-second simulation run, accordingly to the traffic generation parameters presented in Section 5.2.3. Thus, these attacks are never carried out when an outage event is occurring. Accordingly, the generated SV data corresponds to the normal line operation whereas GOOSE messages carry a control block status that corresponds to a fault.

#### 5.2.4.4 Masquerade Attacks towards Equipment Damage (UC04)

This use case follows the same logic as UC03. However, instead of sending malicious GOOSE messages to report a fake fault during a normal situation, it sends malicious GOOSE messages reporting a normal status operation during a fault. The attacker transmits fabricated messages that also mimic the GOOSE bursting (described in Section 5.2.3), *StNum* and *SqNum* field values to cause the same behavior shift as the correspondent legitimate messages (see Figure 5).

This use case inserts fake messages in the time window between 500 *ms* and 600 *ms* of each 1-second simulation run (*i.e.*, the same period during which faults are set up to occur).

#### 5.2.4.5 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 (*i.e.*, without observing its consistency with the IEC-61850 standard) or with modifications that comply with the standard.

In contrast to previous use cases, in the UC05 the attacker does not need to capture previous messages. It can just inject new fabricated messages. Thus, these messages are not likely to have valid values for field as *sqNum* and *stNum*.



#### 5.2.4.6 High Status Number Attack (UC06)

This use case represents a poisoning attack that 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 put random values (from 10,000 to 100,000) in this field. This range was set to reach a trade-off considering that the higher the forged value is, the easier it is to detect the attack, but larger values cause more damage as it takes more time for legitimate messages to reach/surpass the fake increased status.

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

#### 5.2.4.7 High-Rate Flooding Attack (UC07)

This use case explores DoS attacks, in which an attacker floods the multicast channel by sending multiple fake messages. Each fake message will contain an increased value of the status number, as expected at the subscriber devices.

Since it is increasing each message's `StNum`, at a later time, a legitimate message will be discarded because its `StNum` is smaller than the current one.

Therefore, the result will be similar to the previous variant, except by the increased difficulty of distinguishing the legitimate messages from the flooded fake messages. Additionally, this attack can cause resource exhaustion.

## 5.3 Feature Extraction and Enrichment

Once the traffic is generated by ERENO, the resulting output is an IEC-61850 dataset in a format ready to be processed by machine learning algorithms (*e.g.*, J48 algorithm). Whereas some features are directly extracted from the network packets (*e.g.*, GOOSE and SV APDU), other features are generated from the enrichment process described below. This process is responsible for generating more representative features from the basic features through their correlation and simple computations involving information on one or more features or messages.

### 5.3.1 SV Features

The SV protocol payload has important data regarding the physical devices, including the measured current (i) and voltage (v) from merging unities and other IEDs. These features can be collected at multiple devices. In our scenario, we assume a three-phase transmission line connecting two substations (see Section 5.2.2 for more details). Therefore, we can extract 13 basic SV features: the current timestamp (1), the current from each phase and substation ( $3 \text{ phases} \times 2 \text{ substations} = 6$ ), and the voltage from each phase and substation ( $3 \text{ phases} \times 2 \text{ substations} = 6$ ). The full list of basic SV features is listed as follows:

1. **time**: The SV timestamp.
2. **isbA**: Current from Samambaia substation (sb) at Phase A.
3. **isbB**: Current from Samambaia at Phase B.
4. **isbC**: Current from Samambaia at Phase C.
5. **ismA**: Current from Serra da Mesa substation (sm) at Phase A.
6. **ismB**: Current from Serra da Mesa at Phase B.
7. **ismC**: Current from Serra da Mesa at Phase C.
8. **vsbA**: Voltage from Samambaia at Phase A.
9. **vsbB**: Voltage from Samambaia at Phase B.
10. **vsbC**: Voltage from Samambaia at Phase C.
11. **vsmA**: Voltage from Serra da Mesa at Phase A.
12. **vsmB**: Voltage from Serra da Mesa at Phase B.
13. **vsmC**: Voltage from Serra da Mesa at Phase C.

The aforementioned features can be used by IDSs to create models that represent the current status of the power grid. These model can be correlated with the substation events (*e.g.*, those reported from GOOSE messages). Any inconsistency in this correlation can be considered suspicious, malicious or anomalous activity.

### 5.3.2 Enriched SV Features

Each current and voltage value represents a snapshot of a reading on a specific timestamp. Therefore, this information by itself may be of little relevance to an IDS. On the other hand, consecutive current and voltage measures can be used to represent an electrical waveform. A typical electrical waveform during the normal operation of the transmission line follows the behavior illustrated in Figure 10.

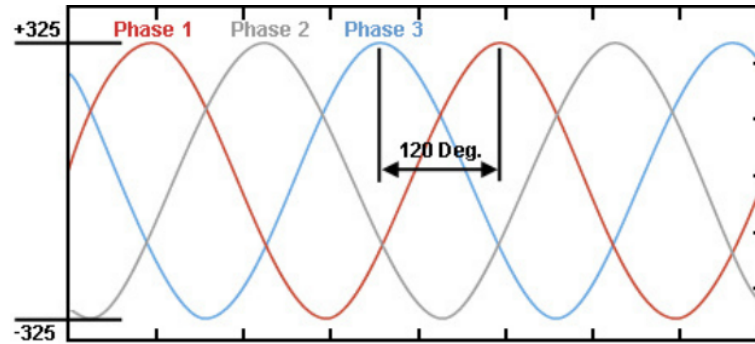


Figure 10: Balanced three-phase readings (OMAKAZI, 2022).

The occurrence of events on the transmission line may affect the current and voltage variables. When multiple consecutive values are analyzed together, it is possible to estimate the current transmission line state and detect anomalies (*e.g.*, electrical faults and line maintenance). A hypothetical situation of anomalous signals in the one of the three phases (*i.e.*, Phase 1) is shown in Figure 11. This is considered anomalous because it does not follow the expected behavior in comparison to the other phases (*i.e.*, its lagging of  $140^\circ$  instead of the expected  $120^\circ$  and its amplitude is higher than the normal range).

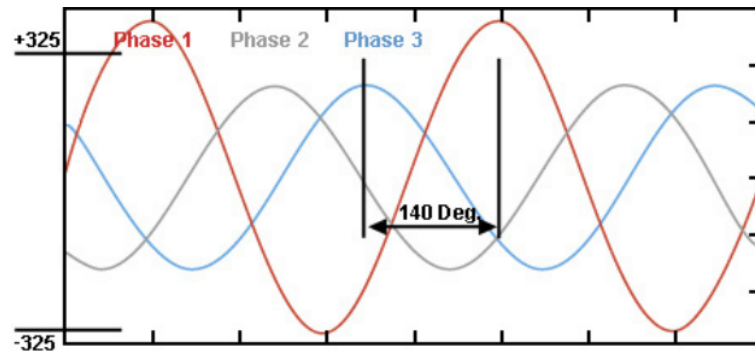


Figure 11: Unbalanced three-phase readings (OMAKAZI, 2022).

In electrical grids protection functions, there are statistical methods to combine multiple readings of the same feature on different timestamps and generate more representative features. We explore two methods to derive novel features: *Trapezoidal Area Sum* (TrapAreaSum) (WEIDEMAN, 2002) and *Root Mean Square* (RmsValue) (SATOH et al.,

2020).

The enriched SV features based on the *RMS Value* method are the following:

14. **isbARmsValue**: Current RMS Value from Samambaia at Phase A.
15. **isbBRmsValue**: Current RMS Value from Samambaia at Phase B.
16. **isbCRmsValue**: Current RMS Value from Samambaia at Phase C.
17. **ismARmsValue**: Current RMS Value from Serra da Mesa at Phase A.
18. **ismBRmsValue**: Current RMS Value from Serra da Mesa at Phase B.
19. **ismCRmsValue**: Current RMS Value from Serra da Mesa at Phase C.
20. **vsbARmsValue**: Voltage RMS Value from Samambaia at Phase A.
21. **vsbBRmsValue**: Voltage RMS Value from Samambaia at Phase B.
22. **vsbCRmsValue**: Voltage RMS Value from Samambaia at Phase C.
23. **vsmARmsValue**: Voltage RMS Value from the voltage from Serra da Mesa at Phase A.
24. **vsmBRmsValue**: Voltage RMS Value from the voltage from Serra da Mesa at Phase B.
25. **vsmCRmsValue**: Voltage RMS Value from the voltage from Serra da Mesa at Phase C.

The enriched SV features based on the *TrapAreaSum* method are the following:

26. **isbATrapAreaSum**: Current Trapezoidal Area Sum from Samambaia at Phase A.
27. **isbBTrapAreaSum**: Current Trapezoidal Area Sum from Samambaia at Phase B.
28. **isbCTrapAreaSum**: Current Trapezoidal Area Sum from Samambaia at Phase C.
29. **ismATrapAreaSum**: Current Trapezoidal Area Sum from Serra da Mesa at Phase A.
30. **ismBTrapAreaSum**: Current Trapezoidal Area Sum from Serra da Mesa at Phase B.

31. **ismCTrapAreaSum**: Current Trapezoidal Area Sum from Serra da Mesa at Phase C.
32. **vsbATrapAreaSum**: Voltage Trapezoidal Area Sum from Samambaia at Phase A.
33. **vsbBTrapAreaSum**: Voltage Trapezoidal Area Sum from Samambaia at Phase B.
34. **vsbCTrapAreaSum**: Voltage Trapezoidal Area Sum from Samambaia at Phase C.
35. **vsmATrapAreaSum**: Voltage Trapezoidal Area Sum from Serra da Mesa at Phase A.
36. **vsmBTrapAreaSum**: Voltage Trapezoidal Area Sum from Serra da Mesa at Phase B.
37. **vsmCTrapAreaSum**: Voltage Trapezoidal Area Sum from Serra da Mesa at Phase C.

### 5.3.3 Basic GOOSE Features

The basic GOOSE features extracted are those that can be directly extracted from the individual GOOSE messages. We propose the extraction of 22 features from GOOSE without any additional processing — similar to the SV Basic Features extraction.

38. **t**: The timestamp of the last state change;
39. **gooseTimestamp**: The GOOSE timestamp;
40. **sqNum**: The GOOSE sequence number;
41. **stNum**: The GOOSE status number;
42. **cbStatus**: Circuit-breaker status on GOOSE;
43. **frameLen**: The GOOSE ethernet frame length;
44. **ethDst**: The GOOSE ethernet destination address;
45. **ethSrc**: The GOOSE ethernet frame source address;

46. **ethType**: The GOOSE ethernet frame type;
47. **gooseTTL**: The time allowed to live;
48. **gooseAppid**: The GOOSE application ID;
49. **gooseLen**: The GOOSE frame length;
50. **TPID**: The tag priority ID;
51. **gocbRef**: The GOOSE control block reference;
52. **datSet**: The IED dataset path;
53. **goID**: The GOOSE flow ID;
54. **test**: The test flag;
55. **confRev**: The configuration revision;
56. **ndsCom**: The GOOSE NDSCOM parameter;
57. **numDatSetEntries**: The number of entries on the datSet;
58. **APDUSize**: The Application Data Unit (APDU) size;
59. **protocol**: The used protocol (expected: GOOSE).

### 5.3.4 Enriched GOOSE Features

The GOOSE traffic may also gain representativity if its features have their values combined along the time. Based on this assumption we compose 10 novel features that can be used to check the consistency between the behavior from multiple consecutive messages and the proposed behavior on the IEC-61850 standards (see Section 2.1.2.2 for more details about the expected GOOSE behavior).

From the 10 enriched GOOSE features, 8 are generated by computing the difference between the feature value of the current message ( $n$ ) and the feature value of the immediately previous message ( $n-1$ ).

60. **stDiff**:  $\text{StNum}\{n\} - \text{StNum}\{n-1\}$ ;
61. **sqDiff**:  $\text{SqNum}\{n\} - \text{SqNum}\{n-1\}$ ;

- 62. **gooseLengthDiff**:  $\text{gooseLength}\{n\} - \text{gooseLength}\{n-1\}$ ;
- 63. **cbStatusDiff**:  $\text{cbStatus}\{n\} - \text{cbStatus}\{n-1\}$ ;
- 64. **apduSizeDiff**:  $\text{apduSize}\{n\} - \text{apduSize}\{n-1\}$ ;
- 65. **frameLengthDiff**:  $\text{frameLength}\{n\} - \text{frameLength}\{n-1\}$ ;
- 66. **timestampDiff**:  $\text{timestamp}\{n\} - \text{timestamp}\{n-1\}$ ;
- 67. **tDiff**:  $t\{n\} - t\{n-1\}$ ;

The last two features are computed combining two different pieces of seemingly unrelated information. First, we compute the difference between the current GOOSE message timestamp (*i.e.*,  $\text{gooseTimesTamp}$ ) and the timestamp of the last GOOSE message with a status change (*i.e.*,  $t$ ). Then, we compute the difference between the current GOOSE message timestamp and the timestamp of the last captured SV message. These features are computed as follows:

- 68. **timeFromLastChange**:  $\text{gooseTimesTamp}\{n\} - t\{n\}$ ;
- 69. **delay**:  $\text{gooseTimestamp}\{n\} - \text{time}\{n\}$ .

The proposed feature enrichment is based on the assumption that multiple consecutive messages can provide more information about the behavioral shifting during malicious actions (*e.g.*, sequence number resetting after the status changes on substation devices). Similarly, the difference between the timestamps from two consecutive messages may reveal messages being transmitted in an improper high-rate.

## 5.4 Feature Selection: GRASP-FS

As the ERENO traffic generator can generate any feature that can be derived from SV and GOOSE protocols — note that we propose 69 features but others can be implemented into the ERENO open source project —, it is important to define which features are more suitable for IDSs to detect precisely multiple types of attacks.

Therefore, as an additional contribution to the ERENO tool, we propose the GRASP-FS as an implementation of the GRASP metaheuristic applied to the Feature Selection (FS) problem. The GRASP metaheuristic (QUINCOZES; PASSOS, et al., 2020) is a

multi-start or iterative process, in which each iteration consists of two phases: construction and local search, as shown in Figure 12.

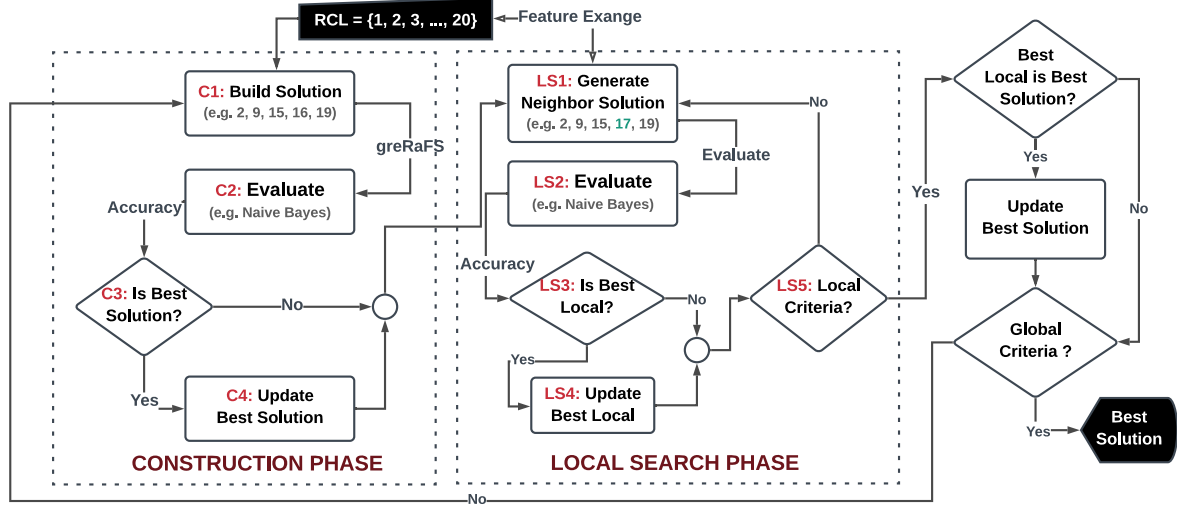


Figure 12: GRASP-based FS process.

The construction phase builds a feasible solution (*i.e.*, using features previously assessed through a filter-based method), at the step C1 in Figure 12, whose neighborhood is investigated until a local optimum solution is found during the local search phase, at the steps LS1 to LS5 in Figure 12. These phases are called in lines 5 and 10 of Algorithm 1.

The use of GRASP applied to the feature selection problem enables the increase of a fitness function (*e.g.*, accuracy, or F1-Score) with a parameterized computational cost. Thus, an optimized feature subset is given as a solution.

Our GRASP implementation adapted to FS problem has the following input parameters: the entire dataset features (`all`), the maximum global number of iterations (`maxTime`), and the desired feature subset length (`|FS|`). Note that the latter is used only when required by the local search method (*e.g.*, bit-flip). Some methods have their own stop criteria (*e.g.*, incremental wrapper methods). To reduce the feature space for analysis, an Restricted Candidate List (RCL) is considered. Also, note that generating an RCL implies excluding features from both construction and local search phases. In our implementation, the RCL is generated once to avoid unnecessary overhead.

The subsets generated in the construction phase are used as seed solutions for the neighborhood exploration, during the local search procedure. This entire process is repeated until the global criteria is reached (*i.e.*, `maxTime`). For each general iteration, the local search runs until the maximum iteration number (`maxIt`) is reached. Each feasible



---

**Algorithm 1:** GRASP-BASED FS ALGORITHM (QUINCOZES; PASSOS, ET AL., 2020)
 

---

```

input : all // all available features
         maxTime // maximum GRASP running time
         L-RCL // qty. of features to compose RCL
         |FS| // initial feature length
         maxIt // maximum number of iteration
output: bestFS // optimized feature subset
1 begin
2   bestFS  $\leftarrow \emptyset$ 
3   bestFS.accuracy  $\leftarrow 0$ 
4   while currentTime < maxTime do
5     greRaFS  $\leftarrow$  construct(|FS|, L-RCL, all)
6     greRaFS.accuracy  $\leftarrow$  eval(greRaFS)
7     if greRaFS.accuracy > bestFS.accuracy then
8       | bestFS  $\leftarrow$  greRaFS
9     end
10    bestLocal  $\leftarrow$  localSearch(greRaFS, RCL, maxIt)
11    if bestLocal.accuracy > bestFS.accuracy then
12      | bestFS  $\leftarrow$  bestLocal
13    end
14  end
15 end
16 return bestFS

```

---

solution found is evaluated through the fitness function and the best overall solution is kept. When the local criteria is achieved, the best local solution is given as result and then compared to the best global solution. After `maxTime` is reached, the best overall result is given as the final output. This output is perhaps a sub-optimal selection because of the stopping criteria. Details of the construction and local phases are presented in the rest of this chapter.

### 5.4.1 Construction Phase

In the construction phase, an initial solution is generated based on a greedy and randomized process. However, to avoid waste of computational and time resources, instead of a full GRASP iteration from a poor seed solution, the traditional GRASP algorithm introduces the concept of RCL. Such elimination of features is typically defined by using some knowledge about the problem at hand (*e.g.*, from a domain expert) or through metrics supported by statistical methods. A small RCL typically provides less diversity and a greater risk of overfitting as well as a potential for missing important features. In contrast, a very large RCL reduces the chance of a good solution being achieved in a reasonable time. Thus, the RCL size is important to the trade-off between diversity and overfitting. In FS, filter-based methods can be used to create the RCL due to their small

computational complexity (ESSEGHIR, 2010; QUINCOZES; PASSOS, et al., 2020), but the features selected to compose the RCL do not guarantee that the generated solutions will have good results (*e.g.*, precision or F1 scores). Thus, using wrapper-based methods may also be an alternative to the RCL generation.

---

**Algorithm 2: CONSTRUCTION PHASE**


---

```

input : all // all available features
         L-RCL // qty. of features to compose RCL
         |FS| // initial feature length
output: greRaFS // Greedy-Random FS subset
1 begin
2   if rclFeatures =  $\emptyset$  then
3     foreach feature  $\in$  all do
4       GR  $\leftarrow$  getGr(feature)
5     end
6     all  $\leftarrow$  attributeRankByGR(all)
7     rclFeatures  $\leftarrow$  selectTopRanked(all, L-RCL)
8   end
9   greRaFS  $\leftarrow$   $\emptyset$ 
10  while (|greRaFS| < |FS|) do
11    greRaFS  $\leftarrow$   $\cup$  selectRandomly(feature  $\in$  rclFeatures )
12  end
13 end
14 return greRaFS

```

---

The Gain Ratio (GR) filter-based method performs feature ranking at the construction phase and then selects the top-ranked features. Therefore, in the first GRASP iteration, when the `rclFeatures` is empty, the GR is computed for each feature (see line 4, Algorithm 2). Once the RCL is generated, it is used to generate the initial solution (*i.e.*, *greRaFS*, in Figure 12) in a greedy and randomized way for each GRASP iteration. In particular, features are chosen randomly from the RCL to be added to *greRaFS* (see line 11, Algorithm 2) until the solution length reaches the threshold `|FS|`. This step is represented as Construction 1 (C1), in Figure 12, and is followed by a wrapper evaluation (C2). The evaluated solution is compared to the current best solution (C3), which is updated when outperformed (C4).

Note that any algorithm or method can be used as an alternative to GR, including the full set of features or manually chosen features. In our substation scenarios, we can choose to use specific GOOSE and/or SV features as part of the RCL, for example.

### 5.4.2 Local Search Phase

Starting from a seed solution, the local search consists of a set of *local movements* to derive neighbor solutions (*i.e.*, new similar feature subsets with few features different from the

seed subset). We use *greRaFS* from the construction phase as a seed to the neighborhood exploration, represented as LS1 (local search) in Figure 12. The next steps are represented by LS#, where # is their sequence number. The local movements are defined by the neighborhood structure (*i.e.*, algorithms that handle a set of local search algorithms) used in each LS#, which can be drawn among the ones that follow. The Bit-Flip (BF) local search explores neighboring solutions by exchanging features between the RCL and *greRaFS* (*e.g.*, adding one feature and removing another per iteration). The Incremental Wrapper-based subset Selection (IWSS) local search performs an iterative feature selection, keeping only those features that present an improvement in the fitness function. The IWSS with replacement (IWSSr) local search also performs an incremental subset selection, but considers also feature replacement instead of just adding new ones (BERMEJO; GAMEZ; PUERTA, 2011; MORADKHANI et al., 2015). Thus, whereas IWSS only discards new features without improvements, IWSSr may remove previously selected features if their replacement by new ones can improve the solution.

---

**Algorithm 3: LOCAL SEARCH**


---

```

input : RCL // Restricted Candidate List of features
         greRaFS // Seed solution to neighbors generation
         maxIt // Maximum number of iteration
output: bestLocal // Best feature set in the local neighborhood
1 begin
2   bestLocal ← greRaFS
3   while currentIteration++ < maxIt do
4     bestNeighborFS ← localMovement (greRaFS, RCL)
5     bestNeighborFS.accuracy ← evaluate(bestNeighborFS, classifier) // eval
6     if bestNeighborFS.accuracy > bestLocal.accuracy then
7       | bestLocal ← bestNeighborFS.accuracy
8     end
9   end
10 end
    return : bestLocal

```

---

In our previous work (QUINCOZES; PASSOS, et al., 2020) we implemented only Bit-Flip algorithm as a simple local search neighborhood structure. In further experiments, we also assess the Variable Neighborhood Descent (VND) (HANSEN; MLADENOVIC, 1999) and the Random Variable Neighborhood Descent (RVND) (PENNA; SUBRAMANIAN; SATORU, 2013) metaheuristics, which can use multiple neighborhood structures, including the Bit-Flip and two others: IWSS and IWSSr. In particular, RVND chooses and removes neighborhoods structure randomly at each iteration, whereas VND does it sequentially. Also, RVND employs a reset procedure: after each local search method execution, if the found solution outperforms the current best solution, the previous methods are put again in the neighborhood structure list and may be reselected (*i.e.*, if it is

randomly selected again).

Regardless of the method used to perform `localMovement`, each generated solution should be evaluated through the selected fitness function, represented as LS2 in Figure 12. If the evaluated solution is the best in the local neighborhood (LS3), the best local solution is updated (LS4). We employ classifier algorithms as *wrapper* methods to perform evaluations and compute the classifier accuracy as a fitness function (see line 5 in Algorithm 3). The local search procedures (*i.e.*, `localMovement` and `evaluate` operations) are repeated until a local stop criteria is reached (LS5), in this case, a maximum number of iterations (`maxIt`).

# 6 Experiments, Results and Discussion

## 6.1 Feature Processing: Selection and Enrichment

In this section, we study two feature processing techniques to improve the IDSs<sup>1</sup> results: Feature Enrichment and Feature Selection. Whereas the former aims at generating novel features to provide more valuable information, the latter aims at discarding irrelevant or redundant features (see Section 3.2). This study aims at answering the following questions:

- How can Feature Selection improve the performance of IDSs in detecting each class of attack?
- Can Feature Enrichment be an alternative for improving the IDSs' performance when Feature Selection is not enough?

### 6.1.1 Feature Selection

We carried out experiments to measure the feature selection performed by GRASP-FS Metaheuristic, as proposed in Section 5.4. Since our scope is on attacks targeting the GOOSE protocol, we take the 22 basic GOOSE features (presented in Section 5.3.3) as our baseline case. These are the features with index 38 to 59 in our proof-of-concept dataset. Therefore, these features were selected to compose the RCL for GRASP-FS. This means that both construction and local search procedures are limited to using the basic GOOSE features.

The parameters used to set up the GRASP-FS algorithm are shown in Table 6. Both training and testing datasets have the same number of samples, generated by using the ERENO tool. The chosen local search algorithms do not require a fixed-length feature subset. We aim at assessing all basic GOOSE features, thus we do not restrict the candidate features from RCL to a smaller subset. We use the F1-Score metric because it can

---

<sup>1</sup>We use the J48 algorithm as an IDS implementation in our results.

Table 6: The chosen parameters for our methodology.

Parameter	Value
Portion of data used for training	50%
Portion of data used for testing	50%
Number of features to select (N)	Dynamic
Number of features to compose the RCL	22 (max GOOSE features)
Basis for RCL generation	All Basic GOOSE Features
Objective Function (decision criteria)	F1-Score
Classifier Algorithm	J48
Local Search Algorithms	IWSSr, BF, and IWSS
Neighborhood Structure	RVND

deal with unbalanced datasets. Based on previous experiments, we choose: the J48 algorithms to perform wrapping evaluations; IWSSr, BF, and IWSS to perform local search; and, RVND neighborhood structure ([QUINCOZES; PASSOS, et al., 2020](#); [QUINCOZES; KAZIENKO; COPETTI, 2018](#)).

### 6.1.2 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 as a baseline and then we perform incremental feature enrichment:

- **GOOSE**: only the basic GOOSE features, presented in Section 5.3.3. These are the features with index 38 to 59 in our proof-of-concept dataset.
- **GOOSE & SV**: the basic GOOSE features (described above) and basic SV features presented in Section 5.3.1. The SV features are those with index 1 to 13 in our proof-of-concept dataset.
- **GOOSE & SV++**: the basic GOOSE and basic SV features (as the previous item) with addition of the enriched SV features presented in Section 5.3.2. The enriched SV features are those with index 14 to 37 in our proof-of-concept dataset.
- **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).

### 6.1.3 Results for Feature Processing

#### 6.1.3.1 Random Replay attacks (UC01)

The detailed results for the UC01 are shown in Table 7.

J48 reached a accuracy (97.44%) and recall (99.95%) upper the 97% for detecting Random Replay attacks (UC01) with the 22 basic GOOSE features. However, its precision was only 78.82%, which leads the F1-Score to 88.14%.

After the GRASP-FS processing, only 3 features were selected (*i.e.*, features 38, 39, and 58). Interestingly, this reduced feature subset improved the F1-Score to 93.06%. The accuracy was improved to 98.58% and precision to 87.05%. The recall was not affected, albeit it was already high. Thus, the main improvements with GRASP-FS were observed for the precision (+8.23%) and F1-Score (+4.92%) metrics.

Table 7: Detection performance of J48 for Random Replay attacks with different input features.

	Accuracy	Precision	Recall	F1-Score	TP	TN	FP	FN
GOOSE	97.44%	78.82%	99.95%	88.14%	38982	360922	10475	18
GRASP-FS	98.58%	87.05%	99.95%	93.06%	38982	365599	5798	18
GOOSE & SV	93.75%	60.32%	99.96%	75.23%	38983	345748	25649	17
GOOSE & SV++	92.56%	56.08%	99.88%	71.83%	38954	340894	30503	46
GOOSE++ & SV++	<b>100%</b>	<b>100%</b>	<b>99.99%</b>	<b>99.99%</b>	<b>38996</b>	<b>371397</b>	<b>0</b>	<b>4</b>

Since the F1-Score reached by J48 with GRASP-FS for the UC01 is still below 94%, we experiment with more features than just the basic GOOSE ones. Both the GOOSE & SV and GOOSE & SV++ features were not able to enhance the IDS performance – they had 75.23% and 71.83% F1-Score. These results show that having more features is not always good (*i.e.*, the additional features may introduce wrong decisions when they are irrelevant). On the other hand, GOOSE++ & SV++ features, without GRASP-FS, lead the F1-Score to 99.99%.

#### 6.1.3.2 Inverse Replay Attacks (UC02)

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

The GRASP-FS procedure selected 3 features from the 22 available (*i.e.*, features 40, 42, and 58). This reduced feature subset improved the F1-Score to 93.85%. The accuracy

was slightly improved to 99.35% and the recall to 91.54%. However, precision was worse (96.28%). Note that GRASP-FS does not yield worse solutions: the chosen selection criteria is F1-Score and, considering both precision and recall, the F1-Score (93.85%) is still slightly higher than using just the basic GOOSE features (93.5%).

Table 8: Inverse Replay attacks feature selection and enrichment assessment.

	Accuracy	Precision	Recall	F1Score	TP	TN	FP	FN
GOOSE	99.33%	98.27%	89.18%	93.50%	27037	526281	476	3282
GRASP-FS	99.35%	96.28%	91.54%	93.85%	27754	525686	1071	2565
GOOSE & SV	99.33%	99.93%	87.77%	93.46%	26612	526738	19	3707
GOOSE & SV++	99.81%	100.00%	96.52%	98.23%	29264	526757	0	1055
GOOSE++ & SV++	<b>99.93%</b>	<b>100.00%</b>	<b>98.80%</b>	<b>99.39%</b>	<b>29954</b>	<b>526757</b>	<b>0</b>	<b>365</b>

Clearly, only feature selection was not enough for significantly enhancing the IDS performance on this scenario. Thus, as an alternative, feature enrichment was used. The GOOSE & SV provided a barely worse performance. However, GOOSE & SV++ and GOOSE++ & SV++ features were able to improve the results. In particular, GOOSE++ & SV++ lead the F1-Score to 99.39% – an gain of 5.89%. The detailed results are shown in Table 8.

### 6.1.3.3 Masquerade Attacks towards 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%) lead the J48 to reach the lowest F1-Score (52.24%) when compared to all other studied attacks. Actually, these results are already expected since masquerade attacks are designed to confuse the IDSs by mimicking the legitimate behavior. Thus, neither GOOSE nor GRASP-FS are able to significantly improve the IDS accuracy – GRASP-FS reached 53.4% F1-Score with 3 features: 39, 41, 42.

Table 9: Masquerade (outage) attacks feature selection and enrichment assessment.

	Accuracy	Precision	Recall	F1Score	TP	TN	FP	FN
GOOSE	94.78%	43.87%	64.55%	52.24%	11102	357194	14203	6098
GRASP-FS	94.66%	43.48%	69.17%	53.40%	11898	355930	15467	5302
GOOSE & SV	95.68%	51.01%	60.31%	55.27%	10374	361435	9962	6826
GOOSE & SV++	97.06%	65.52%	70.65%	67.99%	12152	365003	6394	5048
GOOSE++ & SV++	<b>99.95%</b>	<b>99.70%</b>	<b>99.22%</b>	<b>99.46%</b>	<b>17065</b>	<b>371345</b>	<b>52</b>	<b>135</b>

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 does not present significative improvement –



the F1-Score improves only 3.03%. In fact, basic SV features are not very representative for masquerade attacks. However, when SV and GOOSE features are enriched, they gain information value and can detect 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 9.

#### 6.1.3.4 Masquerade Attacks towards Equipment Damage (UC04)

Although Masquerade Attacks towards Equipment Damage (UC04) detection is not as hard as the UC03, it follows a similar logic. J48 reached a low performance when using the GOOSE features. The low precision (56.16%) and recall (80.1%) lead the F1-Score to 65.96%. As in UC03, these results are also already expected by the same reason: masquerade attacks are designed to confuse the IDSs by mimicking the legitimate behavior. Although GRASP-FS presented an improvement on the F1-Score (+6.56%) with four features (*i.e.*, 39, 40, 42, and 58), this metric is still low (72.52%).

Table 10: Masquerade (equipment damage) attacks feature selection and enrichment assessment.

	Accuracy	Precision	Recall	F1Score	TP	TN	FP	FN
GOOSE	96.30%	56.06%	80.10%	65.96%	13954	360461	10936	3466
GRASP-FS	97.22%	65.16%	81.75%	72.52%	14241	363781	7616	3179
GOOSE & SV	97.67%	66.02%	99.02%	79.22%	17250	362518	8879	170
GOOSE & SV++	<b>99.99%</b>	<b>99.94%</b>	<b>99.88%</b>	<b>99.91%</b>	<b>17399</b>	<b>371386</b>	<b>11</b>	<b>21</b>
GOOSE++ & SV++	99.98%	99.87%	99.79%	99.83%	17383	371375	22	<b>37</b>

Thus, similarly to the UC03, the only way to overcome the challenge imposed by masquerade attacks is to combine GOOSE and SV features. In contrast to UC03 detection, only the SV features need to be enriched to detect UC04. Whereas GOOSE & SV features provide a 79.22% F1-Score, GOOSE & SV++ and GOOSE++ & SV++ can represent the masquerade attacks with a high performance: they reached 99.91% and 99.83% F1-Scores, respectively. In particular, the enrichment of SV features (SV++) for this attack is more relevant than the enrichment of GOOSE because SV++ can provide information about the actual transmission line status (*i.e.*, if there is a electrical fault occurring or not). The detailed results are shown in Table 10.

#### 6.1.3.5 Random Message Injection (UC05)

In contrast to the previously discussed attacks, the Random Message Injection (UC05) is easily detected by J48 even when using only the GOOSE features. It reached 100% on

all metrics. To keep the consistency with the discussion of other attacks, we show the detailed found results in Table 11.

Table 11: Random Message Injection attacks feature selection and enrichment assessment.

	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1Score</b>	<b>TP</b>	<b>TN</b>	<b>FP</b>	<b>FN</b>
GOOSE	100.00%	100.00%	100.00%	100.00%	39000	371397	0	0
GRASP-FS	100.00%	100.00%	100.00%	100.00%	39000	371397	0	0
GOOSE & SV	100.00%	100.00%	100.00%	100.00%	39000	371397	0	0
GOOSE & SV++	100.00%	100.00%	100.00%	100.00%	39000	371397	0	0
GOOSE++ & SV++	100.00%	100.00%	100.00%	100.00%	39000	371397	0	0

According to the presented results of **GOOSE**, it would be not necessary to use any advanced feature processing method except if the goal is to reduce the amount of information needed to process this class of attack.

### 6.1.3.6 High-Status Number Attack (UC06)

J48 reached a perfect recall (100%) and a reasonable good accuracy (98.84%) for detecting the High-Status Number (UC06) attacks. Nevertheless, its precision was only 89.12% and, consequently, its F1-Score was the metric was impaired. Thus, in this case, feature enrichment and feature selection may be assessed to improve such results. The results are shown in Table 12.

Table 12: High-Status Number attacks feature selection and enrichment assessment.

	<b>Accuracy</b>	<b>Precision</b>	<b>Recall</b>	<b>F1Score</b>	<b>TP</b>	<b>TN</b>	<b>FP</b>	<b>FN</b>
GOOSE	98.84%	89.12%	100.00%	94.25%	39000	366636	4761	0
GRASP-FS	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>39000</b>	<b>371397</b>	<b>0</b>	<b>0</b>
GOOSE & SV	98.84%	89.12%	100.00%	94.25%	39000	366636	4761	0
GOOSE & SV++	98.84%	89.12%	100.00%	94.25%	39000	366636	4761	0
GOOSE++ & SV++	98.84%	89.12%	100.00%	94.25%	39000	366636	4761	0

According to Table 12, none of those assessed feature subsets could provide more information regarding the High-Status Number attack. Consequently, they could not improve the results of our IDS implemented by the J48 algorithm.

On the other hand, the use of **GRASP-FS** resulted in a reduced subset of 5 features (*i.e.*, 40, 42, 47, 54, 57) that was able to reach 100% on all metrics. Thus, the **GOOSE** features were already enough to represent this attack but there were features that hindered or confused the analyses, and therefore needed to be discarded through the FS – that was where **GRASP-FS** excelled.

### 6.1.3.7 High-Rate Flooding Attack (UC07)

Similarly to the UC05, the High-Rate Flooding (UC07) attacks are easily detected by J48. It also reached 100% for all metrics, either using GOOSE, GOOSE & SV, GOOSE & SV++, or GOOSE++ & SV++. Thus, we recommend using only GOOSE to reduce the amount of information to be processed. The results are shown in Table 13.

Table 13: High-Rate Flooding attacks feature selection and enrichment assessment.

	Accuracy	Precision	Recall	F1Score	TP	TN	FP	FN
GOOSE	100.00%	100.00%	100.00%	100.00%	18570	371397	0	0
GRASP-FS	100.00%	100.00%	100.00%	100.00%	18570	371397	0	0
GOOSE & SV	100.00%	100.00%	100.00%	100.00%	18570	371397	0	0
GOOSE & SV++	100.00%	100.00%	100.00%	100.00%	18570	371397	0	0
GOOSE++ & SV++	100.00%	100.00%	100.00%	100.00%	18570	371397	0	0

## 6.2 Dataset Features Assessment

In this section, we study the generated dataset features to answer the following question:

- How well features are used for detecting each attack class?

The J48 classifier was chosen for such an analysis because it generates a single decision tree, thus enabling the assessment of the feature used to make decisions through the generated tree. Note that the J48 is a decision tree algorithm that employs a pruning method as an embedded feature selection (that is why each tree in this section has a different height). The expected results are the correct usage of the generated features to build accurate decision trees. Note that our goal is not to assess the J48 algorithm: we aim at checking if the generated features can enable J48 to build consistent decisions trees to make decisions (*e.g.*, without any bias).

In the following figures, the nodes (ellipses) represent features and arrows represent the decision taken by J48 according to their values compared to a threshold. The leaves (gray boxes) represent the J48 output classes and the number of instances classified in such class that followed the path from the root to that leaf.

The decision tree model generated by J48 for Random Replay attacks (UC01), shown in Figure 13, 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,

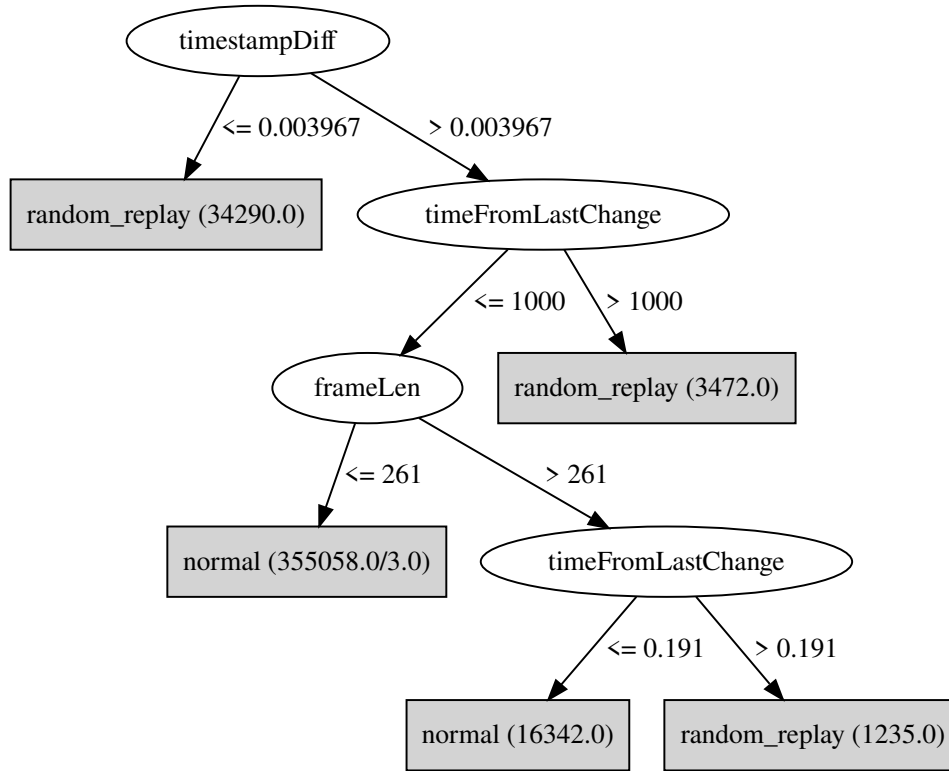


Figure 13: Random Replay Attacks (UC01)

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). Nevertheless, some attack instances eventually have consistent timestamps (*e.g.*, when a replay attack is sent at the expected time interval from the original message). Therefore, additional features (*e.g.*, `frameLen`), that may reveal messages with additional content, were used by J48 to build its decision trees in such cases. The combination of the aforementioned features enables the J48 decision tree to detect UC01 attacks with a very high F1-Score (99.99%). The precision and recall are 100% and 99.99%, respectively.

J48 used `sqDiff` at the root of the decision tree for the UC02, as shown in Figure 14. This feature enables J48 to decide whether and how much the `SqNum` was changed from the last GOOSE message. According to Section 2.1.2.2, 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. The `timeFromLastChange` and `timestampDiff` features are also used in the lower levels of the J48 decision tree. In this use case, J48 generated more false negatives (*i.e.*, 365) than for UC01 (*i.e.*, 4). Thus, the resulting accuracy, recall, and F1-Score were slightly lower: 99.93%, 98.80%, and 99.39%, respectively. The precision was 100% since precision itself does not consider false

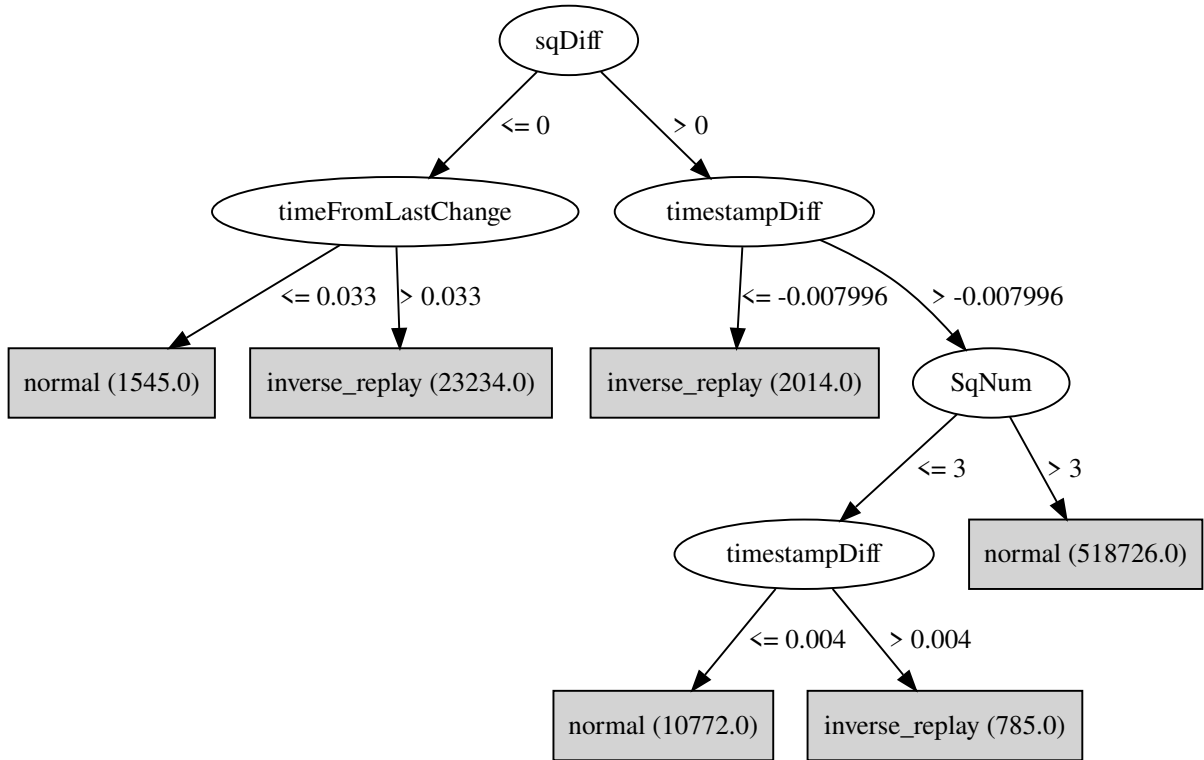


Figure 14: Inverse Replay Attacks (UC02)

negatives.

The UC03 and UC04 model more intelligent intruder behavior: instead of simply replaying old messages, it follows the requirements of IEC-61850 and attempts to imitate the legitimate users' behavior. Therefore, the decision trees built by J48 to detect these attacks are more complex and require a deeper analysis (more features need to be considered).

A very important contribution of our traffic generation tool, as well as the resulting generated dataset, is the possibility of cross referencing two IEC-61850 protocols by IDs: GOOSE and SV. With such a possibility, even very sophisticated GOOSE attacks can be identified based on the correlation of 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 `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 outage by reporting fake electrical faults (UC03). An outage

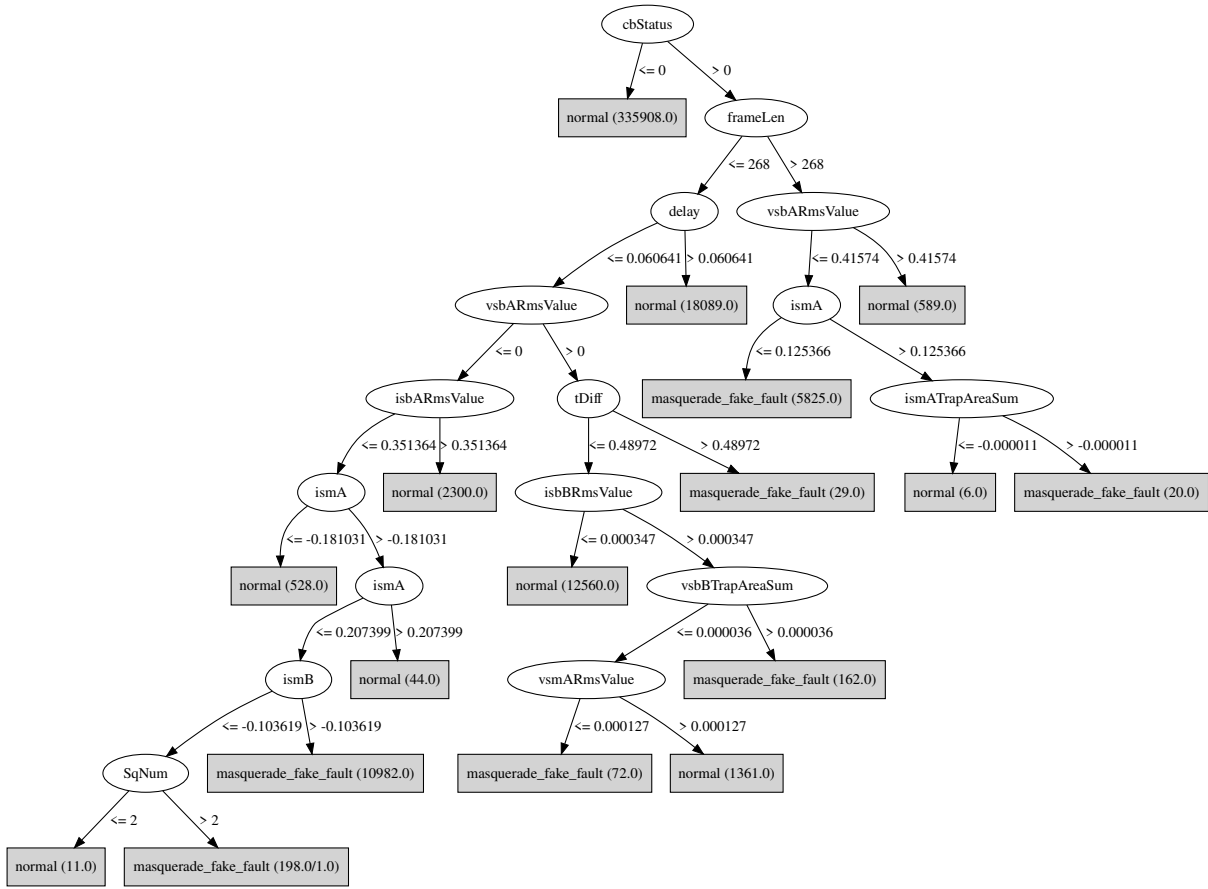


Figure 15: Masquerade Attacks - Outage (UC03)

may happen when an actuator IED process a modified GOOSE and responds by opening the circuit-breaker during normal operation. As shown in Figure 15, the root node is `cbStatus` that represent the fake status imposed by the masquerade attack. Whenever `cbStatus` represent a closed circuit-breaker (i.e., `cbStatus = 0`), J48 concludes that any UC03 is not being ran (i.e., it happens 335,908 times). Otherwise, other features are analyzed to check the its consistency to the `cbStatus` value. As already expected, most of these features are those derived from SV messages, such as the `vsbARmsValue` and `ismATrapAreaSum`. 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 occurring J48 concludes that a fake fault event is being sent by the masquerade attack. Besides, the `delay` feature is also used to improve the classification performance. J48 reached a 99.46% F1-Score, 99.70% precision, 99.22% recall, and 99.95% accuracy for the UC03 attack.

The Masquerade Attacks reporting fake normal situations (UC04) have similar logic to UC03, but a very different decision tree model was generated (see Figure 16). As in the UC03 decision tree model, the most predominant features are those related to the

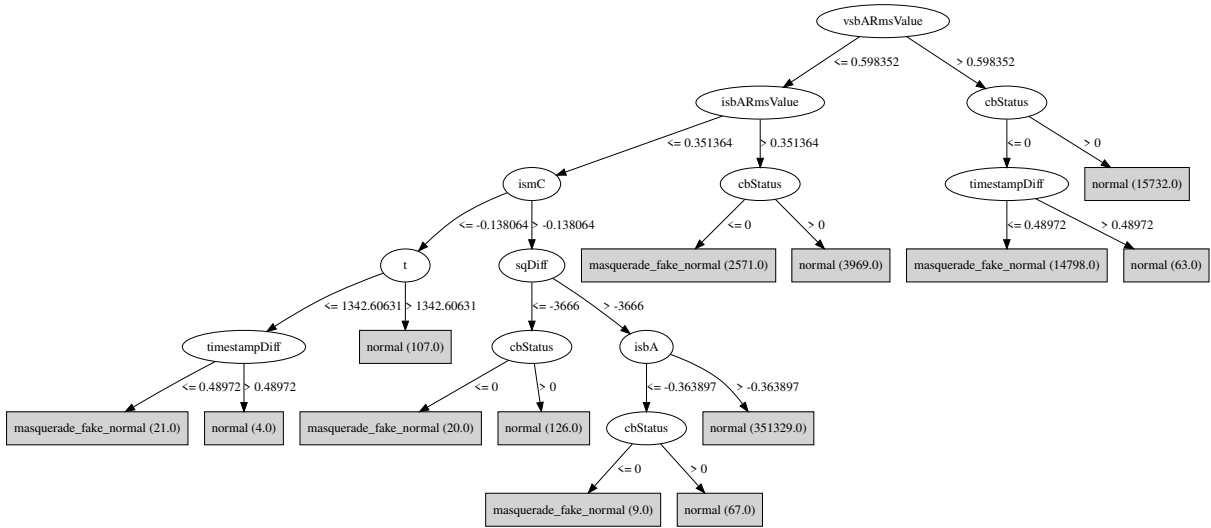


Figure 16: Masquerade Attacks - Equipment Damage (UC04)

electrical signals carried by SV and to the `cbStatus`. However, the root feature was the `vsbARmsValue`. Other electrical signals were used next to the root (e.g., `isbARmsValue`, `ismC`, and `isbA`). Besides, the final decision 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 electric fault. Therefore, the decision tree model generated by J48 also corresponds to the expected attacker behavior mapping for this attack. J48 reached an accuracy of 99.98%, precision of 99.87%, recall of 99.79%, and a F1-Score of 99.83%.

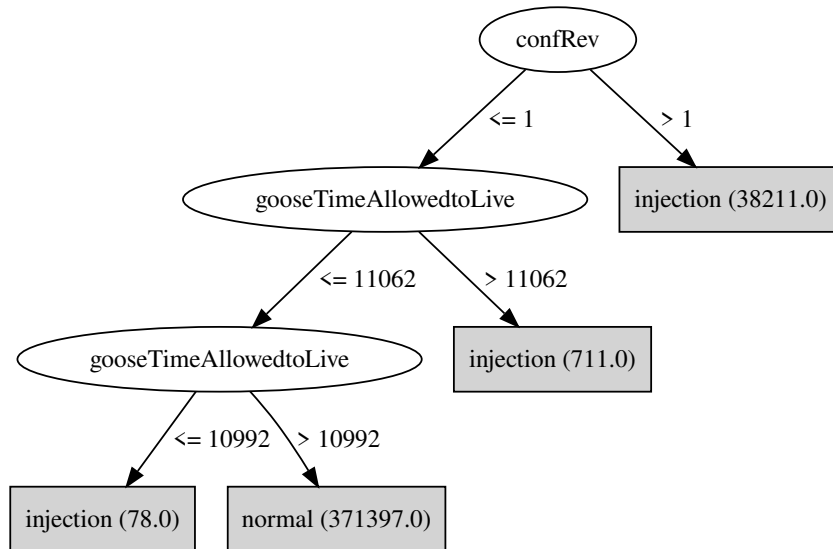


Figure 17: Random Message Injection attacks (UC05)

The Random Message Injection attacks (UC05) are simpler to detect than others due to the naive attacker behavior and lack of information about the targeting environment.

Therefore, there are multiple features that are inconsistent with the expected domain features (see Figure 17). 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 are detected by using the `gooseTimeAllowedtoLive` range. Note that when deploying an IDS, the legitimate value for `confRev` would change over the time. Only these two features are enough to enable a 99.99% accuracy, 100% precision, 99.99% recall, and 99.99% F1-Score.

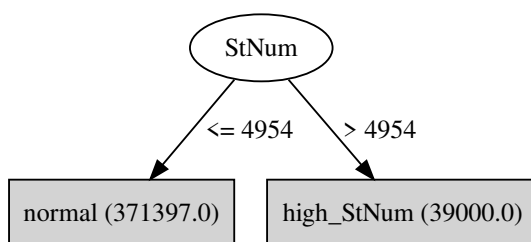


Figure 18: High Status Number attacks (UC06)

The decision tree built by J48 to classify the High Status Number attacks (UC06) is very simple (see Figure 18). Only the `StNum` feature is used to check whether 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 every message containing the `StNum` higher than 4954 is classified as an attack. However, although it reached 100% of recall, its precision was 89.12%. Similarly, its accuracy was 98.84% and its F1-Score is 94.24%. This result shows a potential point of improvement in the classification performance. Note that our scope is limited to ensuring a consistent dataset. Improving or assessing the classifier itself is not included in our goals, thus can be addressed by future work.

Finally, detecting the Poisoned High-Rate Attack (UC07) is also easy because the attacker sends too many messages in a very short period of time. Such a behavior originate a clearly anomalous `timestampDiff` (see Figure 19). In a few cases, the `frameLen` may help the decision because it maps messages with excessive length caused by the high `stNum` transmitted. The J48 algorithm generated zero false positives and negatives. Thus, accuracy, precision, recall, and F1-Score are 100%.

We summarize all aforementioned results for individual use cases analyses in Table 14. Based on these results we can conclude that the proposed features generated by ERENO



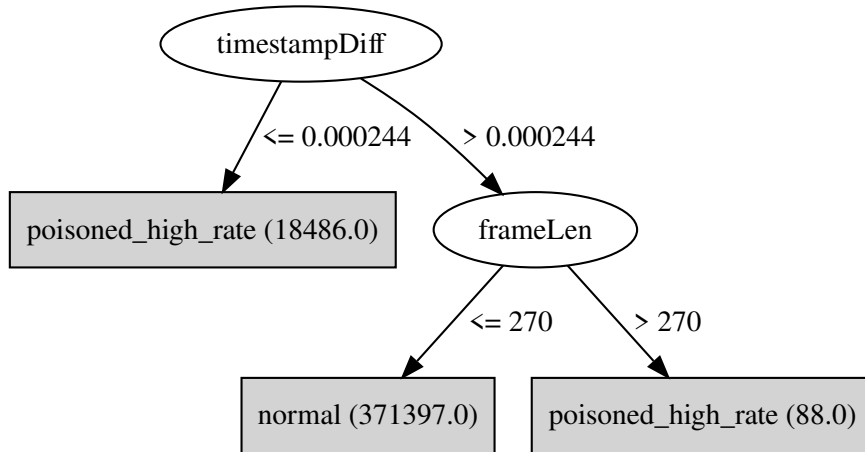


Figure 19: Poised High-Rate Attack (UC07)

Table 14: Results for each individual attack class

Use Case	Accuracy	Precision	Recall	F1Score
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%

are very representative to detect most of the attack use cases. This conclusion is in accordance with our hypothesis: by providing realistic datasets we described a range of 7 attacks and enabled an IDS based on the J48 classifier algorithm to reach good results.

## 6.3 Multi-class

In the previous sections we studied how each feature can be used to perform intrusion detection for different attack types individually (Section 6.2) and how feature enrichment and feature selection can improve the intrusion detection, also for each individual attack (Section 6.1). In both analysis we did not consider what happens when multiple attack classes are put together and how each attack class impacts the detection of each other.

Thus, in this section we assess a multi-class IDS implemented through the J48 algorithm. This analysis considers the same feature subsets previously studied for individual attack classes (GOOSE, GOOSE & SV, GOOSE & SV++, or GOOSE++ & SV++), but now considering all classes together. So, instead of performing multiple experiments with each

attack, we use a larger single dataset containing all of them and process it once. This study aims at answering the following questions:

- Can feature enrichment improve the multi-class intrusion detection?
- How does each attack class impact the performance of an IDS to detect other attacks classes?

In Section 6.3.1 we explore the confusion matrix and the weighted metrics for each feature subset.

### 6.3.1 Analyzing the Confusion Matrices

To assess how each attack impacts the detection of the others for each of the four feature subsets, we provide two pieces of information:

- The resulting confusion matrix. The rows in a confusion matrix represent the expected class and the columns represent the result of the classification. The main diagonal of the matrix represents the number of instances classified correctly;
- Detailed information regarding the metrics that can be extracted from it. Below the confusion matrix, we present both absolute and weighted metrics for each feature subset. The weighted metrics are used to fairly represent the performance of the IDS to detect attacks in multi-class scenarios, where each class is weighted according to its percentage contribution to the total number of instances in the dataset. These metrics are computed through the weighted average of the precision, recall, and F1-Score metrics for each attack according to the confusion matrix data.

Our experiments reveal that the weighted average of the F1-Score metric from all attack increases as the feature enrichment process is done. A summary of our analysis is presented in Table 15. As shown, the weighted F1-Score for the basic *GOOSE* features has the lowest value, less than 75%, whereas the *GOOSE++* & *SV++* features reach more than 99% for this metric. We present the detailed results and their confusion matrices in Tables 16, 17, 18 and 19.

In Table 16, we show the results for the basic *GOOSE* features, which enable the J48 classifier to reach an weighted F1-Score of 72.84% (bottom right) as result of its multi-class analysis. When taken separately, the hardest attack use case is the UC03 (with only

	Accuracy	Precision	Recall	F1Score
GOOSE	95.23%	75.98%	75.22%	72.84%
GOOSE & SV	97.18%	83.62%	84.45%	82.20%
GOOSE & SV++	98.29%	86.72%	89.66%	87.22%
GOOSE++ & SV++	99.88%	99.76%	99.29%	99.52%

Table 15: The impact of feature enrichment on multi-class detection.

13.73% F1-Score). The confusion matrix shows that most of these masquerade attack samples are being incorrectly classified into the normal class (UC00). On the other hand, most of the UC05 instances are correctly classified into the expected class and, thus, have a F1-Score of 99.64%.

Table 16: The resulting confusion matrix for J48 when using only the GOOSE features.

Exp. / Res.	UC00	UC01	UC02	UC03	UC04	UC05	UC06	UC07
UC00	<b>2710866</b>	11396	119	18930	13828	0	0	0
UC01	1411	<b>35321</b>	357	187	105	0	0	1619
UC02	2728	7441	<b>10547</b>	1399	0	641	0	7563
UC03	13001	836	297	<b>2927</b>	0	0	0	139
UC04	7514	408	0	0	<b>9260</b>	0	238	0
UC05	11	2	0	0	0	<b>38987</b>	0	0
UC06	6	36	0	0	0	0	<b>38958</b>	0
UC07	128	67	55	1983	0	0	0	<b>16337</b>

	Absolute Metrics				Weighted Metrics		
	Instances	Prec.	Rec.	F1Sc.	W-Prec.	W-Rec.	W-F1Sc.
UC01	39000 (19.45%)	90.57%	63.63%	74.75%	17.62%	12.38%	14.54%
UC02	30319 (15.12%)	34.79%	92.72%	50.59%	5.26%	14.02%	7.65%
UC03	17200 (8.58%)	17.02%	11.51%	13.73%	1.46%	0.99%	1.18%
UC04	17420 (8.69%)	53.16%	39.93%	45.60%	4.62%	3.47%	3.96%
UC05	39000 (19.45%)	99.97%	98.38%	99.17%	19.44%	19.14%	19.29%
UC06	39000 (19.45%)	99.89%	99.39%	99.64%	19.43%	19.33%	19.38%
UC07	18570 (9.26%)	87.98%	63.67%	73.88%	8.15%	5.90%	6.84%
					<b>75.98%</b>	<b>75.22%</b>	<b>72.84%</b>

In Table 17, we show that the GOOSE & SV features enable the J48 classifier to reach an weighted F1-Score of 82.20% as result of its multi-class analysis. When taken separately, the hardest attack use case is still the UC03 (but now J48 has reached a higher F1-Score of 38.15%). The confusion matrix shows that most of these masquerade attack samples are still being classified into the normal class (UC00). On the other hand, most of the UC05 instances are correctly classified into the expected class and, thus, have a F1-Score of 99.96%.

Table 17: The confusion matrix for J48 when using only the GOOSE &amp; SV features.

Exp. / Res.	UC00	UC01	UC02	UC03	UC04	UC05	UC06	UC07
UC00	<b>2737532</b>	6049	1	8935	2622	0	0	0
UC01	1424	<b>36669</b>	405	289	106	0	80	27
UC02	2750	9503	<b>12874</b>	1799	0	0	0	3393
UC03	9435	124	504	<b>7119</b>	0	0	0	18
UC04	167	408	0	0	<b>16613</b>	0	232	0
UC05	24	3	2	1	0	<b>38970</b>	0	0
UC06	1	19	4	6	4	0	<b>38965</b>	1
UC07	18	62	63	1970	0	0	0	<b>16457</b>

	Absolute Metrics				Weighted Metrics		
	Instances	Prec.	Rec.	F1Sc.	W-Prec.	W-Rec.	W-F1Sc.
UC01	39000 (19.45%)	94.02%	69.40%	79.86%	18.29%	13.50%	15.53%
UC02	30319 (15.12%)	42.46%	92.93%	58.29%	6.42%	14.05%	8.81%
UC03	17200 (8.58%)	41.39%	35.38%	38.15%	3.55%	3.04%	3.27%
UC04	17420 (8.69%)	95.37%	85.88%	90.37%	8.29%	7.46%	7.85%
UC05	39000 (19.45%)	99.92%	100%	99.96%	19.44%	19.45%	19.44%
UC06	39000 (19.45%)	99.91%	99.21%	99.56%	19.43%	19.30%	19.36%
UC07	18570 (9.26%)	88.62%	82.72%	85.57%	8.21%	7.66%	7.92%
					<b>83.62%</b>	<b>84.45%</b>	<b>82.20%</b>

In Table 18, we show that the GOOSE & SV++ features enable the J48 classifier to reach an weighted F1-Score of 87.22% as result of its multi-class analysis. Again, when taken separately, the hardest attack use case is the UC03. The F1-Score of UC03 was higher than the previous analysis but it is still low (47.27%). The confusion matrix shows that, with these enriched SV features, more masquerade attack samples are being classified into the correct class (8,618 in UC03) than the normal class (3,375 in UC00). However, there are still many instances classified into other classes such as UC01 and UC02, contributing to the low F1-Score of UC03. The F1-Score of the UC05 was approximately the same (0.2% lower) as the F1-Score provided by GOOSE & SV, but this still is the easiest attack detected by J48. Its F1-Score was 99.94%.

In Table 19, we show that the GOOSE++ & SV++ features enable the J48 classifier to reach a very high weighted F1-Score of 99.52% as result of its multi-class analysis. The hardest attack remains the same (*i.e.*, UC03), but its F1-Score was enhanced to 94.92%, which is much more acceptable than the previous assessments. The confusion matrix shows that most of the masquerade attack samples are being classified correctly but part of these attacks are confused with normal class (1,327 in UC00) and a few are classified as UC05. All other attacks have a 100% value (or a very close 99.77%) for their F1-Score – and similar behavior for the other metrics. These results reveal that, as expected, the

Table 18: The confusion matrix for J48 when using only the GOOSE &amp; SV++ features.

Exp. / Res.	UC00	UC01	UC02	UC03	UC04	UC05	UC06	UC07
<b>UC00</b>	<b>2751443</b>	321	0	3375	0	0	0	0
<b>UC01</b>	1193	<b>37146</b>	508	138	1	0	0	14
<b>UC02</b>	30	4920	<b>15907</b>	5504	0	0	77	3881
<b>UC03</b>	6919	1038	519	<b>8618</b>	22	0	0	84
<b>UC04</b>	39	0	0	0	<b>17381</b>	0	0	0
<b>UC05</b>	38	4	3	0	0	<b>38955</b>	0	0
<b>UC06</b>	4	26	5	4	0	0	<b>38960</b>	1
<b>UC07</b>	0	0	39	1623	0	0	0	<b>16908</b>

	Absolute Metrics				Weighted Metrics		
	Instances	Prec.	Rec.	F1Sc.	W-Prec.	W-Rec.	W-F1Sc.
<b>UC01</b>	39000 (19.45%)	95.25%	85.48%	90.10%	18.53%	16.63%	17.52%
<b>UC02</b>	30319 (15.12%)	52.47%	93.68%	67.26%	7.93%	14.16%	10.17%
<b>UC03</b>	17200 (8.58%)	50.10%	44.74%	47.27%	4.30%	3.84%	4.05%
<b>UC04</b>	17420 (8.69%)	99.78%	99.87%	99.82%	8.67%	8.68%	8.67%
<b>UC05</b>	39000 (19.45%)	99.88%	100%	99.94%	19.43%	19.45%	19.44%
<b>UC06</b>	39000 (19.45%)	99.90%	99.80%	99.85%	19.43%	19.41%	19.42%
<b>UC07</b>	18570 (9.26%)	91.05%	80.95%	85.70%	8.43%	7.50%	7.94%
					<b>86.72%</b>	<b>89.66%</b>	<b>87.22%</b>

proposed feature enhancement was able to provide a very significantly improvement on the overall IDS' performance.

## 6.4 Results Discussions and Lessons Learned

In this section, we summarize the answers for the key questions raised for *feature processing* (Section 6.1), *dataset features assessment* (Section 6.2), and *multic-class* (Section 6.3) experiments. The main lessons learned are the following:

- **Feature selection can improve the IDSs' detection performance:** our assessment shown how feature selection improved the performance of an IDS based on the J48 algorithm. In particular, the results show in Section 6.1 that using GRASP-FS as a feature selection method for reducing the basic GOOSE features to smaller subsets can improve the F1-Score metric for J48 detecting most attacks. Our detailed results show the following improvements for each attack class use case: UC01 (+4.92%), UC02 (+0.35%), UC03 (+1.16%), UC04 (+6.56%), and UC06 (+5.75%). The use cases UC05 and UC07 were not improved because they were easily detected by J48, which already had a 100% F1-Score.

Table 19: The confusion matrix for J48 when using all features (GOOSE++ &amp; SV++).

Exp. / Res.	UC00	UC01	UC02	UC03	UC04	UC05	UC06	UC07
UC00	<b>2753812</b>	0	0	1327	0	0	0	0
UC01	1	<b>38999</b>	0	0	0	0	0	0
UC02	0	0	<b>30319</b>	0	0	0	0	0
UC03	297	0	0	<b>16741</b>	0	162	0	0
UC04	0	0	0	0	<b>17420</b>	0	0	0
UC05	13	0	0	5	0	<b>38981</b>	1	0
UC06	0	0	1	0	0	1	<b>38998</b>	0
UC07	0	0	0	0	0	0	0	<b>18570</b>

	Absolute Metrics				Weighted Metrics		
	Instances	Prec.	Rec.	F1Sc.	W-Prec.	W-Rec.	W-F1Sc.
UC01	39000 (19.45%)	100%	100%	100%	19.45%	19.45%	19.45%
UC02	30319 (15.12%)	100%	100%	100%	15.12%	15.12%	15.12%
UC03	17200 (8.58%)	97.33%	92.63%	94.92%	8.35%	7.95%	8.14%
UC04	17420 (8.69%)	100%	100%	100%	8.69%	8.69%	8.69%
UC05	39000 (19.45%)	99.95%	99.58%	99.77%	19.44%	19.37%	19.41%
UC06	39000 (19.45%)	99.99%	100%	100%	19.45%	19.45%	19.45%
UC07	18570 (9.26%)	100%	100%	100%	9.26%	9.26%	9.26%
					<b>99.76%</b>	<b>99.29%</b>	<b>99.52%</b>

- **Feature enrichment can improve the IDSs' detection performance:** our assessment shown that using feature enrichment can significantly improve IDSs' performance. In particular, we assessed the feature enrichment through different combinations and correlations of GOOSE and SV features. Our detailed results show the following improvements for each attack class use case: UC01 (+11,85%), UC02 (+5.89%), UC03 (+47.22%), UC04 (33.95%), and UC06 (5.75%). In some cases, the feature enrichment process reached very high results (*e.g.*, 99.99% F1-Score for UC01). In contrast to the feature selection, the additional information provided by feature enrichment does not improve the UC06. This evidences that both feature enrichment and feature selection should be considered. As discussed, the use cases UC05 and UC07 already had a 100% F1-Score. The additional insight about them is that the enriched features do not negatively affect the detection performance.
- **The proposed features can describe multiple attacks classes:** the overall dataset features assessment reveal how well the proposed features are used for J48 detecting each attack use case. We show that J48 successfully classified UC01, UC02, UC03, and UC04 with F1-Scores above the 99%, and UC05 and UC07 with a 100% F1-Score. The lowest result was to UC06 (94.25%). In fact, the UC06

processing was too simple since it considered only one feature (*i.e.*, the `StNum` feature). Whereas this feature is expected to be very representative because this attack class is characterized by high *StNums*, more information is needed to avoid premature decisions.

- **Feature enrichment can improve the multi-class analysis:** the assessment of the whole dataset (*i.e.*, 7 attack classes and 1 normal class) that feature enrichment can consistently improve the IDSs' detection performance. The only combination of GOOSE and SV features resulted in an improvement from 72.84% to 82.20% on the F1-Score. When enriching the SV features, the F1-Score went up to 87.22%. Finally, with the enriched GOOSE and SV features, we reached a 99.52% F1-Score.
- **Multiple attacks impact the detection of each other:** the confusion matrices analysis reveals that when multiple attack classes are simultaneously considered for training and testing an IDS, some attack instances are classified as belonging to other attack classes – especially when the features are not much representative. Even though these cases are less harmful than confusing an attack to the normal class, it still results in more false positives and false negatives. The basic features were the most affected by false alarms (*e.g.*, 24.54% of the UC02 samples were classified as UC01). On the other hand, the feature enrichment can overcome this issue, as it resulted in a very low number of false alarms (*e.g.*, the average precision and recall are above the 99%).

# 7 Conclusion

The integration of communication technologies with the traditional power grid system and the deployment of novel communication protocols exposes digital substation networks based on the IEC–61850 standard to various threats and security challenges to digital substations.

In this thesis, we performed an in-depth survey on the field of IDSs for IEC–61850-based substations and mapped their multiple aspects to understand (i) the range of attacks covered by the state-of-art IDSs, (ii) the available data source for IDSs analysis, (iii) the evaluation methods employed, (iv) the metrics adopted, as well as the IDSs (v) architectures, (vi) approaches, (vii) type of analysis, and (viii) response actions. As result, we found a critical issue on building robust IDSs: the lack of data available for training, testing, and assessing them.

With focus on the lack of available datasets for intrusion detection in communication systems and networks in the context of electrical substations we proposed the ERENO traffic generation tool. As an additional contribution, we made it available both the ERENO source code and a baseline dataset composed of 7 state-of-art attack classes and 1 legitimate traffic class which are represented through 69 features. These features include both basic SV and GOOSE features and enriched features that provides additional and more representative information for IDSs from these protocols.

The dataset generated by ERENO is realistic because it is based on a real substation modeled with the PSCAD simulation tool and on state-of-art attack classes modeled as use cases. The behavior for all classes are designed considering both electrical faults and normal scenarios.

Furthermore, we proposed a novel implementation of the GRASP metaheuristic, named GRASP-FS, to perform feature selection on the generated dataset.

Our results reveal that the proposed tool can successfully generate datasets for training, testing, and evaluating IDSs based on the IEC–61850 network communication pro-



protocols. The proposed feature enrichment provided novel features for GOOSE and SV protocols that demonstrated to be efficient. The full set composed by 69 features improved the J48 classifier algorithm precision, recall, accuracy, and F1-Score up to 19.47%, 12.21%, 2.28%, and 16.02%, respectively.

## 7.1 Contributions

Our main contributions are summarized as follows:

- We introduced a novel methodology for simulating both normal and faulty scenarios on power grids through the PSCAD (LTD, 2019) tool, reproducing a real transmission line;
- We presented a study of the current attack scenarios targeting IEC-61850 systems;
- We proposed the ERENO tool to generate realistic GOOSE and SV traffic features, taking as input the modeled real scenario in PSCAD;
- We identified and extracted the features of the electrical and computer networks domains that are correlated with malicious actions. We also composed novel enhanced features;
- We implemented 8 use cases on the ERENO tool to generate 7 attack classes and one class of benign normal traffic;
- We made the ERENO-IEC-61850 dataset, which was generated by the ERENO tool, available publicly<sup>1</sup>;
- We proposed the GRASP-FS, a novel implementation of the GRASP metaheuristic for the feature selection.

## 7.2 Publications

Along the development of this thesis, we documented each step of our study through academic articles published in international journals and national/international conferences. Part of our research had a direct contribution to this thesis (see Section 7.2.1). Another part is composed by productions which do not compose this thesis (*i.e.*, that are beyond

---

<sup>1</sup><https://github.com/sequincozes/erenno>

the scope), but study correlated themes. Therefore, they are also mentioned here because they are indirect products of the doctoral research (see Section 7.2.2).

### 7.2.1 Production of Direct Results

1. QUINCOZES, S. E., PINHEIRO, J. L., PASSOS, D., DE ALBUQUERQUE, C. V. N., MOSSE, D. ERENO: A Framework for Generating Realistic IEC-61850 Intrusion Detection Datasets. Planned target: **Transactions on Dependable and Secure Computing**, expected to 2022 (in progress).
2. **Production under Review**: QUINCOZES, S. E., PASSOS, D., DE ALBUQUERQUE, C. V. N., MOSSE, D. An Extended Assessment of Metaheuristics-based Feature Selection for Intrusion Detection in CPS Perception Layer. Submitted target: **Annals of Telecommunications**, expected to 2022 (under review).
3. **Published Production** (QUINCOZES; ALBUQUERQUE, et al., 2021): QUINCOZES, S. E., PASSOS, D., DE ALBUQUERQUE, C. V. N., MOSSE, D. A survey on intrusion detection and prevention systems in digital substations. **Computer Networks**, v. 184, p. 107679, 2021.
4. **Published Production** (QUINCOZES; MOSSE, et al., 2021): QUINCOZES, S. E., PASSOS, D., DE ALBUQUERQUE, C. V. N., MOSSE, D., OCHI, L. S., SANTOS, V. F. On the Performance of GRASP-Based Feature Selection for CPS Intrusion Detection. **IEEE Transactions on Network and Service Management**, 2021.
5. **Published Production** (QUINCOZES; PASSOS, et al., 2020): QUINCOZES, S. E., PASSOS, D., DE ALBUQUERQUE, C. V. N., MOSSE, D., OCHI, L. S. GRASP-based Feature Selection for Intrusion Detection in CPS Perception Layer. **In: 2020 4th Conference on Cloud and Internet of Things (CIoT)**. IEEE, 2020. p. 41-48.

### 7.2.2 Production on Correlated Themes

6. DESLFINO, W. O., QUINCOZES, S. E., VIEIRA, J. L., PASSOS, D. P., SAADE, M. D., ALBUQUERQUE, C.V.N., Fault Recovery on Software Defined Network. Planned target: **IEEE Access**, expected to 2022 (in progress).

7. QUINCOZES, S. E., QUINCOZES, V.E., KAZIENKO, J. F. An Extended Evaluation on Machine Learning Techniques for Denial-of-Service Detection in Wireless Sensor Networks. Planned target: **IEEE Internet of Things Journal**, expected to 2022 (in progress).
8. QUINCOZES, V. E., QUINCOZES, S. E., MANSILHA, R., KREUTZ, D., KAZIENKO, J. F. Fault A Mobile Application for Scheduling Health Services on Demand. Submitted target: **Simpósio Brasileiro de Sistemas de Informação**, expected to 2022 (under review).
9. (QUINCOZES; RANIERY, et al., 2021): QUINCOZES, S. E., RANIERY, C., CERETTA, R., PASSOS, D., DE ALBUQUERQUE, C., MOSSE, D., Counselors network for intrusion detection. **International Journal of Network Management (IJNM)**, v. 31, n. 3, p. e2111, 2021.
10. (SOARES et al., 2021): ZOPELLARO, A. A., SOARES, L., MATTOS, D. P., PINHEIRO, P., QUINCOZES, S. E, ..., PASSOS, D., DE ALBUQUERQUE, C. V. N., et al. Enabling Emulation and Evaluation of IEC 61850 Networks With TITAN. **IEEE Access**, v. 9, p. 49788-49805, 2021.
11. (ZOPELLARO SOARES et al., 2021): ZOPELLARO, A. A., VIEIRA, J. L., QUINCOZES, S. E., ..., PASSOS, D., DE ALBUQUERQUE, C. V. N., et al. SDN-based teleprotection and control power systems: A study of available controllers and their suitability. **International Journal of Network Management (IJNM)**, v. 31, n. 3, p. e2112, 2021.
12. (VIEIRA et al., 2021): VIEIRA, J. L., FERREIRA, V. C., BASTOS, I. V., QUINCOZES, S. E., ..., PASSOS, D., DE ALBUQUERQUE, C. V. N., et al. THANOS: Teleprotection Holistic Application for ONOS Controller. In: **2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)**. IEEE, 2021. p. 818-823.
13. (QUINCOZES; QUINCOZES, et al., 2021): QUINCOZES, V. E., QUINCOZES, S. E., et al. Identifica ISP: Autenticação Mútua entre Múltiplas Entidades para Serviços de Suporte Técnico Prestados por ISPs. In: **Anais do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas (SBSeg)**. Sociedade Brasileira de Computação (SBC), 2021. p. 26-33.
14. (QUINCOZES, V E; QUINCOZES; KAZIENKO, 2021): QUINCOZES, V. E., QUINCOZES, S. E., KAZIENKO, J. F. **Livro de Minicursos da VII Escola**

- Regional de Sistemas de Informação (ERSI-RJ)**, Capítulo 7. 1ed. Porto Alegre: Sociedade Brasileira de Computação (SBC), 2021, p. 250-284.
15. (QUINCOZES, Vagner E; QUINCOZES; KAZIENKO, 2021): QUINCOZES, V. E., QUINCOZES, S. E., KAZIENKO, J. F. Avaliando a Sobrecarga de Mecanismos Criptográficos Simétricos na Internet das Coisas: Uma Comparação Quantitativa entre os Protocolos MQTT e CoAP. **In: Anais do XX Workshop em Desempenho de Sistemas Computacionais e de Comunicação**. Sociedade Brasileira de Computação (SBC), 2021. p. 13-24.
  16. (UCHÔA et al., 2020): UCHÔA, L., QUINCOZES, S. E., VIEIRA, J. L., PASSOS, D., DE ALBUQUERQUE, C. V. N., MOSSE, D. Analysis of smart grid fault recovery protocols. **In: IEEE/IFIP Network Operations and Management Symposium (NOMS)**. IEEE, 2020. p. 1-8.
  17. (BORGIANI et al., 2020): BORGIANI, V. M., MORATORI, P., KAZIENKO, J. F., TUBINO, E. E., QUINCOZES, S. E. Toward a Distributed Approach for Detection and Mitigation of Denial-of-Service Attacks Within Industrial Internet of Things. **IEEE Internet of Things Journal**, v. 8, n. 6, p. 4569-4578, 2020.
  18. (QUINCOZES; KAZIENKO, 2020): QUINCOZES, S. E., KAZIENKO, J. F. Machine Learning Methods Assessment for Denial of Service Detection in Wireless Sensor Networks. **In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT)**. IEEE, 2020. p. 1-6.
  19. (KREUTZ et al., 2020): KREUTZ, D. L., MANSILHA, R.B., QUINCOZES, S. E., et al. Introdução a verificação automática de protocolos de segurança com scyther. **Minicursos da XVIII Escola Regional de Redes de Computadores**, Sociedade Brasileira de Computação (SBC), 2020.
  20. (QUINCOZES; TEMP, et al., 2020): QUINCOZES, V. E., Temp, D., QUINCOZES, S.E., et al. Sistema para Autenticação entre Clientes, Técnicos e ISPs. **In: Anais da XVIII Escola Regional de Redes de Computadores**. Sociedade Brasileira de Computação (SBC), 2020. p. 116-122.
  21. (QUINCOZES; SOARES, et al., 2019): QUINCOZES, S. E., ..., PASSOS, D., DE ALBUQUERQUE, C. V. N., et al. Survey and Comparison of SDN Controllers for Teleprotection and Control Power Systems. **In: Latin American Network Operations and Management Symposium (LANOMS)**. 2019.

22. ([QUINCOZES; SANTOS, et al., 2019](#)): QUINCOZES, S. E., RANIERY, C., CERETTA, R., PASSOS, D., DE ALBUQUERQUE, C., MOSSE, D. A Counselors-Based Intrusion Detection Architecture. **In: Latin American Network Operations and Management Symposium (LANOMS)**. 2019.
23. ([QUINCOZES; EMILIO; KAZIENKO, 2019](#)): QUINCOZES, S. E., TUBINO, E. E.; KAZIENKO, J. F., Mqtt protocol: Fundamentals, tools and future directions. **IEEE Latin America Transactions**, v. 17, n. 09, p. 1439-1448, 2019.
24. ([QUINCOZES; KAZIENKO, 2019](#)): QUINCOZES, S. E., KAZIENKO, J. F. Experimental evaluation of a secure and ubiquitous architecture for electronic health records retrieval. **International Journal of E-Health and Medical Communications (IJEHMC)**, v. 10, n. 4, p. 39-53, 2019.
25. ([JUNIOR; QUINCOZES; KAZIENKO, 2019](#)): JUNIOR, C. R., QUINCOZES, S. E., KAZIENKO, J. F., LegitimateBroker: Mitigando Ataques de Personificação em Broker MQTT na Internet das Coisas. **In: Anais do XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais**. SBC, 2019. p. 141-154.
26. ([QUINCOZES; KAZIENKO; COPETTI, 2018](#)): QUINCOZES, S. E., et al. Avaliação de Conjuntos de Atributos para a Detecção de Ataques de Personificação na Internet das Coisas. **In: Anais Estendidos do VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais**. SBC, 2018.

### 7.3 Open Issues and Future Works

The ERENO tool is modular and open-source, it is easy to use as a basis for future researches on IDS. In future works, we plan to extend the ERENO with more use cases to cover novel attacks. Besides, we plan to deploy real-time processing techniques to process more complex traffic generated by ERENO.

Although in the last years a few studies addressed IDSs in the context of Smart Grid, research on intrusion detection in digital substations is still at an early stage. Thus, several research topics remain open:

- *More general IDSs*: current IEC-61850-based IDSs rely on expert knowledge about the substation components, the standard, and its protocols. In particular, specification-based IDSs have limited attack detection capabilities ([HONG; LIU, 2019](#);

HONG; LIU, C.; GOVINDARASU, 2014; HONG; LIU, C.-C.; GOVINDARASU, 2014; YANG; XU, et al., 2016). As shown in Section 4.3, many detection rules fail to detect all or part of attacks due to their high specialization. With ERENO-based datasets, it will be possible to develop more robust IDS proposals by using novel machine learning algorithms;

- *Add Preventive Measures to IDSs*: traditional IDSs are focused on **detecting** malicious behavior. Accordingly, the intrusions attempts are logged or a warning is issued. However, due to the critical role of substation networks, it is important to replace them with IDPSs, which may take proper actions to **prevent** the attack instead of just detecting the intruders' behavior. Clearly, issues of cost, timeliness, performance, and overhead must come into play as well;
- *Big Data Issues*: improving the accuracy from current specification-based IDSs may require combining multiple and heterogeneous data sources (*e.g.*, SCADA-level logs, GOOSE commands, SV measures, MMS reports). On the other hand, the IDSs' processing time should be low enough to detect intrusions timely. Such data volume, variety, and velocity characterize a Big Data challenge (DAI, HONG-NING AND WONG, RAYMOND CHI-WING AND WANG, HAO AND ZHENG, ZIBIN AND VASILAKOS, ATHANASIOS V, 2019), even though it is at the electric digital substation scale.

Based on the aforementioned open issues, we point out some potential future directions. These research topics are based on novel approaches that are still not well explored into substation networks and may be useful to address most of the existing issues on detecting and preventing intrusions in IEC-61850 substations.

- *Smart IDS*: an IDS based on more sophisticated techniques, such as machine learning, analyzing multiple sources of information may be promising for dealing with the most challenging scenarios in which anomaly and specification-based IDSs are ineffective (*e.g.*, for detecting masquerade attacks). El Mrabet *et al.* (EL MRABET *et al.*, 2019) adopted a deep learning architecture to automatically extract features and make a predictive classification in other Smart Grid environments (*i.e.*, AMI). Applying it to substation networks may also yield good results, but that approach has not been explored;
- *Proactive Blocking*: a promising approach to analyze the network traffic looking for malicious patterns in a timely manner consists of using additional hardware

between the devices. This idea was previously introduced by Kim and Park ([KIM; PARK, 2018](#)). They proposed an FPGA-based IDS to process IEC-61850 packets and detect intrusions by rule matching. This idea has the potential of blocking malicious traffic before it arrives at the target device;

- *SDN and IDSs*: Researchers have proposed building IDSs based on Software Defined Network (SDN) to enable general proactive flow blocking and forwarding suspicious traffic to IDSs ([HA et al., 2016](#)). Since SDN enables the flow forwarding through software applications, an implementation of this approach in digital substation networks may be interesting to handle suspicious traffic and blocking messages when they are detected as malicious;
- *Real-time IDS*: initial efforts of building a real-time IDS for Smart Grids were carried out by M. Faisal *et al.* ([FAISAL et al., 2014](#)). They used the MOA Framework to process streaming data and detect malicious traffic. However, they did not consider substation networks, only addressing AMI communication. Employing similar techniques in IEC-61850 networks may be a promising research direction.

# REFERENCES

- ADEPU, Sridhar; KANDASAMY, Nandha Kumar; MATHUR, Aditya. Epic: An electric power testbed for research and training in cyber physical systems security. In: *COMPUTER Security*. [S.l.]: Springer, 2018. p. 37–52.
- AGRAWAL, Prachi et al. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019). *IEEE Access*, IEEE, v. 9, p. 26766–26791, 2021.
- AHMED, Arman et al. Cyber physical security analytics for anomalies in transmission protection systems. *IEEE Transactions on Industry Applications*, IEEE, v. 55, n. 6, p. 6313–6323, 2019.
- AMINANTO, Muhamad Erza et al. Weighted feature selection techniques for detecting impersonation attack in Wi-Fi networks. In: *PROC. Symp. Cryptogr. Inf. Secur.(SCIS)*. [S.l.: s.n.], 2017. p. 1–8.
- BELLOVIN, S. **RFC1948: Defending against sequence number attacks**. [S.l.]: RFC Editor, 1996.
- BERMEJO, Pablo; GAMEZ, Jose A; PUERTA, Jose M. A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets. *Pattern Recognition Letters*, Elsevier, v. 32, n. 5, p. 701–711, 2011.
- BIFET, Albert et al. MOA: Massive misc analysis. *Journal of Machine Learning Research*, v. 11, May, p. 1601–1604, 2010.
- BISWAS, Partha P et al. A synthesized dataset for cybersecurity study of IEC 61850 based substation. In: *IEEE. 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. [S.l.: s.n.], 2019. p. 1–7.
- BORGIANI, Vladimir et al. Toward a Distributed Approach for Detection and Mitigation of Denial-of-Service Attacks Within Industrial Internet of Things. *IEEE Internet of Things Journal*, IEEE, v. 8, n. 6, p. 4569–4578, 2020.



- BOSTANI, Hamid; SHEIKHAN, Mansour. Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. **Computer Communications**, Elsevier, v. 98, p. 52–71, 2017.
- C, IEC. **IEC 62351 Security**. [S.l.]: IET, 2010.
- CHANDRASHEKAR, Girish; SAHIN, Ferat. A survey on feature selection methods. **Computers & Electrical Engineering**, Elsevier, v. 40, n. 1, p. 16–28, 2014.
- COMMISSION, International Electrotechnical. **Communication networks and systems in substations - ALL PARTS**. [S.l.]: IET, 2019.
- COSTA, Nattane Luiza da; LIMA, Marcio Dias de; BARBOSA, Rommel. Evaluation of feature selection methods based on artificial neural network weights. **Expert Systems with Applications**, Elsevier, v. 168, p. 114312, 2021.
- CUPELLI, Marco; CARDET, C Doig; MONTI, Antonello. Voltage stability indices comparison on the IEEE-39 bus system using RTDS. In: IEEE. 2012 IEEE International Conference on Power System Technology (POWERCON). [S.l.: s.n.], 2012. p. 1–6.
- DAI, HONG-NING AND WONG, RAYMOND CHI-WING AND WANG, HAO AND ZHENG, ZIBIN AND VASILAKOS, ATHANASIOS V. Big data analytics for large-scale wireless networks: Challenges and opportunities. **ACM Computing Surveys**, ACM New York, NY, USA, v. 52, n. 5, p. 1–36, 2019.
- DIEZ-PASTOR, Jose F; GARCIA-OSORIO, Cesar, et al. GRASP Forest: a new ensemble method for trees. In: SPRINGER. INTERNATIONAL Workshop on Multiple Classifier Systems. [S.l.: s.n.], 2011. p. 66–75.
- DIEZ-PASTOR, Jose-Francisco; GARCIA-OSORIO, Cesar; RODRIGUEZ, Juan J. Tree ensemble construction using a GRASP-based heuristic and annealed randomness. **Information Fusion**, Elsevier, v. 20, p. 189–202, 2014.
- EL HARIRI, Mohamad et al. misc false data detection and lost packet forecasting system using time series neural networks for IEC 61850 sampled measured values. In: IEEE. POWER & Energy Society Innovative Smart Grid Technologies Conference (ISGT). [S.l.: s.n.], 2017. p. 1–5.
- EL MRABET, Zakaria et al. Deep Learning-Based Intrusion Detection System for Advanced Metering Infrastructure. In: PROCEEDINGS of the 2nd International Conference on Networking, Information Systems & Security. [S.l.: s.n.], 2019. p. 1–7.

- ELGARGOURI, Ahmed; ELMUSRATI, Mohammed. Analysis of Cyber-Attacks on IEC 61850 Networks. In: IEEE. 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT). [S.l.: s.n.], 2017. p. 1–4.
- ESSEGHIR, Mohamed Amir. Effective wrapper-filter hybridization through GRASP schemata. In: FEATURE Selection in Data Mining. [S.l.: s.n.], 2010. p. 45–54.
- FAISAL, Mustafa Amir et al. Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: A feasibility study. **IEEE Systems journal**, IEEE, v. 9, n. 1, p. 31–44, 2014.
- FERNANDES, Chilton; BORKAR, Samarth; GOHIL, Jignesh. Testing of goose protocol of IEC61850 standard in protection IED. **International journal of computer applications**, Foundation of Computer Science, v. 93, n. 16, 2014.
- GANAPATHY, Sannasi et al. Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. **EURASIP Journal on Wireless Communications and Networking**, Springer, v. 2013, n. 1, p. 271, 2013.
- GONT, F; BELLOVIN, SM. **RFC6528: Defending against Sequence Number Attacks**. [S.l.]: RFC Editor, 2012.
- HA, Taejin et al. Suspicious flow forwarding for multiple intrusion detection systems on software-defined networks. **IEEE Network**, IEEE, v. 30, n. 6, p. 22–27, 2016.
- HADELI, Hadeli et al. Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. In: IEEE. 2009 IEEE Conference on Emerging Technologies & Factory Automation. [S.l.: s.n.], 2009. p. 1–8.
- HADELI et al. Generating configuration for missing traffic detector and security measures in industrial control systems based on the system description files. In: IEEE. IEEE Conference on Technologies for Homeland Security. [S.l.: s.n.], 2009. p. 503–510.
- HAHN, Adam; SUN, Chih-Che; LIU, Chen-Ching. Cybersecurity of SCADA within Substations. **Smart Grid Handbook**, Wiley misc Library, p. 1–17, 2016.
- HALL, Mark Andrew. **Correlation-based feature selection for machine learning**. 1999. PhD thesis – The University of Waikato.
- HANSEN, Pierre; MLADENOVIC, Nenad. An introduction to variable neighborhood search. In: META-HEURISTICS. [S.l.]: Springer, 1999. p. 433–458.
- HONG, J.; LIU, C. Intelligent Electronic Devices With Collaborative Intrusion Detection Systems. **IEEE Transactions on Smart Grid**, v. 10, n. 1, p. 271–281, 2019.

- HONG, Junho; LIU, Chen-Ching; GOVINDARASU, Manimaran. Integrated anomaly detection for cyber security of the substations. **IEEE Transactions on Smart Grid**, IEEE, v. 5, n. 4, p. 1643–1653, 2014.
- HONG, Junho; LIU, ChenChing; GOVINDARASU, Manimaran. Detection of cyber intrusions using network-based multicast messages for substation automation. In: IEEE. INNOVATIVE Smart Grid Technologies (ISGT). [S.l.: s.n.], 2014. p. 1–5.
- HOYOS, Juan; DEHUS, Mark; BROWN, Timothy X. Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure. In: IEEE. 2012 IEEE Globecom Workshops. [S.l.: s.n.], 2012. p. 1508–1513.
- IEC. **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**. [S.l.]: IET, 2004.
- IEC, International Electrotechnical 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**. [S.l.]: IET, 2003.
- JUNIOR, Charles Rampelotto; QUINCOZES, Silvio; KAZIENKO, Juliano. LegitimateBroker: Mitigando Ataques de Personificação em Broker MQTT na Internet das Coisas. In: SBC. ANAIS do XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. [S.l.: s.n.], 2019. p. 141–154.
- KABIR-QUERREC, Maelle et al. Power Utility Automation Cybersecurity: IEC 61850 Specification of an Intrusion Detection Function. In: CRC PRESS. 25TH European Safety and Reliability Conference (ESREL 2015). [S.l.: s.n.], 2015.
- KALEEM, Danish; FERENS, Ken. A cognitive approach for attribute selection in internet dataset. In: IEEE. COGNITIVE Informatics & Cognitive Computing (ICCI CC), IEEE 16th International Conference on. [S.l.: s.n.], 2017. p. 319–328.
- KANAKARAJAN, Navaneeth Kumar; MUNIASAMY, Kandasamy. Improving the accuracy of intrusion detection using GAR-Forest with feature selection. In: SPRINGER. PROCEEDINGS of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA). [S.l.: s.n.], 2016. p. 539–547.
- KANG, BooJoong; MAYNARD, Peter, et al. Investigating cyber-physical attacks against IEC 61850 photovoltaic inverter installations. In: IEEE. 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). [S.l.: s.n.], 2015. p. 1–8.

- KANG, BooJoong; MCLAUGHLIN, Kieran; SEZER, Sakir. Towards A Stateful Analysis Framework for Smart Grid Network Intrusion Detection. In: PROCEEDINGS of the 4th International Symposium for ICS & SCADA Cyber Security Research. [S.l.: s.n.], 2016. p. 124–131.
- KARIYAWASAM, Sachintha; RAJAPAKSE, Athula D; PERERA, Nuwan. Investigation of using IEC 61850-sampled values for implementing a transient-based protection scheme for series-compensated transmission lines. **IEEE Transactions on Power Delivery**, IEEE, v. 33, n. 1, p. 93–101, 2017.
- KIM, Junsik; PARK, Jaehyun. FPGA-based network intrusion detection for IEC 61850-based industrial network. **ICT Express**, Elsevier, v. 4, n. 1, p. 1–5, 2018.
- KIM, S.; JO, W.; SHON, T. A novel vulnerability analysis approach to generate fuzzing test case in industrial control systems. In: IEEE. INFORMATION Technology, Networking, Electronic and Automation Control Conference. [S.l.: s.n.], 2016. p. 566–570.
- KREUTZ, Diego Luis et al. Introdução a verificação automática de protocolos de segurança com scyther. In: MINICURSOS da XVIII Escola Regional de Redes de Computadores (ERRC). [S.l.]: Sociedade Brasileira de Computação, 2020. chap. 3, p. 43–68.
- KUSH, Nishchal et al. Poisoned GOOSE: exploiting the GOOSE protocol. In: AUSTRALIAN COMPUTER SOCIETY, INC. PROCEEDINGS of the Twelfth Australasian Information Security Conference (AISC 2014). [S.l.: s.n.], 2014. p. 17–22.
- KWON, YooJin et al. A behavior-based intrusion detection technique for smart grid infrastructure. In: IEEE. 2015 IEEE Eindhoven PowerTech. [S.l.: s.n.], 2015. p. 1–6.
- LANCASTER, Henry Oliver; SENETA, Eugene. **Chi-square distribution**. [S.l.]: Wiley Online Library, 1969.
- LI, Jundong et al. Feature selection: A data perspective. **ACM Computing Surveys (CSUR)**, ACM, v. 50, n. 6, p. 94, 2017.
- LOUNGE, Geek. **Capture files from 4sics geek lounge**. [S.l.: s.n.], 2019. Visited on: 17 Aug. 2021.
- LTD, Manitoba Hydro International. **The World’s Most Advanced Tool for Power Systems EMT Simulations**. [S.l.: s.n.], 2019. Available at <https://www.pscad.com/software/pscad/overview>. Visited on: 20 Aug. 2021.

- MACKIEWICZ, Ralph E. Overview of IEC 61850 and Benefits. In: IEEE. 2006 IEEE Power Engineering Society General Meeting. [S.l.: s.n.], 2006. 8–pp.
- MACWAN, Richard et al. Collaborative defense against data injection attack in IEC61850 based smart substations. In: IEEE. 2016 IEEE Power and Energy Society General Meeting (PESGM). [S.l.: s.n.], 2016. p. 1–5.
- MORADKHANI, Mostafa et al. A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm. **Applied Soft Comp.**, Elsevier, v. 35, p. 123, 2015.
- MORRIS, Robert Tappan. A weakness in the 4.2 BSD Unix TCP/IP software. **AT&T Bell Labs, Tech. Rep. Comput. Sci.**, v. 117, 1985.
- MOSHKI, Mohsen; KABIRI, Peyman; MOHEBALHOJEH, Alireza. Scalable feature selection in high-dimensional data based on GRASP. **Applied Artificial Intelligence**, Taylor & Francis, v. 29, n. 3, p. 283–296, 2015.
- MOUSSA, Bassam; DEBBABI, Mourad; ASSI, Chadi. A detection and mitigation model for PTP delay attack in an IEC 61850 substation. **IEEE Transactions on Smart Grid**, IEEE, v. 9, n. 5, p. 3954–3965, 2016.
- MOUSTAFA, Nour; SLAY, Jill. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: IEEE. 2015 military communications and information systems conference (MilCIS). [S.l.: s.n.], 2015. p. 1–6.
- NIST. **National Institute of Standards and Technology: National Vulnerability Database (NVD)**. [S.l.]: NIST, 2021. Available in: <https://nvd.nist.gov/vuln/>.
- NIXON, Christopher; SEDKY, Mohamed; HASSAN, Mohamed. Practical Application of Machine Learning based misc Intrusion Detection to Internet of Things Networks. In: IEEE. 2019 IEEE Global Conference on Internet of Things (GCIoT). [S.l.: s.n.], 2019. p. 1–5.
- NOCE, Julia et al. Identifying vulnerabilities in smart grid communication networks of electrical substations using geese 2.0. In: IEEE. 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). [S.l.: s.n.], 2017. p. 111–116.
- O’RAW, John; LAVERTY, David M; MORROW, D John. IEC 61850 substation configuration language as a basis for automated security and SDN configuration. In: IEEE. POWER & ENERGY SOCIETY GENERAL MEETING. [S.l.: s.n.], 2017. p. 1–5.

- OMAKAZI. **Voltage Imbalance (Unbalance) Study**. [S.l.: s.n.], 2022. <https://www.omazaki.co.id/en/voltage-imbalance-unbalance-study/>. [Online; accessed 15-January-2021].
- OPERADOR NACIONAL DO SISTEMA ELÉTRICO, ONS. **Submódulo 2.11 Requisitos mínimos para os sistemas de proteção, de registro de perturbações e de teleproteção**. [S.l.: s.n.], 2021. Available in [http://apps08.ons.org.br/ONS.Sintegre.Proxy/ecmprsite/ecmfragmentsdocuments/Subm%C3%B3dulo%202.11-RQ\\_2020.12.pdf](http://apps08.ons.org.br/ONS.Sintegre.Proxy/ecmprsite/ecmfragmentsdocuments/Subm%C3%B3dulo%202.11-RQ_2020.12.pdf), accessed in 2021.
- PATEL, Shivam. **IEC-61850 Protocol Analysis and misc Intrusion Detection System for SCADA Networks using Machine Learning**. 2017. PhD thesis – University of Victoria.
- PENNA, Puca Huachi Vaz; SUBRAMANIAN, Anand; SATORU, Luiz Satoru. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. **Journal of Heuristics**, Springer, v. 19, n. 2, p. 201–232, 2013.
- PINHEIRO, Paulo Henrique et al. Detailed modelling and analysis of digital mho distance relay with single-pole operation. **Acta Polytechnica**, v. 61, p. 537–551, Aug. 2021.
- POPOVIC, Miroslav et al. iPRP—The parallel redundancy protocol for IP networks: Protocol design and operation. **IEEE Transactions on Industrial Informatics**, IEEE, v. 12, n. 5, p. 1842–1854, 2016.
- PREMARATNE, Upeka Kanchana et al. An intrusion detection system for IEC61850 automated substations. **IEEE Transactions on Power Delivery**, IEEE, v. 25, n. 4, p. 2376–2383, 2010.
- QUINCOZES, Silvio; EMILIO, Tubino; KAZIENKO, Juliano. Mqtt protocol: Fundamentals, tools and future directions. **IEEE Latin America Transactions**, IEEE, v. 17, n. 09, p. 1439–1448, 2019.
- QUINCOZES, Silvio E; ALBUQUERQUE, Celio, et al. A survey on intrusion detection and prevention systems in digital substations. **Computer Networks**, Elsevier, v. 184, p. 107679, 2021.
- QUINCOZES, Silvio E; KAZIENKO, Juliano F. Experimental evaluation of a secure and ubiquitous architecture for electronic health records retrieval. **International Journal of E-Health and Medical Communications (IJEHMC)**, IGI Global, v. 10, n. 4, p. 39–53, 2019.

- QUINCOZES, Silvio E; KAZIENKO, Juliano F. Machine Learning Methods Assessment for Denial of Service Detection in Wireless Sensor Networks. In: IEEE. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). [S.l.: s.n.], 2020. p. 1–6.
- QUINCOZES, Silvio E; KAZIENKO, Juliano F; COPETTI, Alessandro. Avaliação de Conjuntos de Atributos para a Detecção de Ataques de Personificação na Internet das Coisas. In: SBC. ANAIS Estendidos do VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais. [S.l.: s.n.], 2018.
- QUINCOZES, Silvio E; MOSSE, Daniel, et al. On the Performance of GRASP-Based Feature Selection for CPS Intrusion Detection. **IEEE Transactions on Network and Service Management**, IEEE, 2021.
- QUINCOZES, Silvio E; PASSOS, Diego, et al. GRASP-based Feature Selection for Intrusion Detection in CPS Perception Layer. In: IEEE. 2020 4th Conference on Cloud and Internet of Things (CIoT). [S.l.: s.n.], 2020. p. 41–48.
- QUINCOZES, Silvio E; RANIERY, Carlos, et al. Counselors network for intrusion detection. **International Journal of Network Management**, Wiley Online Library, v. 31, n. 3, e2111, 2021.
- QUINCOZES, Silvio E; SANTOS, Carlos Raniery Paula dos, et al. A Counselors-Based Intrusion Detection Architecture. In: LATIN American Network Operations and Management Symposium (LANOMS). [S.l.: s.n.], 2019.
- QUINCOZES, Silvio E; SOARES, Arthur Albuquerque Zopellaro, et al. Survey and Comparison of SDN Controllers for Teleprotection and Control Power Systems. In: LATIN American Network Operations and Management Symposium (LANOMS). [S.l.: s.n.], 2019.
- QUINCOZES, V E; QUINCOZES, Silvio E; KAZIENKO, Juliano F. Desvendando a Camada de Aplicação na Internet das Coisas: Teoria, Prática e Tendências. In: LIVRO de Minicursos da VII Escola Regional de Sistemas de Informação (ERSI-RJ). [S.l.]: Sociedade Brasileira de Computação, 2021. chap. 7, p. 250–284.
- QUINCOZES, Vagner E; QUINCOZES, Silvio E; KAZIENKO, Juliano F. Avaliando a Sobrecarga de Mecanismos Criptográficos Simétricos na Internet das Coisas: Uma Comparação Quantitativa entre os Protocolos MQTT e CoAP. In: SBC. ANAIS do XX Workshop em Desempenho de Sistemas Computacionais e de Comunicação. [S.l.: s.n.], 2021. p. 13–24.

- QUINCOZES, Vagner E; QUINCOZES, Silvio E, et al. Identifica ISP: Autenticação Mútua entre Múltiplas Entidades para Serviços de Suporte Técnico Prestados por ISPs. In: SBC. ANAIS do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. [S.l.: s.n.], 2021. p. 26–33.
- QUINCOZES, Vagner E; TEMP, Daniel, et al. Sistema para Autenticação entre Clientes, Técnicos e ISPs. In: SBC. ANAIS da XVIII Escola Regional de Redes de Computadores. [S.l.: s.n.], 2020. p. 116–122.
- RADOGLOU GRAMMATIKIS, P. I.; SARIGIANNIDIS, P. G. Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems. **IEEE Access**, p. 46595–46620, 2019.
- RADOGLOU GRAMMATIKIS, Panagiotis I; SARIGIANNIDIS, Panagiotis G. Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems. **IEEE Access**, IEEE, v. 7, p. 46595–46620, 2019.
- RASHID, Muhammad Talha Abdul et al. A review of security attacks on IEC61850 substation automation system network. In: IEEE. PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY AND MULTIMEDIA. [S.l.: s.n.], 2014. p. 5–10.
- SANS, Institute. **DRAGON – An Intrusion Detection System**. [S.l.], 2002.
- SATOH, Hayato et al. Root-Mean Square Model of Three-Phase Photovoltaic Inverter for Unbalanced Fault. **IEEE Open Access Journal of Power and Energy**, IEEE, v. 7, p. 501–513, 2020.
- SHARAFALDIN, Iman; LASHKARI, Arash Habibi; GHORBANI, Ali A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. **ICISSp**, v. 1, p. 108–116, 2018.
- SOARES, Arthur Albuquerque Zopellaro et al. Enabling Emulation and Evaluation of IEC 61850 Networks With TITAN. **IEEE Access**, IEEE, v. 9, p. 49788–49805, 2021.
- SOLOMIN, EV; TOPOLSKY, DV; TOPOLSKY, ND. Arrangement of data exchange between adaptive digital current and voltage transformer and SCADA-system under IEC 61850 standard. **Procedia engineering**, Elsevier, v. 129, p. 207–212, 2015.
- STOLFO, SJ et al. **KDD Cup 1999 Dataset**. [S.l.: s.n.], 1999.



- STROBEL, Maximilian; WIEDERMANN, Norbert; ECKERT, Claudia. Novel weaknesses in IEC 62351 protected smart grid control systems. In: IEEE. 2016 IEEE International Conference on Smart Grid Communications (SmartGridComm). [S.l.: s.n.], 2016. p. 266–270.
- TATBUL, Nesime et al. Precision and Recall for Time Series. In: BENGIO, S. et al. (Eds.). **Advances in Neural Information Processing Systems 31**. [S.l.]: Curran Associates, Inc., 2018. p. 1920–1930.
- TAVALLAEE, Mahbod et al. A detailed analysis of the KDD CUP 99 data set. In: IEEE. 2009 IEEE symposium on computational intelligence for security and defense applications. [S.l.: s.n.], 2009. p. 1–6.
- THASEEN, I Sumaiya; KUMAR, Ch Aswani; AHMAD, Amir. Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers. **Arabian Journal for Science and Engineering**, Springer, v. 44, n. 4, p. 3357–3368, 2019.
- TONG, W. et al. A Survey on Intrusion Detection System for Advanced Metering Infrastructure. In: 2016 Sixth International Conference on Instrumentation Measurement, Computer, Communication and Control (IMCCC). [S.l.: s.n.], 2016. p. 33–37.
- TURNER, Aaron. **Tcp replay: Pcap editing and replay tools for NIX**. [S.l.: s.n.], 2005.
- UCHÔA, Luana et al. Analysis of smart grid fault recovery protocols. In: IEEE. NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. [S.l.: s.n.], 2020. p. 1–8.
- USTUN, Taha Selim; FAROOQ, Shaik Mullapathi; HUSSAIN, SM Suhail. A Novel Approach for Mitigation of Replay and Masquerade Attacks in Smartgrids Using IEC 61850 Standard. **IEEE Access**, IEEE, v. 7, p. 156044–156053, 2019.
- VIEIRA, Juan Lucas et al. THANOS: Teleprotection Holistic Application for ONOS Controller. In: IEEE. 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM). [S.l.: s.n.], 2021. p. 818–823.
- WEIDEMAN, J Andre C. Numerical integration of periodic functions: A few examples. **The American mathematical monthly**, Taylor & Francis, v. 109, n. 1, p. 21–36, 2002.

- YANG, Cheng-Huei; CHUANG, Li-Yeh; YANG, Cheng-Hong, et al. IG-GA: a hybrid filter/wrapper method for feature selection of microarray data. **source: Journal of Medical and Biological Engineering**, v. 30, 2010.
- YANG, Q; KECKALO, D, et al. Testing IEC 61850 Merging Units. In: PROCEEDINGS OF THE 44TH ANNUAL WESTERN PROTECTIVE RELAY CONFERENCE, WASHINGTON, DC, USA. [S.l.: s.n.], 2017. p. 17–19.
- YANG, Qiang; HAO, Weijie, et al. FARIMA model-based communication traffic anomaly detection in intelligent electric power substations. **IET Cyber-Physical Systems: Theory & Applications**, IET, v. 4, n. 1, p. 22–29, 2019.
- YANG, Yi; MCLAUGHLIN, Kieran, et al. Intrusion detection system for IEC 61850 based smart substations. In: IEEE. 2016 IEEE Power and Energy Society General Meeting (PESGM). [S.l.: s.n.], 2016. p. 1–5.
- YANG, Yi; XU, Hai-Qing, et al. Multidimensional intrusion detection system for IEC 61850-based SCADA networks. **IEEE Transactions on Power Delivery**, IEEE, v. 32, n. 2, p. 1068–1078, 2016.
- YOO, Hyungkuk; SHON, Taeshik. Novel approach for detecting network anomalies for substation automation based on IEC 61850. **Multimedia Tools and Applications**, Springer, v. 74, n. 1, p. 303–318, 2015.
- YUSTA, Silvia Casado. Different metaheuristic strategies to solve the feature selection problem. **Pattern Recognition Letters**, Elsevier, v. 30, n. 5, p. 525–534, 2009.
- ZOPELLARO SOARES, Arthur A et al. SDN-based teleprotection and control power systems: A study of available controllers and their suitability. **International Journal of Network Management**, Wiley Online Library, v. 31, n. 3, e2112, 2021.