Attacking Learning-based Models in Smart Grids: Explainability as a Double-Edged Sword

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Gustavo Sánchez Collado
aus Casas del Castañar, Spanien

Tag der mündlichen Prüfung:	27.04.2026
1. Referent:	Prof. Dr. Veit Hagenmeyer
2. Referentin:	Prof'in. Dr. Barbara Hammer

Abstract

Smart grids increasingly rely on learning-based components for monitoring, control, and security-critical decision making. To address concerns regarding transparency, accountability, and regulatory compliance, eXplainable Artificial Intelligence (XAI) techniques are often integrated into these systems. However, the security implications of explainability in adversarial environments remain insufficiently understood. This dissertation investigates explainability from an attacker-centric perspective and analyzes how XAI alters the threat landscape of learning-based smart grid systems.

Adopting a proactive adversarial methodology, the thesis combines threat analysis, a review of existing attacks, and extensive empirical evaluations conducted in realistic smart grid testbeds. Across multiple use cases (including intrusion detection, power quality recognition and industrial vision pipelines) the work demonstrates that XAI can be systematically exploited to amplify adversarial attacks against integrity, availability, and confidentiality of targeted models. Explanation methods are shown to reduce attacker uncertainty, guide feature-space and problem-space perturbations, enable efficient data poisoning, and facilitate model extraction and covert data exfiltration.

Beyond demonstrating attacks, the thesis explores defense directions that explicitly account for XAI-induced attack surfaces. A lightweight Moving Target Defense strategy is evaluated, showing that redistributing feature importance and eliminating brittle correlations can significantly improve robustness without incurring substantial overhead.

Overall, this work establishes explainability as a double-edged sword in smart grid security: while enhancing transparency for legitimate stakeholders, it simultaneously expands the attacker's capability space. The findings highlight the need for security-aware explainability and adversarially informed design principles in future learning-based critical infrastructure systems.

Acknowledgement

I want to express my gratitude and appreciation to my supervisors at different levels: Dr. Kaibin Bao as group leader; Andreas Hoffmann as institute director; and Prof. Veit Hagenmeyer as "Doctor Father" for believing in me, respecting my best interests, and their continuous support.

I want to thank my dear colleagues Max, Ramadan, Gustav, Hermenegildo, Moritz, Yilin, Qi Zhao, Achyut, Daniel, Dan, Philipp, Aneeqa, Nicolai, Hemanth, Sine, Arman and Richard, who always supported me through my journey both in technical and emotional challenges.

I also thank the administrative colleagues (Frau Sauer, Frau Lehmann, Frau Lang, etc.) for their invaluable support.

I want to thank my former colleague and life partner Niki for her unlimited happiness, love and encouragement; she really gave me the strength to overcome challenges and become a better version of myself.

Por último, quiero dar las gracias a mis padres y a mi hermano por su continuo apoyo y cariño; a mis abuelos, por enseñarme lecciones importantes en la vida, como el trabajo duro y la humildad; y al resto de mi familia y amigos, por estar siempre a mi lado y apoyarme incondicionalmente.

List of Acronyms

IDS Intrusion Detection System

XAI eXplainable Artificial Intelligence

MMS Manufacturing Message Specification

SV Sampled Values

ML Machine Learning

AI Artificial Intelligence

LLMs Large Language Models

PQR Power Quality Recognition

GNSS Global Navigation Satellite System

IED Intelligent Electronic Device

CNN Convolutional Neural Network

CIA Confidentiality, Integrity and Availabilty

PLC Programmable Logic Controller

ICS Industrial Control Systems

SHAP SHapley Additive exPlanations

LIME Local Interpretable Model-agnostic Explanations

LRP Layer-wise Relevance Propagation

MDI Mean Decrease in Impurity

RF Random Forest

SVM Support Vector Machine

MLP Multi-Layer Perceptron

RAIM Receiver Autonomous Integrity Monitoring

FGSM Fast Gradient Sign Method

PQD Power Quality Disturbance

DNN Deep Neural Network

MITM Man-In-The-Middle

MTD Moving Target Defense

FDI False Data Injection

DLP Data Loss Protection

NIDS Network Intrusion Detection Systems

APT Advanced Persistent Threat

LSB Least Significant Bit

PRNG Pseudorandom Number Generator

SOC Security Operation Center

IoC Indicators of Compromise

API Application Programming Interface

DCT Discrete Cosine Transform

SSIM Structural Similarity Index

PSNR Peak Signal-to-Noise Ratio

NLP Natural Language Processing

SCADA Supervisory Control and Data Acquisition

Contents

1 Introduction

> *If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.*
>
> **— Sun Tzu**
> *The Art of War, 5th century BC*

In the domain of critical infrastructure, security failures can result in severe economic, societal, and safety consequences. Traditionally, cybersecurity in such systems has been largely reactive: countermeasures are deployed only after vulnerabilities have been discovered or incidents have already occurred. This paradigm is increasingly recognized as insufficient for complex, interconnected infrastructures, where attacks may propagate rapidly and have cascading effects [CG11]. Consequently, significant research efforts have shifted toward proactive cyber defense, aiming to anticipate future attack strategies and incorporate these insights into system design and protection mechanisms.

Within this context, offensive security research plays a pivotal role. Understanding how systems can be attacked is a prerequisite for building robust defenses. This is particularly relevant for the smart grid, where the growing integration of learning-based components introduces novel and often poorly understood attack surfaces [Zha+24a]. The security of such systems must therefore be analyzed according to the well-established "three golden rules" of adversarial machine learning [BR18]: (i) modeling the adversary and their capabilities, (ii) adopting a proactive, attack-aware perspective, and (iii) designing mechanisms that enable self-protection and resilience.

Adversaries exploit vulnerabilities in learning-based models through adversarial attacks, which can be viewed as a refined and adaptive form of False Data Injection (FDI) tailored to intelligent algorithms [Son+21]. These attacks encompass a wide range of strategies, including evasion, poisoning, and information extraction, each

differing in objectives, assumptions, and operational constraints. In safety- and mission-critical domains such as smart grids, even small degradations in model performance or trustworthiness can have disproportionate consequences.

To address concerns regarding opacity and accountability of learning-based systems, eXplainable Artificial Intelligence (XAI) techniques have been proposed as a means to enhance transparency, interpretability, and trust in automated decision-making [Arr+20]. In critical infrastructures, XAI is often regarded as a facilitating technology for validation, operator acceptance, and regulatory compliance. However, while explanations may improve human understanding, they also expose internal model properties that were previously hidden. This dual role raises a fundamental question: can explainability itself become an attack vector?.

Smart grids constitute highly structured and controlled environments in which data-driven techniques are increasingly deployed for operational and security-related tasks. Representative applications include short-term load forecasting [Kon+17], power quality assessment, and the detection of cyber-physical attacks such as FDI [JHL18]. Despite the extensive literature demonstrating the effectiveness of learning-based approaches in these settings, comparatively little attention has been devoted to their resilience under adversarial conditions, particularly when attackers can exploit explanation mechanisms. Studying adversarial attacks in smart grids is therefore not merely an exercise in understanding attacker behavior, but a necessary step toward identifying systemic weaknesses and informing the design of effective countermeasures.

Learning-based security analysis in smart grids differs substantially from conventional machine-learning domains due to the cyber-physical nature of energy systems. Data encountered in electrical substations and grid monitoring environments is predominantly structured, tabular, and protocol-driven, originating from industrial communication standards. Unlike image or text data, these signals encode physical processes governed by electrical laws, timing constraints, and operational safety requirements. Perturbations must therefore remain physically plausible and protocol-compliant to remain realistic.

Furthermore, smart-grid environments operate under strict availability and safety constraints: false alarms may disrupt operations, while missed detections can propagate into physical instability. These characteristics create a unique adversarial setting in which attacks must balance stealth, physical feasibility, and operational continuity. Consequently, evaluating adversarial machine learning in this domain requires problem-space experimentation rather than purely synthetic feature-space manipulation.

Beyond the power systems domain, adversarial attacks against learning-based models have been studied extensively. Early work by Dalvi et al. [Dal+04] demonstrated how learning-based email spam filters could be systematically circumvented by adaptive adversaries. Subsequent research has largely focused on adversarial perturbations in perceptual domains, including computer vision [CW17] and audio processing [CW18]. More recently, the widespread deployment of Large Language Models (LLMs) has spurred significant interest in adversarial attacks in textual environments [Sha+23]. These studies highlight that adversarial vulnerabilities are not incidental, but intrinsic to many learning paradigms.

Crucially, adversarial strategies against learning-based models are highly domain-dependent. Differences in data modalities, system constraints, observability, and operational semantics fundamentally shape both the feasibility and the impact of attacks [Son+21]. As a result, insights derived from generic benchmarks or perceptual domains cannot be directly transferred to cyber-physical infrastructures such as smart grids. This necessitates domain-specific threat models, attack methodologies, and evaluation frameworks, which form the foundation of the present dissertation.

The growing adoption of explainability is strongly influenced by regulatory developments, particularly within the European Union. The EU AI Act emphasizes transparency, traceability, and human oversight for high-risk AI systems [LWM24], a category that explicitly includes critical infrastructure such as energy networks. As operators increasingly deploy explainability tools to satisfy auditing and compliance requirements, explanation outputs may become accessible to multiple stakeholders, including external auditors and system integrators.

This regulatory context elevates explainability from an optional diagnostic tool to an operational requirement [Pan+23]. The central hypothesis of this thesis is therefore that mandatory explainability introduces a new interaction channel between models and adversaries. When explanations are available, attackers may incorporate them directly into their optimization loop, enabling what we term XAI-in-the-loop or XAI-guided adversarial attacks.

Rather than focusing on a single attack category, this dissertation adopts a holistic security perspective structured around the Confidentiality, Integrity, and Availability (CIA) triad. Each dimension corresponds to a distinct adversarial objective and operational risk. Integrity attacks manipulate model decisions, availability attacks degrade operational reliability, and confidentiality attacks extract sensitive knowledge from deployed systems.

To capture these dimensions comprehensively, the thesis investigates multiple learning paradigms and deployment contexts, including network intrusion detection, power quality recognition, and industrial vision pipelines. This multi-use-case approach enables a systematic analysis of how explainability influences adversarial capabilities across fundamentally different learning scenarios.

1.1 Methodology

Modern critical infrastructures are undergoing a rapid transition toward data-driven operation, where learning-based components increasingly support monitoring, optimization, and security decisions. While these systems promise improved efficiency and automation, they simultaneously introduce new attack surfaces that differ fundamentally from those of traditional rule-based systems. In particular, the behavior of learning models depends on statistical correlations learned from data rather than explicitly specified logic, making their failure modes difficult to anticipate using classical security analysis. As a consequence, proactive security evaluation must extend beyond software vulnerabilities and consider how intelligent models themselves can be manipulated by adversaries.

XAI has emerged as a key mechanism to address transparency and accountability concerns in such systems. However, the very information that explanations reveal about model behavior may also reduce uncertainty for attackers. This observation motivates the central premise of this thesis: explainability is not purely a defensive capability but a security-relevant interface that reshapes the attacker's capability space. The research questions formulated in this work therefore investigate how XAI alters the attack surface of learning-based smart-grid systems and how explanation mechanisms can be exploited (or constrained) to improve security.

This thesis adopts a proactive and adversarial research methodology aimed at systematically analyzing the security implications of explainability in learning-based smart grid systems. The methodological approach is structured in four successive stages, each building upon the insights of the previous one.

First, a global analysis of cyber threats targeting the energy sector is conducted. This analysis combines geopolitical context, cyber threat intelligence, and incident reporting, with a particular focus on Artificial Intelligence (AI)-enabled attacks affecting critical infrastructure. The purpose of this stage is not to introduce new threat intelligence, but to motivate a realistic attacker model. The analysis supports the hypothesis that relevant threat actors are persistent, well-resourced, and often

state-level, and therefore capable of exploiting any exposed signal or auxiliary information. In regulatory environments where XAI is mandated for auditing, validation, or reporting purposes, explanation outputs become such an exploitable signal.

Second, the thesis surveys and systematizes existing scientific work on adversarial attacks against learning-based components in smart grids. A total of 34 peer-reviewed studies are analyzed and categorized according to attack objectives and threat models. In addition, a reproducibility assessment is performed to evaluate the practical feasibility of reported attacks. Based on this analysis, a conceptual mapping between the Confidentiality, Integrity, and Availability (CIA) triad and the Authentication, Authorization, and Accounting (AAA) security framework is established. This taxonomy provides a unifying lens through which attacks against learning-based smart grid systems are analyzed throughout the remainder of the thesis.

Third, the core contribution of this dissertation investigates XAI from an adversarial perspective. Rather than treating explainability as a purely defensive or trust-enhancing mechanism, XAI is analyzed as a potential attack enabler. Across multiple smart grid use cases and threat models, the thesis examines how explanation methods can reduce attacker uncertainty, guide attack optimization, and facilitate adversarial actions targeting integrity, availability, and confidentiality. All attacks are evaluated in smart grid environments, with an emphasis on feasibility, attacker effort, and operational impact.

Finally, the thesis explores defensive directions informed by the identified attack vectors. Our survey of existing literature reveals a strong concentration of research efforts on integrity attacks. This predominance suggests that integrity violations represent the most immediate and practically relevant threat surface for learning-based smart-grid systems. Because explanation methods directly expose feature importance and decision rationale, integrity attacks are especially likely to benefit from XAI guidance. Motivated by this observation, the defense exploration in this thesis focuses on integrity scenarios. A lightweight defense strategy based on Moving Target Defense (MTD) principles is proposed and evaluated in a smart grid intrusion detection scenario. The objective is not to provide a universal mitigation, but to demonstrate how defense mechanisms can be designed to explicitly account for XAI-induced attack surfaces and to increase the cost and uncertainty faced by adversaries.

1.2 Research Questions

This thesis investigates explainability in learning-based smart grid systems used for adversarial purpose. While XAI is commonly introduced to improve transparency, trust, and accountability, its impact on the security of such systems remains insufficiently understood. The central research question addressed in this dissertation is therefore:

How does explainable artificial intelligence (XAI) alter the attack surface of learning-based systems in smart grids, and to what extent can explanation mechanisms be exploited to compromise system security?

To address this question, the thesis investigates adversarial attacks across the three dimensions of the CIA triad. Each topic examines how attackers can leverage learning-based models and, in particular, explanation mechanisms to achieve malicious objectives in realistic smart grid scenarios.

We group more specific research questions into four topics, all within the overarching theme of an attacker that leverages XAI for malicious purposes. In Topic-1, we investigate integrity attacks. In Topic-2, we explore attacks against the availability of models. In Topic-3, we address confidentiality compromises. This thesis serves as an encyclopedia of XAI-in-the-loop offensive strategies against learning-based models within smart grid environments, including a compelling defense strategy.

1.2.1 Topic 1: Attacks Against Integrity

Integrity attacks aim to induce incorrect model behavior while preserving the apparent normal operation of the underlying system. In smart grids, such attacks are particularly dangerous, as they may allow adversaries to evade detection or manipulate control decisions without triggering alarms.

- **RQ-1: To what extent can evasion attacks in the problem space of smart grids compromise the integrity of learning-based models, and how can such attacks be mitigated in practice?**

Adversarial examples, introduced during the model's deployment phase (i.e., at inference time), are engineered to mislead the smart grid's decision-making algorithms, which are critical for real-time monitoring and control, among others. Developing realistic and practical implementations of proof-of-concepts for adversarial examples in electrical substations is important to understand the extent of this threat. To

answer RQ-1, we test the security of learning-based models within realistic contexts by implementing feasible adversarial examples. Additionally, we provide a solution to defend against this kind of attack, i.e., by employing a moving target defense technique.

- **RQ-2: How can targeted data poisoning attacks be constructed to compromise the integrity of learning-based smart grid models during training, and what are their practical implications?**

Targeted data poisoning, performed against a model during its training phase, degrades the model's performance, potentially leading to situations such as intrusion detection failures, which could have cascading effects on grid security. Indeed, an attacker able to inject *backdoors* or *trojans* during data collection can later rely on these manipulations to deceive the model with carefully crafted elements at inference. When developers build datasets for training, it is in their best interest to avoid miss-labeling (in supervised settings) and/or pollution of the normal profile (in unsupervised settings). A malicious actor could purposely inject malicious data points to compromise the model's performance. The goal is to maximize targeted classification errors (e.g., misclassifications into a specific target class) by including poisoned data into the training set. To answer RQ-2, we investigate the injection of malicious data during training stage of learning-based models to produce targeted mistakes in smart grid use cases, including Intrusion Detection System (IDS)s in electrical substations. Additionally, we investigate how to perform data poisoning in realistic smart grid scenarios.

RQ-1 and RQ-2 jointly investigate integrity compromises at both inference time and training time, providing a comprehensive view of integrity threats to learning-based smart grid systems.

1.2.2 Topic 2: Attacks Against Availability

Availability attacks aim to degrade or disrupt the reliable operation of learning-based components, preventing legitimate users from obtaining timely and correct system functionality. In smart grid environments, such attacks are especially critical due to real-time constraints and resource limitations of field devices.

- **RQ-3: How can indiscriminate data poisoning attacks be leveraged to compromise the availability of learning-based systems in smart grids by inducing denial-of-service conditions?**

Indiscriminate poisoning, by corrupting training data, undermines the model's predictive accuracy, rendering it less effective in real-time applications by causing denial of service conditions.

To answer RQ-3, this thesis investigates training-time poisoning strategies that degrade model reliability and responsiveness, thereby causing operational disruptions in learning-based components deployed in electrical substations.

1.2.3 Topic 3: Attacks Against Confidentiality

Confidentiality attacks target the unauthorized extraction of information from learning-based models, including proprietary model parameters, training data characteristics, and sensitive operational patterns.

- **RQ-4: How can learning-based models deployed in smart grid environments be reverse-engineered or stolen, and what confidentiality risks arise from such attacks?**

Model stealing attacks replicate the functionality of smart grid models, potentially revealing proprietary algorithms that are vital for grid security and efficiency. These attacks not only threaten the privacy of the smart grid's operational data but also risk violating consumer trust and regulatory compliance. If an attacker obtains data used to train a given model, it would be possible to train surrogate models known to be highly similar to the original. This situation allows an attacker to generate transferable adversarial examples, along with compromising the original model's intellectual property. To answer RQ-4, we reverse-engineer learning-based models within threat models feasible in use cases related to electrical substations.

Collectively, these research questions frame a systematic investigation of XAI-enabled adversarial attacks against learning-based smart grid systems, spanning integrity, availability, and confidentiality, while grounding all analyses in realistic threat models and operational constraints.

1.3 Contributions of the Present Work

This dissertation investigates security vulnerabilities of learning-based components deployed in smart grids, with a particular focus on adversarial settings in which

attackers can exploit explainability mechanisms. The primary experimental environment is the KASTEL Security Lab Energy testbed, complemented by open-source models and datasets to ensure generality and reproducibility.

Integrity Attacks - Evasion (RQ-1). In addressing **RQ-1**, this thesis demonstrates that explainability can be systematically exploited to facilitate evasion attacks against learning-based intrusion detection systems in smart grid environments. Specifically, attacks targeting Modbus TCP traffic are implemented and evaluated against a machine-learning-based IDS. By applying XAI methods, features with high influence on model decisions are identified and selectively perturbed, resulting in significantly increased evasion success with minimal modification of input data. The findings show that perturbations guided by explanation outputs are substantially more effective than unguided alternatives. This XAI-guided evasion strategy is further generalized to industrial computer vision pipelines, where automated, explanation-driven physical perturbations are introduced in monitoring systems for industrial control environments.

Integrity Attacks - Targeted Poisoning (RQ-2). In response to **RQ-2**, this work introduces XAI-guided targeted data poisoning attacks against learning-based models in smart grid contexts. The proposed attacks are evaluated on intrusion detection systems for Manufacturing Message Specification (MMS) and Sampled Values (SV) traffic compliant with the IEC 61850 standard, as well as on an open-source Power Quality Recognition (PQR) use case. By leveraging explanation methods during the poisoning process, the attacks selectively manipulate influential features, enabling precise integrity violations such as targeted misclassifications and backdoor behaviors. The results demonstrate that XAI significantly increases the effectiveness and efficiency of poisoning attacks.

Availability Attacks (RQ-3). In answering **RQ-3**, the thesis investigates indiscriminate data poisoning attacks aimed at compromising the availability of learning-based smart grid systems. Using XAI-in-the-loop strategies, poisoning attacks are conducted against IDS models monitoring SV, MMS, and S7Comm traffic, as well as against PQR classifiers. Two realistic problem spaces (GNSS time spoofing and the Siemens S7Comm protocol) are formalized to demonstrate end-to-end poisoning attacks that degrade model reliability and operational usability. The results show that explanation-guided poisoning can effectively induce denial-of-service conditions

by broadly degrading model performance while maintaining plausibility within smart grid operational constraints.

Confidentiality Attacks (RQ-4). In addressing **RQ-4**, this dissertation demonstrates that explainability can also facilitate attacks against model confidentiality. First, a model extraction attack is performed against an S7Comm-based IDS, showing how explanation-enhanced attacker capabilities can accelerate model stealing and enable subsequent transferable attacks. Second, a novel steganographic model-stealing scenario is introduced, in which an advanced persistent threat embeds model secrets into network-traffic image representations. Both saliency-based and non-saliency-based strategies are evaluated, enabling covert exfiltration of model information with and without observable degradation of classifier performance. This contribution introduces the concept of *steganographic adversarial attacks*.

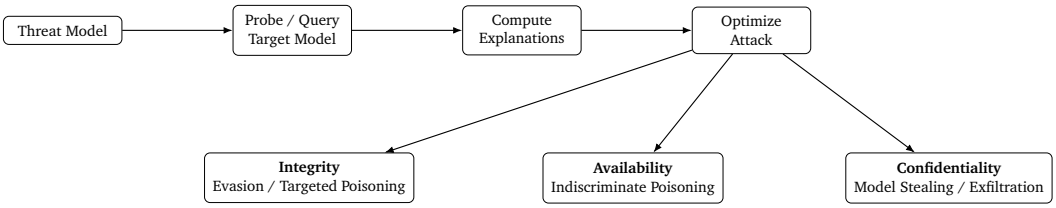


Fig. 1.1: XAI expands the attack surface by providing structured information that enables adversarial optimization across the CIA triad.

Before delving into technical details, it is important to clarify the scope of this work. The goal of this dissertation is not to propose explainability as inherently insecure, nor to introduce fundamentally new adversarial algorithms. Instead, the contribution lies in empirically demonstrating how explainability reshapes adversarial reasoning in realistic cyber-physical environments and in identifying design principles that reconcile transparency with security. The subsequent Chapters therefore combine threat analysis, experimental validation, and defensive insights to understand explainability as a double-edged component of learning-based critical infrastructure systems. Table 1.1 provides a thesis overview, including the use cases investigated.

The investigations conducted throughout this thesis lead to several overarching conclusions. First, explanation mechanisms systematically reduce attacker uncertainty by revealing structured information about model behavior. Second, XAI-guided attacks achieve stronger effects with fewer perturbations compared to unguided baselines, particularly in integrity and availability scenarios. Third, confidentiality risks emerge when explanation outputs act as auxiliary side channels that accelerate

Tab. 1.1: Overview of attacks studied in this thesis, categorized by security objective (CIA triad), attack timing, and Research Question (RQ) addressed.

Use Case	CIA	Timing	RQ
Evading Modbus TCP IDS (evasion / mimicry) + Moving Target Defense	Integrity	Test-time	RQ1
Autonomous XAI-guided physical adversarial perturbations in industrial vision pipelines	Integrity	Test-time	RQ1
Targeted poisoning against PQR, SV IDS, and MMS IDS (targeted misclassification of anomalous signals)	Integrity	Train-time	RQ2
Indiscriminate poisoning against PQR, MMS IDS, and SV IDS in feature space (availability degradation)	Availability	Train-time	RQ3
Indiscriminate poisoning against SV IDS in problem space	Availability	Train-time	RQ3
Model stealing against S7 IDS (query-based extraction for poisoning via fine-tuning)	Confidentiality	Test/Train-time	RQ4
Data exfiltration for model stealing in MMS IDS (steganographic embedding of model secrets)	Confidentiality	Test-time	RQ4

model extraction. Finally, defenses that distribute feature importance and introduce controlled variability can significantly mitigate these risks without eliminating explainability altogether.

As represented in Fig. 1.1, learning-based security mechanisms introduce new attack surfaces. Explainability, while intended to improve trust and robustness, systematically enlarges the attacker's capability space across integrity, availability, and confidentiality.

1.4 Thesis Structure

Chapter 2: Preliminaries

This Chapter introduces the foundational concepts required for the thesis, including smart grid communication protocols, learning-based applications in smart grids, adversarial machine learning, and explainable artificial intelligence. It establishes the terminology, threat models, and security frameworks used throughout the dissertation.

Chapter 3: Threat Landscape and Related Work

This Chapter provides the broader context for the thesis. It first presents a global analysis of cyber threats targeting the energy sector, motivating realistic and well-

resourced attacker models. It then surveys and critically analyzes existing research on adversarial attacks against learning-based smart grid components, including a reproducibility assessment that highlights current limitations in the state of the art.

Chapter 4: Testbed

This Chapter describes the experimental environments and learning-based models used as controlled testbeds for the conducted attacks. It introduces the intrusion detection systems, power quality recognition models, and route choice prediction models that serve as representative smart grid use cases.

Chapter 5: Attacking Integrity

This Chapter investigates evasion and targeted data poisoning attacks that compromise the integrity of learning-based models in smart grids. Particular emphasis is placed on how explainability methods can be exploited to guide and optimize integrity attacks under realistic operational constraints.

Chapter 6: Attacking Availability

This Chapter analyzes indiscriminate poisoning attacks aimed at degrading the availability of learning-based smart grid systems. It demonstrates how XAI-guided strategies can be used to induce denial-of-service conditions by broadly reducing model reliability and responsiveness.

Chapter 7: Attacking Confidentiality

This Chapter focuses on attacks against the confidentiality of learning-based models, including model extraction and steganographic exfiltration techniques. It shows how explanation mechanisms can facilitate information leakage and enable further downstream attacks.

Chapter 8: Defense Directions

This Chapter explores defense directions informed by the identified attack vectors, with a focus on a lightweight Moving Target Defense strategy. The Chapter evaluates how such defenses can increase attacker uncertainty and mitigate risks introduced by explainability mechanisms.

1.5 Publications

This dissertation is supported by a set of peer-reviewed publications that directly contribute to the research questions and core findings presented in this thesis.

Published work is referenced throughout the thesis with a leading letter "P", and in **blue** color to distinguish original contributions from other work.

Core Publications Supporting This Thesis

The following publications form the primary scientific basis of this dissertation and directly support its research questions:

- **G. Sánchez**, G. Elbez, V. Hagenmeyer: *"Attacking Learning-based Models in Smart Grids: Current Challenges and New Frontiers."* In 15th ACM International Conference on Future Energy Systems, 2024. **[P1]**

  Provides a structured overview of adversarial threats to learning-based smart grid components, motivates the CIA-based analysis adopted in this thesis, and supports RQ-1 by investigating XAI-powered evasion.

- **G. Sánchez**, G. Elbez, V. Hagenmeyer: *"Explainable AI in Data Poisoning Threat Models Across the CIA Triad: A Smart Grid Study."* In 7th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, 2025. **[P2]**

  Supports RQ-2 and RQ-3 by demonstrating XAI-guided poisoning attacks affecting integrity and availability.

- **G. Sánchez**, G. Elbez, V. Hagenmeyer: *"Currents of Conflict: A Global Analysis of Cyber Threats to the Energy Sector."* In *atp magazin*, Issue 09/2025. **[P3]**

  Motivates realistic, well-resourced attacker models used throughout the dissertation.

- **G. Sánchez**, G. Elbez, V. Hagenmeyer: *"Lightweight Moving Target Defense for Robust Intrusion Detection in Smart Grids."* In Energy Informatics Academy Conference, 2025. **[P4]**

  Supports RQ-1 by evaluating a defense strategy informed by XAI-enabled attack surfaces.

- **G. Sánchez**, M. Qasim, G. Elbez, V. Hagenmeyer: *"Steganographic Data Exfiltration for Model Stealing: A Case Study on Energy Critical Infrastructure IEC 61850 Datasets."* In IEEE International Conference on Big Data, 2025. **[P5]**

Supports RQ-4 by introducing XAI-based steganography for confidentiality attacks.

- **G. Sánchez**, L. Wei, V. Hagenmeyer: *"LaserTag: A Tool for Autonomous XAI-Guided Physical Adversarial Perturbations in Industrial Vision Pipelines."* In The 56th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2026. **[P6]**

  Supports RQ-1 by performing integrity attacks guided by XAI.

Additional Supporting Publications

The following publications provide additional context, experimental infrastructure, or complementary perspectives relevant to the thesis:

- A. Mumrez, **G. Sánchez**, G. Elbez, V. Hagenmeyer: *"On Evasion of Machine Learning-based Intrusion Detection in Smart Grids."* In IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, 2023. **[P7]**

  Provides a proof of concept supporting RQ-1 by showcasing XAI-powered evasion. A. Mumrez collected the data, and G. Sánchez trained the models and implemented the adversarial attacks.

- G. Elbez, **G. Sánchez**, S. Canbolat, S. Corallo, C. Fruböse, F. Lanzinger, N. Kellerer, G. Keppler, F. Neumeister, B. Beckert, A. Koziolek, M. Zitterbart, and V. Hagenmeyer: *"Insights and Lessons Learned from a Realistic Smart Grid Testbed for Cybersecurity Research"*. In ACM International Conference on Future Energy Systems, 2025. **[P8]**

  Provides a description of the testbed where part of the experiments are carried out.

- N. Kellerer, **G. Sánchez**, H. Alberto, G. Elbez, V. Hagenmeyer: "Attacks on the Siemens S7 Protocol Using an Industrial Control System Testbed". In ACM International Conference on Future Energy Systems, 2025. **[P9]**

  Provides preliminary attack detection results on the Siemens S7 Protocol. N.

Kellerer performed data collection, and G. Sánchez trained the models and analyzed feature importance.

- **G. Sánchez**, F. Ünal, A. Wins: *"Route Choice Prediction Through User Behavior Analysis: Towards Robustness Assessment Under External Perturbations"*. In 7th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications, 2025. **[P10]**

  Preliminary study of navigation route choice prediction and its robustness to small, realistic perturbations, with emphasis on attack surfaces that matter for transport and energy infrastructure.

Parallel and Methodologically Related Work

In parallel to the main dissertation contributions, the following publications address security and AI in adjacent domains:

- J. Caballero, G. Gomez, S. Matic, **G. Sánchez**, S. Sebastián, and A. Villacañas. "The Rise of GoodFATR: A Novel Accuracy Comparison Methodology for Indicator Extraction Tools." In Future Generation Computer Systems 144, 2023. **[P11]**

- **G. Sánchez**, O. Olayinka, A. Pashikani: *"Web Application Penetration Testing with Artificial Intelligence: A Systematic Review"*. In 23rd IEEE International Symposium on Network Computing and Applications 2024. **[P12]**.

- H. Alberto*, **G. Sánchez***, G. Elbez, V. Hagenmeyer: *"Masquerading IEC 61850 GOOSE Protocol: Cyber-Physical Experiments and Detection."* In 16th ACM International Conference on Future Energy Systems, 2025 (*double first authorship*). **[P13]**

- **G. Sánchez** and A. Lundqvist : *"Poster: Towards Intelligent Assurance for Autonomous AI Pentesters: Concurrent Compliance Auditing and Self-Augmentation via Execution Trace Analysis"*. In the 32nd ACM Conference on Computer and Communications Security (CCS) 2025. **[P14]**

- **G. Sánchez** and S. Zhang: *"`Wasabi`: Leveraging Cross-Lingual Pseudo-Homophones to Evade NLP Moderation Systems"*. Under submission. **[P15]**

- M. Schwarzer, **G. Sánchez**, J.F. Loevenich, R.R.F. Lopez, V. Hagenmeyer: *"Synthesize, Adapt, Steal: A Few-Shot Domain Adaptive Model Stealing Attack for Tabular Data"*. In the IEEE Conference on Artificial Intelligence (CAI) 2026. **[P16]**
- M. Schwarzer, J.F. Loevenich, **G. Sánchez**, R.R.F. Lopez, V. Hagenmeyer : *"AI Model Extraction Attacks: Bypassing Single-Client Assumptions in Defenses"*. In The International Conference on Military Communication and Information Systems (ICMCIS), 2026. **[P17]**

2 Preliminaries

Learning-based mechanisms in smart grids operate at the intersection of cyber, physical, and human domains, requiring a precise and shared understanding of both system architectures and adversarial capabilities. This Chapter introduces the foundational concepts necessary to frame the remainder of the dissertation. It first presents the communication protocols and operational components commonly used in smart grids, followed by an overview of learning-based applications deployed in these environments. The Chapter then formalizes key notions from adversarial machine learning, including threat models, evaluation metrics, and security objectives, with particular emphasis on the CIA triad and its relation to classical security frameworks. Finally, explainable artificial intelligence is introduced, not only as a transparency mechanism, but as a source of structured information that may alter the attacker's capabilities. Together, these preliminaries establish the conceptual and technical baseline upon which all subsequent analyses are built.

2.1 Smart Grid Communication Protocols

The increasing digitalization of electrical substations has been largely enabled by the adoption of the IEC 61850 standard [Int13], but there are other protocols present in these environments.

2.1.1 IEC 61850 Standard

This international standard defines a communication architecture for substation automation systems, specifying data models, services, and protocols to ensure interoperability across vendors. IEC 61850 not only covers the data exchange between intelligent electronic devices (IEDs), but also introduces a flexible object-oriented information model that supports advanced applications in the smart grid.

Together, GOOSE, MMS, and SV form the backbone of modern substation communication under IEC 61850, providing a unified framework for both time-critical

protection applications and supervisory monitoring/control, thereby enabling the integration of next-generation smart grid technologies.

GOOSE. A key element of IEC 61850 is the *Generic Object Oriented Substation Event* (GOOSE) mechanism, which enables the fast and reliable multicast transmission of time-critical messages, such as protection trips or interlocking signals [Bru08]. GOOSE messages are mapped directly onto the Ethernet layer, bypassing higher-level protocols to minimize latency and guarantee real-time performance.

MMS. Another major service defined in IEC 61850 is the *Manufacturing Message Specification* (MMS), which provides a client/server communication framework for supervisory control, configuration, and monitoring tasks [Int03]. Unlike GOOSE, which focuses on event-driven peer-to-peer messaging, MMS operates over TCP/IP and offers a flexible mechanism to access the standardized data objects and attributes of IEDs.

SV. In addition, IEC 61850 specifies the *Sampled Values* (SV) service, which allows the transmission of digitized analog measurements, such as currents and voltages, over the substation network [Ing+12]. This enables the replacement of copper wiring with fiber-optic communication, leading to reduced cost, increased flexibility, and improved accuracy in protection and monitoring functions.

2.1.2 Siemens S7 Protocol

The Siemens S7 protocol is a proprietary communication protocol used by Siemens programmable logic controllers (PLCs) for industrial automation. It enables functions such as reading and writing process data, controlling inputs and outputs, and performing diagnostic operations [Dzu+05]. The protocol operates primarily over TCP/IP (known as S7comm over ISO on TCP) and is widely deployed in industrial control systems (ICS). Due to its prevalence in manufacturing and critical infrastructure, the S7 protocol has been the subject of extensive research on cybersecurity, particularly in the context of vulnerabilities and targeted malware such as Stuxnet [Lan11]. Despite being proprietary, reverse engineering efforts have led to greater understanding of its structure and security challenges.

2.1.3 Modbus TCP

Modbus is one of the most widely used communication protocols in industrial automation. Originally introduced in 1979 for serial communication, Modbus has since

been extended to operate over TCP/IP networks, known as Modbus TCP [Mod06]. It follows a master/slave (client/server) architecture, where the client initiates transactions and the server responds with requested data. Modbus TCP encapsulates traditional Modbus messages within TCP frames, making it suitable for Ethernet-based communication while preserving backward compatibility. Its simplicity and openness have driven adoption across a broad range of devices and vendors, but the lack of built-in authentication and encryption mechanisms also makes it highly vulnerable to cyberattacks in modern networked environments [Cha+20]. Consequently, Modbus TCP often represents both a practical integration solution and a significant security concern in industrial networks.

2.2 Artificial Intelligence Applications in Smart Grids

AI and Machine Learning (ML) have become essential tools in the modernization of smart grids. They provide methods for analyzing vast amounts of heterogeneous data generated by sensors, Intelligent Electronic Device (IED)s, and supervisory systems, thereby enabling advanced functionalities such as anomaly detection, predictive maintenance, and demand-side management. This section provides an overview of key application domains where AI/ML has had significant impact.

2.2.1 Intrusion Detection

Smart grids are increasingly exposed to cyber threats due to their reliance on IP-based communication and interconnected infrastructures. ML-based IDS have been proposed to identify malicious activities such as denial-of-service attacks, false data injection, and unauthorized access. By learning patterns from network traffic and system logs, these models can complement signature-based approaches and detect novel attack vectors. Recent advances include the use of learning-based models for feature extraction [Moh+24] and ensemble methods [Als+25] to improve detection accuracy and reduce false positives.

2.2.2 Power Quality Recognition

Ensuring power quality is critical for both industrial and residential consumers. Voltage sags, harmonics, transients, and other disturbances can degrade equipment

performance and even lead to outages. AI approaches have been applied to automatically recognize and classify such events using data from phasor measurement units (PMUs) and smart meters [Tia+21]. Techniques such as Convolutional Neural Network (CNN) models are capable of identifying complex temporal patterns in waveform data [Tia+21]. These models enable real-time monitoring and can support preventive maintenance strategies.

2.2.3 Interdisciplinary Use Cases

Beyond conventional grid operations, AI/ML applications in smart grids extend to interdisciplinary domains that leverage techniques originally developed for other fields. Three notable examples are natural language processing, computer vision, and recommendation systems.

Natural Language Processing

Natural Language Processing (NLP) methods can be used in the smart grid for tasks such as processing incident reports **[P3]**, extracting knowledge from maintenance logs [Li+19a], and rationalizing alarms [AA24]. By transforming unstructured text into structured information, NLP assists operators in decision-making and supports compliance monitoring [Li+24b; Wu+24; Na+24]. Emerging research also explores conversational agents for human–machine interaction in the energy domain [Cam+25]. Adversarial studies against NLP within the smart grid domain also exist [Don+21].

Computer Vision

Computer vision contributes to condition monitoring and asset management in the grid [Zho+24; Lv+24]. Applications include visual inspection of transmission lines, insulators, and substations using drones or fixed cameras [Nel+24; Li+24c; Li+24a; Yin+24]. Deep convolutional neural networks enable automatic detection of defects such as corrosion, cracks, or vegetation encroachment [Li+24c]. This reduces the need for manual inspection and enhances the safety and reliability of infrastructure maintenance, but they are vulnerable to adversarial attacks [Akh+21].

Recommendation Systems

Recommendation system methodologies are increasingly applied in demand response and consumer engagement [Luo+16]. By analyzing consumption patterns and external factors such as weather forecasts, ML models can suggest optimized schedules for appliance usage or electric vehicle charging [Cao+17]. These systems balance user comfort with grid stability, helping to flatten peak demand and integrate renewable energy sources. Such personalized recommendations contribute to a more resilient and efficient smart grid ecosystem, but they have their own vulnerabilities as well [DNM21].

2.3 Security of Machine Learning

Outside the power systems domain, attacks against learning-based methods is a wide research area. This field began to evolve with the influential work of Dalvi *et al.* [Dal+04] in 2004, which explored methods to circumvent learning-based email spam filters. More recent research has predominantly focused on adversarial perturbations in visual and auditory data, as seen in studies pertaining to image [CW17] and audio [CW18] domains. Additionally, there is a growing interest in other areas, including malware detection [Yan+17], biometric systems [Sha+16], and text classification [Li+19c]. By mapping the attacker's capabilities and goals, we obtain a comprehensive overview of the different attacks against learning-based methods, as depicted in Table 2.1. The table categorizes attacks against learning-based methods in smart grids according to the CIA Triad (Confidentiality, Integrity, Availability) and the AAA Framework (Authentication, Authorization, Accounting), but the scope of the models can be further clarified by connecting these attacks to specific model types and their roles within smart grid systems. For example, supervised learning models like Support Vector Machines (SVMs) or Random Forests are often used for anomaly detection and are particularly vulnerable to poisoning attacks during the training phase, which compromise data integrity. Deep neural networks, commonly employed in predictive maintenance or demand forecasting, are susceptible to sponge attacks, which target availability by exhausting computational resources. Similarly, all models face risks of confidentiality breaches, such as model extraction. The aforementioned attacks, whether targeting test data or training data, exploit model-specific weaknesses and can propagate through interconnected systems, amplifying their impact.

Arp et al. [Arp+22] identify 10 common pitfalls that can lead to over-optimistic results, incurring in a false sense of achievement that hinders the adoption of ML for Computer Security in academia and industry. This situation becomes even more critical due to the fact that processes are often undermined by adversaries that actively aim to bypass analysis and break systems. In spite of these difficulties, Arp et al. [Arp+22] encourage the community to explore the challenges and chances of embedding machine learning in real-world security systems.

One of the main lines of research that relates to the intersection of ML and IT security is the topic of Adversarial ML, as it constitutes a significant threat to the possibility of deploying ML solutions in-the-wild, specially in security-critical scenarios. An adversarial attack consists of carefully crafted inputs that deceive a ML algorithm, causing the model to make mistakes. Pierazzi et al. [Pie+20] address the challenge of modifying real input-space objects into adversarial samples; the main difficulty resides in the inverse feature-mapping problem, because depending on the domain at hand it could be not feasible to convert a feature vector into a problem-space object. When the feature-mapping function is neither invertible nor differentiable, the direct applicability of gradient-driven feature-space attacks to find problem-space adversarial examples is in most cases prevented (or at least complicated). Additionally, it should be robust to preprocessing, that is, remain in a way that reaches the final model. Besides, the modified object needs to be valid and inconspicuous according to the specific domain. As an example, in a problem-space setting such as malware, the search to transform a sample in order to bypass a ML-based detection system cannot be purely gradient-based, because there is a high risk of causing critical modifications that, for instance, would eliminate the malicious capabilities making the piece of malware useless from the attacker's point of view. Pierazzi et al. [Pie+20] consider this context to formalize general problem-space evasion attacks, including the definition of a comprehensive set of constraints on available transformations, preserved semantics, robustness to preprocessing and plausibility. These contributions enable comparison between different approaches and set a baseline for more principled designs in future work.

When it comes to evading or misleading ML algorithms, one option could be to perform a feature-space attack on the classifier, that is, modifying certain values directly in the feature vector. A feature vector is a numerical representation of an object in a dataset; essentially, a list of values where each one corresponds to a feature or attribute of the object. This attack could be done using a variety of techniques (e.g., based on gradients, heuristics, etc.) with the objective of identifying what feature modifications would mislead a classifier. This process produces a mimicry attack on theoretical grounds, but in practice, the most difficult task is to

apply the required modifications to the real-world object. In some scenarios such as detection of spam emails or malicious PDF files, it is straightforward to manipulate the given real-world object to obtain the desired feature vector of the optimal attack [Big+13]. This is done without creating disruption that could compromise the adversarial capabilities (i.e., avoiding a modification of the malicious component that would make it ineffective). However, other complex environments are more demanding. An attacker may address some difficult feature mappings such as *n*-gram features [Fog+06] where the feature mapping is not easily inverted. In these cases, obtaining a real-world object mimicking normal behaviour while producing evasion becomes challenging. The concept of mimicry attacks was first introduced as a new type of attack by Wagner and Dean [WD01], and it is described as an adversarial effort to fool an IDS by camouflaging the malicious code in a way that it behaves much like the application would.

2.3.1 Evaluation Metrics

In the context of AI security for smart grids, several fundamental metrics are used to assess model performance. They are typically derived from the confusion matrix, which summarizes predictions across classes.

Confusion Matrix. For a binary classification task, the confusion matrix is defined as:

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

where TP (true positives), TN (true negatives), FP (false positives), and FN (false negatives) denote the respective counts of predictions.

Accuracy. Accuracy measures the overall proportion of correctly classified instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.1}$$

While widely used, accuracy can be misleading for imbalanced datasets, where one class dominates.

Precision. Precision (or positive predictive value) quantifies how many predicted positives are actually correct:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{2.2}$$

High precision is important to reduce false alarms in intrusion detection.

Recall. Recall (or true positive rate) measures how many actual positives are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{2.3}$$

High recall ensures that malicious events or faults are not overlooked.

F1 Score. The F1 score balances precision and recall through their harmonic mean:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.4}$$

It is particularly suitable in AI security contexts, where both false alarms and missed detections must be minimized.

2.3.2 Threat Models in Smart Grids

Firstly, it is important to understand the existing threat models of adversarial attacks.

Attacker's Goal. In the context of smart grids, comprehending the potential objectives of attackers is critical for ensuring robust system security. The CIA triad, a widely recognized model in information technology security, provides a framework for understanding these objectives. Reflecting on these three security violation grounds: *confidentiality* could be compromised when an attacker maliciously interacts with a learning algorithm with the objective of reverse-engineering it; *integrity* is compromised when performance is impacted without affecting normal operation; compromising *availability* refers to making the normal learning-based system functionalities unavailable to legitimate users.

Attacker's Knowledge. According to the information available to the attacker at the time of inception, there exist three main paradigms. *White box*: In the context of smart grids, this represents an extreme, worst-case scenario, often linked to insider threats. It is critical for testing the resilience of learning-based systems against those

who have full access to their inner workings. *Grey Box*: This paradigm reflects a more common scenario in real-world applications, where some system details might be obtained or publicly known. *Black Box*: Assessing the resilience of smart grids against black box attacks is crucial, as it represents the common challenge of defending against external threats with minimal system information. In smart grids, it is important to investigate all three categories, motivated by the fact that *security by obscurity* is not reasonable in this context, i.e., it is not good practice to expect security by code secrecy.

Attacker's Capability. Depending on the phase that an attacker influences the algorithm (i.e., during training or test time) there are different naming conventions to classify approaches. Identifying available transformations, preserving semantics, ensuring robustness to pre-processing and general plausibility are problem-space constraints [Pie+20] that represent major challenges to attackers. These constraints outline the boundaries within which attackers operate, and highlight the complex nature of securing learning-based systems in the critical infrastructure of smart grids.

2.3.3 Mapping CIA and AAA

By mapping the attacker's capabilities and goals, we observe a comprehensive overview of the different attacks against learning-based methods, as depicted in Table 2.1. Furthermore, the AAA framework addresses the main attributes of policy enforcement and access control to resources. AAA was designed to be applied in network security, but the principles provide a useful lens in the context of data governance within data-driven methods in smart grids. The integration of AAA is also included in Table 2.1.

- *Authentication* in ML security involves ensuring the legitimacy of smart grid data used for training and inference.
- *Authorization* relates to enforcing what data can influence the model and who can access the model's predictions and knowledge.
- *Accounting* involves monitoring and logging model access and usage, which is crucial for detecting and responding to attacks.

From a defense perspective: for attacks that involve poisoning, an authorization step that addresses what data can influence the model would increase robustness. In relation to authentication, data governance measures should ensure that input data

Tab. 2.1: Categorization of attacks against learning-based methods in smart grids according to the CIA Triad and AAA Framework [P1]. Legend: Authentication (Δ), Authorization (Ω), Accounting (Σ).

	Integrity	Availability	Confidentiality
Test Data	Evasion/ Adversarial examples, Test-time Poisoning (Δ)	Sponge Attack (Δ)	Model Extraction and Inversion, Membership Inference (Ω)(Σ)
Training Data	Poisoning (e.g., backdoors or trojans) (Δ) (Ω)	Indiscriminate Poisoning (i.e., DoS), Sponge Poisoning (Δ) (Ω)	Model Inversion with Poisoning (Δ) (Ω) (Σ)

is legitimate with techniques such as validation and pre-processing, what would contribute towards increasing robustness against both evasion and poisoning attacks. Additionally, resource-aware authentication measures would contribute towards ensuring availability of learning-based models. Attacks against data confidentiality should be avoided by authorization policies that control access to the model's predictions and knowledge. Furthermore, these breaches are related to the system's accountability mechanisms; potential deficiencies in tracking and auditing access to and usage of the data facilitate attack success.

However, these measures are not always considered in smart grids and, therefore, adversaries take advantage.

2.3.4 The Inverse Feature Mapping Problem

In many machine learning applications, input data undergoes a series of preprocessing and feature extraction steps before being provided to a model. While this improves performance, it also introduces an important security and privacy challenge: the *inverse feature mapping problem* [Pie+20]. This refers to the task of reconstructing or approximating the original input data from the features or intermediate representations used by the model.

In the context of smart grids and AI security, this problem is particularly relevant. For example, intrusion detection systems or anomaly detectors often operate on engineered features derived from network traffic or power system measurements. If an adversary gains access to these features, the ability to invert them back to raw signals (such as network packets or load profiles) could enable sensitive information leakage, including operational patterns or consumer data. Similarly, in side-channel

analysis or model inversion attacks, partial knowledge of feature representations can be exploited to infer confidential inputs.

The inverse feature mapping problem is therefore not only a theoretical challenge in representation learning, but also a practical concern for privacy preservation and adversarial robustness in critical infrastructures like the smart grid. Understanding when and how features can be inverted provides valuable insight into the trade-off between model utility and data confidentiality, guiding the design of secure feature extraction pipelines and privacy-enhancing mechanisms.

2.4 Explainable Artificial Intelligence

Our work focuses on proactively testing models in different attack scenarios, using XAI methods to support attacker endeavors.

First, we introduce the XAI methods that we will use in the scenario involving neural networks (PQR).

Local Interpretable Model-agnostic Explanations (LIME). LIME [RSG16] is a model-agnostic method that explains predictions by approximating the original model locally with a simpler, interpretable surrogate. Given an input x and a black-box model f, LIME constructs a local model g that mimics f around x. Formally, it solves:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.5}$$

where G is a set of interpretable models, $\mathcal{L}(f, g, \pi_x)$ measures how well g approximates f locally using the proximity kernel π_x, and $\Omega(g)$ penalizes complexity to maintain interpretability.

SHapley Additive exPlanations (SHAP). SHapley Additive exPlanations (SHAP) [LL17] is a model-agnostic approach for explaining the output of any ML model. It connects optimal credit allocation with local explanations using the classical Shapley values from game theory and their related extensions. The SHAP value definition is as follows:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} \left[v(S \cup \{i\}) - v(S) \right] \tag{2.6}$$

where ϕ_i is the SHAP value for feature i, N is the set of all features, S is a subset of N excluding i, and $v(S)$ is the value function (the model's prediction) for the subset S.

Layer-wise Relevance Propagation (LRP). LRP [Mon+19] is a model-specific technique used to explain the predictions of deep neural network models by propagating the output backward through the neural network layers, attributing relevance scores to individual neurons and input features. For a neural network with layers l and neurons i in layer l and j in layer $l+1$, the relevance $R_j^{(l+1)}$ of neuron j in layer $l+1$ is propagated to neuron i in layer l as follows:

$$R_i^{(l)} = \sum_j \frac{a_i w_{ij}}{\sum_k a_k w_{kj}} R_j^{(l+1)} \tag{2.7}$$

where a_i is the activation of neuron i, w_{ij} is the weight connecting neuron i in layer l to neuron j in layer $l+1$, a_k is the activation of neuron k in layer l that connects to neuron j in layer $l+1$, w_{kj} is the weight connecting neuron k in layer l to neuron j in layer $l+1$, and $R_j^{(l+1)}$ is the relevance of neuron j in layer $l+1$.

It is important to note that LRP cannot be applied to decision trees; in addition to LIME and SHAP, we consider another two XAI methods applicable to decision trees. Combining the following two methods ensures a more holistic understanding of feature relevance by balancing internal model mechanics (Gini) with performance-based validation (permutation). This way, we cover both model-agnostic and model-specific XAI methods in uses cases that rely on Random Forest (RF) classifiers.

Permutation Importance. Permutation Importance [Bre01] is a model-agnostic method for assessing the importance of individual features in a predictive model by measuring the impact of permuting the feature values on the model's performance. The intuition behind this method is that shuffling a feature's values should disrupt the model's predictions if the feature is important, while having little to no effect if the feature is irrelevant. Formally, for a model f, a dataset X, and a performance metric M, the permutation importance of feature i is calculated as:

$$\text{Importance}(i) = M(f, X) - M(f, X^{\text{perm}(i)}) \tag{2.8}$$

where $X^{\text{perm}(i)}$ is the dataset X with the values of feature i randomly permuted across samples, disrupting its relationship with the target variable. This method provides a straightforward and intuitive way to rank features based on their contribution to the model's predictions. However, it can be computationally expensive for large datasets and may struggle with collinear features, where the importance of one feature can be distributed among its correlated counterparts.

Gini Importance. Gini Importance, also known as Mean Decrease in Impurity (MDI) [Bre01], is a model-specific method used to quantify feature importance in tree-based models such as RF and Gradient Boosted Trees. This method relies

on the reduction in impurity achieved by each feature during the construction of the decision trees. Formally, for a decision tree, the importance of a feature i is calculated as:

$$\text{Importance}(i) = \sum_{s \in \text{Splits}(i)} \Delta\text{Impurity}(s) \tag{2.9}$$

where Splits(i) is the set of all splits in the tree that involve feature i. ΔImpurity(s) is the reduction in impurity (e.g., Gini Impurity or Entropy) at split s. The Gini Importance for a feature is aggregated across all trees in the ensemble —e.g., RF— by summing its importance scores over all splits in all trees. Gini provides a computationally efficient way to rank features in tree-based models.

Selecting The Top Features. With **SHAP**, to select the top X features, we rank the features by the absolute magnitude of their SHAP values and then select the top X features from this ranking. Let $\{\phi_1, \phi_2, \ldots, \phi_M\}$ be the SHAP values for the M features, the absolute SHAP values are $|\phi_i| \quad$ for $i = 1, 2, \ldots, M$. We sort the features by their absolute SHAP values in descending order. To select the top X features based on their relevance scores computed by **LRP**, we rank the features by their relevance scores and then select the top X features from this ranking. We sort the features by their relevance scores in descending order $R_{(1)} \geq R_{(2)} \geq \ldots \geq R_{(M)}$, and select the top X features corresponding to the largest relevance scores just like for SHAP. With **LIME**, to select the top X features, we rank the features by their aggregated importance scores across instances. The aggregated score for feature i is computed as the sum of the absolute LIME importance scores over all instances. With **Gini**, to select the top X features, we rank the features by their Gini importance scores, which measure the reduction in Gini impurity caused by splits on a feature in a tree-based model. With **Permutation Importance**, to select the top X features, we rank the features by their permutation importance scores, which measure the decrease in model performance when the values of a feature are randomly shuffled.

> Explainability is traditionally framed as a defensive or compliance mechanism. In this thesis, we adopt an adversarial perspective and treat explanation outputs as exploitable signals.

3 Threat Landscape and Related Work

> *A conqueror is always a lover of peace; he would like to make his entry into our state unopposed.*
>
> — **Carl von Clausewitz**
> *On War, 1832*

Security analyses of learning-based smart grid systems cannot be conducted in isolation from the broader threat environment in which these systems operate. This Chapter situates the dissertation within a realistic adversarial context by combining a global analysis of cyber threats to the energy sector with a critical review of existing scientific literature. The first part examines geopolitical trends, incident reports, and threat intelligence to motivate attacker models that are persistent, well-resourced, and adaptive. The second part surveys state-of-the-art research on adversarial attacks against learning-based components in smart grids, including a reproducibility assessment that highlights practical limitations and recurring assumptions. By identifying both threat drivers and research gaps, this Chapter motivates the need for domain-specific, empirically grounded analyses of XAI-enabled adversarial attacks.

3.1 A Global Analysis of Cyber Threats to the Energy Sector: "Currents of Conflict" from a Geopolitical Perspective

In the evolving landscape of cybersecurity, geopolitics plays a defining role in shaping cyber threats, attack motivations, and the strategies employed by both state and non-state actors [Noc18]. The emergence of regional hotspots has led to the formation of cyber alliances, where nations align their cyber defense and offensive capabilities with strategic partners. Countries within alliances such as NATO and BRICS often engage in coordinated cyber operations [Sme19; Bel21; Fra20]. Target selection in cyber conflicts is deeply rooted in geopolitical tensions, with adversarial states

strategically targeting government agencies, critical infrastructure, and economic assets of rival nations. Thus, scientific literature [LGS24] has identified the role of malware as a pivotal geopolitical tool in the twenty-first century's ever-evolving landscape of international relations and cybersecurity. Cyberattacks in general are often used as instruments of power projection, deterrence, and asymmetric warfare [Tod09], allowing smaller nations or non-state actors to challenge more technologically advanced adversaries.

However, there is a lack of systematic analysis to verify this intuition across different sources. Gathering insights from different sources is important to avoid potential bias. Beyond direct cyber conflicts, geopolitical factors also shape national cybersecurity practices. Governments have increasingly adopted cyber resilience policies, enforcing stricter regulations on digital infrastructure and cybersecurity cooperation with allies [Fra20]. Meanwhile, adversarial states have developed sophisticated cyber espionage and disruption campaigns, often backed by state-sponsored threat actors [HAG21]. In this regard, Artificial Intelligence (AI) components are now a well-established part of the cyber toolbox, both for defensive and offensive purposes [TDZ20]. One of the most vulnerable sectors to geopolitical cyber threats is the energy industry [MDB19; Pol24; Whi+17]. As nations seek to control global energy markets, cyberattacks against energy infrastructure have become a critical component of geopolitical maneuvering.

The increasing reliance on digital control systems within the energy sector [Bai+21] has made it a prime target for cyber-physical attacks, raising concerns over the resilience of national energy security in the face of rising geopolitical tensions. However, existing databases and analytical frameworks often fall short in effectively capturing and analyzing the geopolitical nuances and sectoral focus of cyber threats. This Section[1] aims to address these challenges through three key contributions:

- We leverage Generative AI to extract, structure and interpret information from raw cyber threat descriptions for analytical purposes.
- We conduct a geopolitical analysis with emphasis on the origins and target regions of threat actors and cyber incidents, comparing general trends across databases with those specific to the energy sector.
- We assess the effectiveness of firewalls in detecting Indicators of Compromise (IOCs) for attacks targeting the energy sector, with a focus on AI-based detection.

[1] This work was peer-reviewed and published in [P3]

Additionally, we share the data and code used for this study with the research community[2].

Tab. 3.1: Databases used in this work.

Type	Database	Country	Samples	Geographical origins and target regions	Target sectors
Actors	MITRE ATT&CK	USA	163	Often reported in group description text	Often reported in group description text
	ThaiCERT	Thailand	499	Reported explicitly in *JSON* format	Often reported in group description text
	Malpedia	Germany	763	Origin reported explicitly in *JSON* format, target sometimes reported in group description text	Often reported in group description text
Incidents	EuRepoC	Germany	3329	Reported explicitly in tabular form	Reported explicitly in tabular form
	CSIS	USA	580	Reported in incident description text	Reported in incident description text
Reports	AIID	USA	825	Sometimes reported within description	Sometimes reported within description
Malware	Malpedia +VirusTotal API	Germany	3166 families +2400 IOCs	Sometimes reported, e.g., via external URLs	Sometimes reported, e.g., via external URLs

3.1.1 Background

Threat Actors in the Energy Domain. The energy sector faces a unique blend of cyber threats stemming from a wide range of actors, including state-sponsored groups, cybercriminal organizations, and hacktivists [San+24]. State-sponsored threat groups are among the most sophisticated adversaries in the energy domain, often leveraging advanced persistent threats (APTs) to conduct espionage, sabotage, or influence operations [Lu+24]. These groups operate with the backing of national intelligence agencies, targeting energy infrastructure to gather intelligence on resource distribution, energy policies, and technological advancements, or to disrupt an adversary's energy supply as a form of economic warfare. The energy sector was ranked 19th out of 25 sectors in terms of actual victims of ransomware attacks in 2023 [PwCnd]. These groups exploit vulnerabilities in operational technology (OT) and industrial control systems (ICS) to extort payments from energy companies [Ngu+24], with some even selling stolen data on dark web marketplaces [Pan+21b]. Hacktivists and ideologically driven actors also pose a significant threat to the energy sector [San+24], often launching attacks to promote environmental causes, protest against fossil fuel reliance, or disrupt operations of multinational energy corporations. Emerging trends such as AI-driven cyber threats indicate that the energy sector will remain a primary target for cyber adversaries.

Evaluation of AI-Based Detection. AI-based detection systems offer several advantages, including their ability to analyze vast amounts of data in real time, identify

[2]https://github.com/gus5298/SecurityThreatsGeopolitics

patterns of malicious behavior, and adapt to new attack techniques through continuous learning. These technologies have significantly improved the detection of sophisticated cyber threats, particularly in identifying zero-day vulnerabilities [Ahm+23], anomalies [Sin+21], and multi-stage attacks [Jia+23] that traditional signature-based detection methods may miss. However, despite their strengths, AI/ML-based cybersecurity solutions face several limitations and challenges [Arp+22]. One major issue is the susceptibility of AI models to adversarial attacks **[P1]**, where threat actors manipulate input data, e.g., to evade detection **[P7]** by an intrusion detection system (IDS). Additionally, the reliance on historical data for training AI models can lead to biases [And+21], making them less effective against novel attack techniques. The interpretability of AI-driven decisions remains another challenge [Neu+22]. While AI enhances threat detection, human analysts are still essential for contextualizing threats, investigating alerts, and making strategic response decisions. The ongoing development of explainable AI (XAI) [Mou+23] and hybrid AI-human collaboration models aims to address these challenges by improving the transparency and reliability of AI-driven cybersecurity solutions. However, to fully realize the benefits of AI in cybersecurity, more effort is required to mitigate its limitations and enhance its adaptability.

Related Work. Recent research has shed light on the multifaceted challenges of detecting and mitigating cyber threats, with some looking into topics related to geopolitics:

Yuan *et al.* [YAC25] offer a comprehensive threefold contribution. Their measurement study assesses the effectiveness of existing Phishing Website Detection (PWD) systems across diverse regions, revealing limitations in current approaches. Building on this, they propose enhancements to adapt PWD techniques for both Western and Chinese websites, and underscore the urgency of the issue by releasing all associated tools and datasets to stimulate real-world solution development. Skopik *et al.* [Sko+24] take a different angle by presenting a tool that categorizes news items through advanced machine learning algorithms. By extracting and indexing key entities (such as company names, products, CVEs, and attacker groups) their tool groups related news into coherent "stories". This approach not only aids in the rapid identification of emerging trends but also automates report summarization and leverages a collaborative ranking system to prioritize critical information. Turning to smart grid security, Sande-Ríos *et al.* [San+24] provide a detailed analysis of the adversaries targeting these critical infrastructures, making reference to geopolitics. They build an adversarial model that considers the attack surface, adversary motivations, goals, and capabilities. Regarding analysis of IOCs, Van Liebergen *et al.* [Lie+23] analyze the VirusTotal file feed over one year, reviewing 328 million

reports for 235 million samples. Their study reveals that despite a feed volume 17 times lower than that of antivirus telemetry, VirusTotal detects eight times more malware.

While these contributions improve our perception of cyber threat detection and response, several research gaps remain. First, comparing diverse data sources across geopolitical contexts (particularly within the smart grid environment) requires further exploration. Second, enhancing comprehension of AI's contribution and performance metrics is necessary. Lastly, despite advancements in automating categorization and summarization, developing systems capable of integrating heterogeneous data sources into cohesive, actionable insights remains a critical requirement. In the present work, we address these gaps.

3.1.2 Parsing Methodology

Databases

We identify relevant databases from varied origins and in different formats. We introduce these sources in this subsection; an overview can be found in Table 3.1.

(1) MITRE ATT&CK [MITnd]. A globally recognized framework created by the Mitre Corporation (United States) that documents known adversary tactics, techniques and procedures. Their ATT&CK Groups database focuses on cataloging APT groups and cybercriminal organizations, detailing their methods, tools, and motivations. This resource is widely used for threat intelligence and security strategy development. They also provide advanced information on malware families. **(2) ThaiCERT APT Groups [ETDnd].** A database maintained by Thailand's national cybersecurity agency. It includes detailed profiles of APT groups, their campaigns, tools, and tactics, with information on their impact. **(3) Malpedia [Frand].** An open platform hosted by Fraunhofer FKIE (Germany) dedicated to providing detailed information on malware families and their associated threat groups. It offers comprehensive descriptions of threat actors, their campaigns, and the malware they deploy, facilitating cross-referencing between malware and the groups behind them. **(4) EuRepoC [EURnd].** The European Repository of Cyber Incidents (EuRepoC) is an independent research consortium established to enhance understanding of the cyber threat landscape within the European Union and globally. Its primary mission is to promote data-driven discussions and policy making in cybersecurity while raising awareness of cyber threats. EuRepoC provides an analytical framework to assess and compare the lifecycle of cyber incidents, with a focus on technical, political, and

legal dimensions. **(5) CSIS Signifcant Cyber Events List [Cennd].** Based in the United States, the Center for Strategic and International Studies (CSIS) think tank maintains a Significant Cyber Events database, tracking major cybersecurity events worldwide. **(6) AIID reports [Incnd].** The Artificial Intelligence Incident Database (AIID) is a centralized repository that documents real-world incidents involving artificial intelligence systems. It includes cases of AI-related vulnerabilities, misuse, and failures in various domains.

Note. The Chinese CERT [CERnd] releases geographical information in monthly textual summaries, but only distinguishes between "national" and "cross-border" incidents, limiting granularity and therefore is not considered in this study.

Data Processing with Generative AI

Parsing unstructured threat intelligence data requires a robust and scalable approach to extract meaningful insights. In this work, we utilize a Generative AI Model (gemini-1.5-flash-latest) to structure raw cyber incidents text data into structured fields for further analysis, as exemplified in Figure 3.1. The parsing strategy is designed to minimize Large Language Model (LLM) token consumption.

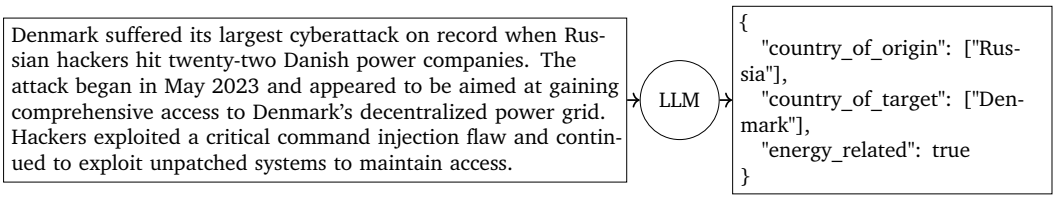


Fig. 3.1: Example of a cyberattack incident description and extracted fields.

As represented in Figure 3.2, the parsing workflow begins with loading the input dataset containing unstructured threat intelligence data (❶). The next step is codifying our target JSON format in a Python *TypedDict*[3], that we name *ThreatParser* (❷). Embedding this schema directly in the system prompt ensures that Gemini outputs precisely the expected keys (and in the correct order), avoiding downstream validation errors. We empirically tuned the model's temperature to 0.1 (balancing variability and determinism) and enforced the target JSON format (❸).

To respect API rate limits and manage token costs, a fixed 7-seconds delay is inserted between calls (❹). All responses undergo a 'try/except' JSON parse: failures (e.g., reaching the API quota limit) are recorded with an error flag (❺). Partial results are written immediately to a JSON file so that parsing can resume after any interruption.

[3] A TypedDict type represents dictionary objects with a specific set of string keys, and with specific value types for each valid key.

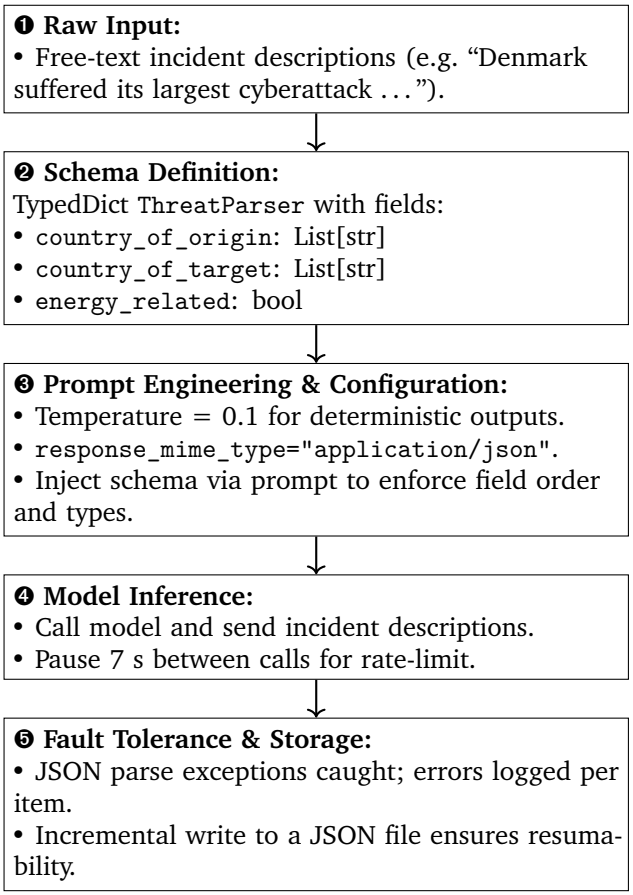


Fig. 3.2: Flow of the generative-AI parsing pipeline, from raw description to structured JSON.

The structured output captures key features, including the attack's origin, target, and domain (i.e., whether it is energy-related or not). Finally, all parsed data is stored in JSON format, preserving structured insights for downstream research and visualization.

Evaluation

To assess the performance of our generative-AI parser in classifying descriptions as energy-related, we created a stratified evaluation set of 200 threat descriptions (100 energy, 100 non-energy) drawn at random from the EuRepoC dataset. Each entry was labeled based on the `receiver category` attribute, and then fed to the same pipeline. The evaluation results yielded an overall accuracy of 84.0%. The following confusion matrix (rows = true class; columns = predicted) provides further details: $\begin{pmatrix} 91 & 9 \\ 23 & 77 \end{pmatrix}$. We observe that 91 true non-energy incidents are correctly classified, 9 non-energy are mislabeled as energy, 23 energy are mislabeled as non-energy, and 77 true energy are correctly identified. From this matrix we compute additional class-specific metrics for the energy class:

$\text{Precision}_{\text{energy}} = \frac{77}{77+9} \times 100\% \approx 89.5\%, \quad \text{Recall}_{\text{energy}} = \frac{77}{77+23} \times 100\% = 77.0\%, \quad F_{1,\text{energy}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \approx 82.7\%.$

Upon inspection, the sentences that are not correctly labeled are (1) those that lack explicit details[4] about the energy targets, and (2) those labeled as "Government", e.g., in the cases where the target is the US Department of Energy. These results demonstrate that our schema-enforced, low-temperature prompting approach achieves strong overall accuracy, high precision on the energy class, and acceptable recall given the challenging, unstructured nature of threat descriptions.

For comparison, the spaCy [Expnd] rule-based baseline (using a `PhraseMatcher` with 16 domain-specific terms[5] such as "energy", "power grid", etc.) reached an overall accuracy of 81%. Its confusion matrix $\begin{pmatrix} 95 & 5 \\ 34 & 66 \end{pmatrix}$, corresponds to a precision of 93% and a recall of 66% for the energy class, giving an F_1 score of 77%.

[4] An example of this is: "Likely Iranian State-sponsored hackers (Crowd strike) have conducted a series of destructive attacks on Saudi Arabia over the last two weeks, erasing data and wreaking havoc in the computerbanks of the agency running the country's airports and hitting five additional targets."

[5] Requires manually gathering keywords via expert knowledge, while our Gen-AI approach only requires to specify the domain itself, e.g.: energy.

In summary, these results demonstrate that our schema-enforced, low-temperature Gen-AI pipeline not only outperforms a rule-based baseline, boosting recall by + 11 percentage points (77% vs. 66%) on energy-related cases, but also maintains comparably high precision, underscoring the practical value of generative AI for accurately extracting nuanced domain signals from unstructured threat descriptions.

Tab. 3.2: Comparison of classification metrics for energy-domain detection.

Metric	Our Gen-AI Parser	spaCy Baseline
Accuracy	84.0%	81.0%
Precision (energy)	89.5%	93.0%
Recall (energy)	77.0%	66.0%
F_1 (energy)	82.7%	77.0%

3.1.3 Geopolitical Big Data Analysis Results

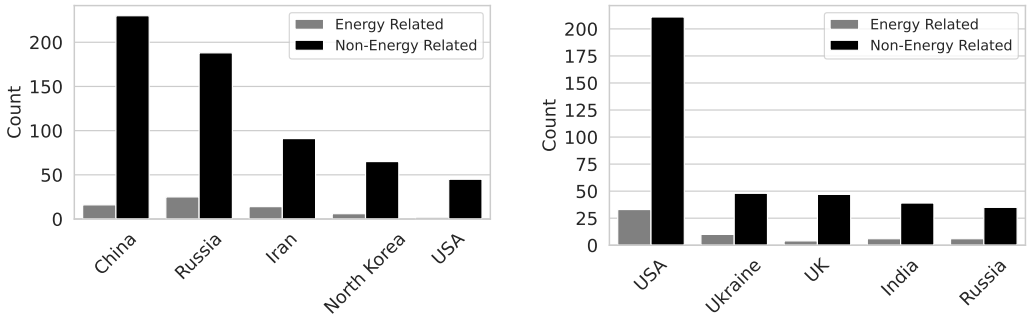
The reporting of geographical origins and target regions varies significantly across different cyber threat databases, reflecting disparities in structure, detail, and accessibility. Table 3.1 summarizes the geographical reporting practices of major cyber threat intelligence resources. Some databases provide structured, machine-readable formats that enable streamlined analysis. For instance, ThaiCERT and Malpedia report geographical origins explicitly in JSON format, facilitating automation and consistency. However, target regions in Malpedia are often embedded within descriptive text, requiring additional processing via the proposed generative AI pipeline. EuRepoC explicitly documents both origins and targets in tabular form, offering a standardized approach allowing comparative analysis. In contrast, databases such as MITRE ATT&CK Groups and CSIS rely on unstructured descriptive text to document geographical information. While rich in qualitative details, these formats necessitate our processing technique to structure the data for analysis. Not all databases include geographical reporting. Notably, AIID omits geographical origins and targets entirely, reducing their utility for geopolitical studies. This omission prevents understanding the spatial distribution of cyber threats. Using our generative AI pipeline, we are still able to extract whether the AI vulnerabilities are related to the energy domain.

Takeaway 1. There is heterogeneity in reporting practices among cyber threat databases. Structured reporting (e.g. JSON), enhances consistency and facilitates cross-database comparisons. The reliance on descriptive text or omission of geographical information hinders the ability to conduct comprehensive geopolitical analyses. Our generative AI pipeline allows for better analysis by parsing relevant unstructured information.

From the provided plots in Figure 3.3, each subfigure shows the top 5 of either threat origins or targets, with bars grouped by energy vs. non-energy related. The three relevant data sources to answer this question (CSIS, Malpedia, and EuRepoC) each provide a slightly different perspective on which countries appear most frequently. In general, the same few countries dominate both energy and general threats (i.e., Russia and China as origins, and the USA as target). However, the bar heights (counts) can differ substantially. For instance, a country might be the top origin overall but drop to a lower position (or even disappear) in the energy-related subset, indicating that some origins are especially active in the general domain but less so in energy. The energy-related bars show a higher concentration in fewer countries (often, one or two countries account for most of the energy-focused incidents) suggesting that certain threat actors specialize in energy infrastructure attacks. By contrast, the general category includes many different targets (government agencies, corporations, etc.). This can make the energy domain appear narrower but more heavily hit by a smaller set of threat actors. The USA appears as a top general target across datasets, while in energy incidents, the Middle East as a region takes the first place by a small difference (according to Malpedia).

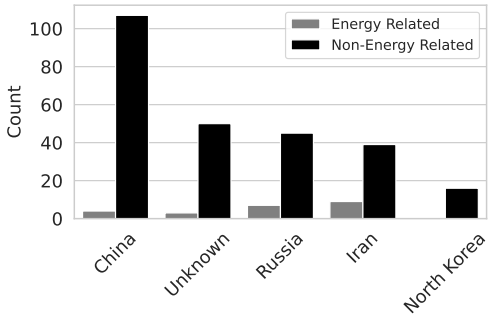
In Figure 3.4, we group each incident's origin (and, analogously, its target) by major geopolitical alliances (namely NATO, BRICS, or "Other") using up-to-date membership rosters as of January 2025. Specifically, our NATO list includes the 32 member states from Albania through the United States (adding Sweden, which joined in July 2023), while the BRICS bloc encompasses not only the original five (Brazil, Russia, India, China, South Africa) but also the ten countries formally participating in the 2024–25 BRICS expansion (Egypt, Ethiopia, Indonesia, Iran, and the United Arab Emirates), providing a relevant, up-to-date overview. Any country not found in either list is classified as "Other".

Concretely, we take each record's country-of-origin field (which may be a list), standardize the country names, and then map each to its alliance via a lookup. We then group by alliance membership; every origin country contributes separately to the aggregate counts, so that each pair (incident, country) becomes its own row. Finally, we assign an alliance label, and plot counts for "Energy Related" versus "Non-Energy Related" in matching bar charts. This clustering by alliance reveals, for example, that BRICS countries dominate non-energy threat origins across all datasets, whereas energy-focused incidents show a comparatively higher concentration in "Other" or NATO members depending on the source. In the Malpedia dataset (Figures 3.4c and 3.4d), we observe the biggest contrast between alliances.

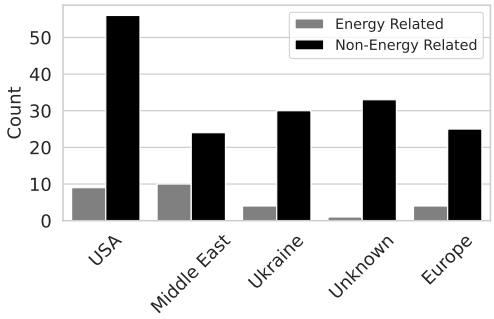


(a) CSIS Origins

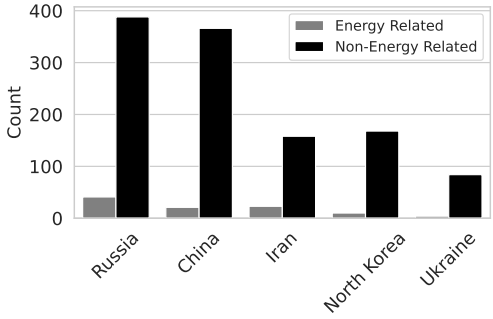
(b) CSIS Targets



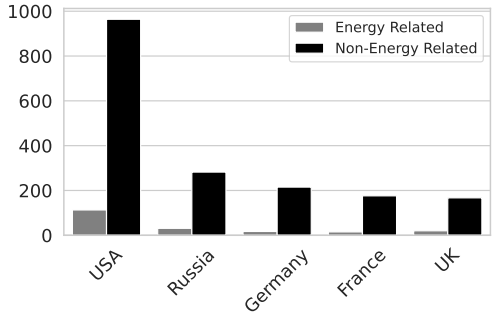
(c) Malpedia Origins



(d) Malpedia Targets



(e) EuRepoC Origins

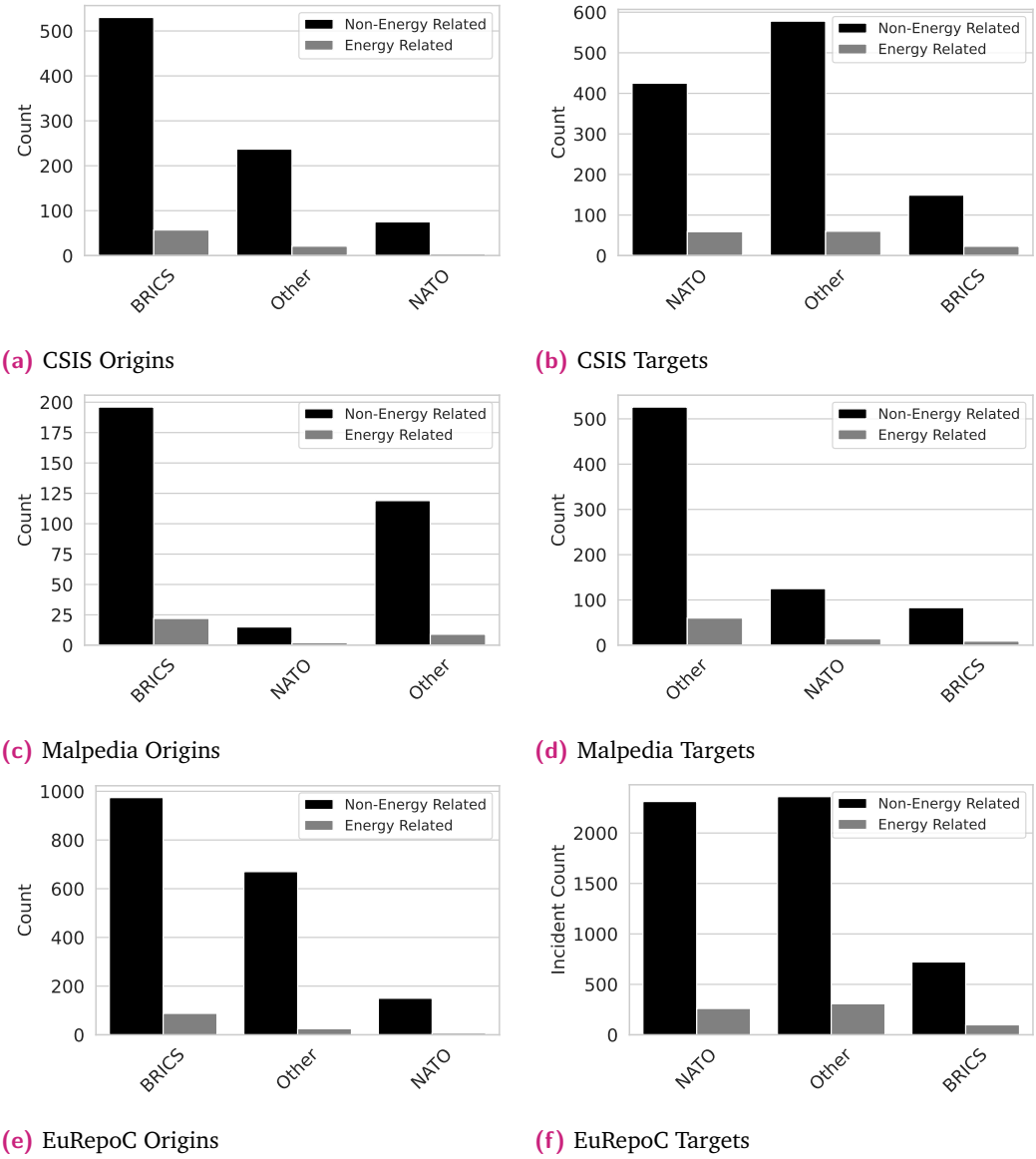


(f) EuRepoC Targets

Fig. 3.3: Top 5 threat origins and targets per dataset.

Takeaway 2. General threats tend to involve a broader set of countries both in terms of origins and targets, whereas energy-related attacks are often more narrowly concentrated. The same few countries dominate the overall ranking (e.g., Russia, China, or the USA) but for energy-focused incidents, certain actors seem to specialize.

From the plots in Figure 3.5, it is clear that cyber incidents in conflict regions are neither uniformly distributed over time nor concentrated in a single geographic theater. Instead, they form distinct clusters that align with specific escalations in each



(a) CSIS Origins

(b) CSIS Targets

(c) Malpedia Origins

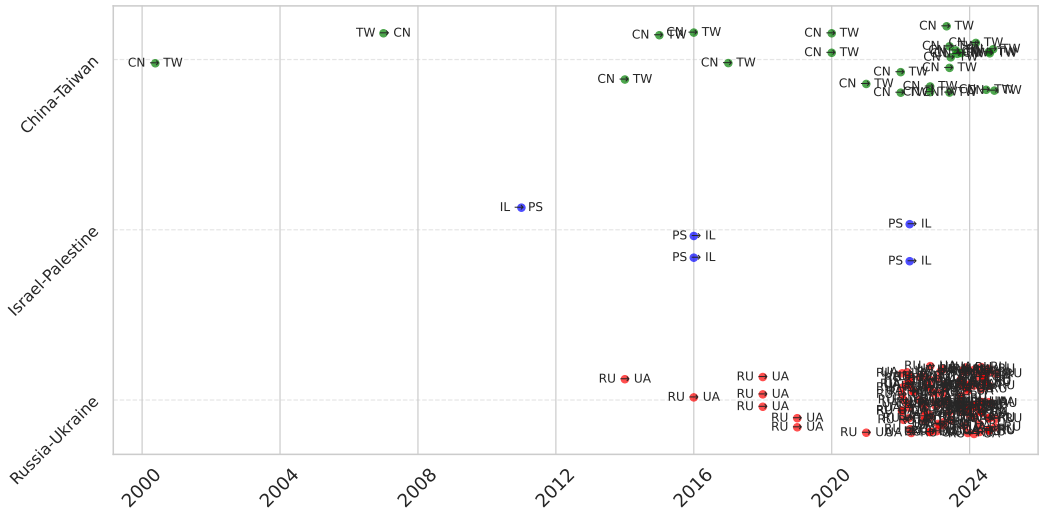
(d) Malpedia Targets

(e) EuRepoC Origins

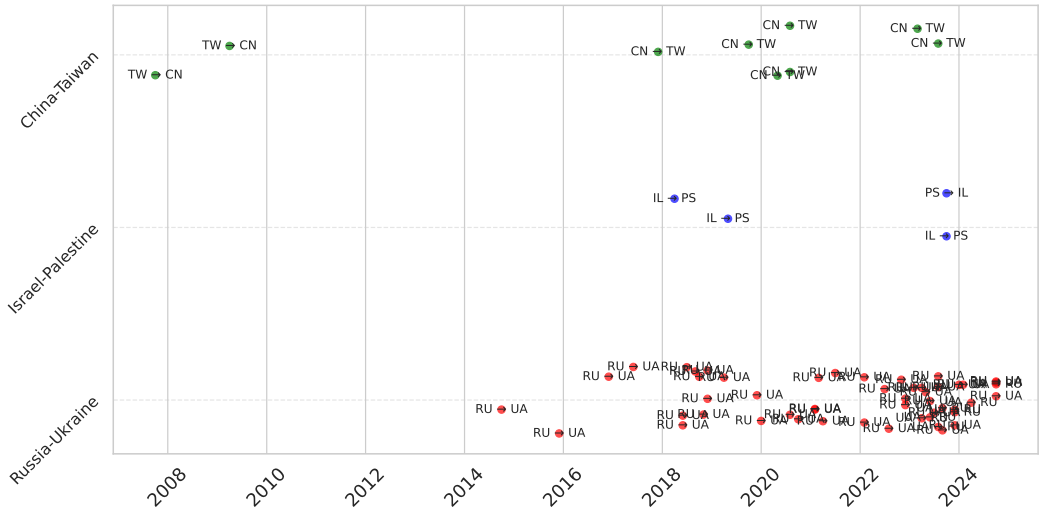
(f) EuRepoC Targets

Fig. 3.4: Top 5 threat origins and targets per dataset, clustered by alliances.

ongoing conflict. In the Russia–Ukraine timeline, a noticeable uptick appears around 2014 (following the annexation of Crimea) and accelerates substantially from 2022 onward. Israel–Palestine incidents show intermittent bursts, fewer overall than Russia–Ukraine, but still recurring during major flare-ups in the region. Meanwhile, the China–Taiwan timeline is comparatively sparser but displays a steady trickle of events spanning multiple years, suggesting a slow-burn pattern of cyber activity. Each conflict region thus exhibits its own rhythm of threat incidents, typically intensifying around key military escalations.



(a) EuRepoC Dataset



(b) CSIS Dataset

	Date	Initiator	Target	Details
CSIS	2015-12	Russia	Ukraine	DoS and SCADA attacks
	2016-12	Russia	Ukraine	Ukrenergo shut down
	2017-06	Russia	Ukraine	Ransomware
	2018-06	Russia	Ukraine	Backdoors
	2020-01	Russia	Ukraine	Infiltration
	2020-05	China	Taiwan	Malware attacks
	2020-10	Russia	Ukraine	Officers indicted (’16 attacks)
	2022-08	Russia	Ukraine	State energy website hack
	2022-11	Russia	Ukraine	Energy and logistics sector hack
	2022-12	Russia	Ukraine	Sandworm APT
	2023-12	Ukraine	Russia	Encryption/deletion of data
EuRepoC	2024-09-19	China	Taiwan	Spear-phishing
	2024-04-19	Russia	Ukraine	Sandworm APT
	2023-06-15	China	Taiwan	CVE-2023-2868
	2023-01-29	Ukraine	Russia	Gazprom hack-and-leak
	2023-01-31	Russia	Ukraine	Sandworm APT
	2022-08-17	Russia	Ukraine	Energoatom website hack
	2022-02-28	Russia	Ukraine	Sandworm APT
	2022-04-12	Russia	Ukraine	Sandworm APT
	2022-04-12	Russia	Ukraine	Sandworm APT

(c) Energy-Related Cyber Incidents in Current Conflicts

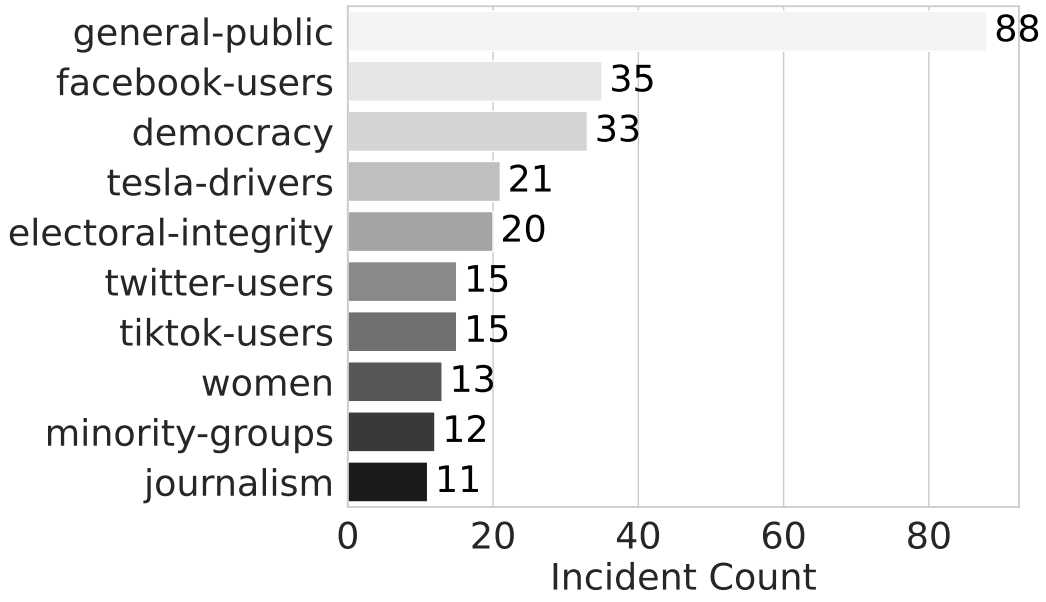
Fig. 3.5: Comparison of conflict chronologies (top row) and energy-related cyber incidents (bottom row).

Takeaway 3. Heightened conflicts correlate with increased frequency of attacks. Russia's persistent cyber campaigns since 2014 reflect ongoing regional disputes. In contrast, the Israel–Palestine conflict shows sharp, periodic spikes in activity that mirror its cyclical hostilities, while China–Taiwan experiences a steady, lower-intensity flow of incidents, suggesting an enduring contest of influence rather than a single, large-scale campaign.

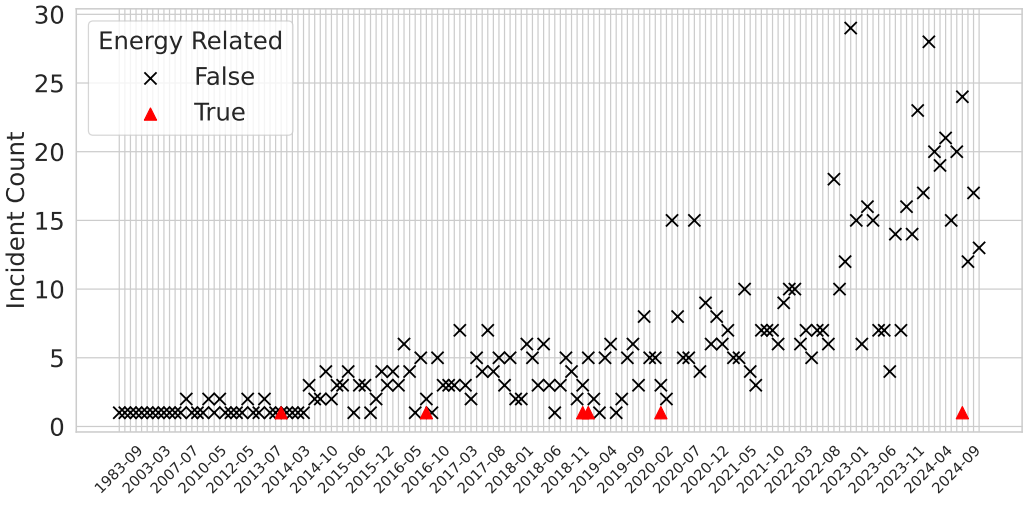
We perform a series of experiments to classify and analyze incidents from the AIID, which contains almost 800 records. The goal is to identify and categorize AI incidents related to specific topics, with a particular focus on the energy domain. Figure 3.6a shows that the top alleged harmed parties in the AIID dataset include a broad range of groups (such as general-public, democracy, or Tesla drivers) while energy- or power-sector victims do not prominently appear among the top ten. It is important to note that, in the case of Tesla cars, AI vulnerabilities lie in the autonomous driving or manufacturing automation processes, and not in the electric vehicle infrastructure. Relative to other domains, explicit AI-based incidents in the energy sector are less common or, at least, less frequently reported. Meanwhile, Figure 3.6b, which plots AI threat incidents over time and highlights energy-related events with red triangles, reveals that such incidents do exist but remain comparatively sparse. To shed light into this, we describe the specific AI and energy related incidents in Table 3.3; in some of these incidents, the energy relation is relatively loose: the generative AI model considers e.g., smart home devices as part of the smart grid, and the climate crisis a subclass of the energy domain. As AI technologies continue to integrate into critical infrastructure, including power grids and smart devices, the potential for energy-targeted attacks will likely grow.

Takeaway 4. These observations imply that AI-related vulnerabilities in the energy domain exist but they appear overshadowed by a wider array of AI threats affecting broader consumer or societal areas.

We analyze the performance of various antivirus engines, with an emphasis on SentinelOne and Acronis, which explicitly report the usage of static machine learning models. We filter out cyber threat groups related to the energy domain. Then, we extract the information about tools that they leverage (e.g., malware families). From the tool names, we fetch IOCs (i.e., file hashes) via Malpedia. These hashes are then submitted to the VirusTotal platform via API to gather detection results from various antivirus engines, both AI-based and traditional. The VirusTotal results were stored locally to avoid redundant queries, preserving API quota for future experiments.



(a) Top 10 Alleged Harmed Parties in the AIID dataset.



(b) AI Threat Incidents Over Time in the AIID dataset.

Fig. 3.6: Analysis of AI Threat Incidents in the AIID dataset.

Tab. 3.3: Most energy-related Cyber Incidents in AIID.

Date	Alleged Developer	Alleged Harmed	Details
2016-10-08	Tesla	Tesla	Poor performance of Tesla factory AI robots (assembling lithium batteries).
2014-01-21	Nest Labs	Fire victims	Smoke + CO alarm could inadvertently silence genuine alarms (smart home).
2019-03-01	Scammers	UK Energy Firm's CEO	Fraudsters used AI to mimic the CEO's voice for social engineering.
2020-04-14	Belgian action group	Belgian government	Deepfake of the Prime Minister urging climate crisis action.
2019-02-01	YouTube	YouTube users	Recommendation algorithm allegedly promoted climate misinformation content.
2024-10-14	Portland Water Bureau	City of Portland	Algorithm reportedly allocated utility bill discounts to high-wealth consumers.

Our analysis shows that only a fraction of IOCs (12.85%) are explicitly tied to energy-focused threats, and that Static ML tools (Acronis and SentinelOne) detect only around 46.8% of malicious indicators, whereas other engines (the *Others* category) detect roughly 88.4%, on par with the overall average (88.4%). This indicates that, for the malware samples used to target the energy domain, traditional solutions are outperforming the ML-based ones.

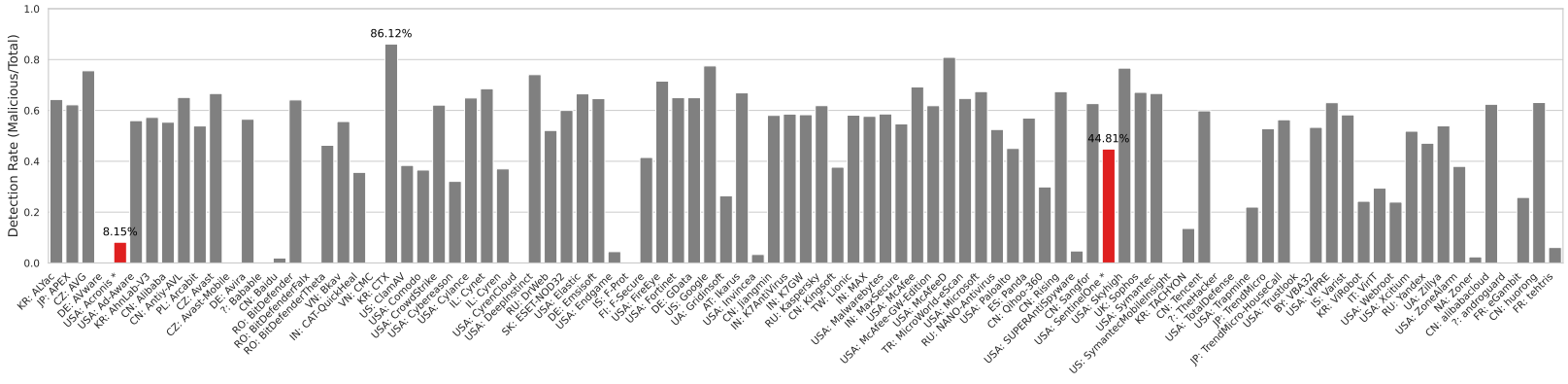


Fig. 3.7: Detection rate by antivirus engine. Static ML engines (Acronis and SentinelOne) are highlighted in red.

Figure 3.7 presents a side-by-side comparison of each antivirus engine's detection rate, with Acronis and SentinelOne (the static ML engines) emphasized in red. The top-performing engine in this dataset achieves a higher detection rate than either of these ML-based solutions.

> **Takeaway 5.** Although ML-based cybersecurity solutions are frequently touted as being especially agile at catching new or sophisticated attacks, the data here suggests they may lag behind more traditional or hybrid approaches when confronting malware associated with the energy domain.

In this Section, we analyze the use of AI (specifically, static machine learning) as a defensive method for detecting IOCs, and generative AI to support natural language tasks. However, the malicious use of AI to enhance cyber operations is an emerging threat (unless done ethically, e.g., for web application penetration testing **[P12]**) with significant geopolitical implications and the potential to impact energy systems. Recent reports [Ope24] have highlighted three cases of AI-related cyber operations involving OpenAI's ChatGPT: **(1) SweetSpecter**: This suspected China-based adversary uses OpenAI's services for reconnaissance, vulnerability research, scripting support, and evasion of anomaly detection, as well as for development. **(2) CyberAv3ngers**: This group, suspected of being affiliated with the Iranian Islamic Revolutionary Guard Corps (IRGC), employs GPT models to conduct research on programmable logic controllers. CyberAv3ngers is known for disruptive attacks on industrial control systems and programmable logic controllers (PLCs) in sectors such as water, manufacturing, and energy. Their targeted infrastructure is typically associated with Israel, the United States, or Ireland. **(3) STORM-0817**: An Iranian threat actor identified as STORM-0817 is developing malware and tools to scrape social media. This actor was found to use OpenAI's models to debug malware, receive coding assistance for building a basic Instagram scraper, and translate LinkedIn profiles into Persian, aimed at identifying potential targets. These three cases —detected based on credible tips— illustrate that while AI offers robust defensive capabilities, adversaries are increasingly harnessing these technologies to advance their offensive operations, raising new challenges for global cybersecurity, especially in critical sectors like energy.

Recommendations. Building on our findings, we propose:

1) **Extension to Other Automation Domains.** To leverage the same generative-AI pipeline to ingest and normalize data from multiple automation sectors. This is applicable to actors, incidents, reports and malware descriptions. While our primary focus is the energy sector, the generative-AI pipeline and subsequent

geopolitical clustering extend readily to other automation environments (such as process industry, automotive manufacturing, or water treatment). This can be easily achieved by swapping the energy-specific keyword for a domain-appropriate one, e.g., using *automotive-related* instead of *energy-related* in the schema definition during prompt engineering (see Figure 3.2). This translates into a simple code update, demonstrating that our method can provide actionable risk insights across diverse automation sectors with minimal reconfiguration. The possibility of summarizing results for different domains into an "Automation Threat Dashboard" for real-time comparative risk assessment is especially interesting.

2) **Dynamic Adversarial Augmentation.** Generating key synthetic examples of adversarially-crafted threat descriptions and including them in the prompts to harden the parser against emerging jargon or evasion tactics.

3.1.4 Summary and Relevance to Explainable Artificial Intelligence

This Section describes different threat intelligence databases and introduces a methodology for parsing their unstructured content using generative AI to enable large-scale analysis. Our findings indicate that cyber threats targeting the energy sector follow distinct patterns compared to general cyber threats. Specialized and often state-aligned threat actors frequently emerge as primary sources of energy-focused incidents, with activity peaks closely aligned with geopolitical tensions and strategic conflicts. Notably, AI-based detection tools (particularly static machine learning solutions) do not consistently outperform traditional antivirus engines in detecting malware observed in the energy domain, suggesting that learning-based defenses introduce complexity without necessarily guaranteeing improved resilience.

Beyond detection performance, our analysis reveals that AI components simultaneously function as both attack vectors and attack surfaces. As learning-based systems become integrated into operational workflows and regulatory processes, adversaries gain incentives to study and exploit their internal behavior. In this context, XAI becomes particularly relevant: explanation mechanisms expose structured information about model reasoning, feature relevance, and decision boundaries, potentially reducing attacker uncertainty. When combined with persistent and well-resourced threat actors identified in the geopolitical analysis, such transparency may unintentionally facilitate more efficient adversarial strategies. This risk becomes particularly relevant when AI model explainability is mandated as a regulatory compliance requirement across different geographical jurisdictions.

These observations motivate the central hypothesis of this thesis: in adversarial environments such as smart grids, explainability can enable XAI-in-the-loop attacks, where explanations are actively leveraged to guide evasion, poisoning, and model extraction processes. The geopolitical threat landscape therefore provides not only contextual motivation but also empirical justification for investigating explainability as a security-relevant component rather than solely a trust-enhancing mechanism. Consequently, improved geopolitical awareness, domain-specific threat intelligence, and security-aware design of explainability mechanisms become essential to mitigate bias exploitation, adversarial manipulation, and information leakage in learning-based critical infrastructure systems.

3.2 Related Work: State-of-the-Art in Smart Grids

In this Section[6], our goal is to enhance the community's comprehension of the challenges, research gaps, and future directions concerning adversarial attacks, specifically those applicable to smart grids. The main contributions can be summarized as follows.

- We categorize existing work on adversarial attacks against learning-based components in smart grids (34 papers), and conduct a detailed assessment of the reproducibility of results.
- We show a correlation between the Confidentiality, Integrity and Availability (CIA) Triad and the Authentication, Authorization, and Accounting (AAA) security framework.

We present our survey, with a focus on works that made their data available. The complete analysis of the state-of-the-art of scientific research in adversarial attacks within smart grids can be consulted in Table 3.4. The reproducibility factors are adapted from [Ols+23].

Reproducibility of Results. We attempted to evaluate the consistency of the claims made in the state-of-the-art by reproducing their results. In the repository provided in [CTZ19] there are files missing, what prevented us from checking the validity of their results (e.g., *data_all.csv* in [CTZ19]). In [MH20] there are missing modules (e.g., *cleverhans_copy.utils*), as well as missing instructions on the execution order and dissimilarity between files (e.g., *attacks.py* and *Attacks.py*). Code instructions are present for [Tia+22] and **[P7]** (while partially described in

[6]Part of this work was peer-reviewed and published in [P1]

Tab. 3.4: Survey of existing scientific work prior to 2024. Legend: ●Fully met, ◐Partially, ○Missing, - Not Applicable.

Ref	Reproducibility										Attack			Other		
	Target Model	Hyper-parameters	Training Info	Dataset Available	Data Split Info	Source Code	Code Instructions	Adv. Attack Code	Code Works	Claims Consistent	Integrity	Availability	Confidentiality	Use of XAI	White/Black/Grey	Publication Year
[LS19]	●	○	◐	○	○	○	-	◐	-	-	●	○	○	○	W	'19
[CTZ19]	●	●	●	●	●	●	◐	●	○	-	●	○	●	○	WB	'19
[Zho+19]	●	●	●	○	●	○	-	◐	-	-	●	○	○	○	W	'19
[SZK20]	●	◐	○	●	○	○	-	-	-	-	●	○	○	○	W	'20
[MH20]	●	●	●	●	●	●	○	●	○	-	●	○	●	○	WB	'20
[Don+21]	●	●	●	●	-	○	-	◐	-	-	●	○	○	○	G	'21
[Tak+20]	●	●	●	●	●	○	-	○	-	-	○	●	○	○	B	'21
[TIS21]	●	●	●	●	●	○	-	○	-	-	●	○	○	○	W	'21
[Son+21]	●	●	●	●	●	○	-	○	-	-	●	○	○	○	WB	'21
[Ham+21]	●	◐	◐	○	◐	○	-	○	-	-	●	○	○	○	WB	'21
[San+21]	●	●	◐	○	○	○	-	○	-	-	●	○	○	○	G	'21
[GS21]	●	●	●	◐	●	○	-	◐	-	-	●	○	○	○	WB	'21
[Wan+21]	●	●	●	○	●	○	-	◐	-	-	●	○	○	○	W	'21
[Pan+21a]	●	●	●	●	●	○	-	○	-	-	●	○	○	○	WB	'21
[Ren+21]	●	●	●	○	○	○	-	◐	-	-	●	○	○	○	WB	'22
[Tia+21]	●	●	●	●	●	●	○	●	○	-	●	○	●	○	B	'22
[BIA22]	●	●	●	●	●	○	-	◐	-	-	●	○	○	○	WB	'22
[ZS22]	●	●	●	◐	●	○	-	◐	-	-	●	○	●	○	WB	'22
[Tia+22]	●	●	●	●	◐	●	●	◐	○	-	●	○	○	○	G	'22
[TIS22]	●	●	●	●	●	○	-	○	-	-	●	○	○	○	WBG	'22
[SW22]	●	◐	◐	○	○	○	-	○	-	-	●	○	○	○	B	'22
[ZQS22]	●	●	●	●	●	○	-	○	-	-	●	○	○	○	B	'22
[Gun+22]	●	○	●	●	●	●	●	○	○	-	●	○	○	○	W	'22
[Bon+23]	●	●	●	●	○	○	-	○	-	-	●	○	○	○	W	'23
[El-+23]	●	●	●	◐	○	○	-	◐	-	-	●	○	○	○	B	'23
[Tak+23]	●	●	●	○	●	○	-	○	-	-	○	●	○	○	B	'23
[Ard+23]	●	◐	●	○	○	○	-	○	-	-	●	○	○	○	W	'23
[HL23]	●	●	●	○	○	○	-	◐	-	-	●	○	○	○	G	'23
[Zhu+23]	●	●	●	●	○	○	-	◐	-	-	●	○	○	○	G	'23
[AA23]	●	◐	●	◐	●	○	-	◐	-	-	●	○	○	○	B	'23
[Naz+23]	●	○	●	○	○	○	-	○	-	-	●	○	○	○	W	'23
[P7]	●	●	●	●	●	●	●	●	●	●	●	○	○	●	W	'23
[WP23]	●	●	●	○	○	○	-	○	-	-	●	○	○	○	G	'23
[Zha+23]	●	●	●	●	○	○	-	○	-	-	●	○	○	○	B	'24

[CTZ19]). Additionally, errors occur when trying to import a component from a library that has been relocated or renamed in a newer version; this would be solved by specifying library version requirements. Adversarial attack code is provided in [CTZ19; MH20; Tia+21] and **[P7]**. Works [LS19; Ren+21; Zho+19; Don+21; GS21; Ren+21; BIA22; ZS22; Tia+22; El-+23; HL23; Zhu+23; AA23] provide pseudo-code of their proposed attack(s) but not an implementation. In the studies [San+21] and **[P7]**, it is mentioned that the code and artifacts would be released upon request. Upon obtaining access to the resources from **[P7]** and subsequent execution, it was observed that their claims were consistent. On the contrary, we contacted the corresponding author of [San+21], who was unable to make data available.

Authors in [Tia+22] do not provide in-depth information about target models due to relying on previous work; we consider this practice acceptable as far as the referenced work is reproducible. In the case of [Tia+22], it was possible to execute the model used [ZWG19] as a target for adversarial attacks only after several code modifications for compatibility (*keras*, *tensorflow* and *matplotlib* scripts), as the library versions used were not specified. Due to the library issues, results were slightly different although the same random seed was used. In [Tia+21], there are files missing (e.g., *20191014universal_pert_5000_1_2.0.npy* and *confusion_matrix_SAA.npy*), and dependencies issues (e.g., *cannot import name 'cast' from partially initialized module 'keras.src.backend'*). In [Gun+22], after following the specified execution instructions by their cited source [Agw+21], pulling data from the cloud failed.

Several papers [CTZ19; MH20; Tak+20; TIS21; Son+21; Tia+21; BIA22; TIS22] and **[P7]** provide comprehensive details about the training stage of learning-based methods. On the other hand, papers [LS19; Naz+23; Gun+22] do not provide information about training hyperparameters. Papers [SZK20; Ham+21; Ard+23; AA23; SW22] provide partial hyperparameter information.

Attack Types. We observed that most published papers investigate attacks compromising the integrity of models, revealing a potential research gap in studying adversarial efforts against the availability and confidentiality of learning-based models.

Authors in [Tak+20] and [Tak+23] deal with attacks against availability. In [Tak+20], it is a reported an analysis on the impact of indiscriminate data poisoning attacks, and how to detect them within the topic of electricity theft. By significantly lowering the accuracy and reliability of the learning models, these attacks have the potential to render the system ineffective for its intended purpose, essentially making the service unavailable or less available to its users. In [Tak+23], we observe a similar

approach, where authors investigate indiscriminate poisoning through different injection levels in the context of FDI detection. Works [CTZ19; MH20; Tia+21; ZS22] compromise confidentiality via the use of surrogate models. In these cases, the objective is to enhance integrity attacks by testing more realistic attacks scenarios (i.e., that do not require access to the inner workings of the models). This is achieved via targeting a substitute model, and then transferring the attacks. However, these approaches assume that attackers have access to data for training testbed models similar to the real targets.

In terms of the amount of information accessible to the attackers, authors in [CTZ19; MH20; Son+21; Ham+21; GS21; Ren+21; BIA22; ZS22; Pan+21a] provide threat models based on both white and black box scenarios; most notably, [TIS22] present white, black and grey box attack approaches. In the rest of the papers, the focus is solely on a single scenario, i.e., either white (10), black (8) or grey (6) box.

Learning-based methods. The vast majority of papers leverage ML and Deep Learning (DL) algorithms, and use tabular data for training. We encountered a number of papers investigating adversarial attacks against Reinforcement Learning (RL) models in smart grid use cases [Wan+21; Pan+21a; SW22; ZQS22; Gun+22; Agw+21; WP23; El-+23]. In RL, an agent learns to make decisions by interacting with an environment to achieve a goal; in Table 3.4, for RL papers the column *Dataset Available* relates to the availability of this whole environment. Furthermore, in [Zha+23], authors present an investigation that focuses on attacks against computer vision applied to smart grids, such as object recognition and defect detection tasks. Another outlier is presented in [Don+21], where authors employ NLP and describe a sentence-level text adversarial attack algorithm, evaluated in the context of a smart grid based on industrial Internet-of-Things.

Other Findings. Only one of the surveyed papers consider XAI in their methodology. Authors in **[P7]** leverage XAI to identify the two most relevant features used by a learning-based Intrusion Detection System (IDS). By focusing their adversarial efforts on adding perturbations exclusively to these features, they attempt to optimize their evasion capabilities against the IDS.

3.2.1 Reproducibility Analysis

We found that only 6 out of the 34 papers surveyed provided source code, constituting approximately 18% of the publications; this is approximately half compared to the baseline [Ols+23], which measures reproducibility of general ML papers in

Tier 1 security conferences, where authors found that 39% of the papers provided code. A lesson we learn from this: not sharing the code used in experiments makes it harder for future researchers to build upon or compare their work with published techniques. Moreover, starting from scratch to develop complex pre-processing tasks, new analytical methods, and sophisticated system designs is a challenging task. For instance, we managed to run further experiments on the code provided by **[P7]**, adding to their investigation. Therefore, making code available not only enhances reproducibility but also supports further innovation and development. From the source codes provided, only one worked out-of-box. This problem also exists in the top tier ML security literature [Ols+23], where 82% of the papers with code required installing further packages, changing paths or directory structures, or fixing errors that appear. In our survey, we identified a lack of code instructions (e.g., a detailed ReadMe file) in approximately 40% of the papers with source code, increasing the difficulty of executing the implementations. Furthermore, 40% of the source codes did not include the corresponding adversarial attack code, being in most cases only reported via pseudo-code as a paper figure.

3.2.2 Confidentiality and Availability

In our survey, we observe a major interest in attacks compromising the integrity of learning-based models. Confidentiality and availability attacks are underrepresented in this body of work, but are equally —or even more— important in the context of critical infrastructure. As a potential reason behind the focus on integrity, particularly through adversarial examples, we consider the existence of immediate and tangible consequences for performance and safety. The initial discovery and exploration of adversarial examples (e.g., in general computer vision) and their effects on integrity in real world scenarios (e.g., in autonomous driving) opened up a new research frontier. As scholars dove into the complexities of these vulnerabilities, a momentum built up around this line of inquiry, leading to a concentration of effort and resources in understanding and mitigating integrity attacks across many different domains, including smart grid security. In 2018, Biggio and Roli [BR18] published a categorization of a decades worth of research in attacks against ML, where one can notice that availability attacks via test data, and confidentiality attacks via training data were still undiscovered.

Nonetheless, the rapid proliferation of new approaches against confidentiality and availability observed in the broad artificial intelligence community will undoubtedly affect the future of attacks tailored to learning models in smart grids. The arms

race between developing more sophisticated attacks and defenses contributes to this dynamic and rapidly evolving field of study.

3.2.3 Focus on Electrical Substations

We observe a lack of investigations in the subtopic of adversarial attacks against electrical substation-related learning-based methods. Electrical substations are pivotal components of the power grid, acting as nodes where transmission lines are connected, transformed, and distributed to various consumers. As part of a nation's critical infrastructure, ensuring their security is vital to prevent disruptions that could have wide-ranging consequences on other sectors such as healthcare, finance, transportation, and water supply.

In consequence, we propose to explore the identified research challenges within the KASTEL Security lab [KAS]. Our experimental environment consists of three key subsystems [Mum+23]: a microgrid, a transmission/distribution substation, and a Software-Defined Network. The transmission/distribution substation is structured into three layers: the station level, equipped with a substation automation system and a human-machine interface for overarching control and surveillance; the bay level, which includes devices for control and protection; and the process level, designed with a test set that simulates the actual physical processes. Both physical and virtual elements are integrated into these subsystems.

3.2.4 Challenges

To the best of our knowledge, research on these scenarios within this contextual framework has not been conducted despite their high potential and importance:

Evasion in the Problem Space. Developing realistic and practical implementations of proof-of-concepts for adversarial examples in intrusion detection use cases related to electrical substations. Specifically, understanding how evasion attacks targeting the problem space of electrical substation-related systems compromise their operational integrity, and what advanced mitigation strategies can be developed to safeguard learning-based components.

Poisoning Training Data. When developers build datasets for training, it is in their best interest to avoid miss-labeling (if supervised) and/or pollution of the normal profile (if unsupervised). Additionally, a malicious actor could purposely inject malicious data to compromise the model's performance. The goal here is to

maximize classification error[7] by injecting poisoning samples into the training set. Furthermore, these malicious data points can be tailored to make the model overfit with the objective of facilitating model inversion.

Sponge Attacks. In smart grids, the availability of systems is the most critical security aspect. Most models that use learning to detect events and control systems are not installed directly on the components found in substations. Instead, field data is gathered and consolidated at a utility data center, where there are sufficient computing resources to process the data, train models, and run applications that make use of these models. Nevertheless, electrical substations can be considered resource-constrained. Future research is warranted to increase our understanding on attacker capabilities when it comes to compromising the availability of learning-based models via attacks that soak up resources.

Model Extraction. If an attacker obtains data used to train a given model, it would be possible to train surrogate models known to be highly similar to the original. This situation allows an attacker to generate transferable adversarial attacks. Further investigation is needed to better understand the impact of model extraction attacks against learning-based components using data from electrical substations, and what are the potential risks to the confidentiality of sensitive information contained in proprietary models.

The objective of proactively testing learning-based models is to eventually increase resilience against these attacks in smart grids . We envision the application of the following techniques to further protect electrical substations:

Adversarial Training. Incorporating adversarially generated examples into the training phase of models helps them recognize and counteract sophisticated attack patterns. By exposing the model to these malicious inputs during training, it becomes better equipped to identify similar threats during operational use, enhancing its defense capabilities against real-world adversarial attacks in the smart grid environment.

Feature Removal. XAI methods can identify vulnerable features in smart grid datasets, which could be strategically removed or altered to strengthen the model's security (i.e., an adversary-aware feature selection step). This might result in a performance trade-off, but enhancing security in critical smart grid operations could outweigh the loss in precision, especially in high-stakes scenarios like grid stability and outage prevention.

[7] The attacker's goal might extend beyond causing misclassification and include any type of misprediction. This is important because decision making and control applications mostly rely on regression models rather than classification models.

Expert Knowledge. Each use case exhibits potential defense directions that are tightly related to a given subdomain. A throughout understanding of subdomain technicalities (e.g., IEC 61850 communication protocols) is critical to understand vulnerabilities prone to be exploited by adversaries.

3.2.5 Summary

This Section categorizes and evaluates work on attacks against learning-based models in smart grids, with a focus on reproducibility and XAI usage. We elaborate on the reasons behind the current limitations and challenges, with the objective of providing further insights to fill the identified research gaps. Our findings serve as a roadmap for the research community to develop stronger and more secure learning-based systems in smart grids, particularly in use cases related to the energy domain.

3.3 Explainable Artificial Intelligence for Offensive Purposes

Previous work has explored adversarial attacks based on importance scores [Li+19b], and the use of counterfactuals [JSJ23]. However, they state several limitations and future work recommendations, which we take on in this work; previous papers use arbitrary XAI methods without systematically comparing which of the available options would perform best for their use case. To avoid arbitrariness, in this thesis, we systematically investigate the capabilities of XAI-powered data poisoning, by comparing different XAI paradigms and their combination for malicious poisoning purposes. The idea is that comparing and combining XAI methods, whether they aim to explain overall model behavior (global) or individual predictions (local), and whether they are designed to work across models (model agnostic) or tailored to specific architectures (model specific), can provide deeper insights into their respective strengths and limitations across different scenarios, as they can be accordingly mapped to black- or white-box threat models. Furthermore, we define the problem space of Global Navigation Satellite System (GNSS) time spoofing based on existing literature [Can+24], demonstrating for the first time a realistic end-to-end XAI-in-the-loop data poisoning scenario in smart grids. Our experiments are taxonomized according to the so-called CIA triad, relating to each principle (Confidentiality,

Integrity and Availability), showcasing a holistic analysis of XAI-guided data poisoning. To put in a concise form, in comparison to existing work we present the following novel contributions: (1) in contrast to [Oka+24; Oka+25] which focus on individual methods, we evaluate the impact of combining different XAI methods for feature relevance identification; (2) [KL21; Paw+24] use counterfactual (CF) explanations (i.e., Latent CF [Bal+20], Permute attack [HF20] and DiCE [MST20]), while we focus on feature relevance methods under the hypothesis that it will help identifying spurious correlations [Arp+22] and facilitate targeted attacks; (3) While we investigate indiscriminate poisoning similarly as in [KL21] (availability compromise), we focus on targeted poisoning (integrity compromise) in more complex scenarios (multiclass classification), and [Oka+24; Oka+25] focus on the effect of evasion attacks at test time, instead of data poisoning at training time; (4) We extend the use of XAI methods to identify key features not only in tabular data, but also in computer vision pipelines within smart grids; (5) contrary to the related work [KL21; Oka+24; Oka+25], the general trends identified in ML [Ols+23] and energy-related AI security literature [P1], we open source artifacts to the research community.

4 Testbed

Evaluating adversarial attacks against learning-based smart grid systems requires experimental environments that faithfully capture protocol semantics, operational constraints, and realistic data distributions. This Chapter presents the testbeds and models used throughout the dissertation to conduct controlled yet representative experiments. The primary environment is the KASTEL Security Lab Energy, which enables protocol-aware experimentation on industrial communication standards. In addition, selected open-source datasets and models are employed to broaden the scope of evaluation. The Chapter introduces the intrusion detection systems, power quality recognition models, and route choice prediction models that serve as targets in subsequent attack Chapters, thereby providing a transparent and reproducible foundation for all empirical results.

4.1 KASTEL Security Lab Energy

The lab has three subsystems[1]: Subsystem 1 for distributed generation, Subsystem 2 for a digital substation based on IEC 61850, and a SDN Subsystem connecting the two.

- **Subsystem 1**, shown in Figure 4.1, represents a distributed generation environment that has a homogeneous structure mainly consisting of Siemens devices. It uses a Hardware-in-the-Loop (HIL) approach with simulated power plants.
- **Subsystem 2**, shown in Figure 4.2, represents a digital substation that has a heterogeneous structure consisting of components from different manufacturers. Here, we can conduct research on the interoperability of these components based on IEC 61850.
- **SDN Subsystem**, connects Subsystems 1 and 2, where the control center in Subsystem 1 is the remote SCADA for Subsystem 2.

[1]The work in this section and more insights are peer-reviewed and published in [P8]

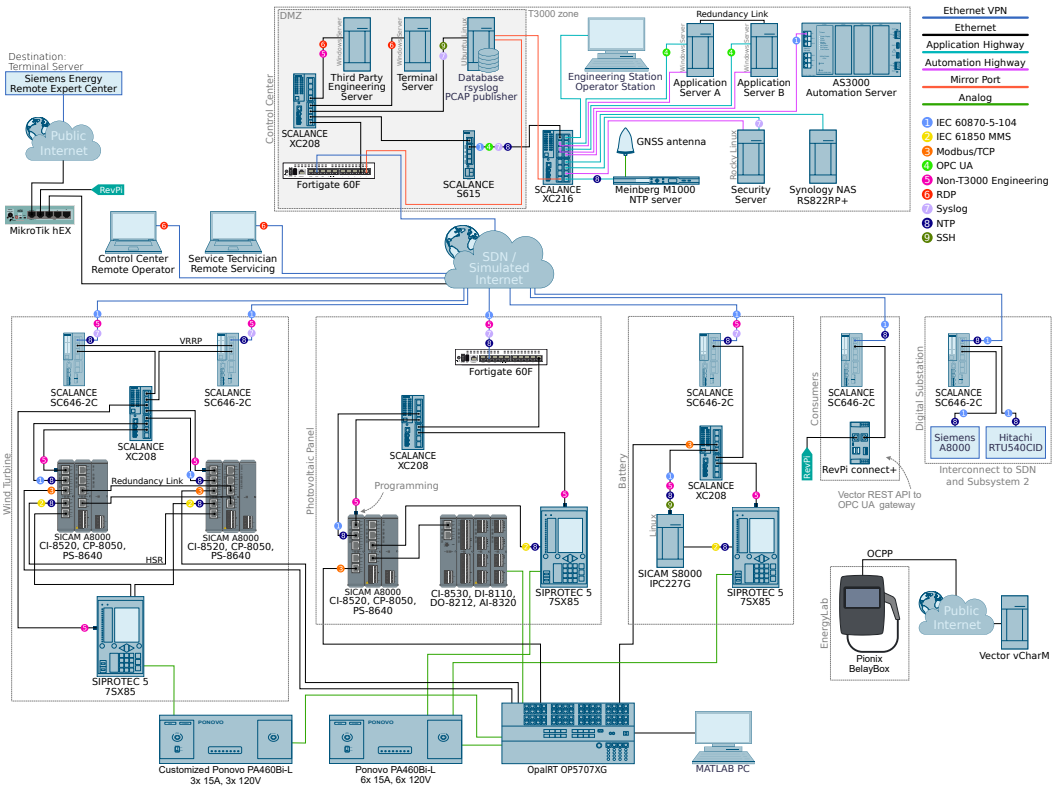


Fig. 4.1: Overview of Subsystem 1 with connection to Subsystem 2 and SDN **[P8]**.

Our lab focuses on the secondary side of the grid. Thus, Subsystem 1 contains simplified models for simulations. Power-plant manufacturers typically rely on subcontractors for Packed Units (PUs), which can be accessed remotely, posing potential vulnerabilities. We simulate this remote access in our Lab but exclude PUs and subcontractor access. For cost efficiency, Subsystem 1 includes only one protection relay per power plant, unlike a real power plant, which would feature multiple relays for both the generator and transformer station. In Subsystem 1, protection relays connect to analog amplifiers, while in Subsystem 2, Merging Units (MUs) convert analog values to Sampled Values (SV) packets for the relays, allowing us to explore both operational modes in the lab.

Uniqueness of Subsystem 1. The testbed features three power plants seamlessly integrated with the T3000 SCADA system over IPsec. Each plant employs a distinct type of Remote Terminal Unit (RTU), providing a realistic control center. Analog signals are generated by a real-time simulator, enhancing the authenticity of the lab. Additionally, the network is organized into VLANs, ensuring adherence to security guidelines and further emphasizing the subsystem uniqueness.

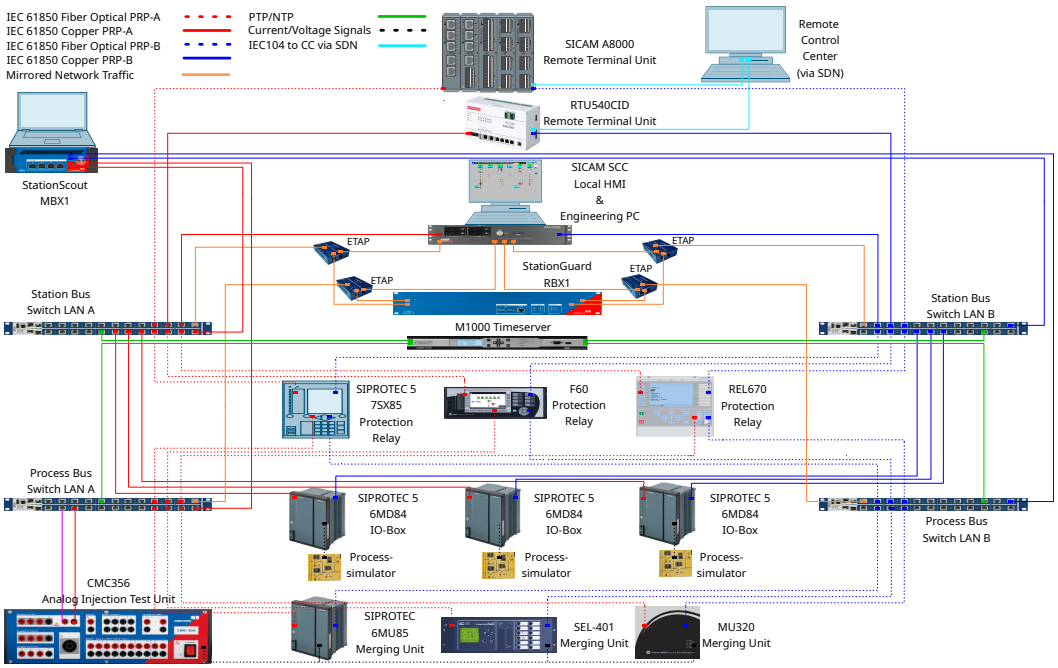


Fig. 4.2: Overview of Subsystem 2 — a digital substation.

Uniqueness of Subsystem 2. The lab is unique in its complexity due to a multivendor setup using real-world equipment and comprehensive integration of IEC 61850 standard. This enables interoperability investigations and the study of digital substation cybersecurity in a holistic network protocol environment.

Discussion. Our lab consists of three subsystems that balance real-setup with research flexibility, utilizing a multi-vendor HIL approach. Its primary advantage is its broad research scope; unlike many testbeds that focus on specific aspects (see Table 4.1), ours examines interactions among transmission, distribution, and communication networks. This enables us to assess the impacts of attacks on one or several subsystems, thereby investigating emergent effects within the increasingly complex and heterogeneous energy grid, for cybersecurity studies. However, our Lab presents certain limitations that we consider relevant to share with the research community: Its construction and maintenance are resource-intensive, compounded by the rapid evolution of cybersecurity requirements, requiring continuous updates. To face these challenges, our adaptable research roadmap responds to such changes, and our interdisciplinary approach fosters large-scale collaboration among projects to optimize resource sharing. Furthermore, navigating complex energy regulations complicates lab operations and may restrict research. Our involvement in standardization committees helps address these challenges. As another limitation, research on inverter firmware in renewable energy plants is currently unfeasible due to the

absence of physical power plants; however, it represents a potential direction for future investigation.

Tab. 4.1: Comparison of different recent Smart Grid testbeds for cybersecurity research (from years 2014 to 2024).

Testbed	Year	Protocols	Architecture	Applications
[Alp14]	2014	• IEC 61850 (MMS, GOOSE, SV) • UDP	HIL	Distribution
[Kou+15]	2015	• IEC 61850 (MMS) • DNP3 • Modbus TCP	HIL	Distribution
[AMF15]	2015	• DNP3 • TCP/IP • IEEE C37.118	HIL	Generation Transmission
[GMC16]	2016	• IEC 61850 (MMS) • IEC 60870-5-104	Simulation	Transmission
[Sri+17]	2017	• Modbus TCP • DNP3 • other OT Prot.	HIL	Distribution
[RNB18]	2018	• IEC 61850 (GOOSE, SV) • IEC 60870-5-104	HIL	Generation
[AKM19]	2018	• IEC 61850 (MMS, GOOSE) • Modbus TCP	Hardware	Generation Transmission Consumer Storage
[KIO18]	2018	• Modbus TCP	HIL	Distribution Storage
[Oye19]	2019	• IEC 61850 (GOOSE) • DNP3 • IEEE C37.118.1a • NTP v.4	HIL	Distribution Transmission
[HEF19]	2019	• TCP/IP	Simulation	Generation Consumer
[Bec+20]	2020	• IEC 61850 (MMS, GOOSE) • DNP3 • IEEE C37.118.2	HIL	Transmission
[Qui+22]	2022	• IEC 61850 (MMS, GOOSE)	HIL	Distribution
[Kan+22]	2022	• IEC 61850 (MMS, GOOSE) • MQTT	Simulation	Generation Transmission Consumer
Our Lab **[P8]**	2025	• IEC 61850 (MMS, GOOSE, SV) • IEC 60870-5-104 • Modbus TCP • PTP • NTP v.4	HIL	Generation Distribution Transmission Storage Consumer

4.2 Target Models

The integration of AI into smart grids introduces critical vulnerabilities to adversarial attacks. In this thesis, we use a number of AI models as targets. Attacking PQR, Modbus TCP, SV, MMS, and S7Comm IDSs in smart grids targets critical components essential for stability, communication, and control. These systems are interconnected, with Modbus TCP, SV, MMS, and S7Comm protocols enabling real-time data exchange and automation, while PQR ensures stable energy delivery. Compromising any of these can disrupt operations, amplify grid instability, and undermine detection mechanisms, creating cascading failures. SV provides measurements, MMS ensures data exchange, S7Comm implements physical control, and PQR ensures the system operates within acceptable limits. Together, they ensure efficiency, reliability, and resilience in critical infrastructure, but their interdependence also makes them collectively vulnerable to cybersecurity threats.

While most research literature focuses on AI security in the images, text and audio domain, tabular data coming from complex systems such as smart grids has its own challenges **[P1]**. Moreover, considering a variety of feature spaces and model architectures is important to better evaluate the capabilities of adversarial attacks. Apart from building our own testing environments, we identify and reproduce several models from existing work [Tia+21; Eyn+24] chosen due to being open source, and being already used to understand feature importance or as target for robustness assessment.

4.2.1 Intrusion Detection Systems

Modbus TCP IDS

In this section[2] we first describe the configuration of our Modbus TCP testbed, including attack scenarios. Finally, we describe in detail the process of generating the dataset for reproducibility purposes, and how we train the learning-based models.

Testbed Architecture. The architecture of our small-scale power generation subsystem at KASTEL Security Lab Energy is depicted in Fig. 4.3. It is a hardware-in-the-loop (HIL) implementation comprising of hardware devices such as Programmable Logic Controllers (PLCs). These devices communicate with simulated models which

[2]Part of this work was peer-reviewed and published in [P7]

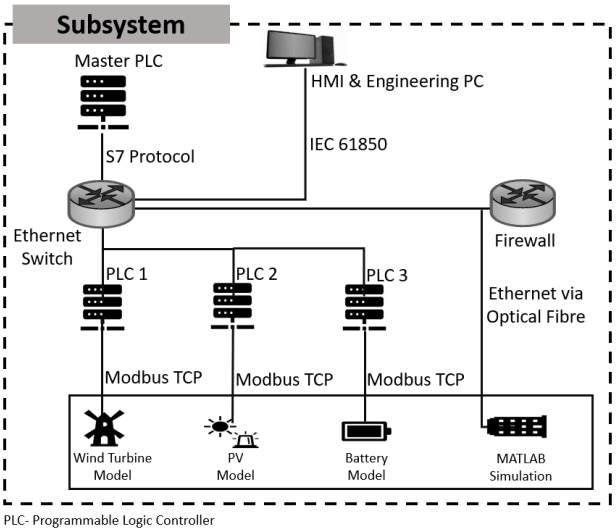


Fig. 4.3: Generation Subsystem at KASTEL Security Lab Energy **[P7]**.

include Photovoltaic (PV), wind turbine and battery models. The former two models function in power generation, while the battery model can work both as load (in case of charging) or supplier (in case of power shortage). PLCs function as Modbus servers, receiving the values for power, current and voltage from simulated models over Modbus TCP. The data received is then transferred to the Master PLC which makes it available to the operator for monitoring purposes. One of the ports for the centralized switch is configured as a mirror port for data collection and monitoring. Table 4.2 presents further details about different devices in the testbed. Remote access to the system is possible via a VPN gateway. This is the point where an adversary with *perfect knowledge* of the system could get into the network, and then proceed to execute different attacks. These potential attacks are described further in this section. It is worth highlighting that details about the initial steps of the attacker procedures are out of the scope of present work.

The architecture for our subsystem could be translated to a 4 layered-automation pyramid, categorising each component according to its function; this is represented in Fig. 4.4. At the lowest level (L0) of field devices executing the physical process, is where simulated MATLAB models are running. Second level (L1) comprises of control devices, which are PLCs for our subsystem. The higher layer of the pyramid relates to the control and monitoring applications such as HMI (Human-Machine Interface).

As mentioned above, the communication between PLCs and simulated models is established over Modbus TCP. This protocol is neither authenticated nor encrypted [XY19], hence it becomes easier for an adversary to exploit these vulnerabilities

Tab. 4.2: Subsystem Elements

Component	Protocol Used	Function
PLC 1	Modbus TCP	Data transmission from Windturbine model to Master PLC
PLC 2	Modbus TCP	Data transmission from PV model to Master PLC
PLC 3	Modbus TCP	Data transmission from battery model to Master PLC
Master PLC	S7	Master PLC controlling slave PLCs and verifying power generation algorithm

and implement different attacks at this level. Considering this, we have chosen to implement attacks against PV model and PLC. It should be noted that other two protocols, Manufacturing Message Specification (MMS) and Siemens S7 also have similar vulnerabilities. The scope of the current study is limited to Modbus TCP, as most of the connections in the subsystem are over this protocol.

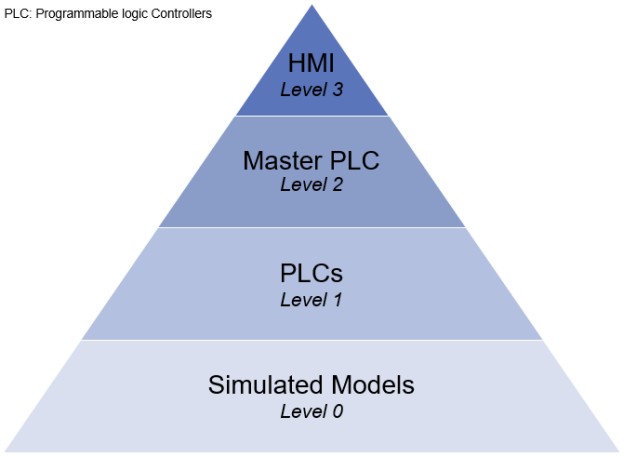


Fig. 4.4: Automation Pyramid for Subsystem **[P7]**.

Several frameworks exist to ensure system's security and management from different perspectives. For instance, the AAA (Authentication, Authorization, and Accounting) framework [Lai+17] addresses the main attributes of policy enforcement and access control to resources. On the other hand, the CIA (Confidentiality, Integrity, and Availability) Triad framework covers the principal attributes for protection of critical information from unintended modifications, keeping it accessible to authenticated users [Vos+22]. We reflect on the three primary pillars of the CIA Triad when implementing different cyber-attacks on Modbus TCP communication between the PLC and PV model. We base our threat model on a *Perfect Knowledge* scenario [Big+13], where we assume that an intruder accessed the network using the remote access feature of our subsystem. This is motivated by the fact that security by obscurity [MN03] is not reasonable, i.e., it is not good practice to expect security by code

secrecy. For initial access, the attacker may use social engineering attacks (e.g, spear phishing) to get the credentials of an authorized user and get into the network. The sequence followed to perform these attacks is demonstrated in Fig. 4.5.

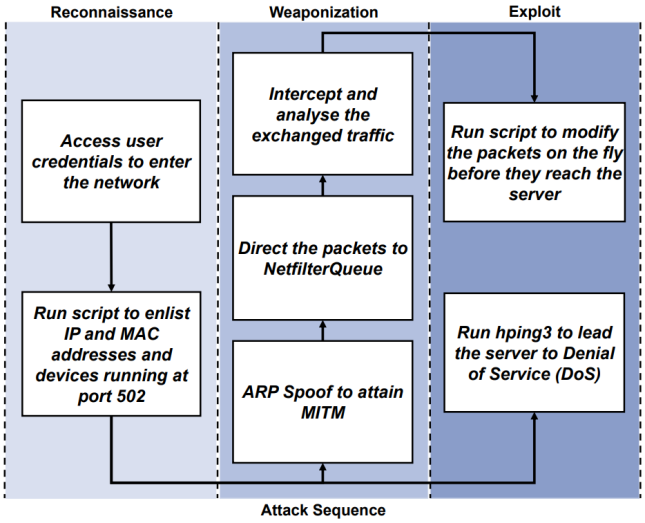


Fig. 4.5: Attack Sequence [P7].

Reconnaissance. Several open source tools such as *NMAP* [Lyo09] are available for discovering the hosts, services, and protocols in a network. Depending upon the information required for implementing further attacks, we implement our own Python script for finding the alive nodes within the network along with enlisting their IP and MAC addresses. Additionally, we identify which of the discovered devices are communicating over port 502 (the default port for Modbus TCP communication) and regard them as servers. This supports in forming a better network architecture from an adversarial point of view.

Data modification Attack. In order to perform any kind of modification or injection attacks, the attacker is supposed to reach a MITM (man-in-the-middle) position to intercept the real-time traffic flow within the network. Here we implement the attack at L0, regarded as the lowest level in the automation pyramid in Fig. 4.4. We perform ARP spoofing to poison the cache of the PLC and PV model while pretending to be the switch between both. This enables us to attain MITM position, and subsequently all the Modbus TCP traffic passes through our attacking machine. We then use *iptables* rules to direct traffic towards *nfqueue* [PyP] where we parse and record the incoming traffic. A customized filter implemented using *scapy* [PGB20] is then used to modify the data in real-time by defined instances while keeping the packet structure intact. TCP payload is modified to zero, which can also be seen in Fig. 4.6. Algorithm 1 describes the steps for attack implementation.

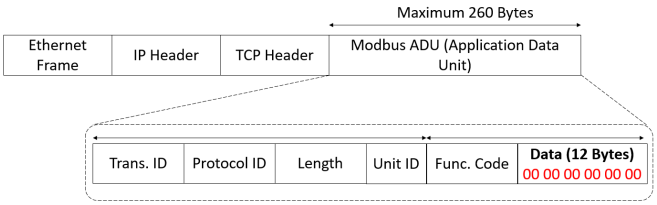


Fig. 4.6: Packet Modification on the Fly **[P7]**.

Algorithm 1 Data Modification on the Fly

Input: Set *iptables* rule
nfqueue.bind(x, process_packet)
try nfqueue.run():
packet.accept()
if packet.haslayer(Raw) and packet.haslayer(TCP): **do**
$setnew\_payload[IP].len \rightarrow None$
$setnew\_payload[IP].chksum \rightarrow None$
$setnew\_payload[TCP].chksum \rightarrow None$
if len(scapy_packet[Raw].load) == y:
$new\_payload[Raw] \rightarrow str[:z] + b'x00'$
packet.set_payload(new_payload)
except nfqueue.unbind():

Denial of Service (DoS). One way to make a server unavailable is by flooding the communication channel with unnecessary packets, exhausting the resources of a server. For this attack, we consider flooding the channel with TCP packets from a large number of invalid IP addresses indicating a connection initiation. The server starts dedicating resources for the incoming connections ending up being exhausted. This results in the unavailability of the server for the legitimate client, and connection termination. We employ the *hping3* tool [Che+15] to flood the channel with TCP packets, which results in connection termination between the PLC and PV model. Algorithm 2 presents the steps to carry out the attack.

Algorithm 2 TCP Sync Flooding

Input: Number of packets, target IP, port
while x<number of packets: **do**
$createpackets \rightarrow packet[IP][TCP]$
if Flag == Ack: **do**
socket.open(targetIP, port)
exit():

Dataset generation is possible following these steps: (1) running the subsystem under normal conditions to gather normal dataset in *pcapng* format, (2) selection

and implementation of well-known attacks, (3) capture of network traffic for under-attack conditions, and (4) feature computation. Extraction of packet-based features, physical process data and flow telemetries is possible from the generated *pcapng* files, but in this work we emphasize the packet-based features.

Run subsystem and export benign data to a file. We first configure port 15 on our switch as a mirror port for monitoring ingress and egress traffic from all other ports. Then, we run our subsystem for approximately four hours to obtain the dataset representing normal operation. This normal traffic data consists of all traffic within the network, and we filtered Modbus TCP traffic for our use case.

Execute a set of representative attacks and export malicious data to a file. For the collection of data for under-attack conditions, we begin by implementing our script for reconnaissance. As a consequence, a number of ARP requests and responses are generated in the network. The next step is to initiate ARP spoofing for attaining MITM. During this time, several TCP/ICMP retransmissions and redirections take place until an attacker achieves MITM position. During the packet modification attack, packets are modified while they pass through the attacker PC. Execution of sync flooding attack results in TCP sync transmissions until the connection is terminated.

Extract features and label transactions. In order to allow for the correct identification of anomalous traffic, it is necessary to select a set of informative features. The analysis of normal traffic yielded 39704 instances. The attacks yielded 1150 instances. Therefore, approximately 97% of the instances are labeled as benign (normal), and approximately 3% of the instances are labeled as malicious (attack); this imbalance will be taken into account at the evaluation stage by using adequate metrics. Modbus TCP packet-based features (*Transaction ID*, *Function Code*, *PDU Length*, *Address*, *Data Values*, *Payload Content Length*) are extracted. We combine these into a dataset. We will use such dataset to train our ML-based IDS.

Prior to training, we apply principal component analysis (PCA) to project the data to a lower dimensional space. Additionally, we standardize features by removing the mean and scaling to unit variance. Finally, based on the best performing models presented in related literature (Section 2), we train RF and SVM classifiers, and perform grid search to identify the best combination of hyperparameters (RF: $max\_depth = 10$, $min\_samples\_leaf = 4$, $min\_samples\_split = 10$, $n\_estimators = 100$; SVM: $C = 0.001$).

SV and MMS IDSs

The SV dataset is created by the authors in [Eyn+24] to simulate and analyze current and voltage measurements in a substation environment. To prepare the dataset, the SV messages are extracted from network capture files (PCAP) and converted into CSV format. Each sample represents a single SV message and includes two types of features: protocol-based and traffic-based. Protocol-based features include details like source and destination MAC addresses, application IDs, and measurements. Traffic-based features capture timestamps and time differences (deltas) between consecutive messages.

The MMS dataset also released by authors in [Eyn+24] includes a larger number of features compared to the SV dataset (107 features for MMS versus 52 features for SV). This difference arises from the nature of the protocols involved. While SV messages are mapped directly to the Ethernet layer, MMS messages are transmitted over TCP/IP, which adds additional protocol headers. As a result, MMS messages contain more detailed information. In the MMS dataset, the classes are Normal, Fault, Data Modification attack, and Delay attack.

Dataset Preprocessing. The input dataset is first preprocessed by cleaning rows with missing feature values, converting feature columns to numeric types, and filling missing values with column means. The dataset is split into training and testing sets using a 70-30 split. The labels in the SV use case are Normal, Fault, Replay attack and Injection attack. In the MMS IDS use case, the pipeline is the same as for the SV use case, with the exception that in the dataset the labels are Normal, Fault, Data Modification attack and Delay attack.

Baseline Model Training. A baseline model is trained using the clean dataset. Specifically, a RF classifier is utilized to ensure robustness against overfitting and ease of interpretability. Model performance is assessed on the test dataset by computing the accuracy.

S7 IDS

To support the present work, we developed a prototype IDS to detect replay attacks on S7comm traffic using a Programmable Logic Controller (PLC) Setup. Attackers aim to manipulate training data so that replay attacks are misclassified as normal traffic. Notably, Siemens SIMATIC S7-1500 CPU devices prior to firmware Version

1.8.3 were vulnerable to such attacks[3], which could bypass replay protection. This vulnerability could be exploited remotely.

In our S7Comm intrusion detection scenario, the attacker's goal is to manipulate the training data through replay attacks, so that the model performance decreases during real-time analysis. The IDS prototype we developed focuses on detecting replay attacks within S7Comm traffic, particularly in an electrical substation network with different protocols, including Siemens PLC communications over TCP port 102. The system starts with a preprocessing phase where network capture files are transformed into structured data formats. This allows for the extraction of both statistical and payload-based features: all possible combinations of 15 ending bytes of the packet raw data, packets per minute, and average payload length. These features help distinguish between normal operations and potential replay attacks. As mentioned, most features are dynamically generated by identifying unique packet endings of a specific length. After grid search for hypermarameters optimization, we identified 15 bytes as the best-performing value for detection. We end up with a dataset containing over 200 features. The packets get aggregated together by using a sliding window technique, that is, each feature vector represents a period of one minute. Then, the statistical feature *packets per minute* gets computed according to this window. Finally, the features that arise from packet endings get assigned a value depending on the occurrences of each packet ending within a given minute of traffic. Two thirds of the sliding windows created from port 102 include S7Comm traffic. After processing, we obtain 773 feature vectors of normal traffic and 1284 of traffic with replay attacks. A RF model is trained using data from normal and attack traffic (leaving 30% for testing), enabling the system to establish a baseline of typical network behavior and replay attack scenarios. The model achieves an accuracy of 98.86% on clean data.

4.2.2 Power Quality Recognition

In [Tia+21], authors present a signal-specific and signal-agnostic algorithm for generating adversarial perturbation of power quality signals in energy systems. The signals and disturbances were generated using the mathematical model presented in [Igu+18], and they use a CNN for classification. Experimental results show that their proposed signal-specific adversarial examples algorithm provides less perturbation compared to the fast gradient sign method [GSS] (which was adapted for power

[3] https://nvd.nist.gov/vuln/detail/CVE-2016-2201. There are no ethical concerns arising from the present work, because Siemens has already patched this vulnerability; we use it for demonstrating an XAI-in-the-loop, real-world adversarial attack model.

signal analysis), while the signal-agnostic algorithm can generate the universal perturbation that can fool learned models. Authors propose adversarial training as a defensive measure.

Preprocessing and Baseline Model Training. The dataset [Tia+21] contains 17 signal classes representing normal signals and different power quality disturbances. The number of features in the dataset is 640. The test accuracy without poisoning is 98.15%. The preprocessing pipeline involves: (1) Loading and normalizing signal data from an Hierarchical Data Format version 5 (HDF5) file. (2) Generating labels for each Power Quality Disturbance (PQD) class and converting them to one-hot encoded format. (3) Shuffling and splitting the dataset into training and validation subsets (split ratio is 80:20), ensuring sufficient class representation in both sets. (4) Expanding the signal data dimensions to meet the input requirements of CNN. The baseline model is a Deep Neural Network (DNN) designed for time-series classification. It employs three convolutional layers with increasing filter sizes, max-pooling, batch normalization, and Rectified Linear Unit (ReLU) activations, followed by dense layers for feature extraction and classification into 17 signal quality disturbances classes. The `Nadam` optimizer and early stopping are used to prevent overfitting.

4.2.3 Industrial Computer Vision

The hardware-in-the loop architecture of the KASTEL Security Lab Energy allows us to perform computer vision tasks using real hardware. The perception model is a compact convolutional network composed of four convolution–batch-normalization–ReLU blocks with dropout for regularization, followed by a global average pooling layer and a linear output head for binary classification. It was trained using standard optimization settings and our own datasets of real panel images (of programable logic controles and other devices) augmented with simple transformations (source code, datasets and other artifacts can be found in the open source repository[4]). Model performance on held-out subsets of the same data distribution is 100%, supporting reliable experiments.

4.2.4 Route Choice Prediction

We present a preliminary study of route choice prediction and its robustness to small, realistic perturbations, with emphasis on attack surfaces that matter for transport

[4]https://github.com/gus5298/XAI-Auto-Laser-Attack-and-Defense

and energy infrastructure. Using a synthetic, richly annotated route dataset, we train per-user Random Forest models (single-model test accuracy 82%) and aggregate explanations via model-specific Gini importances and Local Interpretable Model-Agnostic Explanations (LIME) attributions. Feature analysis shows that distance and duration dominate the global Gini ranking while LIME highlights temporal indicators as highly explanatory at the instance level. We probe sensitivity with nearest-neighbor counterfactuals and find that modest edits, e.g., sub-kilometre distance shifts (0.4–1.2 km) or small hierarchy changes can flip predicted choices; an ensemble majority vote over ten user models generalizes to a held-out user with 80% accuracy. Building on these findings, we formalize a problem-space threat model, discuss the inverse feature-mapping challenge that an attacker must solve to translate feature goals into map/sensor edits, and consider realistic vectors such as fake charger listings and traffic spoofing. We conclude with mitigation directions and a discussion of regulatory implications under the EU AI Act, and outline next steps including in-vehicle validation.

5 Attacking Integrity

Integrity attacks aim to manipulate the behavior of learning-based systems while preserving the appearance of normal operation, making them particularly dangerous in safety-critical infrastructures. This Chapter investigates integrity violations in smart grid environments by analyzing both evasion attacks at inference time and targeted poisoning attacks during training. A central focus is placed on the role of explainability: explanation methods are leveraged to identify influential features, guide perturbation strategies, and reduce attacker uncertainty. Through a series of protocol-aware experiments spanning network intrusion detection, industrial vision pipelines, and user behavior analysis, this Chapter demonstrates that XAI can significantly amplify the effectiveness of integrity attacks under realistic smart grid constraints.

5.1 Evading Modbus TCP Intrusion Detection at Test Time

In general, adversarial attacks directly affect the *integrity* of the target IDS, as the attacker's objective is to intentionally bypass detection at test time. This is done by subtly customizing traditional attacks (i.e., DoS, MITM, data modification), making the detection system unreliable while maintaining the original malicious capabilities of traditional attacks. The *availability* of the IDS is not directly compromised by the adversarial examples. However, due to a big amount of classification errors (both false negatives and false positives), the system becomes effectively unusable. This situation may force the system administrator to disable the IDS [Bar+06]. In our scenario, an attacker wants to access and harm the smart grid, which is protected behind the IDS. Therefore, from an adversarial viewpoint, it is beneficial to keep the IDS up and running to avoid incident response measures. Moreover, adversarial examples are a medium to perform conventional attacks, which may affect availability, (e.g., as a consequence of successful DoS conditions). To highlight the importance of addressing domain-specific constraints when performing evasion

attacks, we present a feature-space attack against our ML model as a baseline[1]. We use this baseline to then present the requirements for a realistic attack, i.e., feasible in the problem space [Pie+20].

Knowledge: For our attacks, we assume that the attacker has perfect knowledge of the target system.

Feature Mapping: Regarding transformation applied, PCA is not directly invertible, but it is possible to approximately reconstruct the original data. On the other hand, standard scaling is invertible. Both PCA and standard scaling can be considered differentiable.

Feature Space: The IDS considers only explicit features contained or inferred within the protocol headers.

Problem Space: The perturbations target Modbus TCP traffic only.

Classifiers: We focus on evading both SVM and RF, to simulate the case where the target IDS is operating as an ensemble of the two classifiers.

Search Strategy: We perform a feature-space attack based on benign and malicious feature modification. By choosing the top-2 most relevant features for SVM and the top-2 most important for RF, we target both classifiers simultaneously.

Side-effect features: Not explicitly considered.

In order to evaluate the performance of IDS, we split 80% of the data for training and 20% for testing. The performance of the classifiers before and after modifications is reported in Table 5.1. Precision- and recall-based metrics are generally recommended to address an imbalanced dataset and to prevent the base rate fallacy [Arp+22]. We observe that the MITM attacks are being undetected in most of the cases by both of the classifiers.

Tab. 5.1: Detection Results (%).
The test set includes normal and malicious instances.
The different attacks are isolated from the test set to show detection results before and after adversarial modifications.

Subset	Classifier	Accuracy	F1	Recall	Precision	FPR
Test Set	SVM	99.6	92.9	86.9	100	0.0
	RF	99.4	89.6	87.3	92.0	7.7
DoS	SVM	94.8 / 0.0	97.4 / 0.0	94.8 / 0.0	100 / 0.0	0.0 / 0.0
	RF	94.8 / 1.0	97.4 / 2.0	94.8 / 1.0	100 / 100	0.0 / 0.0
Data modification	SVM	88.9 / 0.0	94.1 / 0.0	88.9 / 0.0	100 / 0.0	0.0 / 0.0
	RF	88.9 / 100	94.1 / 100	88.9 / 100	100 / 100	0.0 / 0.0
MITM	SVM	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
	RF	5.6 / 0.0	1.1 / 0.0	5.6 / 0.0	100 / 0.0	0.0 / 0.0

[1] Part of this work was peer-reviewed and published in [P7; P1]

The second value shows the evasion capabilities of our adversarial attack against each classifier. For this, we applied modifications to all samples from the test set that were manually labeled as malicious, and isolate them to report metrics. For those with both 0% recall and False Positive Rate (FPR), all the instances are classified as False Negative (FN). The data modification instances remain detected by the RF, which is the most robust model against the proposed perturbations. The instance of MITM attack initially identified by the IDS is now evading detection.

Explaining the model. As an exploratory step towards understanding what features are relevant for detection/evasion purposes, we produce a ranking according to importance metric. In the case of SVM, an attacker can use the vector weights as indicator. Due to its linearity, SVM will be more likely to misclassify an attack if the most benign features are prevalent. For RF, an attacker relies on the Mean Decrease in Impurity (MDI) as importance metric, or more advanced explanation methods such as SHAP [LL17] to identify those features that will contribute towards our target (i.e. benign) classification output. In this case, the top-2 features for both classifiers are same: *Modbus Address* and *PDU Length*. Our evasion attack aims to mislead the classifiers using minimal transformations to avoid disruptions (i.e., mimicry attack). Therefore, the results reported in Table 5.1 involve a maximum modification of 59.3% for *Modbus Address* and 40.1% for *PDU Length*. These percentages relate to the comparison between the adversarially modified value and the original range for a given feature. For example, if the smallest value of a target feature is x and the maximum is y, the range of action for modifications z is the difference between y and x. As a result, the maximum modification allowed would be z (i.e. 100%); this is to avoid forming outliers that would highly likely be unfeasible in the real-world object.

Due to the application of PCA and standard scaling, it is necessary to revert the modified data to its original to observe the real-world object. In the case of PCA, it is possible to reconstruct an approximation of the original data, but it will not be exactly the same. This would require an extra step to make sure that the reconstructed values are correct (e.g., rounding or truncating numbers to maintain original format). Regarding standard scaling, it is possible to apply the inverse scaling to bring the data back to its original range. In our experiments, we ensure that the perturbations do not go beyond potentially feasible values by limiting a minimum and maximum value for each feature. We achieve this by extracting the smallest and biggest value present in the original dataset (for each feature) and ensuring that the adversarial examples do not exceed these. In practice, these limits force that the adversarial perturbations keep the values within a viable range.

In this work, we have presented a feature-space attack. The problem space, on the other hand, encompasses the entire extent of the target, including the data, objectives, constraints, and real-world context:
Preserved Semantics: While implementing attack scenarios, care must be taken to keep the packet structure intact.
Available Transformations: Changing features while modifying the packet requires a complete understanding of protocol specifications. These specifications reveal how different features are interlinked, and how malicious packets could be crafted and included into the network without being identified. This involves understanding the sequence and acknowledgment numbers for the TCP stream, function codes and related register addresses, packet memory structures, etc.
Plausibility: For injecting a packet with function code 3 to read the holding register, it is necessary to specify the correct register address to be accepted. Similarly, changing the length of transmitted data will require updating the header. For this matter, the usage of alternative, sophisticated attacks such as stealthy False Data Injection Attacks (FDIA) should be explored. Robustness to preprocessing is an additional constraint that must be taken into account.

Explainability in the threat model. XAI techniques are currently used in domains such as computer vision [KL21] to detect the presence of adversarial directions in images. For instance, saliency maps [YXP19] have been long used for detecting adversarial perturbations in the ML literature. However, we envision the use of XAI for crafting more powerful adversarial attacks against learning-based models based on heterogeneous, tabular data from SGs. This Section presents an initial experiment and analysis of how XAI can be utilized from an adversarial perspective.

Experimental Setup. We explore an attacker's potential to analyze a target model, either directly (in a white-box approach) or through a surrogate model (black-box approach) due to the transferability of attacks [Dem+19]. We use the code and dataset presented in Section 4.2.1. The dataset is from Modbus TCP traffic, used for intrusion detection in a electrical substation testbed [KAS]. We focus on the proposed Random Forest (RF) model, chosen for its complexity compared to linear SVMs.

SHAP Summary Plot. We employ SHapley Additive exPlanations (SHAP) [LL17] to produce a summary plot (see Figure 5.1). This type of plot is used to show the contribution of each feature to the output of the model. The plot uses a *bee swarm* style to display the density of the points, avoiding overlaps in order to see each point clearly. Each row represents a feature from the dataset. Features are ranked by their importance, which can be inferred by the spread and color intensity of the

points. The X-axis represents the SHAP value for each feature. This value indicates the impact of a feature on the model's output. A higher absolute SHAP value means a higher impact on the model output. Points placed to the right of the vertical line (zero impact) indicate a positive impact on the model output, while points to the left indicate a negative impact. The color of the points represents the value of the feature (not the SHAP value). In Figure 5.1, *High* feature values are colored in pink and *Low* feature values are colored in blue. This means that high values of a feature tend to push the model output higher if the SHAP value is positive or lower if the SHAP value is negative. The spread of the points along the X-axis shows the distribution of the impacts each feature has across the data. A wide spread means the feature has varied effects depending on the context (other feature values in the vector).

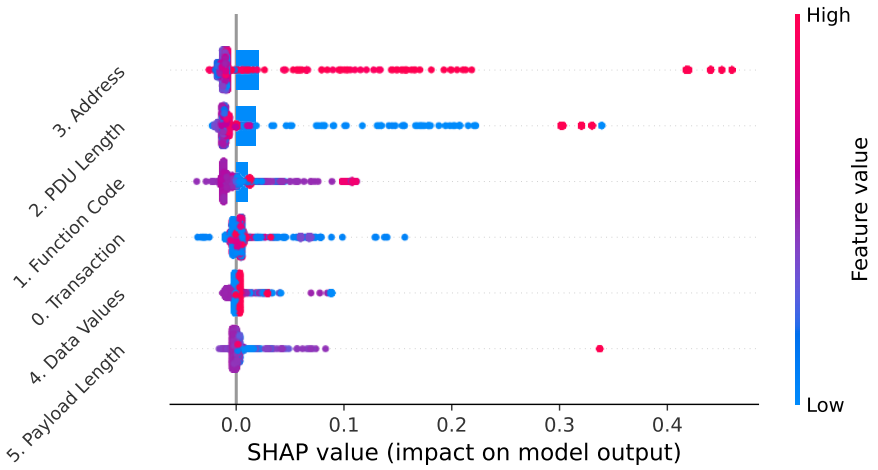


Fig. 5.1: SHAP summary of the RF model.

Adversarial Insights. The SHAP summary plot in Figure 5.1 provides insights that can be exploited to craft adversarial examples. By determining the direction to alter the most relevant feature's values, an attacker can optimally steer predictions. In our experiment, we observe that including higher values of *Modbus Address* would lower prediction success. Furthermore, we see that *Payload Length* is not as relevant, and should remain unchanged. This is because higher values of addresses are linked to the learned profile of the system under normal operation. Therefore, an adversary would include increased values of modbus addresses deliberately to maximize evasion probability and stealthiness. The SHAP values give a sense of how much a feature should be changed. Features with a wider spread of SHAP values might be more sensitive to changes, and even small perturbations could lead to significant impacts on the output. To empirically demonstrate this, we applied the same perturbation ($\varepsilon = 2.7$, as per the initial experiments) to all possible pairs of

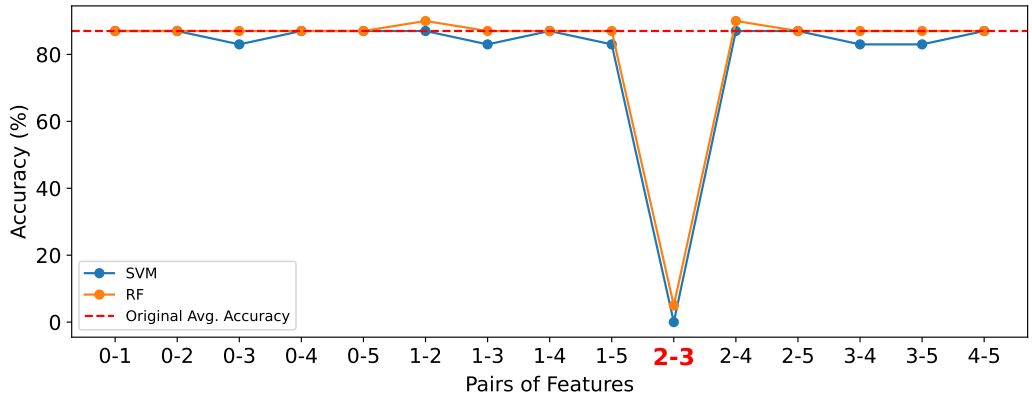


Fig. 5.2: IDS accuracy under adversarial attack targeting different pairs of features.

features. As can be seen in Figure 5.2, the most impactful manipulation corresponds to features with index 2 and 3 (i.e., *Modbus Address* and *PDU Length*), which are the most important—and therefore, vulnerable— ones according to SHAP. The feature that correspond to each index can be consulted in Figure 5.1 (on the Y-axis labels).

> In summary, the attacker can leverage insights on natural feature variability to guide subtle, hard-to-detect changes, thus crafting stealthy attacks that can potentially bypass SG security. Nevertheless, the trade-off between impact and stealthiness of the attack must be addressed.

In our experiments, we attempt to address this by ensuring that perturbations remain within realistic bounds, that is, setting minimum/maximum limits for each feature. This is accomplished by identifying the smallest/largest values for each feature in the original dataset and ensuring that the values of the adversarial examples do not surpass these boundaries. Consequently, these constraints ensure that the adversarial modifications maintain feature values within a practical range. In practice, it may occur that in order to keep an attack feasible, a combination of features need to be perturbed to mislead the predictor, not only the most important one(s). However, authors do not show a realistic implementation (i.e., end-to-end exploit in the problem space) of the evasion attack, limiting their study to the feature space.

Through XAI, it is also possible for an attacker to identify spurious correlations [Arp+22] in learning models. These occur when unrelated data artifacts mistakenly guide the model in classifying tasks, leading it to rely on irrelevant patterns rather than addressing the actual problem. For instance, in a network intrusion detection scenario, if most attacks in the training data come from a specific network region, the model might wrongly focus on identifying attacks based on IP ranges rather than

the true nature of the attacks. This issue is a ML pitfall [Arp+22] exploitable by sophisticated attackers.

Within this Section, we implemented several attacks against Modbus TCP and developed an ML-based IDS for detecting those attacks. The results from the evaluation of this IDS against our mimicry attack showed a concerning decrease in its performance. This revealed features potentially vulnerable to be abused for evasion with minimal perturbations.

5.2 Targeted Poisoning against PQR, SV IDS and MMS IDS.

When an attacker's objective is to force specific misclassifications (e.g., *malware* as *goodware* but keeping real *goodware* decisions accurate) via data poisoning, the poisoning is *targeted*. This affects the integrity of the model.

5.2.1 Poisoning Power Quality Recognition

Preprocessing and Baseline Model Training. The dataset [Tia+21] contains 17 signal classes representing normal signals and different power quality disturbances. The number of features in the dataset is 640. The test accuracy without poisoning is 98.15%. The preprocessing pipeline involves: (1) Loading and normalizing signal data from an Hierarchical Data Format version 5 (HDF5) file. (2) Generating labels for each PQD class and converting them to one-hot encoded format. (3) Shuffling and splitting the dataset into training and validation subsets (split ratio is 80:20), ensuring sufficient class representation in both sets. (4) Expanding the signal data dimensions to meet the input requirements of CNN. The baseline model is a DNN designed for time-series classification. It employs three convolutional layers with increasing filter sizes, max-pooling, batch normalization, and Rectified Linear Unit (ReLU) activations, followed by dense layers for feature extraction and classification into 17 signal quality disturbances classes. The `Nadam` optimizer and early stopping are used to prevent overfitting.

Explainability Methods for Feature Importance. The framework[2] utilizes SHAP, LRP, and LIME to identify influential features in the signals. SHAP values quantify

[2]Part of this work was peer-reviewed and published in [P2]

the contribution of individual features to the model's predictions based on cooperative game theory. Kernel SHAP is applied to approximate feature contributions using a surrogate model trained on a subset of the input signals. LRP propagates relevance scores backward through the DNN to attribute importance to input features. Relevance scores are calculated using gradients and are aggregated over multiple samples to identify critical features. LIME generates local surrogate models to approximate the decision boundary of the DNN for specific instances. Perturbations are applied to the input signals, and the corresponding feature importances are derived from the surrogate models.

Explainability-Guided Data Poisoning. The top features identified by each explainability method are perturbed to evaluate the model's robustness. The perturbation pipeline includes: (1) Selecting the top 25% of features based on importance scores. (2) Applying Gaussian noise with a defined magnitude (50%) to the selected features. (3) Retraining the model on the poisoned data and evaluating its performance on clean test data.

Evaluation of Union and Intersection Attacks. To simulate combined attacks, unions and intersections of the top features from SHAP, LRP, and LIME are computed. For each union and intersection, the corresponding features are perturbed, the model is retrained on the poisoned data, and finally, performance metrics are calculated to assess the attack's impact.

Tab. 5.2: Detection results with a perturbation magnitude of 50% affecting top features (out of 640) for PQR (CNN model).

Poisoning Method	Poisoned Features	Accuracy (%)	Target (C-1) Misclassifications
No Poison	0	98.15 ± 0.79	26 ± 5
Random	160	69.45 ± 14.17	1063 ± 743
SHAP	160	79.73 ± 9.14	1056 ± 840
LRP	160	47.82 ± 20.76	189 ± 213
LIME	160	38.88 ± 10.23	141 ± 120
SHAP $\cup$ LRP	277	60.11 ± 14.17	248 ± 341
SHAP $\cup$ LIME	281	50.29 ± 8.86	124 ± 72
LRP $\cup$ LIME	260	34.92 ± 4.19	57 ± 31
All $\cup$	353	76.68 ± 18.23	561 ± 304
SHAP $\cap$ LRP	43	97.05 ± 2.41	276 ± 372
SHAP $\cap$ LIME	39	96.42 ± 1.48	227 ± 135
LRP $\cap$ LIME	60	78.22 ± 11.53	470 ± 414
All $\cap$	15	97.70 ± 0.80	96 ± 38

In Table 5.2, the union-based combinations ($\cup$) under poisoned conditions exhibit significant performance degradation, with the broad inclusion of features amplifying the vulnerability to adversarial effects. Conversely, intersection-based strategies ($\cap$) offer less adversarial impact. When the most important features of C-1 (normal)

are altered, the model's decision boundaries shifts in ways that achieves the target misclassifications, that is, signals with disturbances labeled as normal signals.

Results. The results confirm that combining XAI methods for poisoning effectively misclassified faulty signals as normal while limiting the number of perturbed features, achieving its intended goal. An attacker may choose to provoke more misclassifications by using SHAP or random poisoning (as will be described in Subchapter 6.1), or remain stealthy (only 1.1% overall model accuracy decrease) by using a targeted SHAP ∩ LRP attack, causing a 962% increase in targeted misclassifications (from 26 in the clean model to 276).

5.2.2 Poisoning SV and MMS IDS

Feature Importance Estimation. The framework leverages multiple explainability techniques: (a) The framework uses the TreeExplainer from SHAP to calculate feature importance scores by aggregating absolute SHAP values across all classes and samples; (b) the permutation importance method measures the decrease in model performance when a feature's values are randomly permuted, thus indicating the feature's predictive contribution; (c) using the mean decrease in accuracy, the importance of features is derived directly from the trained RF model using the decrease in the Gini impurity metric during the construction of decision trees; (d) LIME approximates the model locally for each instance by fitting interpretable surrogate models.

Feature Set Analysis. The framework computes unions and intersections of the top features identified by SHAP, permutation importance, Gini importance, and LIME. These feature sets provide insight into the agreement or divergence among different XAI methods.

Data Poisoning. We limit the maximum number of perturbed features to the top 25% in the SV (max. 8 out of 32) and MMS (max. 13 out of 52) datasets (RF model). The identified sets are perturbed using Gaussian noise with a specified magnitude of 50%. New poisoned datasets are generated for each feature set.

Model Fine-tuning and Evaluation. The baseline model is fine-tuned on the poisoned datasets. For each poisoning scenario, the accuracy is computed on the test dataset. Each scenario is executed 5 times to compute average values and standard deviations.

Table 5.3 showcases insights into the interplay between feature selection and adversarial attack resilience. The high accuracy achieved by the models, specially the

Tab. 5.3: Detection results (accuracy in %) with a perturbation magnitude of 50%. PF: Poisoned Features.

Method	**SV**	Target	**MMS**	Target
No Poison	99.53 ± 0 PF:0	81 ± 0	100 ± 0 PF:0	0
Random	99.41±0.20 PF:8	97 ± 4	99.97 ± 0.06 PF:10	0
SHAP	99.03±0.20 PF:8	194 ± 6	99.11 ± 0.18 PF:10	0
Perm.	99.09±0.29 PF:8	184 ± 8	98.97 ± 0.23 PF:10	0
Gini	99.02±0.28 PF:8	197 ± 9	98.96 ± 0.20 PF:10	0
LIME	99.09±0.20 PF:8	182 ± 5	98.92 ± 0.22 PF:10	0
SHAP ∪ Perm.	98.96±0.15 PF:8	80 ± 1	98.99 ± 0.11 PF:13	0
SHAP ∪ Gini	99.00±0.15 PF:8	83 ± 2	98.82 ± 0.20 PF:10	0
SHAP ∪ LIME	98.99±0.17 PF:8	82 ± 2	98.72 ± 0.20 PF:10	0
Perm. ∪ Gini	98.97±0.15 PF:8	81 ± 0	98.85 ± 0.22 PF:13	0
Perm. ∪ LIME	99.07±0.14 PF:8	77 ± 1	98.68 ± 0.24 PF:13	0
Gini ∪ LIME	99.01±0.15 PF:8	82 ± 2	98.93 ± 0.16 PF:10	0
All ∪	98.98±0.15 PF:8	80 ± 1	98.90 ± 0.21 PF:13	0
SHAP ∩ Perm.	99.08±0.15 PF:8	182 ± 2	98.75 ± 0.24 PF:7	0
SHAP ∩ Gini	98.98±0.14 PF:8	190 ± 1	98.98 ± 0.18 PF:10	0
SHAP ∩ LIME	99.03±0.15 PF:7	186 ± 6	98.83 ± 0.17 PF:10	0
Perm. ∩ Gini	99.08±0.14 PF:8	185 ± 3	99.00 ± 0.22 PF:7	0
Perm. ∩ LIME	99.10±0.15 PF:8	182 ± 3	98.77 ± 0.25 PF:7	0
Gini ∩ LIME	99.09±0.14 PF:7	179 ± 2	98.76 ± 0.21 PF:10	0
All ∩	99.09±0.15 PF:5	180 ± 6	98.83 ± 0.22 PF:7	0

MMS IDS under these conditions demonstrate that the overall model remains robust when perturbations are constrained to top-ranked features. The marginal differences in performance between different combinations, emphasize the importance of precise class targeting for maintaining malicious capabilities. Intersection-based strategies outperform the rest when it comes to forcing the SV IDS to misclassify network attacks into normal traffic.

5.2.3 Summary

The XAI-guided targeted poisoning attack against the SV IDS more than doubles the malicious capabilities of random poisoning (see subchapter 6.1) and limits the number of features to be perturbed (specially successful are the intersections of methods). The MMS IDS resisted against targeted data poisoning (0 attack instances misclassified into normal) and reduced its overall accuracy by less than 2%.

5.3 Autonomous XAI-Guided Physical Adversarial Perturbations in Industrial Vision Pipelines

Cyber-physical systems (CPS) that underpin critical infrastructures (e.g., industrial, energy, etc.) increasingly rely on computer vision for monitoring the physical state of

devices [He+22]. From in-process tool wear detection in manufacturing [LAC20] to monitoring remote substations [SY14; Zhe+16; Liu+18], visual analytics pipelines now automate the recognition of indicators (such as LED status lights [EY20] or dial positions [Pei+23]) to enhance situational awareness and reduce operator workload. While these perception modules improve efficiency, they also introduce a new attack surface: an adversary capable of manipulating the visual scene can compromise the dependability of system diagnostics and control decisions.

In the present work[3], we introduce `LaserTag`, an open-source toolchain for studying and demonstrating *physical-world adversarial attacks* on computer-vision monitoring in industrial control systems. `LaserTag` combines eXplainable Artificial Intelligence (XAI) with a reproducible hardware platform to investigate how small optical perturbations, such as those produced by consumer-grade laser diodes, can mislead learning-based classifiers tasked with recognizing normal and abnormal states of industrial panels.

`LaserTag` integrates two complementary components: (1) light-weight computer vision models and XAI visualization interface for real-time classification of panel states, and (2) a Raspberry-Pi-based pan–tilt actuator that directs laser beams toward regions of high model saliency. The XAI module thus guides the physical attack by identifying image regions most influential to the classifier's decision, allowing targeted color-channel interference against the camera sensor.

The primary contribution of this work is a publicly available[4] experimental framework that bridges computer-vision dependability research and adversarial physical testing. The framework allows for live demonstration with real hardware.

As case studies, we evaluate the system on front panels of three representative industrial devices: a *Siemens SIMATIC S7-1500*, a *SIMATIC S7-400*, and a *Siemens Energy Omnivise T3000* control system. These platforms exemplify the diversity of indicator geometries and optical characteristics encountered in operational technology environments. By systematically exploring the interaction between explainability maps, environment variables and different device states, `LaserTag` enables quantitative assessment of vision model robustness and facilitates the design and evaluation of countermeasures.

[3] Part of this work was peer-reviewed and published in [P6]
[4] https://github.com/gus5298/XAI-Auto-Laser-Attack-and-Defense

5.3.1 Background and Motivation

Related Work

Modern industrial operations increasingly rely on computer vision to interpret analog indicators [Hua+20]—for example, reading the state of light-emitting diodes (LEDs) [SLL21] on Programmable Logic Controllers (PLCs) or security controllers. Such perception pipelines add an additional layer of security, promising higher scalability and reduced cognitive load for operators, yet their *dependability* hinges on robustness to environmental shifts and intentional manipulation. Research over the past decade has shown that deep models, even when highly accurate, can be fooled by carefully crafted perturbations in the physical world [Wei+24].

Early demonstrations of physical attacks established that adversaries can realize targeted misclassification with printable accessories or scene modifications: Sharif *et al.* showed that inconspicuous eyeglass frames can reliably evade or impersonate subjects in face-recognition systems [Sha+16]. Eykholt *et al.* developed RP2 to create robust perturbations for traffic signs that survive viewpoint and lighting changes [Eyk+18]. Brown *et al.* introduced the *adversarial patch*, a universal, robust overlay that compels classifiers toward an attacker-chosen label across diverse scenes [Bro+17]. These results crystallize a central insight for dependability: physical perturbations that are easy to deploy operationally can invalidate model assumptions.

Beyond static artifacts, light-based and projection-based attacks demonstrate that *instantaneous* optical perturbations can subvert perception without leaving physical residue. Nguyen *et al.* used adversarial light projections to fool face-recognition in real time [Ngu+20]. Work on 'phantom' objects and projector spoofing showed that injecting patterns into a camera stream can mislead Advanced Driver-Assistance Systems (ADAS) [Nas+20]. Complementary research has revealed that lasers can couple into sensors in unexpected ways: Sugawara *et al.*'s *Light Commands* exploited laser-induced audio injection to control voice assistants, underscoring how optical energy can traverse unconventional attack paths [Sug+20]. Recent studies further refine *laser-spot* attacks that precisely modulate small image regions to cause misclassification under daylight conditions [Hu+23].

Despite this progress, **tools** for *systematic, reproducible* experimentation on industrial devices reading remain scarce. The dependability community lacks open, end-to-end frameworks that: (i) couple interpretable vision models with optical actuation, (ii) map explainability signals to physical target points, and (iii) operate on real

devices and indicators found in operational technology (OT) environments. Practical constraints (panel geometries, LED optics, camera exposure control, and safety) demand specialized testbeds that go beyond generic adversarial-image pipelines.

`LaserTag` addresses this gap by providing an XAI-guided, physically grounded platform to study and *improve* the dependability of industrial vision pipelines. By aligning Grad-CAM [Sel+17] saliency with pan–tilt laser actuation, researchers can probe failure modes where color channel saturation, specular highlights, or small bright spots steer the decision boundary. Using representative hardware—including Siemens SIMATIC S7-1500 and S7-400 PLCs and a Siemens Energy Omnivise T3000—`LaserTag` enables controlled studies of robustness, repeatability, and countermeasures under realistic optics and lighting.

Threat Model

The `LaserTag` framework focuses on physical perturbations that influence the visual appearance of indicator lights on industrial equipment. The adversary considered in this work is *physically proximate* to the target device and capable of directing a low-power laser toward its front-panel LEDs; in the real world, this can be achieved by, e.g., an adversary flying a drone into an industrial location, or an adversary physically placing the required hardware onsite. The attacker does not modify the device, its firmware, or its network interfaces; instead, the goal is to alter what the *camera-based monitoring system* perceives. This reflects scenarios where visual inspection is used as an auxiliary or redundant information channel and where optical manipulations may interfere with automated decision support or operator situational awareness.

The attacker is assumed to have access to a camera feed that observes the panel, either by directly viewing the image stream or by probing the classifier behaviour through repeated observations. We assume the ability to perform limited trial-and-error adjustments, either manually (e.g., with a gamepad) or through XAI-based autonomous scanning, to find laser positions that influence the vision model. No assumptions are made about the internal parameters of the classifier; the available visual explanations in our testbed model may not exist in a deployed system, and are used in the present work solely to study how explainability could guide perturbation strategies. The adversary's capabilities are intentionally restricted to keep the threat model aligned with realistic laboratory conditions. The attacker uses a commodity laser diode of limited power and wavelength and cannot introduce arbitrary pixel-level changes. Laser illumination is instantaneous, localized, and

constrained by optical reflections, camera exposure, and the physical geometry of the device under test. The adversary cannot control environmental conditions such as ambient lighting, glass covers, or panel materials, although experiments are conducted under different lighting scenarios (room lights on, room lights off) to evaluate robustness.

The attacker's objective is to influence the classifier's interpretation of device states—for example, by making an indicator appear active when it is not, or by oversaturating the sensor such that a lit LED becomes difficult to detect. Achieving a specific misclassification is not assumed to be fully deterministic; rather, the aim is to explore which perturbations *can* affect the model and to characterize the conditions under which such disruptions occur. Impact on downstream control logic is explicitly out of scope; we evaluate only perception-stage effects.

Finally, we do not consider stronger adversaries who can compromise internal logic, tamper with the PLC, or manipulate digital communication protocols. The threat model is intentionally narrow to isolate and study the role of visual perturbations in industrial controllers monitoring pipelines and to enable controlled exploration of potential dependability weaknesses without overstating real-world risk.

5.3.2 System Architecture

Figure 5.3 summarizes the overall architecture of the `LaserTag` framework. The conceptual view in Figure 5.3a shows how responsibilities are distributed across four domains: the attacker's analysis workstation, the on-site embedded actuation platform, the industrial device under test, and the defender's monitoring environment. The attacker operates a model (i.e., the original or a surrogate of the model on the Defender's side) together with an explainer and can steer the laser manually using a USB gamepad, or deploy a fully autonomous attack in which the laser is pointed at the most important location for a given state classification without human intervention. The saliency information is used to guide the attacker to aim a laser beam toward sensitive locations on the target panel; in autonomous mode, the attacker does not require manual interaction.

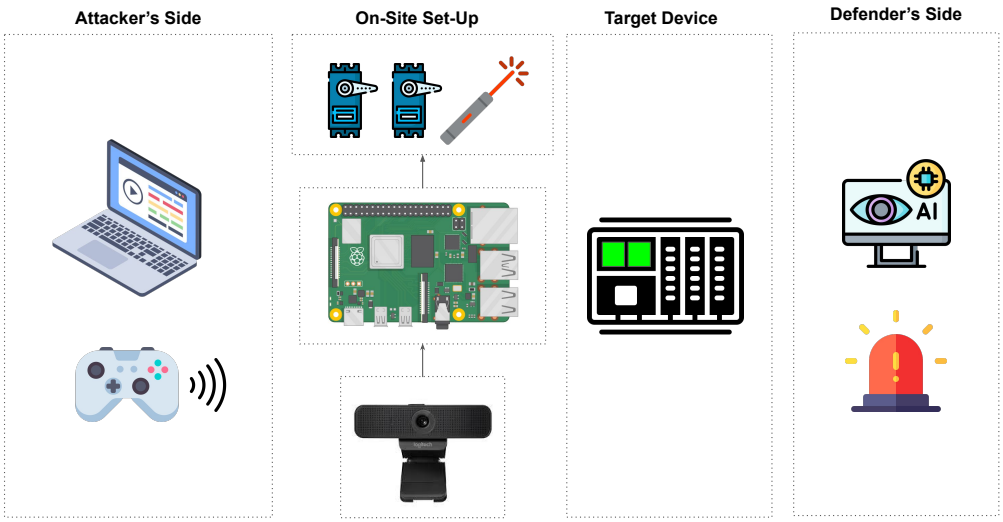
The real-world setup used in our experiments is shown in Figure 5.3b. A consumer-grade RGB camera positioned on a flexible tripod observes the front panel of an advanced Distributed Control System (DCS) called Omnivise T3000, which is used for ensuring cybersecurity in power plants and hybrid energy systems. The camera and laser are mounted on a two-axis servo platform powered by a Raspberry Pi.

The Pi generates PWM signals for the servos and switches the laser diode through a MOSFET driver, whereas the host PC performs all inference and visualization tasks. A USB gamepad enables intuitive teleoperation, and the physical separation between perception (PC) and actuation (Pi) helps maintain a clear boundary between the sensing and intervention components of the system.

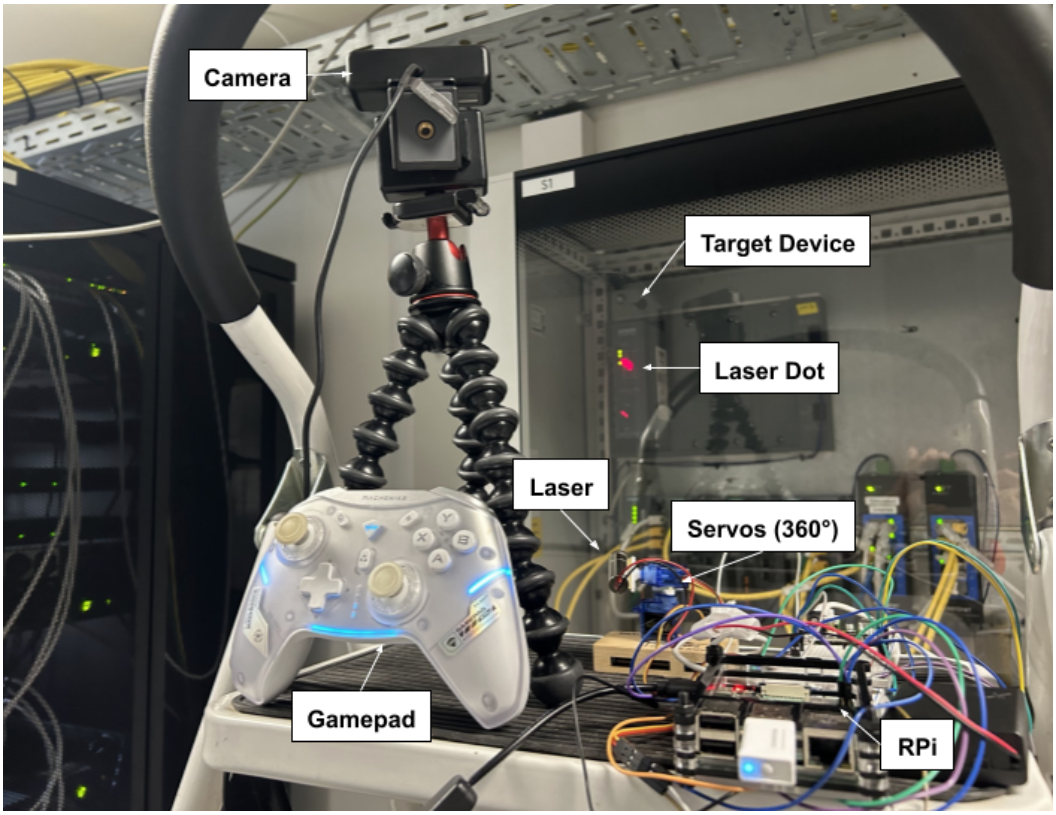
Operation proceeds as a continuous pipeline: the camera captures frames that are forwarded to the host PC, where an ROI is selected and processed by a Convolutional Neural Network (CNN) classifier. Grad-CAM produces a saliency map highlighting regions most influential for the predicted LED state. These regions can be used directly by the attacker during manual aiming, or used by our algorithmic solution to autonomously aim the laser. Whenever laser commands are issued, the Pi adjusts the pan–tilt servos accordingly and activates the diode for short, controlled illumination bursts directed at the target device.

From a hardware perspective, the setup consists of three main elements: a visual subsystem built around a 1080p RGB camera; an actuation subsystem composed of two SG90 servos and a low-power red or green laser diode (lasers color can be chosen based on desired goal) driven through a MOSFET; and the computational layer, split between the PC and the Pi. Communication between the two devices uses a simple network interface, though both components can function independently, which reduces coupling and supports safer experimentation. All configuration parameters (including camera exposure, region-of-interest settings, and servo calibration) are stored to support reproducibility, and deterministic inference settings can be enabled to stabilize evaluation.

The design is deliberately modular. The CNN may be substituted by any PyTorch-compatible classifier, and the explainability layer can be extended with additional XAI methods such as Integrated Gradients (IG) [STY17b] or Layer-Wise Relevance Propagation (LRP) [Bac+15]. Similarly, the servo subsystem can be replaced with alternative actuators without modifying the overall workflow. This flexibility allows the `LaserTag` framework to be integrated into broader testbeds or adapted to different classes of devices.



(a) The attacker-side module runs CNN-based inference and XAI analysis, allowing for manual aiming of the laser via a gamepad (in addition to a novel, autonomous aiming mode). The on-site Raspberry Pi controls a pan–tilt platform and laser diode for physical perturbation of the target device's indicator lights, confusing the Defender and raising/hiding alarms.



(b) The physical prototype deployed in a lab environment, showing the camera, pan–tilt servos, gamepad interface, laser module, and Raspberry Pi mounted near an actual industrial device.

Fig. 5.3: `LaserTag` conceptual overview and experimental setup.

5.3.3 Implementation Details

The `LaserTag` prototype relies entirely on low-cost, readily available components that make the setup easy to reproduce. The perception pipeline runs on a workstation and receives video frames from an RGB camera facing the industrial panel. A Raspberry Pi 4 controls the physical actuation hardware, which consists of two SG90 micro servos mounted in a pan–tilt configuration and a low-power laser diode driven through an AO3400A MOSFET. A small step-down regulator provides a stable supply voltage for the laser circuitry, and a keyed hardware switch together with protective eyewear ensures safe operation during experiments. All grounds are tied together to avoid electrical offsets between the servo supply and the laser driver.

To provide interpretability, Grad-CAM is implemented with backward hooks on the final convolutional block. For each processed frame, the viewer computes a saliency map and overlays it on the ROI with adjustable transparency. The system also records the model's probability vectors and the associated heatmaps, enabling later analysis of classifier behaviour and explanation consistency.

Servo motion is controlled through the `pigpio` library, which provides stable PWM outputs on the Raspberry Pi. The same control loop also triggers the MOSFET gate for activating the laser diode. Pulsewidths are mapped directly to the physical travel range of the servos, allowing predictable motion during manual or automated aiming. In demonstration mode, the PC and Pi operate as loosely connected components: the PC performs local inference and may transmit target coordinates or activation signals to the Pi, which can additionally log timestamps and servo positions to support post-hoc inspection. This loose coupling simplifies debugging and enhances fault tolerance by preventing perception-side failures from affecting the actuation logic.

Reproducibility is supported by storing metadata, including exposure settings, calibration values, and random seeds. Maintaining these parameters ensures that runs can be repeated under comparable conditions, which is essential for evaluating physical-world attack behaviour. Safety considerations are incorporated throughout the design, as the system complies with recommended guidelines for operating low-power lasers in indoor laboratory environments. Experiments are carried out with beam stops and mandatory protective eyewear, making the platform suitable for research and teaching within the dependability community.

XAI-Guided Closed-Loop Aiming

The autonomous attack loop runs entirely on the host PC. It builds on the same binary classifier and Grad-CAM explainer described earlier, but closes the loop with the Raspberry Pi through a simple TCP JSON protocol.

First, the operator manually selects a ROI around the relevant part of the panel by dragging a rectangle in the live camera view and pressing `s` to lock it. From this point on, the system operates in a closed loop on the ROI only. Each frame is cropped to the ROI and passed through the trained model (either `SmallCNN` or `ResNet18Classifier`), using the same normalization and input size that were used during training. The script keeps a mapping of class indices to labels (by default `"normal"` and `"network_failure"`) and a confidence threshold to decide whether the current state should be treated as normal.

Within the ROI, the script applies a multi-scale Grad-CAM procedure. Several center crops at different scales (e.g., 100 %, 85 %, 70 %) are evaluated; for each crop, the classifier output is computed and the probability for a given state is recorded. The crop that maximizes this probability is retained as the most informative view. Grad-CAM is then applied to the last convolutional layer of the model for that crop, producing a saliency heatmap over the crop, which is upsampled back to the full ROI resolution. The pixel with the maximum saliency value in this heatmap is taken as the *target point* $(u_\mathrm{t}, v_\mathrm{t})$ inside the ROI.

In *automatic* mode, this target point is determined once, on the first frame after the ROI is locked, and then kept fixed for the rest of the run. This design choice is reflected directly in the code: a variable is set on first use and not updated, even if Grad-CAM changes later. This prevents the aiming point from drifting if the classifier output fluctuates while the laser is already approaching the LED region. In *manual* mode, by contrast, the operator can click inside the ROI to override the Grad-CAM suggestion and explicitly set the target pixel to be aimed at with the gamepad, which is stored. Both modes share the same closed-loop control logic.

At each iteration, the system also performs color-based detection of the laser spot within the ROI. The crop is converted to HSV (Hue, Saturation, Value) space, and a simple threshold is used to isolate either red or green pixels, depending on the current classifier decision: if the model considers the state to be normal (class index equal to the configured `normal_idx` and confidence above a threshold), the code expects a red laser dot; if the model considers the state abnormal, it expects a green dot. This convention matches the intended attack semantics (red to inject a failure appearance, green to mask a failure), but is implemented purely as a choice of color

mask. The largest connected component in the thresholded mask is extracted, and its centroid yields the current laser position $(u_\mathrm{l}, v_\mathrm{l})$ inside the ROI, if any.

Given the Grad-CAM target $(u_\mathrm{t}, v_\mathrm{t})$ and the detected laser position $(u_\mathrm{l}, v_\mathrm{l})$, the script computes the pixel error

$$e_u = u_\mathrm{t} - u_\mathrm{l}, \qquad e_v = v_\mathrm{t} - v_\mathrm{l}.$$

If no laser is detected in the ROI, the error is left undefined and no actuation command is sent, and the internal stability counter is reset. Otherwise, the horizontal and vertical errors are compared against a configurable pixel threshold. When $|e_u|$ or $|e_v|$ exceed the threshold, the script determines the sign of the required correction in yaw and pitch and sends incremental step commands to the Raspberry Pi over TCP. The Pi-side server receives each JSON message, calls `ServoAPI.step_yaw` or `ServoAPI.step_pitch` once, and returns a simple status reply. Each call moves the pan–tilt platform by one discrete step, so the PC implements a very simple proportional controller that adjusts yaw and pitch one increment at a time based solely on the sign of (e_u, e_v). Two inversion flags (`INVERT_YAW`, `INVERT_PITCH`) in the script compensate for wiring and mechanical orientation.

To avoid overshooting and to robustly detect convergence, the script maintains a counter of consecutive frames for which both $|e_u|$ and $|e_v|$ remain within the alignment threshold. If this condition holds for a configured number of frames, the system enters a locked state (variable set), and no further step commands are sent. The user can manually reset this lock by pressing keyboard key `r`, which allows the servos to re-adjust while keeping the same Grad-CAM target point.

The resulting behaviour is a closed-loop, XAI-guided aiming mechanism: Grad-CAM is used once to determine a physically meaningful target location within the ROI, and a purely image-based feedback loop then steers the laser spot toward that location using only camera observations and the detected dot position. No information about servo angles or absolute geometry is required; the entire process is driven by the combination of the classifier's explanation and the observed laser footprint on the panel.

Tab. 5.4: Performance of physical-world attack experiments conducted on the three target devices. A ✓ denotes a fully successful manipulation for both attacker variants (bruteforce sweep and XAI-guided targeting). "Trigger Failure" corresponds to inducing an alarm state when the device was normal; "Hide Failure" denotes suppressing a real fault indication.

Device	Trigger Failure (Normal → Alarm)	Hide Failure (Alarm → Normal)	Notes
S7–1500	✓	✓	Digital Display
S7–400	✓	✓	Green/Red LEDs
T3000	✓	✓	Green/Red LEDs

5.3.4 Experimental Findings

Device Characteristics

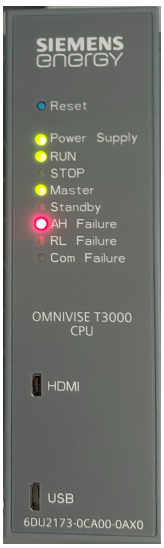
Figure 5.4 presents the two operating states of the Siemens Energy Omnivise T3000 CPU used in our evaluation. Panels (a) and (b) show the full device front under normal and failure-indicating conditions, respectively. Panels (c) and (d) display the corresponding ROI regions overlaid with Grad-CAM heatmaps, revealing how the classifier focuses on the active LED cluster in each scenario. This pairing enables direct comparison between true device states and the visual evidence that drives the model's decisions.

Figure 5.5 presents representative front-panel states for the Siemens S7–1500 (top row) and the Siemens S7–400 (bottom row). The S7–1500 exposes its operating condition through a compact digital display: normal operation is shown with a green-highlighted header, whereas failure conditions activate a red alarm banner. This produces a large, contiguous illuminated region on the LCD panel, which becomes the primary focus region for the vision classifier and the main target area for laser-based perturbations.

In contrast, the S7–400 reports its status using discrete LEDs arranged in a vertical light column. Normal operation is characterized by green indicator LEDs (e.g., `RUN`, power rails), while fault states activate a specific red LED associated with the failing subsystem (e.g., `BUS5F`). Compared to the display-based S7–1500, the LED-based S7–400 yields sharper, point-like optical signatures with a more limited illuminated area.

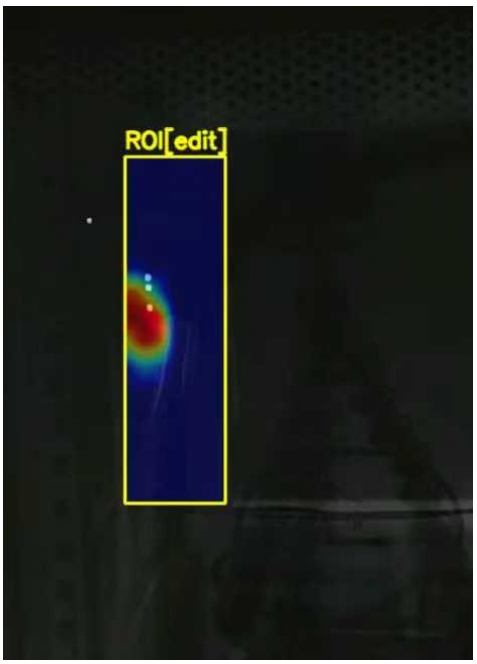
These modality differences (i.e., LCD display versus discrete LEDs) are important for evaluating physical-world robustness because the attack surface differs in size, intensity distribution, and reflection behaviour. Both devices, however, can be manipulated successfully using the same laser actuation platform.

(a) Normal state.



(b) Failure state.

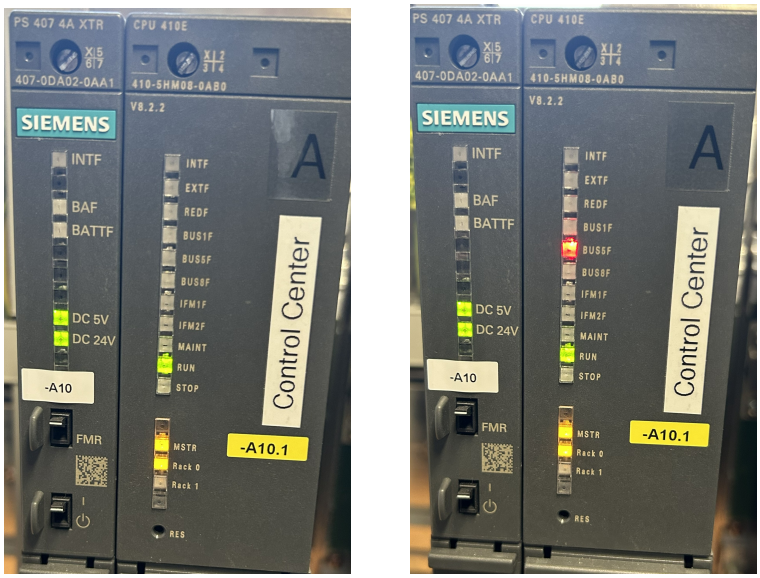
(c) Grad-CAM (normal state).



(d) Grad-CAM (failure state).

Fig. 5.4: Comparison of normal and failure operating states on a Siemens Energy Omnivise T3000 CPU. (a) The *Power Supply*, *RUN*, and *Master* LEDs are active, indicating normal operation. (b) A distinct condition where the *AH Failure* LED is illuminated. (c)–(d) Corresponding Regions of Interest (ROI) with Grad-CAM overlays highlighting the pixels most influential for the classifier's decision in each state.

(a) S7–1500 (Normal) **(b)** S7–1500 (Stop/Fail)



(c) S7–400 (Normal) **(d)** S7–400 (Failure)

Fig. 5.5: Front-panel states of the Siemens S7–1500 (top row) and Siemens S7–400 (bottom row). Green denotes normal operation, while red indicates a failure/stop condition. These states form the basis for the computer-vision classifier and the physical laser perturbation experiments.

Results

Table 5.4 summarizes the performance of experiments executed on the three industrial devices. For each platform we evaluated two adversarial objectives: (i) *Trigger Failure*, where a laser perturbation forces the classifier to report a network-failure or alarm condition even though the device was operating normally; and (ii) *Hide Failure*, where an active fault indicator is suppressed to appear normal.

Across all devices and lighting conditions, both attacker models (bruteforce sweep and XAI-guided targeting) achieved full success. The Siemens Energy T3000 and Siemens S7-400 use discrete LED indicators arranged along the front panel, allowing the laser perturbation to target individual light sources directly. In contrast, the S7-1500 exposes status information through a small digital display rather than separate LEDs. For the S7-1500 device, the model is trained to read the highlighted region (green for normal, red/orange for stop/failure) on the display. Accordingly, the physical attack is directed at that illuminated screen area, where optical saturation causes the classifier to misinterpret the displayed state.

Transferability Through Black-Box Surrogate Attacks

Table 5.5 compares the behaviour of the main CNN and a black-box ResNet18 surrogate under identical inputs. In the S7–1500 Failure scenario, the green-laser perturbation causes both models to misclassify a real failure as a normal state when no digital defense is deployed. The same holds for the T3000 Normal scenario, where the green laser reliably induces false alarms (Normal → Failure).

Digital Defense: Filters

We evaluate a set of lightweight digital defenses that fall into four main families: RGB suppression, HSV-based attenuation, inpainting-based reconstruction, and grayscale transformations. *RGB suppression* removes the laser's chromatic advantage by zeroing the corresponding channel in the input (e.g., `remove_green`, `remove_red`), while the *strong* variants additionally apply broad hue masking and aggressive brightness reduction in HSV space to eliminate both the laser core and its surrounding halo (e.g., `remove_green_strong`, `remove_red_strong`). *HSV blocking* selectively dims highly saturated, high-brightness pixels in a target hue range without removing the underlying scene structure (`hsv_block_green`, `hsv_block_red`), thereby attenuating laser highlights while preserving natural image content. *Inpainting defenses* provide the

most precise correction: the laser region is segmented through an HSV mask, dilated to include reflection artifacts, and then reconstructed using OpenCV's Navier–Stokes interpolation (`laser_inpaint_green`, `laser_inpaint_red`, and their stronger variants with expanded masks and larger inpainting radii). These filters effectively erase the laser artifact while maintaining input characteristics expected by the classifier. In addition, the *color-agnostic highlight clipping* filter (`clip_highlights`) attenuates all bright and saturated regions regardless of hue, suppressing red or green laser spots equally. Finally, *grayscale transformations* (`gray`, `gray_blur`) remove color entirely and optionally blur fine structures, neutralizing color-specific perturbations but potentially introducing distribution shifts that affect classifier accuracy. Together, these defenses span targeted chromatic suppression, brightness attenuation, artifact reconstruction, and full color removal, enabling systematic study of robustness and defense transferability across model architectures.

The results show clear differences between classes of digital defenses. On the S7–1500, effective defenses must preserve the visibility of the red failure icon while suppressing the green-laser contamination. Channel-based filters such as `remove_green` and `remove_green_strong`, as well as artifact-targeted approaches (`clip_highlights` and the green inpainting filters), successfully restore correct predictions for both the CNN and the surrogate model. In contrast, grayscale transformations collapse chromatic cues entirely, causing the classifier to misinterpret the clean failure screen as normal. Thus, they cannot be used in scenarios where colour carries semantic information about the true state. In the T3000 case, the true state is conveyed by green LEDs, and the red laser introduces a localized, high-intensity red artifact. Filters that remove or neutralize red components (`remove_red`, `remove_red_strong`), inpaint the laser spot, or clip saturated regions all maintain correct "Normal" predictions under attack for both models. Here, even grayscale methods succeed, though they introduce unnecessary distribution shift and thus are less desirable for long-term deployment.

Overall, targeted colour removal and localized inpainting provide the most stable and interpretable digital countermeasures across both devices and both attack scenarios.

Transferability becomes evident when analysing which filters correct the prediction for both models. On the S7–1500 panel, the filters `remove_green`, `remove_green_strong`, `clip_highlights`, and the two green inpainting variants restore correct predictions simultaneously for the CNN and ResNet18, demonstrating high cross-model robustness. In contrast, `hsv_block_green` protects only the CNN and fails for ResNet18, indicating poor transferability. The grayscale-based filters (`gray`, `gray_blur`) distort

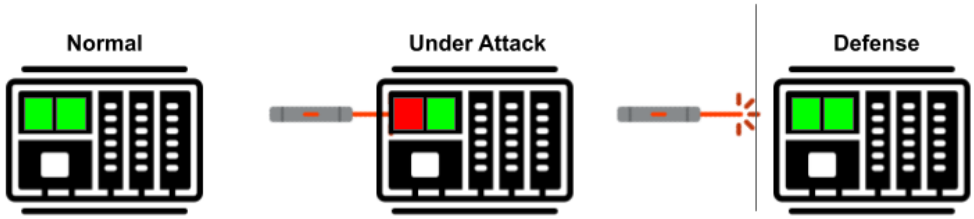


Fig. 5.6: Conceptual illustration of a physical fail-safe defense against laser-based manipulation of Computer-Vision (CV)-based industrial controllers monitoring. **(Left) Normal:** The panel displays a stable visual state (e.g., green indicators), and the computer-vision system matches the corresponding digital telemetry. **(Center) Under Attack:** An optical adversary attempts to inject a false LED state (e.g., a forged red fault signal). The camera observes a manipulated region inconsistent with the expected device state. **(Right) Defense:** Upon detecting a mismatch between digital readings and CV-based status interpretation, the system deploys a physical *curtain* that occludes the panel.

the clean failure image itself, breaking baseline performance and therefore cannot be considered transferable defenses. For the T3000, a broader set of filters transfers well across models. `hsv_block_red`, red-channel removal, highlight clipping, and the red inpainting variants all fully cancel the induced false alarm for both models. Although grayscale transforms also work, they slightly perturb the distribution of the input image and thus are less suitable for deployment.

Defenses that remove or locally correct the physical artifact (green/red channel suppression, inpainting) transfer best across models, while defenses that globally alter the input (e.g., grayscale) transfer poorly or break baseline accuracy

Physical Defense: Smart Curtain

To mitigate optical attacks, we propose a simple but effective physical–digital hybrid defense (Fig. 5.6). Whenever the device status reading inferred by the CNN disagrees with the device's digital telemetry, the system could trigger a mechanical curtain that covers the panel surface. The curtain presents a uniform white background to the camera; the CNN (or alternative models) can be trained trained to recognize this white screen as a *diagnostic mode*. Any appearance of colored light patterns (such as a laser-induced red or green patch) on the white surface is then unambiguously treated as an adversarial interference attempt rather than a legitimate panel state. This creates a verification channel in which CV-based status readings cannot be forged without leaving high-contrast artifacts against the white screen, thereby enabling robust detection of optical spoofing.

Tab. 5.5: Digital-filter evaluation under two representative device states: (i) SIMATIC S7–1500 in a true *Failure* condition (goal: model should predict Failure), and (ii) Siemens Energy T3000 in a true *Normal* condition (goal: model should predict Normal). A tick (✓) indicates a correct prediction; a cross (✗) indicates misclassification. "NoLaser" denote the correctness of the CNN and surrogate ResNet18 on the clean input. "Laser" denote correctness after applying the opposite–color laser perturbation to the most salient region.

SIMATIC S7–1500 (Failure → Hide Failure)

Filter Mode	NoLaser CNN	NoLaser ResNet18	Laser CNN	Laser ResNet18
None	✓	✓	✗	✗
hsv_block_green	✓	✓	✓	✗
remove_green	✓	✓	✓	✓
gray	✗	✗	✗	✗
remove_green_strong	✓	✓	✓	✓
clip_highlights	✓	✓	✓	✓
laser_inpaint_green	✓	✓	✓	✓
laser_inpaint_green_strong	✓	✓	✓	✓
gray_blur	✗	✗	✗	✗

Siemens Energy T3000 (Normal → Trigger Failure)

Filter Mode	NoLaser CNN	NoLaser ResNet18	Laser CNN	Laser ResNet18
None	✓	✓	✗	✗
hsv_block_red	✓	✓	✓	✓
remove_red	✓	✓	✓	✓
gray	✓	✓	✓	✓
remove_red_strong	✓	✓	✓	✓
clip_highlights	✓	✓	✓	✓
laser_inpaint_red	✓	✓	✓	✓
laser_inpaint_red_strong	✓	✓	✓	✓
gray_blur	✓	✓	✓	✓

5.3.5 Summary

`LaserTag` introduces an effective platform for exploring how explainability methods can inform the study of physical adversarial attacks in industrial vision systems. By coupling computer vision and explainable artificial intelligence visualizations with a Raspberry Pi–based pan–tilt laser module, the toolkit allows dependability researchers to visualize, automatically provoke, and document perception failures in a transparent and reproducible way. Across multiple real devices, our experiments show that `LaserTag` can reliably induce both false alarms and hidden failures, while digital color–filter defenses (particularly channel-removal, highlight clipping, and laser inpainting) offer strong and transferable protection across models.

5.4 User Behavior Analysis in Energy Infrastructure: Towards Robustness Assessment of Route Choice Prediction

Route recommendation systems increasingly interact with critical transportation and energy infrastructures—e.g., Electric Vehicle (EV) charging networks. While such systems aim to optimize travel time and user utility, they can also become an attack surface for both economic manipulation (e.g., steering drivers toward specific chargers) and cyber campaigns (e.g., APTs exploiting compromised chargers). Understanding how users select routes in navigation systems is crucial for designing reliable, user-centric guidance [Win+24] and for anticipating vulnerabilities to malicious or accidental perturbations. Traditional route recommendation focuses on optimizing travel time or distance [Bie98], but ignores how small changes in route attributes might influence user choices (and therefore system performance) under adversarial conditions.

In this Section[5], we make four key contributions that map directly onto a cyclic pipeline (see Figure 5.7): ❶ we curate a large synthetic route dataset annotated with rich structural features, enabling model training and analysis; ❷ we apply eXplainable AI (XAI) techniques to interpret feature relevance and pinpoint those most susceptible to real-world perturbations; ❸ we model real-world threats encompassing malicious actors, in addition to evaluating how small, feasible changes can affect recommendations; and ❹ we discuss defense directions and EU AI Act compliance considerations to ensure our framework can guide the development of secure recommendation systems.

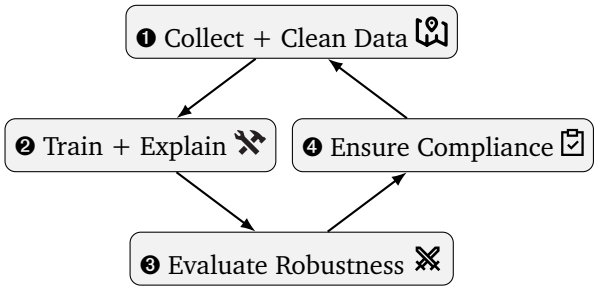


Fig. 5.7: This continuous feedback loop ensures that route-recommendation systems evolve toward ever-stronger, more transparent, and regulation-compliant performance.

[5]Part of this work was peer-reviewed and published in [P10]

Background and Related Work

Route Choice Modeling. Traditional discrete choice models [Bie98] estimate utility functions. Other work leverages learning methods to capture non-linear interactions [Are13], and most recent work explore the integration of Large Language Models (LLMs) [Ge+25; Kik+24]. These studies lack robustness analysis under adversarial conditions.

Explainable AI for Navigation. Model-agnostic XAI yields feature attributions for black-box models, while model-specific methods leverage a model's internal structure for white-box explanations. Prior work uses XAI in vehicle routing, e.g., [Kik+24] present a post-hoc framework that quantifies each edge's influence on a generated route, but it does not tackle robustness or threat modeling.

Adversarial Robustness in Spatial Contexts. Adversarial attacks on map-based services have been studied [Zen+18], but user behavior models remain untested. Counterfactual explanations [Cai+24] enable minimal changes that change model outputs.

Dataset

A synthetic dataset was generated to simulate realistic user route choices. First, sets of candidate routes for various origin-destination pairs were obtained from a real-world navigation service.

Route Feature Extraction. From the dataset of routes we decode polylines (i.e., lists of coordinates, where line segments are drawn between consecutive points). Each route is described by a series of features, including:

- *Hierarchy*: Measures how well a route aligns with the intended hierarchical design of the road network [RKG21];
- *Interconnection Density*: Calculates route complexity [RKG21];
- *Continuity*: Captures the inconvenience related to changing between different levels of the road network hierarchy [RKG21];
- *Turn Complexity*: Measures how complex a route is for a driver to follow [RKG21];
- *Rush-Hour Slot*: Categorizes the route's departure time into rush-hour (morning/evening) or non-rush-hour periods.

Next, user decisions were simulated based on the Random Utility Maximization (RUM) framework, a common approach in travel behavior modeling [McF74]. A population of agents was created, with each agent assigned a distinct utility function. The preference weights for these functions were sampled from distributions informed by established travel behavior literature, which documents the heterogeneity of traveler preferences [HG03]. For each set of candidate routes, an agent selects the route that maximizes their utility. A stochastic component was added to the decision model to simulate the variability and unobserved factors inherent in real-world choices.

Threat model

We now give a formalization of the threat model used in our analysis.

Notation. Let $\mathcal{M}$ denote the problem space (map and sensor data: vector of map entities, charger metadata, live traffic reports, signage, etc.). The routing/preprocessing pipeline computes a feature vector $x \in \mathbb{R}^d$ for each candidate route via a deterministic (but possibly complex) mapping

$$g : \mathcal{M} \to \mathcal{X} \subseteq \mathbb{R}^d, \qquad x = g(M),$$

where $M \in \mathcal{M}$ is the current problem-space state. Our trained classifier $f : \mathcal{X} \to [0, 1]$ returns a score or probability $p = f(x)$ that a given route is preferred; for an option set $\mathcal{R} = \{r_1, \ldots, r_k\}$ the model produces scores $p_i = f(x_{r_i})$ and the recommender selects $\arg\max_i p_i$ (or equivalently uses these scores as ranking/probabilities).

Attacker objective. An adversary aims to modify the system so that a *target route* $r_t \in \mathcal{R}$ (or a route that visits a target Points of Interest (POIs) such as an EV charger) is selected more frequently. We distinguish two attacker utilities:

- **Targeted selection utility:** increase the model score of r_t relative to alternatives:

  $$\text{success:} f\big(g(M')_{r_t}\big) \geq \max_{j \neq t} f\big(g(M')_{r_j}\big),$$

  where M' is the modified problem state.

- **Economic / propagation utility:** maximize expected traffic $T(M')$ through the target location (a function of selection probabilities across option sets). In abstract form:

$$U(M') = \mathbb{E}_{\text{options}}\left[\, \mathbf{1}\{\arg\max_i f(g(M')_{r_i}) = r_t\} \,\right] \cdot V,$$

where V is the adversary's per-user value (financial gain or propagation benefit).

Digital (feature-space) attack. A digital adversary that can perturb feature vectors directly solves a constrained optimization in feature space. For a chosen norm $\|\cdot\|$ and budget ε,

$$\min_{\delta_x} \|\delta_x\| \quad \text{s.t.} \quad f(x_{r_t} + \delta_x) \geq \max_{j \neq t} f(x_{r_j}), \quad \|\delta_x\| \leq \varepsilon.$$

This is the formulation used in our numerical sensitivity experiments (we also consider ℓ_∞ bounds and percentage bounds, e.g., $\|\delta_x\|_\infty \leq 0.05\|x\|_\infty$).

Problem-space attack (inverse feature mapping). Real attackers act in $\mathcal{M}$ by changing map entries, traffic reports, or charger metadata. The problem-space manipulation δ_M induces feature changes via g:

$$x' = g(M + \delta_M) = g(M) + \Delta x, \quad \Delta x := g(M + \delta_M) - g(M).$$

The attacker solves

$$\min_{\delta_M \in \mathcal{S}} C(\delta_M) \quad \text{s.t.} \quad f\big(g(M + \delta_M)_{r_t}\big) \geq \max_{j \neq t} f\big(g(M + \delta_M)_{r_j}\big),$$

where $C(\cdot)$ is a cost function (effort, monetary cost, or exposure risk) and $\mathcal{S}$ is the set of *feasible, semantically-valid* problem-space edits (for example: add a charger entry at a specific geo-cell, report a traffic jam using crowd reports, or change a map tag subject to validation rules). Feasibility constraints encode semantic consistency and preprocessing robustness, e.g.,

$$\mathcal{S} = \{\delta_M : \ \Phi(M + \delta_M) = \text{true}, \text{and} \|\Delta x_p\| \leq \eta\},$$

where $\Phi(\cdot)$ enforces semantic constraints (e.g., timestamps, mutually exclusive categorical flags such as `rush_hour_morning` vs. `non_rush_hour`), and η captures

plausible feature perturbation bounds (domain knowledge such as $\pm 5\%$ on distance/-duration).

Relation between digital and problem-space attacks. The problem-space formulation and inverse feature mapping relate via

$$\exists\, \delta_M \in \mathcal{S} \text{ s.t. } g(M + \delta_M) = x + \delta_x \quad \Longrightarrow \quad \text{digital attack } \delta_x \text{ is realizable in } \mathcal{M}.$$

Because g is generally nonlinear and nonconvex, finding such δ_M is an inverse mapping problem that may be infeasible, non-unique, or high-cost in practice.

Attack types and knowledge. We consider the following attacker knowledge levels:

- **White-box surrogate**: attacker has approximate access to f or trains a surrogate using public APIs / probe queries.
- **Black-box**: attacker can only observe system outputs (rankings, reported crowdedness) and must apply query-based or problem-space heuristics.

Our experiments emulate the digital (feature-space) setting for sensitivity quantification and then discuss plausibility and constraints for problem-space realizations.

Targeted vs. untargeted. The objective above encodes a *targeted* attack (promote r_t). An untargeted attacker can instead seek to reduce overall accuracy or force suboptimal routing by solving:

$$\max_{\delta_M \in \mathcal{S}} \mathbb{E}_{\text{options}}\Big[\mathbf{1}\{\arg\max_i f(g(M + \delta_M)_{r_i}) \neq r_i^*\}\Big],$$

where r_i^* denotes the true preferred route under the benign model.

Practical constraints (semantics and detection). Realistic problem-space attacks must preserve obvious semantics (e.g., a route cannot be simultaneously `rush_hour_morning=True` and `non_rush_hour=True`), must avoid simple cross-source validation checks, and must survive preprocessing (map-layer deduplication, POI validation). These constraints are captured by $\mathcal{S}$ and $C(\cdot)$ in the optimization above and motivate the need for defenses such as cross-source verification and provenance checks.

5.4.1 Experiments

Dataset Preprocessing. We merge the routes and the annotated route selection dataset to form choice sets. Each record includes all candidate routes (same `option_id`) with extracted features and a binary label for the user's selected `route_id`. The dataset includes data from 5000 users. We split the dataset into 70% train, 15% validation, 15% test.

ML Training. A simple Random Forest (100 trees, max depth 10, class weights) is trained via grid search. On clean test data we achieve 82.3% accuracy (F1 79%).

Feature Importance via XAI. We combine two complementary explainability methods to reveal which route attributes drive model decisions:

1. *Random-Forest Gini importances (model-specific, global)*: Table 5.6 shows that distance and duration are the top predictors on average, as measured by mean decrease in impurity.

2. *LIME importances (model-agnostic, averaged local)*: Table 5.6 reports the mean of LIME's instance-level absolute attributions across all users, revealing that rush-hour indicators emerge as the most influential overall.

Random-Forest Gini importances thus provide a true global, model-specific ranking, while LIME (by averaging many local, model-agnostic explanations) yields a complementary global perspective grounded in individual prediction sensitivity.

Tab. 5.6: Feature Importances: Gini vs. LIME

Feature	Gini Importance	LIME Importance
distance_km	0.2490	0.0607
duration_minutes	0.2080	0.0415
turn_complexity	0.1880	0.0673
interconnection_density	0.1540	0.0565
hierarchy	0.0650	0.0249
rush_hour_morning	0.0400	0.1566
rush_hour_evening	0.0380	0.1549
non_rush_hour	0.0380	0.0000
continuity	0.0200	0.0293

Nearest-Neighbor Counterfactuals. To probe sensitivity, we retrieve for a test instance nearest neighbors of the opposite predicted labels in feature space. Table 5.7 shows one example and compares each feature's original and counterfactual values (the rest remained unmodified).

Across all users, we observe that:

Tab. 5.7: An example of Counterfactual (CF) Differences

Feature	Original	CF	Difference
distance_km	147.7740	146.5875	-1.1865
duration_minutes	84.7189	84.4600	-0.2589
hierarchy	9.7554	9.0301	-0.7253
interconnection_density	24	25	+1
continuity	6	5	-1
turn_complexity	11	12	+1
non_rush_hour	False	True	Flip
rush_hour_morning	True	False	Flip

- **Distance shifts** of 0.4–1.2 km often flip predictions.
- **Hierarchy changes** of 4–10% likewise yield alternative preferred routes.
- Other features (e.g. continuity, turn complexity) require larger relative changes (> 20%) to alter outputs.

This demonstrates that small, plausible perturbations in real-world features can change the model's choice recommendation.

Generalization to Unseen Users. We evaluated 10 user-specific models on a held-out 11th user. The individual model accuracies ranged from 20% to 86%, averaging 72%; a simple majority-voting ensemble achieved 80% accuracy on the held-out user. These results confirm that while personalization boosts performance, ensemble strategies can provide robustness to user variability.

Problem-Space Attack Feasibility. In the problem space [Pie+20], an adversary's goal is to induce desired changes in route features by editing real-world map components, such as adding/removing/mislabeling EV chargers, altering map metadata, or injecting fake traffic data. To execute such an attack in the problem space, an adversary must respect real-world constraints, only applying transformations that are available, semantically valid, and robust to the routing engine's preprocessing. For example, Table 5.7 shows that when the *non_rush_hour* flag flips to `True`, the *rush_hour_morning* feature must simultaneously flip to `False`, since it would be semantically impossible for a route to be both in and out of rush hour at the same time. Doing this in a way that directly affects the recommendation requires solving an *inverse feature-mapping* problem: finding concrete edits that yield a target change in the model's input features. Inverse feature mapping is challenging because the relationship between map edits and feature values is generally non-linear and non-convex. For example, to alter *hierarchy*, one could relabel secondary roads as primary in the map database, but this demands intimate knowledge of the map schema and risks detection by data-validation pipelines. Moreover, adversaries can perturb

real-time traffic data (which is used to compute features such as time and distance) via "crowdedness" attacks. As demonstrated in prior work [**eryonucu2022sybil**], attackers can use multiple emulated or scripted devices to report fake traffic jams, making roads or POIs appear congested. These attacks are easy to mount at scale due to the open, contributory nature of participatory sensing features. By spoofing congested conditions, an attacker can steer users away from certain routes or POIs, influence feeder traffic to preferred locations, or even disrupt entire transit corridors in a cyberattack scenario. A particularly subtle avenue of manipulation lies in *rush-hour spoofing*. In our LIME-based analysis, temporal features related to rush hour were the most influential decision factors for route choice, therefore, they are strong candidates for problem-space manipulation. These features depend on the departure timestamps. An adversary could exploit this by spoofing GPS data. Additionally, many navigation platforms use crowdsourced incident-confirmation pop-ups, prompting drivers to verify accidents or roadblocks in real time; this mechanism is similarly vulnerable: coordinated false confirmations from multiple users can corrupt the collaborative sensing pipeline, causing widespread misinformation and misguided rerouting.

5.4.2 Discussion

Our findings show that critical features (e.g., distance, time, turn complexity) are prime targets for adversaries. We hypothesize and demonstrate with initial experiments that XAI-guided attacks are more efficient (lower perturbation magnitudes) than naïve methods. To counter this, defenses should explore: (1) regularization of high-importance features, avoiding over-reliance on brittle features by reweighting feature importance; and (2) map data verification via cross-source validation of charger locations and road hierarchy, among other checks.

Regulatory context. AI systems that perform management or operational tasks for critical infrastructure and road traffic can be classified as *high-risk* under the EU AI Act (classification rules and Annex III). Whether a particular route recommender is considered high-risk depends on the deployment context (e.g., embedded as a safety component in a vehicle/OT system or operating as an authoritative traffic manager). We therefore recommend that deployers perform a documented Article-6 risk assessment and, if necessary, prepare the technical documentation and robustness testing required for high-risk systems. Compliance requires; (1) rigorous *data governance* and verification of input data (e.g. map layers) to prevent poisoning; (2) *transparency* via XAI explanations for end users; and (2) *robustness assessments* against foreseeable manipulations, as demonstrated in the present work.

Limitations. Our synthetic dataset and simulated problem-space attacks may not capture all real-world complexities. Inverse feature mapping remains an open challenge; realizing a desired change in high-level features requires solving an inverse mapping to actual map edits, e.g., adding billboards that mislabel a secondary road as a motorway, or installing decoy chargers. This mapping is non-convex and may be infeasible without detailed knowledge of the routing engine's preprocessing pipeline.

Future Work. Future work will involve testing our framework in real-world driving scenarios under controlled map perturbations, safeguard user privacy via data augmentation and synthetic datasets, extend our analysis to advanced adversarial attacks (e.g. generative, multi-objective, temporal), and explore adaptive defenses like adversarial training and map-data cross-validation.

5.4.3 Summary

The present Section provides a preliminary investigation on the adversarial robustness of route recommendation systems, with an explicit focus on electric vehicle charging infrastructure. By integrating explainable AI and realistic problem-space threat modeling, we lay the groundwork for robustly assessing the security of learning-based models for route recommendation. Our framework guides practitioners toward EU AI Act compliance and more secure, transparent navigation services.

6 Attacking Availability

While integrity attacks seek controlled manipulation, availability attacks aim to broadly degrade system functionality, rendering learning-based components unreliable or unusable for legitimate operators. In smart grids, such attacks are particularly disruptive due to strict real-time requirements and resource-constrained deployment environments. This Chapter examines indiscriminate data poisoning attacks that compromise the availability of learning-based models by degrading their overall performance. By incorporating explainability mechanisms into the attack process, poisoning strategies are optimized to maximize operational impact while maintaining plausibility. The results demonstrate that XAI-guided availability attacks can induce denial-of-service conditions in smart grid learning systems without requiring excessive attacker resources.

6.1 Indiscriminate Poisoning against PQR classification, MMS IDS and SV IDS in the Feature Space

In the same PQR use case as presented in Subchapter 5.2.1[1], we perform an experiment on random (indiscriminate) poisoning, meaning that we do not aim to cause a targeted missclassification but rather a general degradation of performance, affecting availability.

Hierarchical clustering dendrogram. Firstly, the dendrogram in Figure 6.1a visualizes the analysis of hierarchical clustering applied to the class representations based on their feature vectors. Hierarchical clustering is a bottom-up approach that iteratively merges the closest clusters based on a chosen distance metric—in this case, Euclidean distance—and the Ward linkage method to minimize variance within clusters. The horizontal axis in Figure 6.1a represents the different classes involved in the analysis. These classes are grouped iteratively based on their similarity. The vertical axis measures the Euclidean distance between clusters. The height at which two clusters are merged reflects their dissimilarity—lower merges indicate more similar clusters, while higher merges indicate greater dissimilarity. At the

[1]Part of this work was peer-reviewed and published in [P2]

bottom of the dendrogram, each class starts as its own individual cluster. As we move upward, closely related classes (those with minimal pairwise distances) are merged first. Classes within the same branch, such as C-1 (normal) and C-5 (transient/impulse/spike), exhibit the highest similarity, indicating that their features in the dataset overlap. This is likely due to the inherent characteristics of these signal types; for instance, transient signals may momentarily resemble normal operating conditions before returning to equilibrium. We explore the data this way to better understand the following behavior: a random poisoning attack (see Figure 6.1b) is likely to make the model struggle to differentiate between classes C-5 and C-1, as misclassifications are influenced by how tightly or loosely coupled classes are in the feature space.

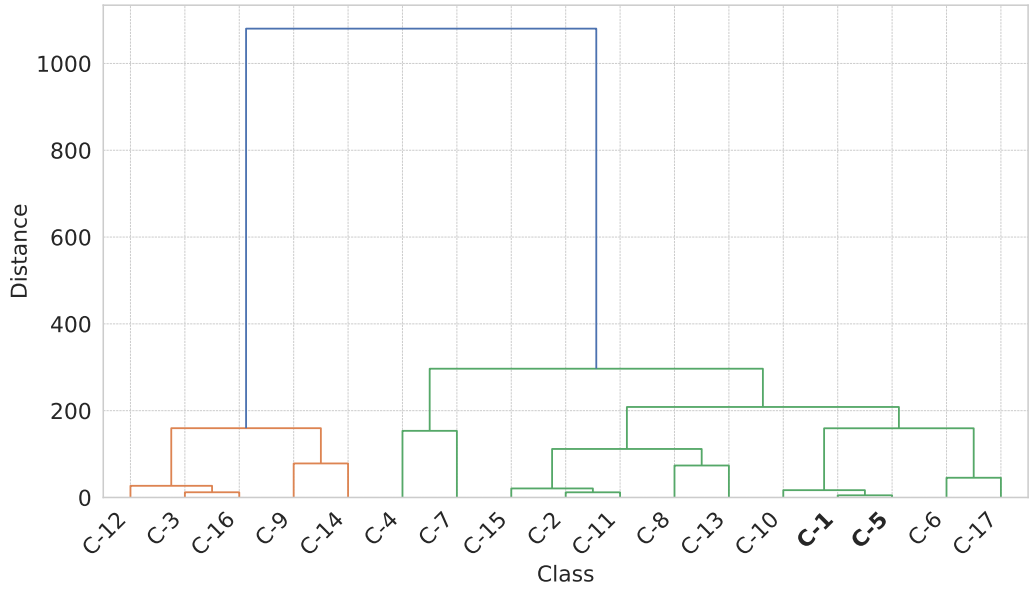
Tables 5.2 and 5.3 show how random (indiscriminate) poisoning compares versus XAI-in-the-loop targeted attacks.

Summary. Our findings confirm that combining XAI-based methods for targeted poisoning achieves the intended effect of misclassifying faulty signals (PQR use case) as normal, while simultaneously constraining the number of perturbed features. Depending on the objective, an attacker may amplify the number of misclassifications through SHAP for indiscriminate poisoning, or act more stealthily: the SHAP $\cap$ LRP configuration reduces overall accuracy by merely 1.1% yet increases targeted misclassifications by 962% (from 26 in the clean model to 276).

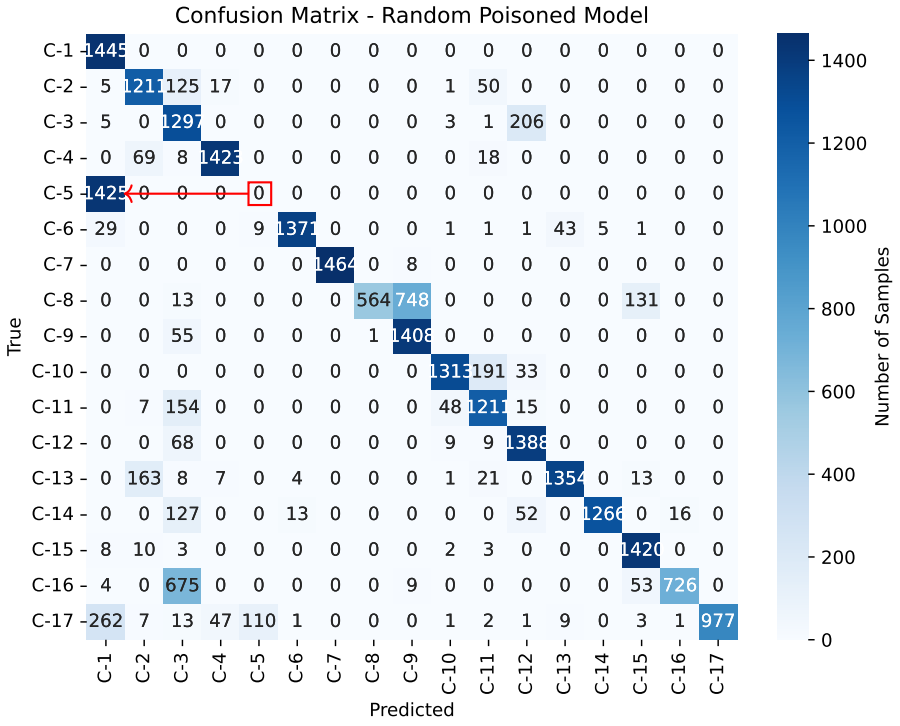
As previously stated, using XAI explanations to guide targeted poisoning more than doubles the attack power relative to indiscriminate poisoning, and it confines perturbations to far fewer features (attacks are most effective when they exploit the intersection of multiple explanation methods). Conversely, the MMS IDS remained resilient to such targeted poisoning: no poisoned instances were misclassified as normal, and the model's overall accuracy declined by less than 2%.

6.2 Indiscriminate Poisoning against SV IDS in the Problem Space

In the energy domain, after having conducted a vulnerability assessment of a time server to GNSS spoofing, authors in [Can+24] manage to experimentally determine the impact resulting from such an attack with their lab setup. Building on existing literature [Can+24], this Section defines the problem space constraints of GNSS time spoofing-based adversarial attacks (to the best of our knowledge, this was never



(a) Hierarchical clustering dendrogram.



(b) Confusion matrix for random poisoning.

Fig. 6.1: (a) Hierarchical clustering dendrogram of power signal classes based on feature representations. (b) Confusion matrix with overlay highlighting misclassifications for a random poisoning example.

formally defined before), including available transformations and attack scenarios. We apply these attacks to the SV IDS use case[2], as it heavily relies on time-related data, as identified by the XAI methods. The dataset preprocessing steps are the same as in the feature space study in the previous Section. As presented in [Can+24], authors successfully conducted time spoofing attacks on a commercial time server, employing both single-constellation (GPS) and multi-constellation (GPS and Galileo) signals. To demonstrate the impact, we use a dataset that provide high-resolution time data. The idea is to manipulate the timestamps in the previously introduced SV dataset to simulate feasible GNSS time spoofing and evaluate the performance of the learning model.

Threat Model. We use the findings presented in [Can+24] to set realistic conditions for testing the impact of GNSS time spoofing in learning-based models within smart grids.

Adversary Goals. The primary goal of the adversary is to disrupt the operations of smart grid systems by introducing timing inaccuracies. This can lead to several detrimental effects, including incorrect load balancing, miscoordination of grid resources, and compromised fault detection and response mechanisms.

Adversary Capabilities. There are various methods to manipulate a GNSS receiver, each differing in effectiveness, obfuscation level, and the required hardware, software, and planning. An asynchronous attack involves broadcasting synthesized or pre-recorded GNSS signals, often preceded by a jamming period to disrupt reception of genuine signals. This disruption can cause the receiver to lose its established correlation and lock onto the counterfeit signal that follows. A synchronous spoofing attack, on the other hand, starts by aligning the counterfeit signal very precisely with the genuine signal, achieving a time accuracy of a few nanoseconds and matching Doppler shifts by a few Hertz. Once the receiver locks onto the more intense counterfeit signal, its properties are gradually altered, guiding the receiver to maintain correlation with the altered signal. Additionally, highly sophisticated spoofing methods might use multiple synchronized simulators, an array of transmitting antennas, or even drones to mimic the angles of signal arrival. However, these attacks are detectable through extensive signal analysis. Unfortunately, most civil-use receivers in the energy grid lack even basic defenses against simple attacks.

In [Can+24], authors subtly shift the broadcast time in the GNSS simulation to manipulate the time server, ensuring the signal remains self-consistent.

[2]Part of this work was peer-reviewed and published in [P2]

Attack Scenarios. The techniques encompass both asynchronous and synchronous spoofing methods.

(1) *Single-Constellation Asynchronous Attack*: Involves applying a constant time shift to a range of timestamps. This attack can be formalized as follows:

$$t' = \begin{cases} t + \Delta t & \text{if } t_{\text{start}} \leq t \leq t_{\text{end}} \\ t & \text{otherwise} \end{cases} \tag{6.1}$$

Here, t represents the original timestamp and t' denotes the shifted timestamp. The parameter Δt is the constant time shift applied during the attack, and t_{start} and t_{end} are the start and end times of the attack window, respectively.

(2) *Synchronous Multi-Constellation Attack*: Applies a time shift that increases linearly over time, starting from an initial point. The mathematical representation is given by:

$$t' = \begin{cases} t + \Delta t_{\text{start}} + r \cdot (t - t_{\text{start}}) & \text{if } t_{\text{start}} \leq t \leq t_{\text{end}} \\ t & \text{otherwise} \end{cases} \tag{6.2}$$

In this context, t is the original timestamp and t' is the altered timestamp. The term Δt_{start} refers to the initial time shift applied at the start of the attack, while r represents the rate of change in the time shift, measured in nanoseconds per second. The attack is applied within the interval $[t_{\text{start}}, t_{\text{end}}]$.

(3) *Stealthy Synchronous Multi-Constellation Attack*: applies a very slow, nearly undetectable time shift that accumulates over time. This attack can be expressed as:

$$t' = \begin{cases} t + r_s \cdot \left(\frac{t - t_{\text{initial}}}{3600}\right) & \text{if } t_{\text{initial}} \leq t \leq t_{\text{end}} \\ t & \text{otherwise} \end{cases} \tag{6.3}$$

Here, t is the original timestamp and t' is the shifted timestamp. The parameter r_s indicates the slow rate of change in the time shift, measured in microseconds per hour. The attack starts at t_{initial} and continues up to t_{end}, with the time shift gradually increasing over this period.

Feasible Transformations. In our experimental setup, we leverage the results presented in [Can+24] where authors successfully conducted time spoofing attacks on a commercial time server. As described, they used GPS signals for single-constellation attacks and included Galileo signals for multi-constellation attacks. The maximum

achievable time drift is limited by the maximum steering speed of the local oscillator. Based on these, we specify the following transformations as feasible:

(1) A simple asynchronous attack was able to adjust the time server's output by a few minutes within an attack duration of less than 5 minutes.

(2) The synchronous multi-constellation attack managed to shift the time server from the true time at a rate of about 50 nanoseconds per second, resulting in a time offset of 10 microseconds in under 5 minutes.

(3) A stealthier synchronous multi-constellation attack proved capable of subtly altering the time without noticeable changes in the announced clock properties, achieving a time shift of about 1 microsecond per hour.

GNSS Spoofing-Based Data Poisoning in the problem space. We investigate the impact of realistic GNSS spoofing attacks on the SV IDS. The dataset includes time-series features, specifically `timestamp` and `timestamp_delta`, which capture absolute timestamps and time intervals between consecutive measurements, respectively. These features are critical for accurate sequence integrity and data processing in time-sensitive systems such as energy grids. We implement the three types of GNSS spoofing attacks: Asynchronous Attack, Synchronous Attack and Stealthy Attack. The number of modified feature vectors is 68303 out of 84000.

Implementation and Dataset Preparation. We apply these GNSS spoofing attacks on the SV dataset consisting of 52 features, including absolute timestamps and deltas as the most important features according to XAI methods. The dataset is preprocessed to handle missing values with column means. We perturb the `timestamp` feature according to the specified attack scenarios and recalibrate the `timestamp_delta` feature to maintain temporal consistency: $\texttt{timestamp\_delta}[i] = \texttt{timestamp}[i] - \texttt{timestamp}[i-1]$. For each attack type, we identify the affected feature vectors based on the specified time window and calculate the number of perturbed vectors relative to the total training set size.

Experimental Pipeline. The experimental pipeline involves: (1) Cleaning and splitting the dataset into training and testing subsets. (2) Training a RF classifier on clean data to establish baseline performance metrics. (3) Applying the asynchronous, synchronous, and stealthy attacks to timestamp data and modifying the `timestamp_delta` individually. (4) Retraining the model on poisoned data to evaluate the degradation in performance. (5) Summarizing attack-specific results and comparing them with the baseline to understand the impact of different types of GNSS spoofing.

Tab. 6.1: Results for Random Forest (Accuracy in %) under GNSS time spoofing attack in the SV use case.

Protocol	Original	Asynchronous	Synchronous	Stealthy
SV	99.52	97.78	93.30	97.71

Summary. The results in Table 6.1 reveal that the model shows decrease in accuracy but experiences varied degradation depending on the type of spoofing attack. The synchronous attack poses the most significant challenge, reducing accuracy to 93.30%, while asynchronous and stealthy attacks result in comparatively moderate reductions to 97.78% and 97.71%, respectively. These findings highlight the model's susceptibility to temporal perturbations and the varying levels of indiscriminate adversarial effectiveness of different attack types.

7 Attacking Confidentiality

Confidentiality attacks target the unauthorized extraction of sensitive information from learning-based systems, including proprietary model parameters and characteristics of the training data. In smart grid contexts, such attacks pose risks not only to intellectual property but also to privacy, regulatory compliance, and downstream system security. This Chapter investigates confidentiality breaches enabled by explainability mechanisms, focusing on model stealing and covert data exfiltration attacks. By exploiting explanation outputs as auxiliary signals, attackers are shown to accelerate model extraction and enable transferable attacks. The Chapter further introduces steganographic adversarial attacks as a means of covertly exfiltrating model secrets within smart grid communication data, highlighting a previously unexplored confidentiality risk.

7.1 Model Stealing S7 IDS

Tab. 7.1: Detection Results (%) for the S7Comm IDS.

Task	Model	Original	1%	2%	3%	4%	5%
S7Comm IDS	RF	98.86	92.39	86.89	70.06	68.44	68.44
	Ensemble	98.38	93.85	86.73	72.81	68.28	68.44

In our S7Comm intrusion detection scenario, the attacker's goal is to manipulate the training data through replay attacks, so that the model performance decreases during real-time analysis[1]. The IDS prototype we developed focuses on detecting replay attacks within S7Comm traffic, particularly in an electrical substation network with different protocols, including Siemens PLC communications over TCP port 102. The system starts with a preprocessing phase where network capture files are transformed into structured data formats. This allows for the extraction of both statistical and payload-based features: all possible combinations of 15 ending bytes of the packet raw data, packets per minute, and average payload length. These features help distinguish between normal operations and potential replay attacks. As mentioned, most features are dynamically generated by identifying unique packet

[1]Part of this work was peer-reviewed and published in [P2]

endings of a specific length. After grid search for hypermarameters optimization, we identified 15 bytes as the best-performing value for detection. We end up with a dataset containing over 200 features. The packets get aggregated together by using a sliding window technique, that is, each feature vector represents a period of one minute. Then, the statistical feature *packets per minute* gets computed according to this window. Finally, the features that arise from packet endings get assigned a value depending on the occurrences of each packet ending within a given minute of traffic. Two thirds of the sliding windows created from port 102 include S7Comm traffic. After processing, we obtain 773 feature vectors of normal traffic and 1284 of traffic with replay attacks. A RF model is trained using data from normal and attack traffic —leaving 30% for testing—, enabling the system to establish a baseline of typical network behavior and replay attack scenarios. The model achieves an accuracy of 98.86% on clean data.

Threat Model. For evasion, the attacker attains a Man-In-The-Middle position between a PLC (S7-1500) for controlling a photovoltaic panel and PCS7 (Master); we assume that she knows the feature space strategy of the model (e.g., via an insider). Furthermore, we assume that the attacker is able to monitor the output of the model (e.g., by observing if the server has been restarted or turned off as an incident response). In contrast with the previous use cases (white box), here we present a grey box scenario.

Available Transformations. Our objective is to use XAI to reveal fragile features and spurious correlations [Arp+22]. Once these are identified, we want to exacerbate the issue via feasible data poisoning. In pursuit of this, it is essential to assess the degree of allowable modifications to the feature representation of samples. These modifications must be achievable through transformations that are acceptable within the problem space [Pie+20]. In this case, we simulate the increase in occurrences of specific feature endings through realistic replay attacks, adapting the number of packets per minute accordingly. This addresses the concept of side-effect features [Pie+20], i.e., the byproduct of the inverse feature-mapping problem, enabling us to prove necessary and sufficient conditions for the existence of problem-space data poisoning attacks.

Preserved Semantics. Unlike feature attacks, in this scenario the poisoned data is embedded in real objects (i.e., S7Comm benign, valid packets), thereby inherently preserving their functionality.

Plausibility and Robustness to Preprocessing. The attacker must ensure that packets are accepted as normal communication and that they do not raise any alarm during data collection. That is, packets must be replayed in a way that they do not

cause disturbances or major network traffic increases as per normal PLC operations. For this reason, we limit the perturbation magnitude (i.e., number of packets that the attacker replays to increase the occurrences of benign payloads) to a maximum of 5% increase from the original feature, which we consider to be conservative. Furthermore, if during a given minute of traffic there were no S7comm packets, we do not replay any to avoid easy identification of outliers.

Methodology. First, we create a surrogate model of the original classification model by creating a new labeled dataset. For this, we simulate a process were the attacker monitors the ongoing normal traffic while replaying benign packets (e.g, stealthy operations such as read values without creating disruptions). The attacker is able to label instances of traffic due to observing if there is an incident response (e.g., network reset) or not. This way, the attacker constructs her own model that mimics the original. The attacker then proceeds with further analysis by fitting the new labeled dataset to a Multi-Layer Perceptron (MLP), training it for 50 epochs. At this point, the attacker integrates XAI techniques, with the objective of obtaining insights into which features most significantly influence the detection decisions. Aggregated importance across XAI methods (*SHAP*, *LRP*, *LIME*) is calculated as follows: Each feature's score is assigned based on its rank within the top features of each method:

$$\text{Score}_{\text{feature}} = \text{Number of Features} - \text{Rank} \tag{7.1}$$

For instance, in a top-5 list: Rank 0 (highest) receives a score of 5 and Rank 1 receives a score of 4, and so on. Scores for each feature are aggregated across all methods:

$$\text{Aggregated Importance}_{\text{feature}} = \sum_{\text{methods}} \text{Score}_{\text{feature}} \tag{7.2}$$

The final output highlights the features that are consistently ranked highly across all methods, emphasizing their overall significance , as can be seen in Figure 7.1. The most important feature is packets per minute, followed by a series of key packet payload endings. Now the attacker understands what are the features that should be targeted in order to decrease the performance of the model. Finally, during a data collection exercise, the attacker replays the benign packets (with a specific packet payload ending, ranked as the second most important feature) to decrease the performance of the model at inference time. By reverse engineering the protocol, we know that the most relevant packet ending relates to a read operation without much security implications (a spurious correlation [Arp+22]), that is, the replay of these specific packets do not cause anomalies, however, when the model is deployed, the performance decreases substantially depending on the perturbation magnitude as depicted in Table 7.1.

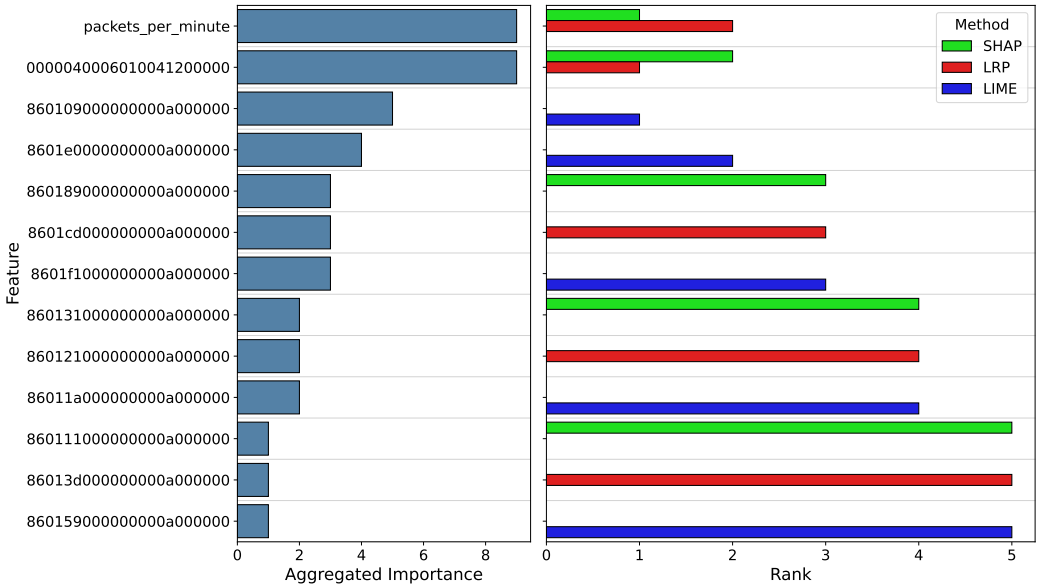


Fig. 7.1: Feature importance comparison in the S7Comm IDS use case. Left: Features by aggregated importance across methods. Right: Features ranked by individual XAI methods.

7.1.1 Summary

The results show a decrease in detection of replay attacks even in cases where the perturbation magnitude is minimal. In Table 7.1, the original RF model compares to the ensemble —a majority-voting IDS by combining RF, MLP and Support Vector Machine (SVM)— algorithms under different perturbation magnitudes against the features vectors that contained at least one S7Comm packet (66.66% of the dataset). By altering important features' values through real world perturbations, attackers can significantly influence the model's fit, pulling the decision curve toward these outliers.

7.1.2 Another S7 Problem Space

In related work [P9], the S7 protocol IDS problem space is characterized diffferently. In this subsection, we show preliminary attack detection results with a proof-of-concept implementation of a different IDS and report feature importance. We describe how the presented dataset in [P9] can be used to train ML models.

As a pre-processing step, we remove feature columns that exhibit a single unique value (e.g., *facility*, *tag*, and *severity* from the log messages) or that may lead to spurious correlations [Arp+22] specific to the use case at hand **[P1]** (e.g., *hostname*,

IP address, *MAC address*) to ensure our models are trained on meaningful and non-redundant information. Our IDS instead leverages process values extracted from the S7 application layer protocol, log messages, and directly from the Supervisory Control and Data Acquisition (SCADA) system for detection of the data modification attacks.

Our proposed IDS aggregates the data from the three sources into fixed time windows and labels each window as an attack if any packet within the window is abnormal (i.e., not labeled as `normal`). We explored multiple candidate time windows (5, 10 and 15 seconds) as a hyperparameter, and through grid search, we identified an optimal aggregation window of 10 seconds, which yielded the best F1 macro score.

In parallel, we tuned the hyperparameters of three baseline classifiers – Logistic Regression, Decision Tree, and RF – using Grid Search with the F1 macro score as the evaluation metric. These tuned models were then combined into a hard-voting ensemble, which, when evaluated on the optimal 15-second aggregation configuration, achieved an overall F1 macro score of 90% on the test set (99% on attack and 80% on normal samples). However, the RF model was able to achieve a F1 macro score of 95% on the test set (100% on attack and 91% on normal samples), outperforming the ensemble. These results are presented in table 7.2.

Tab. 7.2: Confusion matrix for each model using a 15s time window (Positive: Attack, Negative: Normal).

Model	FN	FP	TN	TP
Logistic Regression	0	6	0	187
Decision Tree	0	2	4	187
Random Forest	0	1	5	187
Ensemble	0	2	4	187

Now, we use explanations to better interpret the best model (RF) via feature importance. The built-in feature importance method for RF is Gini Importance, also known as Mean Decrease in Impurity (MDI) [Bre01]. This method relies on the reduction in impurity achieved by each feature during the construction of the decision trees. The average importance in table 7.3 is scaled by 10^3 for clarity.

The feature ranking reveals the most critical features for detecting cyberattacks in our smart grid dataset. Notably, the feature `X_new_value` from the logs data achieved the highest average normalized importance. Following this, features `X_old_value` (also from logs data) and `in_batt_actual_charge_power` from process data rank highly.

Tab. 7.3: Top 10 Features for the best-performing model (Random Forest)

Feature	Avg. Importance	Source
X_new_value	97.0	Logs
X_old_value	95.9	Logs
in_batt_actual_charge_power	87.6	Process
in_batt_temperature	86.8	Process
in_pv_cell_temperature	83.5	Process
in_pv_wind_speed	76.1	Process
in_pv_poa_direct	54.8	Process
in_batt_current	45.3	Process
in_batt_state_of_charge	44.2	Process
in_pv_inverter_ac_power	43.3	Process

Overall, the RF feature importance ranking indicates that a combination of energy measurements and log metadata collectively contributes to a robust detection mechanism, providing an advanced view of system behavior under potential cyberattack scenarios.

7.2 Data Exfiltration for Model Stealing in MMS IDS

Modern organizations face an ever-growing threat landscape in which adversaries continuously refine covert techniques for data exfiltration. Traditional Data Loss Protection (DLP) solutions rely on pattern matching and business rules to block sensitive data transfers [Kim21], but steganographic channels (especially those embedded within image or multimedia streams) evade signature-based detection [Cox+07; ML05]. Meanwhile, ML–based Network Intrusion Detection Systems (NIDS) have gained traction due to their ability to detect complex attack patterns in high-volume traffic [Hol+21]. To leverage state-of-the-art vision models, some NIDS implementations convert raw packet streams into grayscale images, enabling learning-based classifiers to distinguish benign from malicious flows [Swa+24]. Converting raw network packets into fixed-size grayscale images enables practitioners to leverage the advances in computer-vision architectures without labor-intensive feature engineering, effectively turning complex temporal and protocol-specific patterns into spatial textures that AI methods can readily learn. By representing packet streams as two-dimensional arrays of byte values, one gains a uniform input format that simplifies model pipelines and makes it straightforward to apply powerful explainability techniques for visual analysis and interpretation, which are otherwise difficult to adapt to purely tabular network features.

This architectural shift, however, introduces a novel attack surface: adversaries can hijack the image representations used for threat detection to exfiltrate proprietary

or sensitive information without raising alarms. Specifically, an Advanced Persistent Threat (APT) with insider access (via malware or compromised credentials) can embed model details (e.g., network architecture, weights, hyperparameters) into the Least Significant Bit (LSB) of these images. Since the CNN used for classification is blind to such minor pixel-level modifications, the images can be disseminated through normal intelligence-sharing channels (e.g., collaborative threat intelligence or cloud storage) and later retrieved by the adversary. We conduct[2] an extensive empirical evaluation, measuring classification accuracy, image quality, and payload capacity for different approaches. We show that while LSB embedding and Pseudorandom Number Generator (PRNG)-based methods preserve model accuracy and image fidelity even for large payloads, the saliency-guided layered approach suffers from scalability issues and increased distortion.

In the present Section, we make the following contributions:

1. We implement and evaluate learning-based NIDS that transforms raw traffic captures into grayscale images, achieving high classification accuracy on IEC 61850 energy critical infrastructure datasets. Our goal is not to surpass detection baselines or achieve state-of-the-art accuracy; rather, we construct a practical image-based detection environment to rigorously demonstrate steganographic exfiltration as a proof-of-concept in safety-critical cyber-physical systems.

2. We introduce a steganographic model-stealing scenario in which an APT embeds ML model secrets within network-traffic images, enabling covert exfiltration without degrading classifier performance.

3. We propose and compare three embedding pipelines: (a) a baseline LSB replacement approach, (b) a dual-layer PRNG-controlled scheme secured by cryptographic hashing, and (c) a saliency-guided layered technique that uses Integrated Gradients (IG) or SHAP as XAI methods to identify high-saliency pixels.

4. We outline practical defenses against steganographic exfiltration in image-based NIDS setups and release our code and artifacts to foster further research[3].

[2] Part of this work was peer-reviewed and published in [P5]
[3] https://github.com/gus5298/stego_stealing

7.2.1 Motivation

Smart grids and critical infrastructures increasingly rely on ML–based NIDS for real-time anomaly detection [Zha+24b]. Protocol standards such as IEC 61850 for energy critical infrastructure facilitate interoperability but also produce high-volume, structured traffic that can be readily converted into image representations [Swa+24]. For example, electric substation traffic often contains packet sequences corresponding to monitoring and control events. By streaming raw bytes into fixed-size buffers, each packet stream yields a grayscale image in which spatial patterns correlate with benign or malicious behavior. While this approach has significantly improved detection rates, it also gives rise to a stealthy exfiltration vector: an insider or compromised endpoint can embed stolen model parameters directly into these images via LSB manipulation. Once images are shared externally (e.g., as part of threat intelligence exchanges) an adversary can recover the hidden payload without triggering any alarms.

In many Security Operation Center (SOC) environments, analysts subscribe to Indicators of Compromise (IoC) [Kim21] feeds and share threat data across organizations. These channels often lack content verification for embedded metadata, assuming that shared images are benign. An APT can exploit this trust to maintain long-term access, exfiltrate high-value assets (e.g., trained models that encode proprietary feature engineering and vendor Intellectual Property), and potentially enable more potent adversarial attacks in the future (e.g., by generating effective adversarial examples) [GBC18]. Given the increasing adoption of image-based NIDS [Swa+24], defending against steganographic exfiltration is an urgent challenge.

7.2.2 Background and Related Work

In this Section, we review relevant prior art on network steganography, model stealing attacks, and image-based anomaly detection for NIDS.

Network Steganography

Steganography refers to techniques for hiding information within a benign cover medium so that the presence of the hidden data is undetectable [Cox+07; Fri09]. Early work by Murdoch and Lewis demonstrated covert channels in network protocols by manipulating packet timings and header fields [ML05]. Subsequent approaches embed data within payloads of HTTP, DNS, and VoIP streams [Fah+24].

Network steganography poses significant challenges for DLP and IDS tools because there are no easily recognizable signatures. In APT campaigns, groups such as Duqu, Regin, and Turla have used steganographic techniques (e.g., hiding payloads in image files posted to social media) to maintain stealthy command-and-control channels [Sym11; Fir14; Kas18]. Defending against such channels requires advanced anomaly detection, content fingerprinting, or active probing, which can be resource-intensive and prone to false positives.

Model Stealing

Model stealing (also called model extraction) is an attack in which an adversary attempts to reconstruct a target model's functionality or parameters by observing its outputs on chosen inputs [Tra+16]. Application Programming Interface (API)-based attacks query a black-box model to infer decision boundaries. In the NIDS context, however, models are usually not exposed via remote APIs, so adversaries resort to side channels, insider threats, or malware to access model files directly [Fan+23]. Stolen models can reveal proprietary feature engineering, training data biases, and enable the generation of adversarial examples [Pap+17]. Prior work has shown meta-learning frameworks that mimic victim models with minimal queries [Fan+23], but these approaches focus on inference queries, not data exfiltration. Our work differs by embedding stolen model secrets directly into network-traffic images via steganography, rather than relying on remote querying **[P2]**, as API-exposed ML-models are not common in critical infrastructure.

Image-Based Anomaly Detection

Recent approaches have leveraged computer vision techniques to detect malicious network traffic by converting packet streams into images. For instance, Swain et al. represent network packets as 235×235 grayscale images by mapping 235-bit feature vectors into square grids [Swa+24]. Their framework further demonstrates how adversarial machine learning techniques from computer vision (CV) can transfer to the network domain. Holland et al. [Hol+21] highlight that standard ML benchmarks are image-oriented, motivating image representations of non-visual data. While image-based NIDS achieve high classification accuracy, they also inherit vulnerabilities from CV models, including susceptibility to gradient-based evasion (adversarial examples) [GBC18]. We build upon these insights and show that hiding data in the same images used for detection remains transparent to the CV model yet threatens model confidentiality.

Steganography in Adversarial Machine Learning

A large body of vision research studies *adversarial examples* [AM18]: small, structured perturbations that induce misclassification while remaining visually subtle. Most such methods optimize for *integrity* violation (i.e., changing a model's decision) and, increasingly, for robustness to common defenses [QQ20]. In contrast, our objective is *confidentiality*: we embed model secrets into image-based network representations so that the in-house detector's predictions remain unchanged, emphasizing payload capacity, recoverability, and imperceptibility rather than attack success rates.

Superpixel- or attention-guided attacks explicitly move away from global, pixel-wise noise. In [Don+20], perturbations are confined to salient regions and made constant within superpixels, a design shown to preserve attack efficacy while improving robustness to image-processing defenses and steganalysis.

A complementary line connects adversarial optimization with steganography. The framework in [Mo+19] starts from a modern embedding cost and iteratively refines it using gradients from steganalytic CNNs, yielding payload placements that are empirically harder to detect. Similarly, [Zen+23] borrows the steganographic *embedding suitability map* to modulate adversarial perturbations, concentrating energy in statistically inconspicuous regions to boost undetectability without sacrificing attack strength.

Our method differs in purpose and evaluation: we use LSB/PRNG-based embedding (and a limited saliency-guided variant) to exfiltrate model secrets while preserving NIDS decisions, and we quantify imperceptibility together with unchanged classification accuracy.

7.2.3 Threat Model

Adversary Capabilities

We assume an APT has already compromised an internal host within a corporate or critical-infrastructure network. The adversary has read/write access to the ML pipelines and can modify the code that generates image representations of packet streams. The adversary does not require direct external connectivity for the model; instead, they rely on normal intelligence-sharing channels (e.g., sending images to a third-party vendor for collaborative threat analysis). We also assume the adversary has obtained the trained model's files (architecture, weights, optimizer

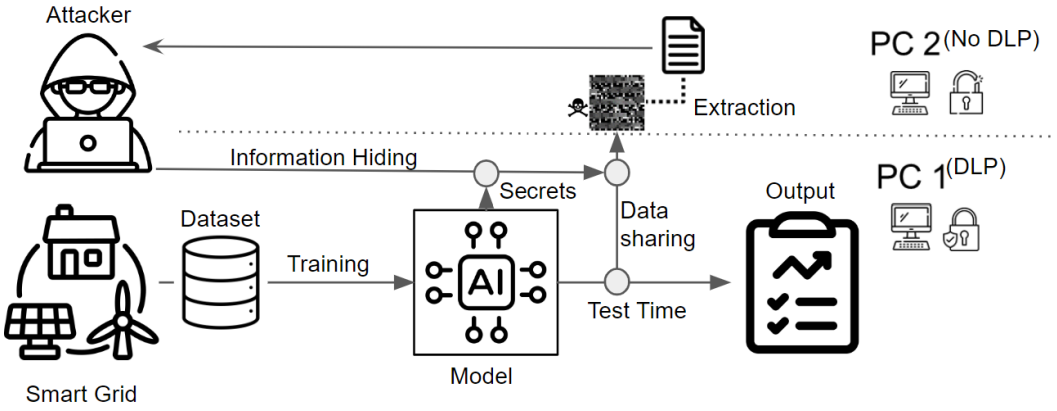


Fig. 7.2: Visualization of the threat model.

state) through insider access or malware. Their goal is to exfiltrate these secrets without detection.

Adversarial Goals

The primary goal is to extract proprietary ML model information (including network architecture, hyperparameters, weight tensors, and optimizer state), embed these secrets within benign-looking images, and transmit the modified images through existing workflows without triggering IDS, DLP, or human scrutiny. The secondary goal is to maintain model integrity and classification performance; that is, the classifier used at the victim's site should continue to perform as before despite the hidden payload.

Assumptions and Scope

We assume:

- The NIDS pipeline converts raw packet PCAPNG files into fixed-size grayscale images (128×128 or 64×64) as input to a CNN or RF classifier.
- Intelligence sharing is part of normal operations, where images are sent offsite for collaborative analysis.
- The adversary controls or influences one or more recipients in the sharing chain (e.g., a compromised third-party vendor) and can retrieve the stego images.

- Standard DLP/IDS tools in place do not perform pixel-level forensic analysis of shared images.

Secret Taxonomy and Bit-Lengths

Let S denote the full set of secrets to be exfiltrated. We partition S into classes c with corresponding bit-length ℓ_c, as shown in Table 7.4.

Tab. 7.4: Taxonomy of Model Secrets

Class	Description	ℓ_c (bits)
Architecture	CNN architecture: number of layers, layer types, activation functions, dropout configurations	$\approx 10^3$
Hyperparameters	Learning rate, batch size, optimizer settings, kernel sizes, number of filters	$\approx 10^2$
Weights & Biases	Quantized weight and bias tensors (e.g., 100K–500K parameters, 8–32 bits per parameter)	$\approx 10^7$
Optimizer State	Adam optimizer momentum vectors (m, v), learning rate schedule	$\approx 10^7$
Training Metadata	Dataset identifiers (e.g., FDIA, MMS), class labels, split ratios, preprocessing details	$\approx 10^3$

Embedding Capacity of Image Cover Channels

Consider a set of N cover images, each with dimensions $H \times W$ pixels and L least significant bits (LSBs) available per pixel. The total embedding capacity is:

$$C_{\text{total}} = N \times (H \times W \times L) \quad \text{bits.}$$

For our experiments, $H = W = 128$ or 64, $L = 1$, and N varies depending on the dataset size.

Payload Selection under Capacity Constraints

Given a budget C_{total}, the adversary selects a subset $\mathcal{S}' \subseteq \{c\}$ so that

$$\sum_{c \in \mathcal{S}'} \ell_c \leq C_{\text{total}}.$$

Define $U(c)$ as the utility (e.g., reconstruction priority) of secret class c. A greedy selection algorithm sorts classes by increasing ratio $\ell_c/U(c)$ and includes each class until capacity is exhausted.

Stealth and Fidelity Metrics

We quantify exfiltration efficacy using:

- **Reconstruction Fidelity**:

$$F = \frac{|\text{Recovered bits}|}{|\text{Original bits}|} \in [0, 1].$$

- **Detection Probability**:

$$P_{\text{det}} = f\big(C_{\text{img}}, \Delta\text{PSNR}, \Delta\text{SSIM}\big),$$

  where ΔPeak Signal-to-Noise Ratio (PSNR) and ΔStructural Similarity Index (SSIM) measure cover distortion.

The adversary aims to maximize F subject to P_{det} remaining below a detection threshold.

7.2.4 Intrusion Detection and Secrets' Embedding

This Section details data preparation, model training, and the three steganographic pipelines.

Data Collection and Image Preparation

We evaluate on two publicly available smart grid datasets:

1. **False Data Injection Attack (FDIA)** [Tan+24; AK20; Sma25]: The well-known Electric Power and Intelligent Control (EPIC) false data injection attack dataset [Sma25] with two classes: Normal and Attack. The EPIC dataset's FDIA1 subset captures network traffic where the SCADA-to-PLC MMS "sync" command's boolean flag is maliciously spoofed to False, causing the plant's synchronization process to fail. Packets are captured in PCAPNG format, each record containing full packet headers and payload. The authors do

not present an IDS. Therefore, to establish a tabular baseline, we train three simple classifiers —a RF, an SVM, and a simple 1D-CNN— on the FDIA labeled tabular dataset (47 FDIA vs. 111 normal samples) provided by the authors in an open-source repository[Sma25], using default network-traffic features. Both the RF and the SVM achieve 99.37% accuracy with near-perfect precision and recall for both classes (FDIA: 0.99 F1, Normal: 1.00 F1). The 1D-CNN, by contrast, reached 91.14% accuracy, showing strong normal detection (1.00 recall) but lower FDIA recall (0.70).

2. **IEC 61850 MMS** [Eyn+24]: A multi-class tabular dataset containing Normal, Faulty, and Attack sessions for smart grid communications. The authors report 100% detection accuracy in a tabular-data approach through an ensemble of RF and Extremely Randomized Trees.

For simpliclty, we will refer to the datasets above as FDIA and MMS respectively, even though both contain MMS traffic.

Each PCAPNG file is streamed into a fixed-size buffer of 128×128 = 16,384 bytes to form a single grayscale image. Using PyShark, we extract raw packet bytes (headers + payload). We sequentially write byte values into the image buffer row-wise; once the buffer fills, the image is saved, and a new buffer is initialized (Figure 7.3). This process preserves temporal packet ordering and standardizes spatial dimensions for CNN input. For the MMS dataset, images are resized further to 64×64 during preprocessing to reduce computational cost.

Learning Models

We compare three classifiers: a CNN, a RF, and a one-class Support Vector Machine (OC-SVM). Each model uses the same image inputs but different preprocessing:

Preprocessing. All images are converted to single-channel (grayscale), resized to 128×128 for FDIA and 64×64 for MMS, normalized to $[0, 1]$, and batched in sizes of 32.

CNN Architecture. Our CNN consists of three convolutional blocks:

- **Conv Block 1**: Conv2D (32 filters, 3×3) → ReLU → BatchNorm → MaxPool (2×2) → Dropout (0.2).

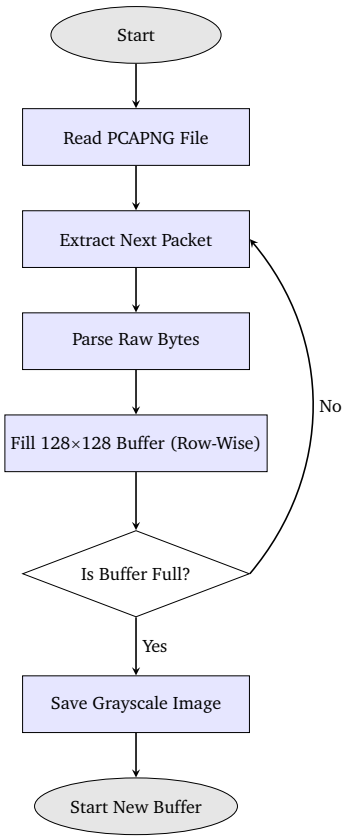


Fig. 7.3: Flowchart of the image creation process.

- **Conv Block 2**: Conv2D (64 filters, 3×3) → ReLU → BatchNorm → MaxPool (2×2) → Dropout (0.3).

- **Conv Block 3**: Conv2D (128 filters, 3×3) → ReLU → BatchNorm → MaxPool (2×2) → Dropout (0.4).

A fully-connected layer follows: Flatten → Dense (256) → ReLU → Dropout (0.5) → Output Dense (1) with Sigmoid activation for binary tasks or Dense (3) with Softmax for three-class classification. We use the Adam optimizer (lr = 10^{-4}), binary cross-entropy or categorical cross-entropy loss as appropriate, and early stopping on validation loss.

Random Forest. We flatten each grayscale image into a 1D feature vector ($H \times W$ features). The RF consists of 100 decision trees with maximum depth chosen via cross-validation. We use Gini impurity as the split criterion and train in a supervised manner on labeled data.

One-Class SVM. OC-SVM is trained solely on the Normal class. We use an RBF kernel and tune the ν parameter (upper bound on the fraction of training errors) via grid search. At inference, samples that fall outside the learned boundary are flagged as anomalies.

For both datasets, we partition images into training (70%), validation (15%), and testing (15%) sets, ensuring class balance. The total number of images per dataset is approximately 4800 for FDIA (2368 Normal, 2020 Attack, plus held-out) and 2400 for MMS (of which 1000 per class).

Table 7.5 presents image-based accuracies:

- **FDIA**: CNN achieves 100% accuracy; RF achieves 97%; OC-SVM achieves 74%.
- **MMS**: CNN achieves 93% accuracy on three classes; RF achieves 67.68%.

Since the CNN (MMS and FDIA) and RF (FDIA) (see Table 7.5) are the best-performing ones, we disregard the rest for the experimental evaluation.

While ML tabular models (e.g., RF/SVM), reported in Section 7.2.4, achieve near-perfect or perfect detection on FDIA (aproximately 99%) and MMS (100%), our image-based CNN matches the tabular baseline on FDIA but drops to 93% on MMS, illustrating a trade-off between leveraging visual representations and maintaining raw classification accuracy.

Note. The objective of our image-based detection is not to challenge existing detection baselines or obtain state-of-the-art accuracy, but rather to build an environment for image steganography as proof of concept in a use case relevant to critical infrastructure.

Tab. 7.5: Classification accuracy on images (without steganography).

Dataset	Model	Classes	Accuracy
FDIA	CNN	Normal, Attack	100%
	Random Forest	Normal, Attack	97%
	One-Class SVM	Normal, Attack	74%
MMS	CNN	Normal, Faulty, Attack	93%
	Random Forest	Normal, Faulty, Attack	67.68%

Baseline LSB Embedding

Least significant bit (LSB) steganography in grayscale images replaces the LSB of selected pixel intensities with payload bits. For an 8-bit pixel P, embedding one bit m yields:

$$P' = (P\&\sim 1)|m,$$

and extraction is $m = P' \& 1$ [Fri09]. The total capacity per image is $H \times W$ bits when using one LSB per pixel. In our baseline pipeline, we embed random alphanumeric strings of length 10, 100, and 1000 characters (80, 800, and 8000 bits) into each image's LSBs. We then evaluate whether the CNN and RF classifiers (trained on unmodified images) maintain their original accuracy on stego images. We also compute image quality metrics—MSE, PSNR, SSIM, Discrete Cosine Transform (DCT) differences—to quantify distortion.

Dual-Layer PRNG-Based Embedding

The baseline LSB approach is straightforward but easily reversible by any adversary who suspects steganography. To improve concealment and scalability, we adopt a dual-layer PRNG framework secured by cryptographic hashing:

Seed Generation. We compute a 32-bit seed by taking SHA-256 over the concatenation of the image's raw pixel bytes and a shared secret key. This seed is then embedded into a small set of reserved pixels (150 for MMS, 300 for FDIA) chosen pseudorandomly by a secondary PRNG seeded with an ephemeral key.

Payload Embedding. Upon decoding and verifying the seed, the primary PRNG initializes with the same seed to pseudorandomly select non-reserved pixels for embedding message bits via LSB substitution. We embed 8 bits per character sequentially across images. This ensures that without knowledge of the secret key and PRNG procedure, an external observer cannot identify which pixels carry the payload.

Figure 7.4 illustrates the embedding process and Figure 7.5 shows the decoding. Tables 7.6 and 7.7 summarize classification accuracy and image quality across payload sizes for both datasets and both models.

Saliency-Guided Layered Embedding

To minimize the number of pixels tampered while ensuring robust recovery, we design a two-layered embedding scheme guided by model saliency:

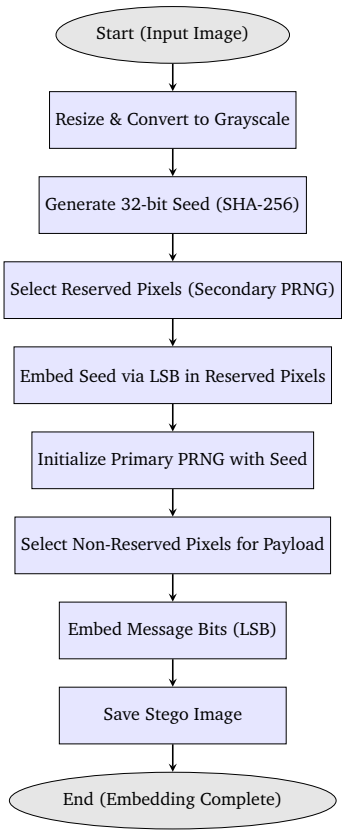


Fig. 7.4: Flowchart of PRNG-based embedding process.

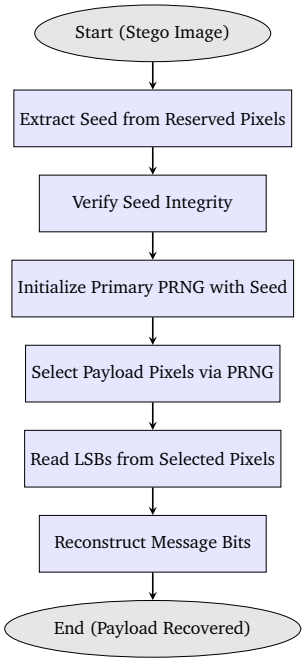


Fig. 7.5: Flowchart of PRNG-based decoding process.

Tab. 7.6: Classification accuracy under PRNG-based steganography (FDIA, MMS).

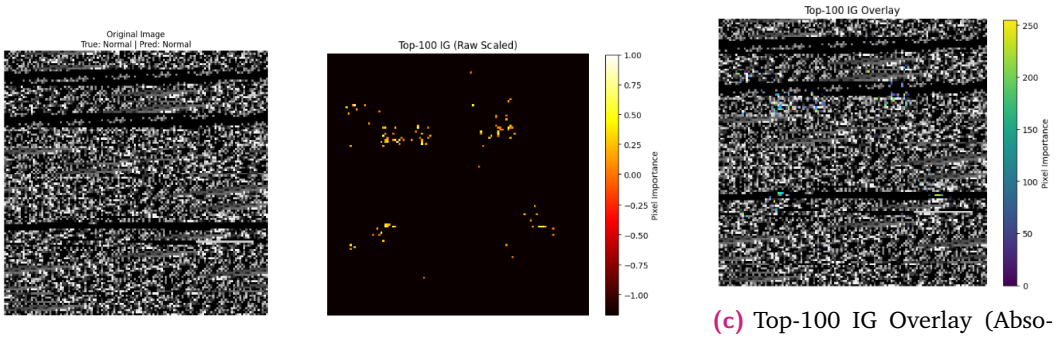
Dataset/Model	Characters	Class	✓	×	Acc. (%)
FDIA / CNN	10	Normal	2368	0	100.00
		Attack	2020	2	99.90
	100	Normal	2368	0	100.00
		Attack	2020	2	99.90
	1000	Normal	2368	0	100.00
		Attack	2019	3	99.85
FDIA / RF	10	Normal	2368	0	100.00
		Attack	2020	2	99.90
	100	Normal	2368	0	100.00
		Attack	2020	2	99.90
	1000	Normal	2368	0	100.00
		Attack	2019	3	99.85
MMS / CNN	10	Normal	764	8	98.97
		Attack	135	9	93.75
		Faulty	873	21	97.64
	100	Normal	764	8	98.97
		Attack	135	9	93.75
		Faulty	873	21	97.64
	300	Normal	764	8	98.97
		Attack	135	9	93.75
		Faulty	873	21	97.64

Tab. 7.7: Image quality metrics under PRNG-based steganography (FDIA, MMS).

Dataset / Model	Class	Characters	Avg. MSE	Avg. PSNR (dB)	Avg. SSIM	DCT (Mean / Max)
FDIA / CNN	Normal	10	0.00	72.86	1.00	0.05 / 0.24
		100	0.03	64.15	1.00	0.13 / 0.94
		1000	0.24	54.31	1.00	0.39 / 8.94
	Attack	10	0.00	72.85	1.00	0.05 / 0.24
		100	0.02	64.16	1.00	0.13 / 1.01
		1000	0.24	54.31	1.00	0.39 / 9.58
FDIA / RF	Normal	10	0.00	72.85	1.00	0.05 / 0.24
		100	0.03	64.15	1.00	0.13 / 0.94
		1000	0.24	54.31	1.00	0.39 / 8.94
	Attack	10	0.00	72.86	1.00	0.05 / 0.25
		100	0.02	64.16	1.00	0.13 / 1.01
		1000	0.24	54.31	1.00	0.39 / 9.60
MMS / CNN	Normal	10	0.01	66.85	1.00	0.09 / 0.44
		100	0.10	58.11	1.00	0.25 / 1.41
		300	0.29	53.45	1.00	0.43 / 3.68
	Attack	10	0.01	66.83	1.00	0.09 / 0.45
		100	0.10	58.11	1.00	0.25 / 1.42
		300	0.29	53.45	1.00	0.43 / 3.25
	Faulty	10	0.01	66.83	1.00	0.09 / 0.45
		100	0.10	58.11	1.00	0.25 / 1.32
		300	0.29	53.44	1.00	0.43 / 2.99

First Layer (Message Embedding). We select the top-k pixels most influential to the classifier's decision, using either Integrated Gradients (IG) [STY17a] for the CNN (see Fig. 7.6 for an example) or SHAP values [LL17] for the RF. A short alphanumeric message (2–10 characters, 16–80 bits) is embedded into the LSBs of these k pixels.

Second Layer (Coordinate Metadata). We encode each selected pixel's (x,y) coordinate into 32 bits (16 bits per axis), yielding $32k$ bits of metadata. This metadata is embedded into the 7^{th} bit of the first N pixels of the image (e.g., $N = 1024$ or 1300),



(a) Original **(b)** Top-100 (Raw Scaled) **(c)** Top-100 IG Overlay (Absolute)

Fig. 7.6: Example of original input, the top-100 most salient pixels according to IG, and the overlay highlighting those pixels.

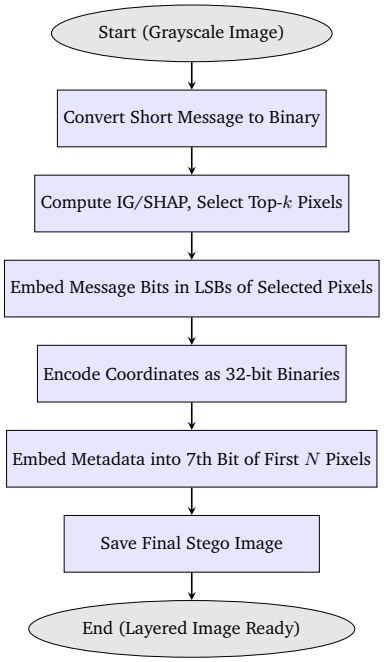


Fig. 7.7: Flowchart of saliency-guided layered embedding.

ensuring the coordinate map can be reconstructed. Figure 7.7 and Figure 7.8 depict embedding and decoding.

While this scheme ensures reversible recovery, it faces severe scalability constraints: for a message of length m characters ($8m$ bits), we require $k = 8m$ selected pixels, and $32k$ bits of coordinate metadata. When m grows, $32 \times 8m = 256m$ bits of metadata quickly exceeds the reserved capacity (≤ 1300 bits), leading to embedding failures, as shown in Tables 7.8 and 7.9.

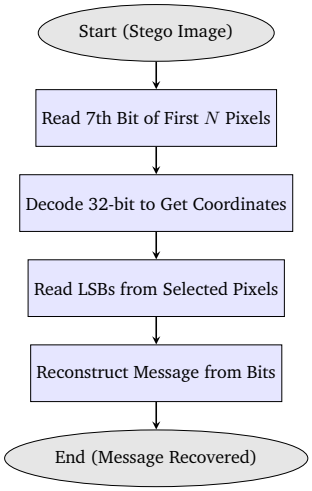


Fig. 7.8: Flowchart of saliency-guided layered decoding.

Tab. 7.8: Classification accuracy under saliency-guided layered embedding (FDIA, MMS).

Dataset/Model	Characters	Class	✓	×	Acc. (%)
FDIA / CNN	3	Normal	2368	0	100.00
		Attack	2019	3	99.85
	5	Normal	2368	0	100.00
		Attack	2019	3	99.85
	10	Normal	2368	0	100.00
		Attack	2000	22	98.91
FDIA / RF	2	Normal	1990	378	84.03
		Attack	1840	182	91.01
	3	Normal	1989	379	83.96
		Attack	1827	195	90.36
	5	Normal	1992	376	84.11
		Attack	1844	178	91.18
MMS / CNN	2	Normal	764	8	98.97
		Attack	135	9	93.75
		Faulty	873	21	97.64
	3	Normal	764	8	98.97
		Attack	135	9	93.75
		Faulty	873	21	97.64
	5	Normal	764	8	98.97
		Attack	135	9	93.75
		Faulty	872	22	97.53

Experimental Evaluation

Impact of LSB Embedding. Table 7.10 shows that embedding up to 1000 characters (8000 bits) in each image yields negligible distortion (PSNR bigger or equal than 64 dB, SSIM = 1.0000) and preserves 100% classification accuracy for both Normal and Faulty classes. Even for long payloads, the CNN remains effectively blind to the steganographic modifications because the LSB-level, PRNG-distributed bit flips (±1/255 intensity) preserve global and local structure (high PSNR/SSIM) and do not induce stable, discriminative patterns in the feature maps, leaving the model's predictions unchanged.

Tab. 7.9: Image quality metrics under saliency-guided layered embedding (FDIA, MMS).

Dataset / Model	Characters	Class	Avg. MSE	Avg. PSNR (dB)	Avg. SSIM	DCT (Mean / Max)
FDIA / CNN	3	Normal	264.59	23.93	0.98	12.03 / 159.41
	5	Normal	441.93	21.69	0.96	15.58 / 261.12
	10	Normal	885.58	18.67	0.91	22.02 / 503.95
	3	Attack	276.49	23.74	0.98	12.32 / 162.77
	5	Attack	460.45	21.51	0.96	15.92 / 265.73
	10	Attack	919.55	18.50	0.91	22.48 / 510.62
FDIA / RF	2	Normal	141.66	26.73	0.99	8.59 / 110.29
	3	Normal	215.40	24.88	0.99	10.63 / 162.27
	5	Normal	363.41	22.57	0.97	13.85 / 257.76
	2	Attack	152.07	26.41	0.99	8.85 / 125.41
	3	Attack	230.45	24.58	0.99	10.96 / 183.32
	5	Attack	386.72	22.30	0.97	14.24 / 290.72
MMS / CNN	2	Normal	576.14	20.56	0.94	17.68 / 197.01
	3	Normal	867.71	18.78	0.90	21.63 / 286.00
	5	Normal	1447.59	16.54	0.81	27.72 / 429.53
	2	Attack	608.67	20.41	0.94	17.96 / 194.36
	3	Attack	921.54	18.58	0.90	22.10 / 281.02
	5	Attack	1536.22	16.36	0.81	28.29 / 427.70
	2	Faulty	574.50	20.60	0.94	17.40 / 206.63
	3	Faulty	1428.02	16.62	0.81	27.32 / 436.97
	5	Faulty	1428.02	16.62	0.81	27.32 / 436.97

Tab. 7.10: CNN accuracy and image quality metrics for baseline LSB embedding.

Characters	Class	Accuracy	Avg. MSE	Avg. PSNR (dB)	Avg. SSIM	Mean DCT	Max DCT
10	Normal	100%	0.00	∞	1.0000	0.000052	0.000557
10	Faulty	100%	0.00	∞	1.0000	0.000053	0.000482
100	Normal	100%	0.00	73.63	1.0000	0.000164	0.001483
100	Faulty	100%	0.00	73.56	1.0000	0.000166	0.001602
1000	Normal	100%	0.00	64.03	1.0000	0.000495	0.006617
1000	Faulty	100%	0.00	64.04	1.0000	0.000494	0.006521

PRNG-Based Embedding Results. As shown in Table 7.6, both CNN and RF maintain $> 99.85\%$ accuracy on FDIA and $> 93.75\%$ on MMS, even with 1000-character payloads. Image quality metrics (Table 7.7) demonstrate minimal distortion: PSNR remains above 54 dB, SSIM = 1.00, and DCT differences stay within acceptable bounds. The dual-layer PRNG approach therefore enables large payloads across multiple images while preserving classifier performance and imperceptibility.

Layered Embedding Results. Table 7.8 shows that for the saliency-guided layered scheme, embedding more than 5 characters (40 bits) yields noticeable degradation. On FDIA, CNN accuracy drops to 98.91% for 10 characters; RF accuracy falls to $\approx 84\%$ for 2 characters. On MMS, a 5-characters payload reduces SSIM to 0.81 and causes misclassifications. Table 7.9 confirms high MSE (> 1400) and low PSNR (< 17 dB) for larger payloads. These results highlight severe scalability limitations: the metadata overhead (32 bits per selected pixel) quickly exhausts reserved capacity, forcing trade-offs that harm image fidelity and classifier accuracy.

Findings and Mitigations

Stealthy Exfiltration. Our experiments reveal a critical weakness in image-based NIDS: CNNs fail to detect minimal LSB perturbations, allowing adversaries to hide substantial secrets without impacting classification. The PRNG-based scheme scales to exfiltrate large model components (e.g., 10^7 bits of weights) across multiple images, exploiting typical intelligence-sharing workflows.

Steganographic Adversarial Examples. By contrast, the saliency-guided layered method fails to scale beyond a few dozen bits and begins to erode classification performance, effectively turning the embedded payload into adversarial perturbations.

Defenders should consider the following mitigations:

Automated Image Forensics. Incorporate tools that compute pixel-level statistics (e.g., RS analysis, sample pairs) to flag images with abnormal LSB distributions [Fri09]. Such tools can be integrated into DLP pipelines to inspect every shared image.

Content Fingerprinting. Maintain a canonical fingerprint (e.g., SHA-256) of known benign images. Any deviation (even at the LSB level) should trigger alerts. This requires strict version control and rapid hash computation but can catch tampering precisely.

Encrypted Image Channels. If sharing is necessary, encrypt images end-to-end and perform content inspection after decryption. This prevents external adversaries from intercepting images in transit but does not stop insider threats.

Big-Data Perspective: The 5Vs

Volume. Operational packet captures are massive; converting streams to images produces large corpora and sizable model artifacts. Our steganographic capacity scales with corpus size via $C_{\text{total}} = NHWL$, so volume directly governs how much can be covertly exfiltrated.

Velocity. The packet→image pipeline and NIDS inference run near real time. Our LSB/PRNG embedding adds negligible latency, enabling low-friction exfiltration within streaming workflows and motivating real-time steganalysis/DLP.

Variety. We handle heterogeneous sources (IEC 61850 MMS traffic; Normal/-Faulty/Attack classes), multiple representations (tabular vs. image), and models (RF/SVM/CNN), as well as distinct stego schemes (baseline LSB, PRNG, layered).

Value. For defenders, image-based NIDS offers actionable forensics (pixel↔packet feature-mapping [Pie+20]) and a controlled testbed to study covert channels; for attackers, the method enables high-value model exfiltration without degrading detection.

Veracity. Steganography undermines the trustworthiness of shared artifacts, creating confidentiality and integrity risks even when accuracy appears unchanged. We therefore report imperceptibility (PSNR/SSIM/DCT) and recommend defenses to preserve data veracity.

Future Work

While our spatial LSB and PRNG-based techniques demonstrate effective exfiltration, frequency-domain approaches (e.g., DCT-based embedding) offer enhanced robustness against compression and steganalysis. Future work should investigate the impact of a variety of techniques on new computer vision models (such as deep convolutional neural networks pretrained on large image benchmarks) and further steganographic technique,s such as embedding in mid-frequency DCT coefficients by partitioning images into 8×8 blocks, applying DCT, and modifying least significant bits of selected coefficients [Cox+07; BMB13]. This would improve imperceptibility under JPEG compression and resist statistical detection. Additionally, exploring adaptive payload distribution strategies (where high-utility bits like architecture meta-data are embedded first, followed by lower-priority bits) could optimize reconstruction fidelity under strict capacity limits.

As further potential improvements, the placement heuristics and gradient-driven cost learning from [Don+20; Mo+19; Zen+23] are complementary to our channel: they could be layered onto our pipeline to further minimize detectability, and, conversely, adapted by defenders to build steganography-aware inspectors that target our embedding strategies.

Regarding future IDS research, we have observed that image-based IDS can be highly actionable because every pixel corresponds to a specific byte (and thus a precise packet field) in the capture. When saliency/activation maps flag a region, it is possible to deterministically map (inverse feature mapping [Pie+20]) those pixels back to the linear byte index , resolve the enclosing packet from stored boundaries, and identify the exact field (e.g., MMS PDU, opcode, or flag) that drove the alert. This pixel/packet mapping turns “black-box” CNN signals into concrete forensic evidence, pinpointing which packets to inspect, what rules to

deploy (e.g., drop/alert on a field value), and where to harden protocol handling without hand-crafted features.

7.2.5 Summary

We present a novel model-stealing threat scenario in which an adversary embeds sensitive learning-based model information (architecture, hyperparameters, weights) within image-based representations of network traffic. Our objective is not to advance detection accuracy but to establish an image-based experimental platform that substantiates steganographic exfiltration in a critical-infrastructure context. Our baseline Least Significant Bit (LSB) experiment shows that Convolutional Neural Networks (CNN) classifiers remain oblivious to embedded payloads, while the Pseudorandom Number Generator (PRNG)-based scheme enables large, stealthy exfiltration with minimal distortion and negligible impact on classification performance. In contrast, a saliency-guided layered approach, although interpretable, suffers from severe scalability constraints and degrades model accuracy when payloads exceed a few dozen bits, turning the hidden message into adversarial examples. We recommend that organizations using image-based Network Intrusion Detection Systems (NIDS) incorporate steganalysis and forensic image inspection into their Data Loss Prevention (DLP) workflows to detect covert channels. Our code and artifacts are publicly available[4] to encourage further research in defending against steganographic exfiltration in Machine learning (ML) pipelines.

[4] https://github.com/gus5298/stego_stealing

8 Defense Directions

The attack vectors identified in the preceding Chapters underscore the need for defense mechanisms that explicitly account for adversarial exploitation of explainability. Rather than proposing universal mitigations, this Chapter explores defense directions that aim to increase attacker uncertainty and operational cost. A lightweight Moving Target Defense strategy is investigated in the context of smart grid intrusion detection, with particular attention to its interaction with feature importance and model robustness. By analyzing how defensive variability affects both attack effectiveness and explanation stability, this Chapter demonstrates how security-aware system design can partially mitigate risks introduced by explainability mechanisms.

8.1 Lightweight Moving Target Defense for Robust Intrusion Detection in Smart Grids

Smart grids rely heavily on network protocols, e.g., classic Modbus TCP for substation communications, yet conventional learning-based intrusion detectors overfit to spurious correlations and crumble under adversarial or distributional shifts. In this Section[1], we introduce a lightweight Moving Target Defense (MTD) proxy that randomizes the Modbus slave address on each TCP session. In our proof-of-concept experiments, a Random Forest detector under MTD maintains 95% detection accuracy while, in the eXplainable Artificial Intelligence (XAI) sense, its reliance on the address field drops, and payload-related features gain prominence. We further demonstrate that simple deterministic checks and dynamic honeypots can complement MTD to protect integrity, availability, and confidentiality with minimal or no machine learning. Our results highlight that even modest MTD interventions can substantially harden smart-grid intrusion detection systems against both inadvertent shifts and targeted evasion.

The modern smart grid has evolved into a complex cyber–physical ecosystem, tightly coupling energy-delivery infrastructure with information and communication technologies. This nexus of power and data systems (often studied under the banner of

[1] Part of this work was peer-reviewed and published in [P4]

Energy Informatics) promises unprecedented efficiency, but also opens new cyber-attack surfaces. Within the energy domain, adversaries can exploit learning-based systems **[P1]** in several ways, e.g.: poisoning training data [MLG24], crafting adversarial examples **[P7]**, or building surrogate models to test their capabilities in an attacker-controlled environment [Tia+21], eventually compromising model *availability*, *integrity* and *confidentiality* (CIA triad). While various countermeasures [Tia+21; Tia+22; Bon+23; Zha+23] have been investigated, most incur significant data, compute, or operational overhead, as they rely on fortifying the learning models themselves.

While IEC 61850 is increasingly adopted in modern substations, classic Modbus TCP remains ubiquitous in many existing deployments [Joh+25]. Its simple, open structure and rich open-source ecosystem make it an ideal use case for Moving Target Defense (MTD) feasibility research. Moreover, the core idea (randomizing a static protocol field to force a learning-based IDS to diversify its decision basis) applies equally to IEC 61850's dataset IDs, MMS object references, and other features. Thus, demonstrating our MTD proxy on Modbus-TCP both ensures reproducibility and paves the way for future extensions to native energy-protocol defenses.

In the present Section, our main contribution is as follows:

1. **Lightweight MTD.** We demonstrate that MTD offers a lightweight yet highly effective layer in the smart-grid security stack. By continuously randomizing Modbus-TCP slave addresses on each client session, MTD breaks the dominant feature correlation that supervised detectors, such as RF, repeatedly exploit.

Our key experimental findings are:

- The detection accuracy does not decrease substantially after applying a lightweight MTD proxy.
- MTD preserves *integrity* and *availability*, sustaining $\approx 95\%$ detection accuracy under adversarially perturbations in a RF classifier.
- XAI-based feature-importance shifts dramatically: payload-length and data-values gain prominence, while reliance on the static address field collapses.

In order to achieve safe energy systems, rather than over-engineering complex, reactive detectors, we advocate for lightweight, early-warning mechanisms that catch attackers at the outset. As secondary contribution, we provide a discussion on these topics:

2. **Instant MITM Detection.** Many intrusion detection datasets (including well-cited smart grid publications, e.g., [AKM19]) rely on attacks achieved via Address Resolution Protocol (ARP) spoofing. A simple, deterministic ARP-table consistency check flags man-in-the-middle attempts in $\mathcal{O}(1)$ time, halting attacks before they unfold. We argue that ML is not required in those cases, as it is often unable to stop attacks and increases the computational overhead and the attack surface.

3. **Proactive Enumeration Defenses.** It is known that attackers are using AI and LLMs in their malicious endeavors, for reconnaissance and even in the cyber-physical world [Ope24]. To counter these, we discuss on-the-fly web-directory honeypots to mislead AI/LLM-powered scanners during reconnaissance of internet-facing assets, protecting backends and harvesting attacker behavior for further analysis.

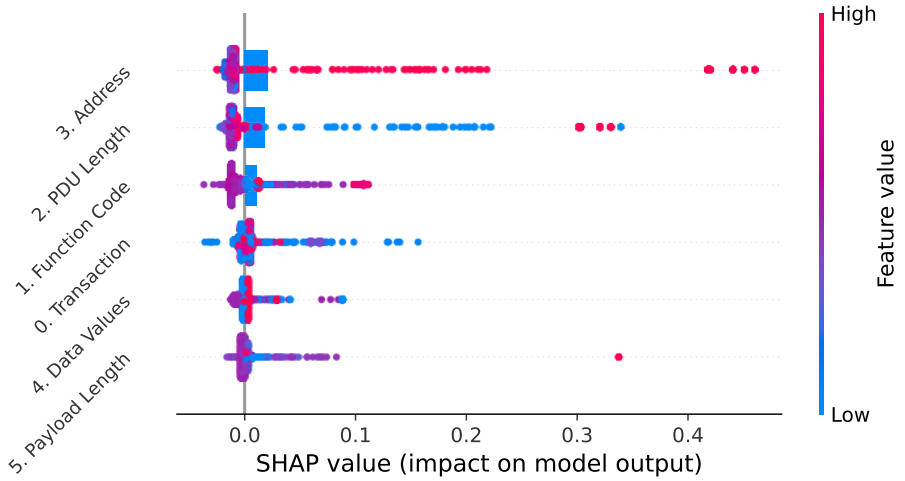
8.1.1 Related Work

ML-based Intrusion Detection in ICS. ML-based detectors have been studied for Modbus-TCP in the energy domain (see Chapter 4.2.1). These approaches train classifiers (e.g., Random Forests, SVMs) on network-level features to detect deviations from normal behavior. However, they require sizable labeled datasets, incur nontrivial inference latency, and remain vulnerable to adversarial evasion and poisoning attacks that undermine integrity and availability.

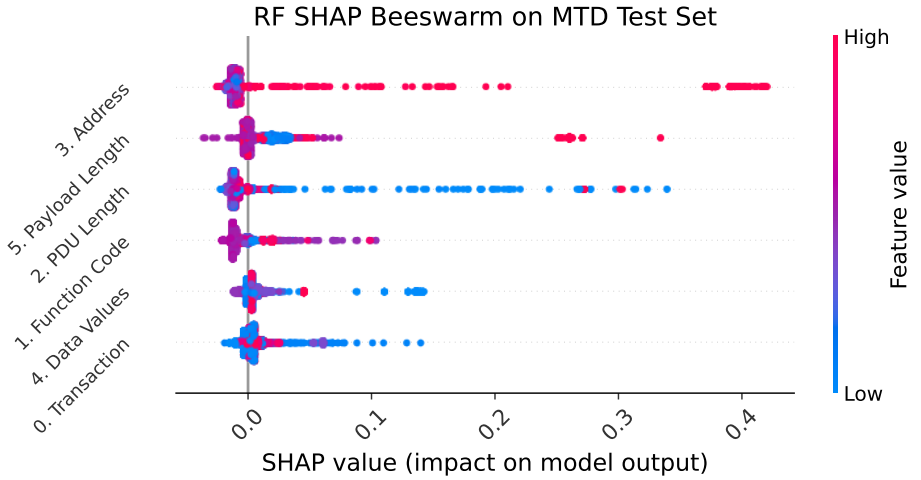
Moving Target Defense (MTD). MTD techniques continuously shift system configurations (such as IP addresses, ports, or protocol parameters) to thwart attacker reconnaissance and impede exploit reliability [Jaj+12]. Regarding Modbus TCP, MTD has been used to secure SCADA communications [Hey18] and improving the resiliency of Industrial Control Systems (ICS) [Cha19]. To the best of our knowledge, there are currently no commercial "plug-and-play" MTD solutions specifically tailored to Modbus-TCP. Most industrial network security offerings (e.g., firewalls with Deep-Packet Inspection, ICS-aware intrusion prevention systems from vendors like Schneider Electric, Nozomi, or Tenable) focus on static signature and anomaly detection rather than on dynamic, per-session address randomization. Prior work has demonstrated MTD benefits against adversarial examples [Roy+19; Mar+21], but practical MTD proxies for Modbus-TCP (randomizing slave addresses at the session level) within have not been validated, and we use the smart-grid domain as relevant use case. To the best of our knowledge, we are the first to study the impact of MTD using XAI techniques.

8.1.2 Impact of Moving Target Defense on Feature Importance and Model Robustness

MTD refers to continuously changing addresses, shifting communication channels, or varying protocol parameters to increase attacker uncertainty. Let $\mathcal{C}(t)$ denote the network configuration at time t. MTD can be modeled as a stochastic process on the configuration space, making $\mathcal{C}(t)$ unpredictable from an attacker's perspective.



(a) Baseline RF SHAP beeswarm from **[P1]**.



(b) RF SHAP beeswarm under MTD.

Fig. 8.1: Comparison of RF feature-importance (SHAP beeswarm) before and after applying MTD.

Proof of Concept. Baseline feature-importance and detection results for learning-based Modbus-TCP intrusion detectors have been reported in Chapter 5.1. We use the well-established method SHAP [LL17] as a model-agnostic approach for explaining the output of ML models. In those experiments, RF models relied heavily on the static `Address` field (mean SHAP ≈ 0.27), achieving high accuracy (99%) but exhibiting brittle behavior when that field was perturbed. Unsupervised detectors (One-Class SVM, Isolation Forest) likewise exploited a fixed address distribution and performed poorly under slight distributional shifts.

In the present Section, we extend those findings by applying a simple MTD that randomizes the Modbus `Address` on each connection, with a focus on the RF model as in the baseline in Chapter 5.1.

Re-weighted Feature Importance. Figure 8.1(a)–(b) contrasts the RF's SHAP plots[2] before and after applying MTD, showing that the `Address` feature's impact diminishes while `PayloadLength` and `DataValues` gain prominence under defense. Table 8.1 reports the mean absolute SHAP values for each feature before and after applying our simple MTD. Although `Address` remains the top contributor, its mean |SHAP| decreases only modestly (0.021→0.019, approx. 9.5%), whereas `Payload Length` rises dramatically (0.002→0.014, +600%) and `Data Values` increases substantially (0.005→0.009, +80%). PDU Length and Transaction also shift by non-negligible amounts. This confirms that randomizing the Modbus TCP address still forces the classifier to draw more signal from secondary fields, even if it does not entirely eliminate address bias.

Feature	Baseline	MTD
0. Transaction	0.005	0.006
1. Function Code	0.011	0.009
2. PDU Length	0.018	0.012
3. Address	0.021	0.019
4. Data Values	0.005	0.009
5. Payload Length	0.002	0.014

Tab. 8.1: Mean absolute SHAP values per feature before and after applying MTD.

Per-Pair Accuracy Comparison. Figure 8.2(a)–(b) repeats the two-feature ablation experiment before and after MTD. While baseline per-pair accuracies hovered around

[2] We use the so-called SHAP beeswarm plot, designed to display an information-dense summary of how the top features in a dataset impact the model's output.

80–90%, MTD raises almost all pairs above 95%. By breaking the dominant address correlation, MTD compels the model to learn more robust multi-feature patterns, improving generalization even with only two features.

Detection Accuracy under MTD. Table 8.2 presents each model's accuracy before (baseline) and after MTD. Supervised classifiers remain highly accurate (Linear SVM stays at 100% and RF drops only to 95%) demonstrating effective compensation via secondary features. In contrast, unsupervised detectors suffer severe degradation (One-Class SVM by 22% and Isolation Forest by 40%), confirming their dependence on a static address distribution.

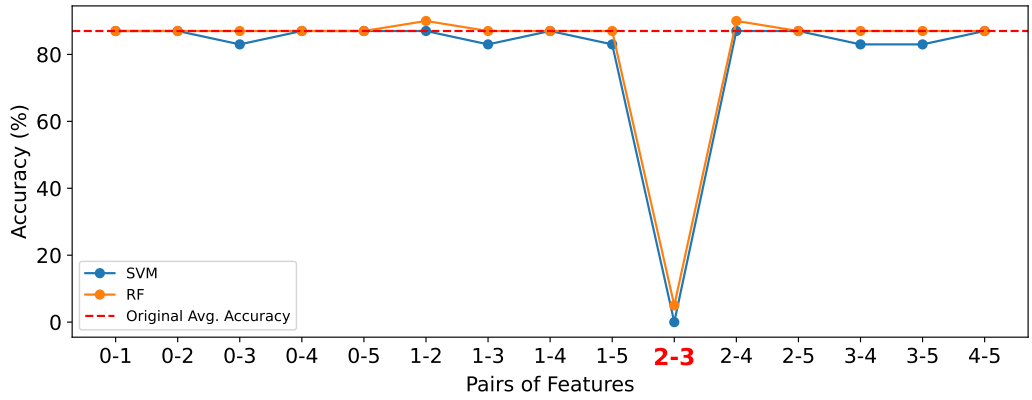
Model	Baseline Acc. (%)	MTD Acc. (%)
One-Class SVM	48	26
Isolation Forest	87	47
Linear SVM	100	100
Random Forest	99	95

Tab. 8.2: Detection accuracy before and after applying MTD.

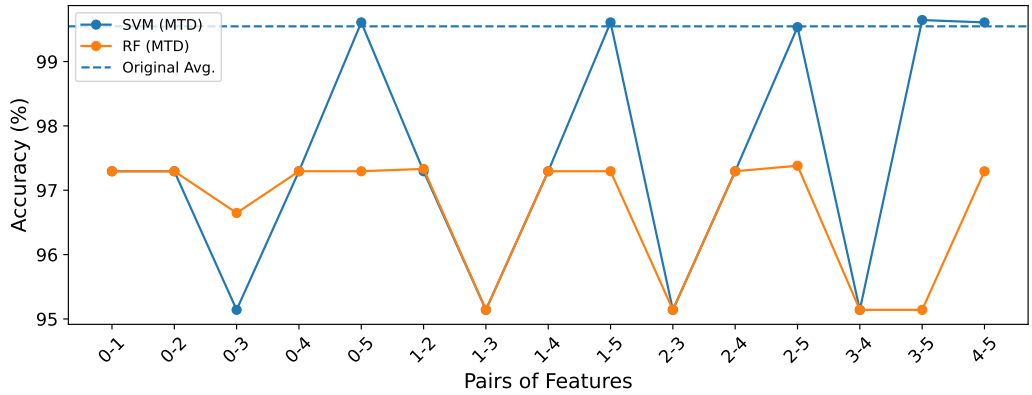
Adversarial Robustness under MTD. We evaluate gradient-based evasion on the MTD test set. Both supervised models remain highly vulnerable (Linear SVM drops to 0% adversarial accuracy and RF to 13%) showing that MTD alone does not mitigate crafted evasion attacks.

Summary of findings:

- A lightweight MTD (randomizing only the Modbus `Address`) forces supervised models to redistribute feature importance and maintain high detection accuracy.
- MTD severely degrades unsupervised detectors that rely on static address distributions.
- MTD does not by itself provide total robustness to adversarial perturbations; therefore, it should be integrated with robust anomaly-detection mechanisms and adversarial training to achieve comprehensive resilience.



(a) Baseline per-pair accuracy from **[P1]**.



(b) Per-pair accuracy under MTD.

Fig. 8.2: Two-feature ablation accuracy for SVM and RF, before and after MTD.

Practical Deployment Considerations. While randomizing the Modbus "Address" field serves as a clean proof-of-concept for MTD, a production-grade implementation must preserve the semantics of Modbus slave addressing. Concretely, one would insert a lightweight MTD proxy or gateway that presents a fresh, randomized slave ID on each client connection and internally maps it back to the true physical address when forwarding to the PLC. This approach transparently breaks any fixed "address" correlation for an attacker while allowing existing devices to operate unchanged. However, it also introduces nontrivial engineering trade-offs: each translation hop adds modest latency, the proxy becomes a potential single point of failure, and operational complexity increases (e.g., managing consistent address mappings across redundant gateways). Thus, while address-randomization demonstrably forces detectors to diversify their decision basis, real-world deployments must balance these security gains against performance, reliability, and maintenance costs.

8.2 Beyond Moving Target Defenses

While our primary contribution focuses on MTD for Modbus-TCP intrusion detection, real-world deployments benefit from a layered active-defense strategy. In this Section, we briefly outline two additional, lightweight techniques that complement MTD: deterministic MITM detection via ARP-table consistency checks and dynamic web-directory honeypots to thwart AI-driven enumeration. Both approaches require minimal or no ML, align with the CIA triad, and can be integrated into smart-grid networks with modest engineering effort.

AI to Detect Man-In-The-Middle attacks (MITM).

ARP-spoofing (also known as ARP cache poisoning) is a classic technique for MITM attacks on Ethernet-based LANs. A compromised node (or network-adjacent attacker) can send crafted ARP replies associating its MAC address with the IP address of a legitimate device (e.g., the substation gateway) thus intercepting traffic between devices and the gateway. There are scientific papers published on the application of ML to detect ARP spoofing-based Man-In-The-Middle (MITM) attacks [EKT16] **[P7]** or detect attacks that include MITM as a necessary step in the threat model **[P7]** [Tan+24]. Instead of using ML to classify network flows or infer suspicious patterns, one can deterministically detect spoofing whenever an ARP table entry violates basic consistency rules. Two simple rules suffice:

(1) If two distinct MAC addresses $\mathrm{MAC}_i \neq \mathrm{MAC}_j$ are associated with the same IP_x in the aggregated ARP records across the LAN, then a spoofing event is present. Formally, define an aggregated/mirror table

$$\mathrm{ARP}_{\mathrm{agg}}(t) = \bigcup_{s \in \mathcal{N}} \mathrm{ARP}_s(t)$$

if $\left|\left\{ \mathrm{MAC} : (\mathrm{IP}_x, \mathrm{MAC}) \in \mathrm{ARP}_{\mathrm{agg}}(t) \right\}\right| > 1$, raise an alert.

(2) Many devices send gratuitous ARP messages periodically (or in response to link events), i.e., "I am IP_x at MAC_x." If a node s receives a gratuitous ARP reply contradicting its currently cached $(\mathrm{IP}_x, \mathrm{MAC}_x)$, this indicates either a legitimate MAC change (rare) or spoofing. For ICS devices, MAC addresses rarely change; thus any gratuitous ARP altering local state is suspicious.

These rules can be monitored by a lightweight network sensor (e.g., a switch with ARP inspection capability or a dedicated appliance) that collects ARP announcements. No ML is needed, just simple set/dictionary operations. ML methods typically require

feature extraction (e.g., packet interarrival times, header fields, flow statistics) and execution of classification models (e.g., neural networks, RF). Training such models on network traffic demands substantial labeled datasets, which may not be readily available for every substation or topology. Inference at line rate can introduce latency. Substations often operate with timing constraints: protective relays may trip in milliseconds if traffic inspection slows forwarding. Many protective relays, IEDs, or field switches have limited CPU and memory. Embedding ML models on these devices is impractical. Even if centralized, the computational resources needed to classify every ARP packet or analyze flows in real time exceed that of a simple ARP-table consistency check. ML classifiers rely on features extracted from traffic patterns. An attacker aware of the detection model can engineer adversarial ARP packets (e.g., timing-based perturbations, spoofed TTLs) that evade classification.

In contrast, the deterministic ARP-table check is not easily fooled: any incorrect IP→MAC binding is immediately flagged. If the AI is trained or updated on data streams, a subversive adversary can inject malicious training samples (e.g., carefully crafted ARP exchanges labeled as "benign") to manipulate the classifier's decision boundary. A formal threat model for ML poisoning uses the notion of an adversarial cap in the training distribution; quantifying robustness against poisoning is nontrivial and requires ongoing validation.

AI-Based Enumeration Prevention via Honeypots.

Modern electrical substations often expose web-based HMIs or asset-management portals to support remote monitoring and firmware updates. Attackers use *directory busting* (either via traditional large wordlists or AI/LLM–based semantic clustering [Ant+24; CCP24] **[P12]**) to discover hidden endpoints (e.g. backup config directories, debug interfaces). Each discovered endpoint leaks information or allows exploitation. By *dynamically* spawning honeypot directories in response to enumeration patterns, defenders can:

- *Deplete attacker resources* (LLM tokens, request quotas, time) by forcing probes into fictitious paths.
- *Detect and log* enumeration tools, including AI-driven scanners, without relying solely on ML classification.
- *Delay* attacker progress, buying time for incident response and mitigation.

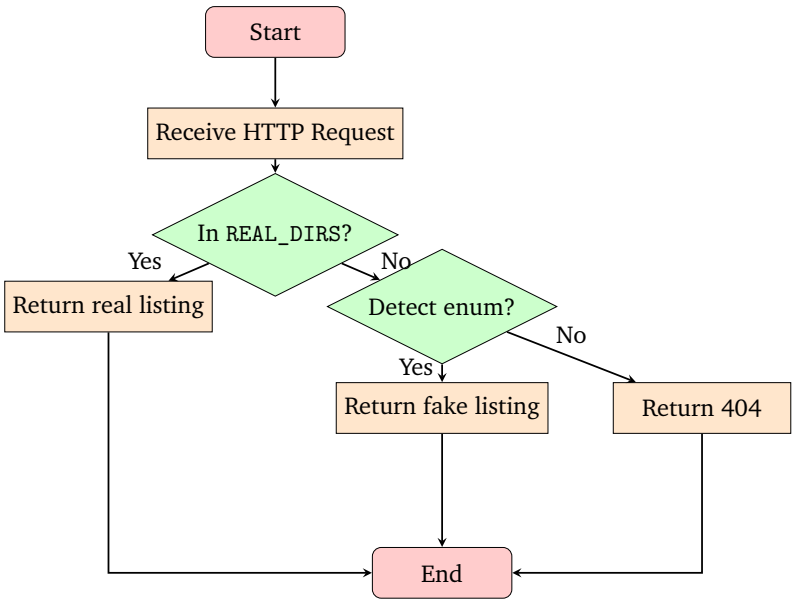


Fig. 8.3: Flowchart of the enumeration detection and honeypot-spawning logic.

Model the attacker's enumeration as a sequence of directory guesses $D = \{d_1, d_2, \dots\}$. Let W be the defender's wordlist of real directories ($|W| = w$), and let $\tilde{W}(t)$ be the set of *honeypot* directories created up to time t. Each probe d_i incurs a cost in tokens or time, $c(d_i)$. The defender's strategy is: for any probe $d_i \notin W$, add d_i to $\tilde{W}$ and respond with a plausible directory listing.

$$\text{Expected cost to attacker for } n \text{ probes:} \quad C(n) = \sum_{i=1}^{n} c(d_i).$$

With honeypots, each *false* guess now yields a valid response, so $C_{\text{honeypot}}(n) = \sum_{i=1}^{n} c(d_i)$ with $d_i \in W \cup \tilde{W}(i-1)$. Since $|\tilde{W}(i)| = i - |\{d_j \in W : j \le i\}|$, the attacker's search space effectively grows by one per false guess, enforcing linear growth in cost and latency. Thus, for any enumeration strategy (random wordlist or LLM-based clustering), the honeypot mechanism guarantees $C_{\text{honeypot}}(n) = \Theta(n)$, while a naive server gives $C_{\text{normal}}(n) = O(n)$ but with zero delay on non-existent paths.

Figure 8.3 illustrates the step-by-step logic to be implemented at the smart-grid substation HMI interface. Incoming HTTP requests are first checked against a whitelist of genuine directories ('/status', '/config', '/logs'). If the path is not recognized, an enumeration-detection heuristic is applied. When enumeration is detected, a new honeypot directory (e.g., extracted from known wordlists) is dynamically spawned and a plausible directory listing is returned, thereby misleading and resource-consuming the attacker's AI-based or wordlist-based scanning tool. If no enumeration is detected, a standard 404 response is returned. This one-column

flowchart succinctly captures the active defense mechanism against directory-busting in smart grid environments.

8.3 Summary

In this Chapter, we show that a minimal Moving Target Defense (MTD), based on randomizing only the Modbus TCP `Address` per connection, significantly improves the robustness of learning-based intrusion detectors in smart-grid environments. Our Random Forest proof-of-concept under MTD retains high detection accuracy despite address modifications, and its explainability analysis reveals a marked shift of importance toward payload-centric features. By forcing the classifier away from a brittle single-feature correlation, MTD enhances integrity without retraining or heavy overhead. Our MTD proxy on Modbus-TCP paves the way for extensions to native energy-protocol defenses. Complementary, non learning-based techniques address robustness at earlier stages of the attack chain with lower costs. As future work, we plan to further measure the success of these three methods in a hardware-in-the-loop IEC 61850 testbed.

9 Conclusion

The present dissertation investigates the security implications of explainability in learning-based smart grid systems from an adversarial perspective. Through a series of empirically grounded studies spanning integrity, availability, and confidentiality, the work demonstrates that explainability systematically alters the attacker's capability space. While XAI improves transparency, interpretability, and trust for legitimate stakeholders, it simultaneously exposes structured information that can be exploited by adversaries to guide, optimize, and scale attacks.

Across network-based intrusion detection, power quality recognition, industrial vision systems, and recommendation models, the results consistently show that explanation mechanisms reduce attacker uncertainty and lower the cost of adversarial actions. XAI-guided evasion and poisoning attacks achieve significantly higher impact with fewer perturbations than unguided baselines, enabling both precise integrity violations and broad availability degradation. In confidentiality-oriented scenarios, explanation outputs act as auxiliary side channels that accelerate model stealing and enable covert exfiltration, including steganographic attacks embedded within smart grid communication data.

At the same time, the dissertation highlights important constraints and asymmetries. The inverse feature mapping problem remains a fundamental obstacle, particularly in black-box and problem-space attacks, limiting the attacker's ability to translate feature-space insights into effective real-world perturbations. Moreover, not all learning-based systems are equally vulnerable: models with more distributed feature reliance and better-aligned data representations exhibit increased resistance to XAI-guided manipulation.

Informed by these findings, the thesis explored defense directions that explicitly consider XAI-induced attack surfaces. A lightweight Moving Target Defense strategy demonstrated that introducing controlled variability can shift feature importance away from brittle correlations and significantly improve robustness without heavy computational overhead. These results suggest that defenses need not eliminate explainability, but rather manage and constrain the information it exposes.

Taken together, the present work reframes explainability as a security-relevant system component rather than a purely auxiliary tool. It underscores the necessity of integrating adversarial thinking into the design, deployment, and regulation of explainable learning-based systems in critical infrastructure.

Main Insights

- Explainability methods systematically reduce attacker uncertainty by exposing structured information about feature relevance, decision boundaries, and model behavior, thereby amplifying adversarial effectiveness across the CIA triad.
- XAI-guided attacks achieve higher impact with fewer perturbations than unguided attacks, enabling both stealthy integrity violations and scalable availability degradation in realistic smart grid environments.
- The inverse feature mapping problem remains a key limiting factor for adversaries, particularly in black-box and problem-space attacks, highlighting a natural friction point between explanation access and real-world exploitability.
- Learning-based systems that rely heavily on a small number of dominant or spurious features are particularly vulnerable; distributing feature importance and avoiding brittle correlations substantially improves robustness.
- Explainability mechanisms can function as implicit side channels, facilitating confidentiality breaches such as model extraction and steganographic data exfiltration when not explicitly secured.

9.1 Answering Research Questions

The research questions posed in Chapter 1.2 are addressed across the core Chapters of this dissertation. Integrity-oriented questions (RQ-1 and RQ-2) are investigated through evasion and targeted poisoning attacks in Chapter 5, availability-related threats (RQ-3) are analyzed via indiscriminate poisoning in Chapter 6, and confidentiality risks (RQ-4) are examined through model stealing and steganographic exfiltration in Chapter 7. Defense implications arising from these findings are explored in Chapter 8.

RQ-1 investigates the extent to which evasion attacks in realistic smart-grid problem spaces can compromise the integrity of learning-based models and how such

attacks can be mitigated. The results demonstrate that integrity can be significantly compromised through carefully crafted perturbations that remain operationally plausible. By leveraging explainability methods, attackers can identify highly influential features and introduce minimal modifications that substantially degrade detection performance while preserving normal system behavior. Across intrusion detection and cyber-physical monitoring scenarios, XAI-guided perturbations consistently outperform unguided attacks in terms of efficiency and stealth. Mitigation experiments further show that defenses which redistribute feature importance (such as lightweight Moving Target Defense) reduce reliance on brittle correlations and improve robustness without requiring costly retraining or architectural changes.

RQ-2 examines how targeted poisoning attacks can be constructed to compromise model integrity during training and evaluates their practical consequences. The findings show that targeted poisoning becomes more effective when guided by explanation mechanisms that reveal feature relevance. XAI-informed attackers can inject strategically crafted samples that induce specific misclassifications while minimally affecting overall accuracy, thereby maintaining stealth. In several smart-grid use cases, targeted poisoning substantially increased the attack effectiveness compared to random poisoning while requiring modifications to only a small subset of features. These results indicate that training pipelines represent a critical security boundary in learning-based smart-grid systems, as poisoned datasets can introduce persistent vulnerabilities exploitable at inference time.

RQ-3 addresses how indiscriminate data poisoning can be used to compromise the availability of learning-based smart-grid systems by inducing denial-of-service conditions. Experimental results confirm that indiscriminate poisoning degrades model reliability rather than enforcing specific misclassifications, leading to widespread prediction instability and reduced operational usefulness. By corrupting training distributions, attackers can increase error rates and uncertainty levels sufficiently to trigger operational fallback mechanisms or overwhelm monitoring workflows, effectively producing denial-of-service effects without directly disrupting infrastructure communication. The impact varies depending on problem-space constraints of the underlying objects, highlighting that availability attacks exploit systemic fragility rather than individual decision boundaries.

RQ-4 investigates how learning-based smart-grid models can be reverse-engineered and what confidentiality risks arise from such attacks. The results demonstrate that deployed models can be effectively approximated through query-based extraction and auxiliary information derived from explanation outputs, which reduce attacker uncertainty during surrogate model training. Furthermore, the thesis introduces a

steganographic exfiltration scenario in which model parameters and architectural information are covertly embedded within data representations used by intrusion detection systems. These findings reveal that explainability mechanisms may function as indirect side channels, enabling intellectual-property leakage and privacy risks even when direct model access is restricted.

9.2 Future Work

Several promising research directions emerge from this work. First, future studies should investigate the security implications of frontier AI models (such as large language models, diffusion models, and multimodal foundation models) when integrated into smart grid operations, decision support, and automation pipelines. These models introduce new forms of explainability, interaction, and memory that may further expand adversarial capability spaces.

Second, deeper investigation is needed into the trade-offs between regulatory-driven transparency requirements and adversarial risk, particularly in the context of the EU AI Act and similar frameworks. Developing security-aware explainability standards that balance interpretability with controlled information exposure remains an open challenge.

Third, future defenses should explore adaptive and protocol-aware mechanisms beyond Moving Target Defense, including explanation obfuscation, selective disclosure, and explanation randomization, evaluated under realistic attacker models.

Finally, extending the presented attacks and defenses to hardware-in-the-loop and cross-domain smart grid deployments will be essential to assess their impact at scale and to inform the design of resilient, explainable, and secure critical infrastructure systems.

Bibliography

[AKM19] Sridhar Adepu, Nandha Kumar Kandasamy, and Aditya Mathur. “Epic: An Electric Power Testbed for Research and Training in Cyber Physical Systems Security”. In: *ESORICS ’19) Workshops*. Springer. 2019, pp. 37–52 (cit. on pp. 62, 145).

[AA23] Afia Afrin and Omid Ardakanian. “Adversarial Attacks on Machine Learning-Based State Estimation in Power Distribution Systems”. In: *Proceedings of the 14th ACM International Conference on Future Energy Systems*. 2023, pp. 446–458 (cit. on pp. 51, 52).

[AMF15] Hossein Ghassempour Aghamolki, Zhixin Miao, and Lingling Fan. “A hardware-in-the-loop SCADA testbed”. In: *NAPS ’15*. IEEE. 2015 (cit. on p. 62).

[Agw+21] Utkarsha Agwan, Lucas Spangher, William Arnold, et al. “Pricing in prosumer aggregations using reinforcement learning”. In: *eEnergy ’21*. 2021 (cit. on pp. 52, 53).

[Ahm+23] Rasheed Ahmad, Izzat Alsmadi, Wasim Alhamdani, and Lo’ai Tawalbeh. “Zero-day attack detection: a systematic literature review”. In: *Artificial Intelligence Review* 56.10 (2023), pp. 10733–10811 (cit. on p. 34).

[AK20] Chuadhry Mujeeb Ahmed and Nandha Kumar Kandasamy. “A comprehensive dataset from a smart grid testbed for machine learning based cps security research”. In: *International Workshop on Cyber-Physical Security for Critical Infrastructures Protection*. Springer. 2020, pp. 123–135 (cit. on p. 129).

[AM18] Naveed Akhtar and Ajmal Mian. “Threat of adversarial attacks on deep learning in computer vision: A survey”. In: *IEEE Access* 6 (2018), pp. 14410–14430 (cit. on p. 126).

[Akh+21] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. “Advances in adversarial attacks and defenses in computer vision: A survey”. In: *IEEE access* 9 (2021), pp. 155161–155196 (cit. on p. 20).

[AA24] Sarah Alabdulhadi and Ali Al-Matouq. “Efficient and Standardized Alarm Rationalization for Cybersecurity Monitoring”. In: *IEEE Access* (2024) (cit. on p. 20).

[P13] Hermenegildo da Conceição Alberto, Gustavo Sánchez, Jean Marie Dembele, et al. “Masquerading IEC 61850 GOOSE Protocol: Cyber-Physical Experiments and Detection”. In: *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*. 2025, pp. 1000–1001 (cit. on p. 15).

[Alp14] Univ. Grenoble Alpes. *G-ICS lab (GreEn-ER8 Industrial Control systems Sandbox)*. Accessed: 2024-06-05. 2014 (cit. on p. 62).

[Als+25] Amjad Alsirhani, Noshina Tariq, Mamoona Humayun, Ghadah Naif Alwakid, and Hassan Sanaullah. “Intrusion detection in smart grids using artificial intelligence-based ensemble modelling”. In: *Cluster Computing* 28.4 (2025), p. 238 (cit. on p. 19).

[And+21] Giuseppina Andresini, Feargus Pendlebury, Fabio Pierazzi, et al. “Insomnia: Towards concept-drift robustness in network intrusion detection”. In: *Proceedings of the 14th ACM workshop on artificial intelligence and security*. 2021, pp. 111–122 (cit. on p. 34).

[Ant+24] Diego Antonelli, Roberta Cascella, Antonio Schiano, Gaetano Perrone, and Simon Pietro Romano. ““Dirclustering”: a semantic clustering approach to optimize website structure discovery during penetration testing”. In: *Journal of Computer Virology and Hacking Techniques* 20.4 (2024), pp. 565–577 (cit. on p. 151).

[Ard+23] Carmelo Ardito, Yashar Deldjoo, Tommaso Di Noia, et al. “Machine-learned Adversarial Attacks against Fault Prediction Systems in Smart Electrical Grids”. In: *arXiv preprint arXiv:2303.18136* (2023) (cit. on pp. 51, 52).

[Are13] Theo A. Arentze. “Adaptive Personalized Travel Information Systems: A Bayesian Method to Learn Users’ Personal Preferences in Multimodal Transport Networks”. In: *IEEE Transactions on Intelligent Transportation Systems* (2013) (cit. on p. 100).

[Arp+22] Daniel Arp, Erwin Quiring, Feargus Pendlebury, et al. “Dos and don’ts of machine learning in computer security”. In: *USENIX Sec*. 2022 (cit. on pp. 22, 34, 58, 74, 78, 79, 118–120).

[Arr+20] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. In: *Information fusion* (2020) (cit. on p. 2).

[Bac+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, et al. “On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation”. In: *PLOS ONE* 10.7 (2015) (cit. on p. 87).

[Bai+21] Sanghita Baidya, Vidyasagar Potdar, Partha Pratim Ray, and Champa Nandi. “Reviewing the opportunities, challenges, and future directions for the digitalization of energy”. In: *Energy Research & Social Science* 81 (2021), p. 102243 (cit. on p. 32).

[Bal+20] Rachana Balasubramanian, Samuel Sharpe, Brian Barr, Jason Wittenbach, and C Bayan Bruss. “Latent-cf: a simple baseline for reverse counterfactual explanations”. In: *arXiv:2012.09301* (2020) (cit. on p. 58).

[Bar+06] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. “Can machine learning be secure?” In: *ACM ASIACCS*. 2006 (cit. on p. 73).

[Bec+20] Tamara Becejac, Crystal Eppinger, Aditya Ashok, Urmila Agrawal, and James O'Brien. "Prime: a real-time cyber-physical systems testbed: from wide-area monitoring, protection, and control prototyping to operator training and beyond". In: *IET Cyber-Physical Systems: Theory & Applications* 5.2 (2020) (cit. on p. 62).

[Bel21] Luca Belli. "Cybersecurity policymaking in the BRICS countries: From addressing national priorities to seeking international cooperation". In: *The African Journal of Information and Communication* 28 (2021), pp. 1–14 (cit. on p. 31).

[BIA22] Shameek Bhattacharjee, Mohammad Jaminur Islam, and Sahar Abedzadeh. "Robust anomaly based attack detection in smart grids under data poisoning attacks". In: *Proceedings of the 8th ACM on Cyber-Physical System Security Workshop*. 2022, pp. 3–14 (cit. on pp. 51–53).

[Bie98] Michel Bierlaire. "Discrete choice models". In: *Operations research and decision aid methodologies in traffic and transportation management*. Springer, 1998, pp. 203–227 (cit. on pp. 99, 100).

[Big+13] Battista Biggio, Igino Corona, Davide Maiorca, et al. "Evasion Attacks against Machine Learning at Test Time". In: *ECML/PKDD*. 2013 (cit. on pp. 23, 65).

[BR18] Battista Biggio and Fabio Roli. "Wild patterns: Ten years after the rise of adversarial machine learning". In: *Pattern Recognition* (2018) (cit. on pp. 1, 54).

[BMB13] Rajib Biswas, Sayantan Mukherjee, and Samir Kumar Bandyopadhyay. "DCT domain encryption in LSB steganography". In: *2013 5th International Conference and Computational Intelligence and Communication Networks*. IEEE. 2013, pp. 405–408 (cit. on p. 140).

[Bon+23] Atef H Bondok, Mohamed Mahmoud, Mahmoud M Badr, et al. "Novel Evasion Attacks against Adversarial Training Defense for Smart Grid Federated Learning". In: *IEEE Access* (2023) (cit. on pp. 51, 144).

[Bre01] Leo Breiman. "Random forests". In: *Machine learning* (2001) (cit. on pp. 28, 121).

[Bro+17] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. "Adversarial patch". In: *arXiv preprint arXiv:1712.09665* (2017) (cit. on p. 84).

[Bru08] Christoph Brunner. "IEC 61850 for power system communication". In: *2008 IEEE/PES Transmission and Distribution Conference and Exposition*. IEEE. 2008, pp. 1–6 (cit. on p. 18).

[P11] Juan Caballero, Gibran Gomez, Srdjan Matic, et al. "The rise of GoodFATR: a novel accuracy comparison methodology for indicator extraction tools". In: *Future Generation Computer Systems* 144 (2023), pp. 74–89 (cit. on p. 15).

[Cai+24] Ruichu Cai, Yuxuan Zhu, Jie Qiao, et al. “Where and how to attack? A causality-inspired recipe for generating counterfactual adversarial examples”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024 (cit. on p. 100).

[Cam+25] Riccardo Campi, Mathyas Giudici, Nicolo Oreste Pinciroli Vago, Marco Brambilla, and Piero Fraternali. “Enhancing Human-AI Collaboration through a Conversational Agent for Energy Efficiency”. In: *Proceedings of the AAAI Symposium Series*. Vol. 5. 1. 2025, pp. 52–55 (cit. on p. 20).

[Can+24] Sine Canbolat, Clemens Fruboese, Ghada Elbez, and Veit Hagenmeyer. “Assessing GNSS Vulnerabilities in Smart Grids”. In: *DIMVA*. 2024 (cit. on pp. 57, 110, 112, 113).

[Cao+17] Yue Cao, Houbing Song, Omprakash Kaiwartya, et al. “Electric vehicle charging recommendation and enabling ICT technologies: recent advances and future directions”. In: *IEEE COMSOC MMTC Communications-Frontiers* 12.6 (2017), pp. 23–32 (cit. on p. 21).

[CW18] Nicholas Carlini and David Wagner. “Audio adversarial examples: Targeted attacks on speech-to-text”. In: *2018 IEEE security and privacy workshops (SPW)*. IEEE. 2018, pp. 1–7 (cit. on pp. 3, 21).

[CW17] Nicholas Carlini and David Wagner. “Towards evaluating the robustness of neural networks”. In: *2017 IEEE symposium on security and privacy (SP)*. IEEE. 2017, pp. 39–57 (cit. on pp. 3, 21).

[CCP24] Alberto Castagnaro, Mauro Conti, and Luca Pajola. “Offensive AI: Enhancing Directory Brute-forcing Attack with the Use of Language Models”. In: *AISec @ ACM CCS*. 2024 (cit. on p. 151).

[Cennd] Center for Strategic and International Studies. *Significant Cyber Incidents*. `https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents`. Accessed: Feb 2025. n.d. (Cit. on p. 36).

[CERnd] CERT China. *CERT China*. `https://www.cert.org.cn/publish/english/55/index.html`. Accessed: Feb 2025. n.d. (Cit. on p. 36).

[Cha+20] Hung-Chang Chang, Chen-Yi Lin, Da-Jyun Liao, and Tung-Ming Koo. “The Modbus protocol vulnerability test in industrial control systems”. In: *2020 International conference on cyber-enabled distributed computing and knowledge discovery (CyberC)*. IEEE. 2020, pp. 375–378 (cit. on p. 19).

[Cha19] Adrian R Chavez. “Moving target defense to improve industrial control system resiliency”. In: *Industrial Control Systems Security and Resiliency: Practice and Theory*. Springer, 2019 (cit. on p. 145).

[Che+15] Bo Chen, Nishant Pattanaik, Ana Goulart, Karen L Butler-Purry, and Deepa Kundur. “Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed”. In: *CQR*. 2015 (cit. on p. 67).

[CTZ19] Yize Chen, Yushi Tan, and Baosen Zhang. “Exploiting vulnerabilities of load forecasting through adversarial attacks”. In: *Proceedings of the tenth ACM international conference on future energy systems*. 2019, pp. 1–11 (cit. on pp. 50–53).

[CG11] Richard Colbaugh and Kristin Glass. “Proactive defense for evolving cyber threats”. In: *ISI ’11*. 2011 (cit. on p. 1).

[Cox+07] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Burlington, MA: Morgan Kaufmann, 2007 (cit. on pp. 122, 124, 140).

[Dal+04] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. “Adversarial classification”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 99–108 (cit. on pp. 3, 21).

[DNM21] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. “A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks”. In: *Acm Computing Surveys (Csur)* 54.2 (2021), pp. 1–38 (cit. on p. 21).

[Dem+19] Ambra Demontis, Marco Melis, Maura Pintor, et al. “Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks”. In: *28th USENIX security symposium (USENIX security 19)*. 2019, pp. 321–338 (cit. on p. 76).

[Don+21] Jialiang Dong, Zhitao Guan, Longfei Wu, Xiaojiang Du, and Mohsen Guizani. “A sentence-level text adversarial attack algorithm against IIoT based smart grid”. In: *Computer Networks* 190 (2021), p. 107956 (cit. on pp. 20, 51–53).

[Don+20] Xiaoyi Dong, Jiangfan Han, Dongdong Chen, et al. “Robust Superpixel-Guided Attentional Adversarial Attack”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020 (cit. on pp. 126, 140).

[Dzu+05] Dacfey Dzung, Martin Naedele, Thomas P Von Hoff, and Mario Crevatin. “Security for industrial communication systems”. In: *Proceedings of the IEEE* 93.6 (2005), pp. 1152–1177 (cit. on p. 18).

[EKT16] Oliver Eigner, Philipp Kreimel, and Paul Tavolato. “Detection of man-in-the-middle attacks on industrial control networks”. In: *2016 International Conference on Software Security and Assurance (ICSSA)*. IEEE. 2016, pp. 64–69 (cit. on p. 150).

[P8] Ghada Elbez, Gustavo Sánchez, Sine Canbolat, et al. “Insights and Lessons Learned from a Realistic Smart Grid Testbed for Cybersecurity Research”. In: *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*. 2025, pp. 805–812 (cit. on pp. 14, 59, 60, 62).

[EY20] Misheel Enkhbaatar and Tatsuya Yamazaki. “Study on Automatic LED Monitoring System for Data Center Devices”. In: *2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT)*. Vol. 1. IEEE. 2020, pp. 340–343 (cit. on p. 83).

[ETDnd] ETDA. *APT Groups*. `https://apt.etda.or.th/cgi-bin/aptgroups.cgi`. Accessed: Feb 2025. n.d. (Cit. on p. 35).

[EURnd] EUREPOC. *EUREPOC Table View*. `https://eurepoc.eu/table-view/`. Accessed: Feb 2025. n.d. (Cit. on p. 35).

[Expnd] Explosion AI. *Rule-based Matching — PhraseMatcher*. `https://spacy.io/usage/rule-based-matching#phrasematcher`. Accessed: 25 July 2025. n.d. (Cit. on p. 38).

[Eyk+18] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, et al. “Robust physical-world attacks on deep learning visual classification”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1625–1634 (cit. on p. 84).

[Eyn+24] Ahmad Eynawi, Aneeqa Mumrez, Ghada Elbez, and Veit Hagenmeyer. “Machine learning-based feature selection for intrusion detection systems in IEC 61850-based digital substations”. In: *SGComm*. IEEE. 2024 (cit. on pp. 63, 69, 130).

[Fah+24] Abdulrahman Fahim, Shitong Zhu, Zhiyun Qian, et al. “DNS Exfiltration Guided by Generative Adversarial Networks”. In: *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2024, pp. 580–599 (cit. on p. 124).

[Fan+23] Kaisheng Fan, Weizhe Zhang, Guangrui Liu, and Hui He. “FMSA: a meta-learning framework-based fast model stealing attack technique against intelligent network intrusion detection systems”. In: *Cybersecurity* 6.1 (2023), p. 35 (cit. on p. 125).

[Fir14] FireEye Labs. *Regin: Sophisticated State-Sponsored Espionage Platform Use of Steganography*. White Paper, FireEye, Inc. 2014 (cit. on p. 125).

[Fog+06] Prahlad Fogla, Monirul Sharif, Roberto Perdisci, Oleg Kolesnikov, and Wenke Lee. “Polymorphic Blending Attacks”. In: *USENIX Sec*. 2006 (cit. on p. 23).

[Fra20] Rodrigo Fracalossi de Moraes. “Whither security cooperation in the BRICS? Between the protection of norms and domestic politics dynamics”. In: *Global Policy* 11.4 (2020), pp. 439–447 (cit. on pp. 31, 32).

[Frand] Fraunhofer FKIE. *Malpedia Actors*. `https://malpedia.caad.fkie.fraunhofer.de/actors`. Accessed: Feb 2025. n.d. (Cit. on p. 35).

[Fri09] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Methods*. Cambridge, UK: Cambridge University Press, 2009 (cit. on pp. 124, 133, 139).

[Ge+25] Shichao Ge, Peijun Ye, Renrui Zhang, et al. “LLM-Driven Cognitive Modeling for Personalized Travel Generation”. In: *IEEE Transactions on Computational Social Systems* (2025) (cit. on p. 100).

[GBC18] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Available at `https://www.deeplearningbook.org`. Cambridge, MA: MIT Press, 2018 (cit. on pp. 124, 125).

[GSS] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *ICLR, 2015*. Ed. by Yoshua Bengio and Yann LeCun (cit. on p. 70).

[GS21] Zhang Guihai and Biplab Sikdar. “Adversarial machine learning against false data injection attack detection for smart grid demand response”. In: *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE. 2021, pp. 352–357 (cit. on pp. 51–53).

[GMC16] Prageeth Gunathilaka, Daisuke Mashima, and Binbin Chen. “Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions”. In: *CPSS ’24*. 2016 (cit. on p. 62).

[Gun+22] Sam Gunn, Doseok Jang, Orr Paradise, Lucas Spangher, and Costas J Spanos. “Adversarial poisoning attacks on reinforcement learning-driven energy pricing”. In: *BuildSys ’22*. 2022 (cit. on pp. 51–53).

[Ham+21] Kian Hamedani, Lingjia Liu, Jithin Jagannath, and Yang Yi. “Adversarial classification of the attacks on smart grids using game theory and deep learning”. In: *Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning*. 2021, pp. 13–18 (cit. on pp. 51–53).

[HEF19] Eman Hammad, Mellitus Ezeme, and Abdallah Farraj. “Implementation and development of an offline co-simulation testbed for studies of power systems cyber security and control verification”. In: *International Journal of Electrical Power & Energy Systems* 104 (2019) (cit. on p. 62).

[HF20] Masoud Hashemi and Ali Fathi. “Permuteattack: Counterfactual explanation of machine learning credit scorecards”. In: *arXiv:2008.10138* (2020) (cit. on p. 58).

[He+22] Yang He, Baisheng Nie, Jianhui Zhang, Priyan Malarvizhi Kumar, and BalaAnand Muthu. “Fault detection and diagnosis of cyber-physical system using the computer vision and image processing”. In: *Wireless personal communications* 127.3 (2022), pp. 2141–2160 (cit. on p. 83).

[HG03] David A Hensher and William H Greene. “The mixed logit model: the state of practice”. In: *Transportation* 30 (2003), pp. 133–176 (cit. on p. 101).

[Hey18] Vahid Heydari. “Moving target defense for securing SCADA communications”. In: *IEEE Access* (2018) (cit. on p. 145).

[Hol+21] Jordan Holland, Paul Schmitt, Nick Feamster, and Prateek Mittal. “New directions in automated traffic analysis”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021, pp. 3366–3383 (cit. on pp. 122, 125).

[Hu+23] Chengyin Hu, Yilong Wang, Kalibinuer Tiliwalidi, and Wen Li. “Adversarial laser spot: Robust and covert physical-world attack to dnns”. In: *Asian conference on machine learning*. PMLR. 2023, pp. 483–498 (cit. on p. 84).

[Hua+20] Jian Huang, Junzhe Wang, Yihua Tan, Dongrui Wu, and Yu Cao. “An automatic analog instrument reading system using computer vision and inspection robot”. In: *IEEE Transactions on Instrumentation and Measurement* 69.9 (2020), pp. 6322–6335 (cit. on p. 84).

[HL23] Rong Huang and Yuancheng Li. “Adversarial Attack Mitigation Strategy for Machine Learning-Based Network Attack Detection Model in Power System”. In: *IEEE Transactions on Smart Grid* 14.3 (2023), pp. 2367–2376 (cit. on pp. 51, 52).

[HAG21] Lance Y Hunter, Craig Douglas Albert, and Eric Garrett. “Factors that motivate state-sponsored cyberattacks”. In: *The Cyber Defense Review* 6.2 (2021), pp. 111–128 (cit. on p. 32).

[Igu+18] Raúl Igual, Carlos Medrano, Francisco Javier Arcega, and Gabriela Mantescu. “Integral mathematical model of power quality disturbances”. In: *ICHQP*. IEEE. 2018 (cit. on p. 70).

[Incnd] Incident Database. *Incident Summaries*. `https://incidentdatabase.ai/summaries/incidents/`. Accessed: Feb 2025. n.d. (Cit. on p. 36).

[Ing+12] David ME Ingram, Pascal Schaub, Richard R Taylor, and Duncan A Campbell. “Performance analysis of IEC 61850 sampled value process bus networks”. In: *IEEE Transactions on industrial informatics* 9.3 (2012), pp. 1445–1454 (cit. on p. 18).

[Int13] International Electrotechnical Commission. *IEC 61850 Communication networks and systems for power utility automation*. Standard, IEC 61850 Series. Available at: `https://webstore.iec.ch/publication/6028`. 2013 (cit. on p. 17).

[Int03] International Organization for Standardization. *ISO 9506: Manufacturing Message Specification (MMS)*. Standard, ISO 9506. Available at: `https://www.iso.org/standard/17834.html`. 2003 (cit. on p. 18).

[Jaj+12] Sushil Jajodia, Anup K Ghosh, VS Subrahmanian, et al. *Moving Target Defense II: Application of Game Theory and Adversarial Modeling*. Vol. 100. Springer Science & Business Media, 2012 (cit. on p. 145).

[JHL18] JQ James, Yunhe Hou, and Victor OK Li. “Online false data injection attack detection with wavelet transform and deep neural networks”. In: *IEEE Transactions on Industrial Informatics* 14.7 (2018), pp. 3271–3280 (cit. on p. 2).

[JSJ23] Guillaume Jeanneret, Loıc Simon, and Frédéric Jurie. “Adversarial counterfactual visual explanations”. In: *CVPR*. 2023 (cit. on p. 57).

[Jia+23] Yan Jia, Zhaoquan Gu, Lei Du, et al. “Artificial intelligence enabled cyber security defense for smart cities: A novel attack detection framework based on the MDATA model”. In: *Knowledge-Based Systems* 276 (2023), p. 110781 (cit. on p. 34).

[Joh+25] Rayner Johnson, Senthil Krishnamurthy, Haltor Mataifa, and Mohammed Esmail. “A comparison study between Modbus and IEC 61850 MMS Protocols”. In: *SAUPEC*. IEEE. 2025 (cit. on p. 144).

[Kan+22] Nandha Kumar Kandasamy, Sarad Venugopalan, Tin Kit Wong, and Nicholas Junming Leu. “An electric power digital twin for cyber security testing, research and education”. In: *Computers and Electrical Engineering* 101 (2022) (cit. on p. 62).

[Kas18] Kaspersky Lab APT Research Team. *Turla: The Use of Image-based Steganography in Advanced Espionage Campaigns*. Technical Report, Kaspersky Lab. 2018 (cit. on p. 125).

[KAS] KASTEL. *KASTEL - Security and Privacy for Future Energy Systems*. Available at `https://www.kastel.kit.edu/english/energie.php(accessed17/08/2023)` (cit. on pp. 55, 76).

[P9] Nicolai Kellerer, Gustavo Sánchez, Hermenegildo Alberto, Veit Hagenmeyer, and Ghada Elbez. “Attacks on the Siemens S7 Protocol Using an Industrial Control System Testbed”. In: *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*. 2025, pp. 770–779 (cit. on pp. 14, 120).

[Kik+24] Daisuke Kikuta, Hiroki Ikeuchi, Kengo Tajiri, and Yuusuke Nakano. “RouteExplainer: An Explanation Framework for Vehicle Routing Problem”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2024 (cit. on p. 100).

[Kim21] Watson Kimberly K. “Deploying Indicators of Compromise (IoCs) for Network Defense”. In: *Cybersecurity Automation and Threat Intelligence Sharing Best Practices, CISA*. 2021 (cit. on pp. 122, 124).

[KIO18] KIOS. *KIOS Testbeds for critical infrastructure systems*. 2018 (cit. on p. 62).

[Kon+17] Weicong Kong, Zhao Yang Dong, Youwei Jia, et al. “Short-term residential load forecasting based on LSTM recurrent neural network”. In: *IEEE transactions on smart grid* 10.1 (2017), pp. 841–851 (cit. on p. 2).

[Kou+15] Georgia Koutsandria, Reinhard Gentz, Mahdi Jamei, et al. “A real-time testbed environment for cyber-physical security on the power grid”. In: *CPS-SPC ’15*. 2015 (cit. on p. 62).

[KL21] Aditya Kuppa and Nhien-An Le-Khac. “Adversarial XAI methods in cybersecurity”. In: *IEEE TIFS* (2021) (cit. on pp. 58, 76).

[Lai+17] Christine Lai, Nicholas Jacobs, Shamina Hossain-McKenzie, et al. “Cyber security primer for DER vendors, aggregators, and grid operators”. In: *Tech. Rep.* 12 (2017) (cit. on p. 65).

[Lan11] Ralph Langner. “Stuxnet: Dissecting a cyberwarfare weapon”. In: *IEEE security & privacy* 9.3 (2011), pp. 49–51 (cit. on p. 18).

[LWM24] Johann Laux, Sandra Wachter, and Brent Mittelstadt. “Trustworthy artificial intelligence and the European Union AI act: On the conflation of trustworthiness and acceptability of risk”. In: *Regulation & Governance* (2024) (cit. on p. 3).

[LGS24] Sozon Leventopoulos, Dimitris Gritzalis, and George Stergiopoulos. “Malware as a Geopolitical Tool”. In: *Malware: Handbook of Prevention and Detection*. Springer, 2024, pp. 251–271 (cit. on p. 32).

[Li+19a] Gang Li, Weiying Wang, Ying Qi, and Min Cui. “Defect text analysis method of electric power equipment based on double-layer bidirectional LSTM model”. In: *2019 IEEE 3rd International Electrical and Energy Conference (CIEEC)*. IEEE. 2019, pp. 1318–1324 (cit. on p. 20).

[Li+19b] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. “TextBugger: Generating Adversarial Text Against Real-world Applications”. In: *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. 2019 (cit. on p. 57).

[Li+19c] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. “Textbugger: Generating adversarial text against real-world applications”. In: *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*. 2019 (cit. on p. 21).

[Li+24a] Li Li, Ke Xu, Fei Ye, et al. “State of the Art and Development Trends for Inspection Robots Applied in Substations”. In: *Proceedings of the 2024 4th International Conference on Control and Intelligent Robotics*. 2024, pp. 44–48 (cit. on p. 20).

[Li+24b] Qiang Li, Feng Zhao, Zhongxu Li, Zhen Qiu, and Xiaofeng Wu. “Feature Extraction Method of Electric Power User-side Metering Data Based on Text Classification”. In: *Proceedings of the 2024 9th International Conference on Cyber Security and Information Engineering*. 2024, pp. 852–857 (cit. on p. 20).

[Li+24c] Yang Li, Yan Li, Zhenli Wang, et al. “Typical Defect Recognition Technology for Substations Based on Improved Deep Learning Algorithms”. In: *Proceedings of the 2024 International Conference on Image Processing, Intelligent Control and Computer Engineering*. 2024, pp. 223–228 (cit. on p. 20).

[Lie+23] Kevin van Liebergen, Juan Caballero, Platon Kotzias, and Chris Gates. “A Deep Dive into the VirusTotal File Feed”. In: *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer. 2023, pp. 155–176 (cit. on p. 34).

[LAC20] Romulo Gonçalves Lins, Paulo Ricardo Marques de Araujo, and Marcio Corazzim. “In-process machine vision monitoring of tool wear for Cyber-Physical Production Systems”. In: *Robotics and computer-integrated manufacturing* 61 (2020), p. 101859 (cit. on p. 83).

[Liu+18] Jing Liu, Ming Nie, Hao Wu, and Xiaoming Mai. “An image-based accurate alignment for substation inspection robot”. In: *2018 International Conference on Power System Technology (POWERCON)*. IEEE. 2018, pp. 4113–4117 (cit. on p. 83).

[LS19] Tian Liu and Tao Shu. “Adversarial false data injection attack against nonlinear ac state estimation with ann in smart grid”. In: *Security and Privacy in Communication Networks: 15th EAI International Conference, SecureComm 2019, Orlando, FL, USA, October 23–25, 2019, Proceedings, Part II 15*. Springer. 2019, pp. 365–379 (cit. on pp. 51, 52).

[Lu+24] Qiuyu Lu, Jun’e Li, Zhao Peng, et al. “Detecting the cyber-physical-social cooperated APTs in high-DER-penetrated smart grids: Threats, current work and challenges”. In: *Computer Networks* 254 (2024), p. 110776 (cit. on p. 33).

[LL17] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Advances in Neural Information Processing Systems 30*. 2017 (cit. on pp. 27, 75, 76, 135, 147).

[Luo+16] Fengji Luo, Gianluca Ranzi, Xibin Wang, and Zhao Yang Dong. “Service recommendation in smart grid: vision, technologies, and applications”. In: *2016 9th International Conference on Service Science (ICSS)*. IEEE. 2016, pp. 31–38 (cit. on p. 21).

[Lv+24] Jun Lv, Hejun Zhang, Xiao Chen, Jin Huang, and Haocheng Zhou. “An image recognition method for intelligent inspection of power grid equipment”. In: *Proceedings of the 2024 International Conference on Power Electronics and Artificial Intelligence*. 2024, pp. 463–467 (cit. on p. 20).

[Lyo09] Gordon Fyodor Lyon. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009 (cit. on p. 66).

[MDB19] Tamara Maliarchuk, Yuriy Danyk, and Chad Briggs. “Hybrid warfare and cyber effects in energy infrastructure”. In: *Connections* 18.1/2 (2019), pp. 93–110 (cit. on p. 32).

[Mar+21] Peter Martin, Jian Fan, Taejin Kim, Konrad Vesey, and Lloyd Greenwald. “Toward effective moving target defense against adversarial ai”. In: *MILCOM*. IEEE. 2021 (cit. on p. 145).

[McF74] Daniel McFadden. “The measurement of urban travel demand”. In: *Journal of public economics* 3.4 (1974), pp. 303–328 (cit. on p. 101).

[MN03] Rebecca T Mercuri and Peter G Neumann. “Security by obscurity”. In: *Communications of the ACM* 46 (2003) (cit. on p. 65).

[MITnd] MITRE. *MITRE ATT&CK Groups*. `https://attack.mitre.org/groups/`. Accessed: Feb 2025. n.d. (Cit. on p. 35).

[Mo+19] Huaxiao Mo, Tingting Song, Bolin Chen, Weiqi Luo, and Jiwu Huang. "Enhancing JPEG steganography using iterative adversarial examples". In: *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE. 2019, pp. 1–6 (cit. on pp. 126, 140).

[Mod06] Modbus Organization, Inc. *Modbus Application Protocol Specification V1.1b*. Available at: `http://www.modbus.org/specs.php`. 2006 (cit. on p. 19).

[MH20] Gautam Raj Mode and Khaza Anuarul Hoque. "Adversarial Examples in Deep Learning for Multivariate Time Series Regression". In: *arXiv preprint arXiv:2009.11911* (2020) (cit. on pp. 50–53).

[MLG24] Hesamodin Mohammadian, Arash Habibi Lashkari, and Ali A Ghorbani. "Poisoning and Evasion: Deep Learning-Based NIDS under Adversarial Attacks". In: *PST*. IEEE. 2024 (cit. on p. 144).

[Moh+24] Saad Hammood Mohammed, Abdulmajeed Al-Jumaily, Mandeep S Jit Singh, et al. "A review on the evaluation of feature selection using machine learning for cyber-attack detection in smart grid". In: *Ieee Access* 12 (2024), pp. 44023–44042 (cit. on p. 19).

[Mon+19] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. "Layer-wise relevance propagation: an overview". In: *Explainable AI: interpreting, explaining and visualizing deep learning* (2019) (cit. on p. 28).

[MST20] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. "Explaining machine learning classifiers through diverse counterfactual explanations". In: *FAT*. 2020 (cit. on p. 58).

[Mou+23] Nour Moustafa, Nickolaos Koroniotis, Marwa Keshk, Albert Y Zomaya, and Zahir Tari. "Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions". In: *IEEE Communications Surveys & Tutorials* 25.3 (2023), pp. 1775–1807 (cit. on p. 34).

[Mum+23] Aneeqa Mumrez, Muhammad M Roomi, Heng Chuan Tan, et al. "Comparative Study on Smart Grid Security Testbeds Using MITRE ATT&CK Matrix". In: *SmartGridComm '23*. 2023 (cit. on p. 55).

[P7] Aneeqa Mumrez, Gustavo Sánchez, Ghada Elbez, and Veit Hagenmeyer. "On Evasion of Machine Learning-based Intrusion Detection in Smart Grids". In: *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. 2023 (cit. on pp. 14, 34, 50–54, 63–67, 74, 144, 150).

[ML05] S. J. Murdoch and S. Lewis. "Embedding Covert Channels in IP Packet Headers". In: *Proceedings of the 5th International Workshop on Privacy Enhancing Technologies (PET)*. Springer. 2005, pp. 147–162 (cit. on pp. 122, 124).

[Na+24] Qionglan Na, Yixi Yang, Dan Su, et al. "Speech Recognition Model Inspired on Large Language Model for Smart Grid Dispatching". In: *Proceedings of the 2024 International Conference on Power Electronics and Artificial Intelligence*. 2024, pp. 439–442 (cit. on p. 20).

[Nas+20] Ben Nassi, Yisroel Mirsky, Dudi Nassi, et al. “Phantom of the adas: Securing advanced driver-assistance systems from split-second phantom attacks”. In: *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. 2020, pp. 293–308 (cit. on p. 84).

[Naz+23] Fatemeh Nazary, Yashar Deldjoo, Tommaso Di Noia, Carmelo Ardito, and Eugenio Di Sciascio. “Smart Electrical grids Under the Lens of Adversarial Attacks”. In: (2023) (cit. on pp. 51, 52).

[Nel+24] Raluca Nelega, Gergo Kovacs, Alexandru Oprea, Romulus Valeriu Flaviu Turcu, and Emanuel Puschita. “Real-Time Monitoring System of Photovoltaic Power Plant Using UAV Thermal Images”. In: *2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE. 2024, pp. 372–377 (cit. on p. 20).

[Neu+22] Subash Neupane, Jesse Ables, William Anderson, et al. “Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities”. In: *IEEE Access* 10 (2022), pp. 112392–112415 (cit. on p. 34).

[Ngu+20] Dinh-Luan Nguyen, Sunpreet S Arora, Yuhang Wu, and Hao Yang. “Adversarial light projection attacks on face recognition systems: A feasibility study”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 814–815 (cit. on p. 84).

[Ngu+24] Lan-Huong Nguyen, Van-Linh Nguyen, Ren-Hung Hwang, et al. “Towards secured smart grid 2.0: exploring security threats, protection models, and challenges”. In: *IEEE Communications Surveys & Tutorials* (2024) (cit. on p. 33).

[Noc18] Julien Nocetti. “The geopolitics of cyberconflict”. In: *Politique étrangère* 2 (2018), pp. 15–27 (cit. on p. 31).

[Oka+24] Satoshi Okada, Houda Jmila, Kunio Akashi, et al. “XAI-driven adversarial attacks on network intrusion detectors”. In: *EICC*. 2024 (cit. on p. 58).

[Oka+25] Satoshi Okada, Houda Jmila, Kunio Akashi, et al. “XAI-driven black-box adversarial attacks on network intrusion detectors”. In: *International Journal of Information Security* 24.3 (2025), pp. 1–15 (cit. on p. 58).

[Ols+23] Daniel Olszewski, Allison Lu, Carson Stillman, et al. “" Get in Researchers; We’re Measuring Reproducibility": A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2023, pp. 3433–3459 (cit. on pp. 50, 53, 54, 58).

[Ope24] OpenAI Threat Intelligence. *Influence and Cyber Operations: An Update*. `https://cdn.openai.com/threat-intelligence-reports/influence-and-cyber-operations-an-update_October-2024.pdf`. Accessed: Feb 2025. Oct. 2024 (cit. on pp. 48, 145).

[Oye19] Ibukun Adesile Oyewumi. *ISAAC: The Idaho Cyber-physical System Smart Grid Cybersecurity Testbed*. University of Idaho, 2019 (cit. on p. 62).

[Pan+21a] Alexander Pan, Yongkyun Lee, Huan Zhang, Yize Chen, and Yuanyuan Shi. "Improving robustness of reinforcement learning for power system control with adversarial training". In: *RL4RL @ ICML '21* (2021) (cit. on pp. 51, 53).

[Pan+23] Cecilia Panigutti, Ronan Hamon, Isabelle Hupont, et al. "The role of explainable AI in the context of the AI Act". In: *FAccT*. 2023 (cit. on p. 3).

[Pan+21b] George Pantelis, Petros Petrou, Sophia Karagiorgou, and Dimitrios Alexandrou. "On strengthening smes and mes threat intelligence and awareness by identifying data breaches, stolen credentials and illegal activities on the dark web". In: *Proceedings of the 16th International Conference on Availability, Reliability and Security*. 2021, pp. 1–7 (cit. on p. 33).

[Pap+17] N. Papernot, P. McDaniel, I. Goodfellow, et al. "Practical Black-Box Attacks against Deep Learning Systems Using Adversarial Examples". In: *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIACCS)*. 2017, pp. 506–519 (cit. on p. 125).

[PGB20] Christopher Parian, Terry Guldimann, and Sajal Bhatia. "Fooling the master: Exploiting weaknesses in the modbus protocol". In: *Procedia Computer Science* 171 (2020) (cit. on p. 66).

[Paw+24] Marek Pawlicki, Aleksandra Pawlicka, Rafał Kozik, and Michał Choraś. "Explainability versus Security: The Unintended Consequences of xAI in Cybersecurity". In: *SecTL@AsiaCCS*. 2024 (cit. on p. 58).

[Pei+23] João Peixoto, João Sousa, Ricardo Carvalho, et al. "End-to-end solution for analog gauge monitoring using computer vision in an iot platform". In: *Sensors* 23.24 (2023), p. 9858 (cit. on p. 83).

[Pie+20] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. "Intriguing properties of adversarial ml attacks in the problem space". In: *2020 IEEE symposium on security and privacy (SP)*. IEEE. 2020, pp. 1332–1349 (cit. on pp. 22, 25, 26, 74, 105, 118, 140).

[Pol24] Miles Pollard. "A case study of Russian cyber-attacks on the Ukrainian power grid: Implications and best practices for the United States". In: *Pepperdine Policy Review* 16.1 (2024), p. 1 (cit. on p. 32).

[PwCnd] PwC Germany. *Under the Lens: The Energy Sector*. `https://www.pwc.de/de/energiewirtschaft/under-the-lens-the-energy-sector.pdf`. Accessed: Feb 2025. n.d. (Cit. on p. 33).

[PyP] PyPI. *NetfilterQueue: Python bindings for libnetfilterqueue*. Available at `https://pypi.org/project/NetfilterQueue/(accessed17/08/2023))` (cit. on p. 66).

[QQ20] Meikang Qiu and Han Qiu. "Review on image processing based adversarial example defenses in computer vision". In: *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity)*. IEEE. 2020, pp. 94–99 (cit. on p. 126).

[Qui+22] Flavio Quizhpi-Palomeque, Freddy Jiménez, Pedro Rivera, Mateo Quizhpi-Cuesta, and Francisco Gómez-Juca. “Implementation of an iec61850 virtual relay network in a protection laboratory”. In: *ROPEC ’22*. IEEE. 2022 (cit. on p. 62).

[RKG21] Karl Rehrl, Stefan Kranzinger, and Simon Gröchenig. “Which quality is a route? A methodology for assessing route quality using spatio-temporal metrics”. In: *Transactions in GIS* 25.2 (2021), pp. 869–896 (cit. on p. 100).

[Ren+21] Chao Ren, Xiaoning Du, Yan Xu, et al. “Vulnerability analysis, robustness verification, and mitigation strategy for machine learning-based power system stability assessment model under adversarial examples”. In: *IEEE Transactions on Smart Grid* 13.2 (2021), pp. 1622–1632 (cit. on pp. 51–53).

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “"Why should i trust you?" Explaining the predictions of any classifier”. In: *SIGKDD*. 2016, pp. 1135–1144 (cit. on p. 27).

[Roy+19] Abhishek Roy, Anshuman Chhabra, Charles A Kamhoua, and Prasant Mohapatra. “A moving target defense against adversarial machine learning”. In: *ACM/IEEE SEC*. 2019 (cit. on p. 145).

[RNB18] Stephan Ruhe, Steffen Nicolai, and Peter Bretschneider. “Modelling and simulation of electrical phenomena in a real time test bench”. In: *UPEC ’18*. IEEE. 2018 (cit. on p. 62).

[SW22] Moein Sabounchi and Jin Wei-Kocsis. “A practical adversarial attack on contingency detection of smart energy systems”. In: *ISGT ’22*. 2022 (cit. on pp. 51–53).

[P1] Gustavo Sanchez, Ghada Elbez, and Veit Hagenmeyer. “Attacking Learning-based Models in Smart Grids: Current Challenges and New Frontiers”. In: *Proceedings of the 15th ACM International Conference on Future Energy Systems*. 2024 (cit. on pp. 13, 26, 34, 50, 58, 63, 74, 120, 144, 146, 149).

[P2] Gustavo Sanchez, Ghada Elbez, and Veit Hagenmeyer. “Explainable AI in Data Poisoning Threat Models Across the CIA Triad: A Smart Grid Case Study”. In: *2025 IEEE 7th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*. IEEE. 2025 (cit. on pp. 13, 79, 109, 112, 117, 125).

[P6] Gustavo Sanchez, Wei Li, and Veit Hagenmeyer. “LaserTag: A Tool for Autonomous XAI- Guided Physical Adversarial Perturbations in Industrial Vision Pipelines”. In: *In The 56th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2026 (cit. on pp. 14, 83).

[P14] Gustavo Sanchez and Adam Lundqvist. “Poster: Towards Intelligent Assurance for Autonomous AI Pentesters: Concurrent Compliance Auditing and Self-Augmentation via Execution Trace Analysis”. In: *Proceedings of the 2025 on ACM SIGSAC Conference on Computer and Communications Security*. 2025 (cit. on p. 15).

[P5] Gustavo Sanchez, Muhammed Qasim, Ghada Elbez, and Veit Hagenmeyer. "Steganographic Data Exfiltration for Model Stealing: A Case Study on Energy Critical Infrastructure IEC 61850 Datasets". In: *2025 IEEE International Conference on Big Data (BigData)*. IEEE. 2025 (cit. on pp. 13, 123).

[P10] Gustavo Sanchez, Fatih Ünal, and Alexandra Wins. "Route Choice Prediction Through User Behavior Analysis: Towards Robustness Assessment Under External Perturbations". In: *2025 IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS)*. IEEE. 2025 (cit. on pp. 15, 99).

[P15] Gustavo Sanchez and Shengzhi Zhang. "Wasabi: Leveraging Cross-Lingual Pseudo-Homophones to Evade NLP Moderation Systems". In: *Under Submission*. 2026 (cit. on p. 15).

[P3] Gustavo Sánchez, Ghada Elbez, and Veit Hagenmeyer. "A Global Analysis of Cyber Threats to the Energy Sector: "Currents of Conflict" from a geopolitical perspective". In: *atp magazin* 67.9 (2025), pp. 56–66 (cit. on pp. 13, 20, 32).

[P4] Gustavo Sánchez, Ghada Elbez, and Veit Hagenmeyer. "Lightweight Moving Target Defense for Robust Intrusion Detection in Smart Grids". In: *Proceedings of the Energy Informatics Academy Conference 2025*. 2025 (cit. on pp. 13, 143).

[P12] Gustavo Sánchez, Olakunle Olayinka, and Aryan Pasikhani. "Web Application Penetration Testing with Artificial Intelligence: A Systematic Review". In: *2024 22nd International Symposium on Network Computing and Applications (NCA 2024)*. Institute of Electrical and Electronics Engineers (IEEE). 2024 (cit. on pp. 15, 48, 151).

[San+24] Javier Sande-Ríos, Jesús Canal-Sánchez, Carmen Manzano-Hernández, and Sergio Pastrana. "Threat analysis and adversarial model for Smart Grids". In: *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2024, pp. 130–145 (cit. on pp. 33, 34).

[San+21] Everton Jose Santana, Ricardo Petri Silva, Bruno Bogaz Zarpelão, and Sylvio Barbon Junior. "Detecting and mitigating adversarial examples in regression tasks: a photovoltaic power generation forecasting case study". In: *Information* 12.10 (2021), p. 394 (cit. on pp. 51, 52).

[SZK20] Ali Sayghe, Junbo Zhao, and Charalambos Konstantinou. "Evasion attacks with adversarial deep learning against power system state estimation". In: *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2020, pp. 1–5 (cit. on pp. 51, 52).

[P17] Maxime Schwarzer, Gustavo Sanchez, Thies Möhlenhof, et al. "AI Model Extraction Attacks: Bypassing Single-Client Assumptions in Defenses". In: *International Conference on Military Communication and Information Systems (ICMCIS)*. 2026 (cit. on p. 16).

[P16] Maxime Schwarzer, Gustavo Sanchez, Thies Möhlenhof, et al. “Synthesize, Validate, Steal: A GAN-based and Query-Efficient Attack on Tabular Models”. In: *2026 IEEE Conference on Artificial Intelligence (CAI)*. IEEE. 2026 (cit. on p. 16).

[Sel+17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626 (cit. on p. 85).

[Sha+16] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. “Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition”. In: *Proceedings of the 2016 acm sigsac conference on computer and communications security*. 2016, pp. 1528–1540 (cit. on pp. 21, 84).

[Sha+23] Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, et al. “Survey of vulnerabilities in large language models revealed by adversarial attacks”. In: *arXiv preprint arXiv:2310.10844* (2023) (cit. on p. 3).

[SLL21] Yufeng Shu, Bin Li, and Hui Lin. “Quality safety monitoring of LED chips using deep learning-based vision inspection methods”. In: *Measurement* 168 (2021), p. 108123 (cit. on p. 84).

[Sin+21] Ilias Siniosoglou, Panagiotis Radoglou-Grammatikis, Georgios Efstathopoulos, Panagiotis Fouliras, and Panagiotis Sarigiannidis. “A unified deep learning anomaly detection and classification approach for smart grid environments”. In: *IEEE Transactions on Network and Service Management* 18.2 (2021), pp. 1137–1151 (cit. on p. 34).

[Sko+24] Florian Skopik, Benjamin Akhras, Elisabeth Woisetschläger, et al. “On the Application of Natural Language Processing for Advanced OSINT Analysis in Cyber Defence”. In: *Proceedings of the 19th International Conference on Availability, Reliability and Security*. 2024, pp. 1–10 (cit. on p. 34).

[Sma25] SmartgridADSC Group. *EPICA_Dataset: Electric Power Intrusion and Cyberattack Dataset*. `https://github.com/smartgridadsc/EPIC_Attack_Datasets/tree/main/EPICA_Dataset`. GitHub repository. 2025 (cit. on pp. 129, 130).

[Sme19] Max Smeets. “NATO Members’ Organizational Path Towards Conducting Offensive Cyber Operations: A Framework for Analysis”. In: *2019 11th International Conference on Cyber Conflict (CyCon)*. Vol. 900. IEEE. 2019, pp. 1–15 (cit. on p. 31).

[SY14] Hui Song and Guofu Yin. “Accurately localize and recognize instruments with substation inspection robot in complex environments”. In: *Sensors & Transducers* 174.7 (2014), p. 211 (cit. on p. 83).

[Son+21] Qun Song, Rui Tan, Chao Ren, and Yan Xu. “Understanding credibility of adversarial examples against smart grid: A case study for voltage stability assessment”. In: *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. 2021, pp. 95–106 (cit. on pp. 1, 3, 51–53).

[Sri+17] Siddharth Sridhar, Aditya Ashok, Michael Mylrea, et al. “A testbed environment for buildings-to-grid cyber resilience research and development”. In: *RWS ’17*. IEEE. 2017 (cit. on p. 62).

[Sug+20] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. “Light commands:{Laser-Based} audio injection attacks on {Voice-Controllable} systems”. In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, pp. 2631–2648 (cit. on p. 84).

[STY17a] M. Sundararajan, A. Taly, and Q. Yan. “Axiomatic Attribution for Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Vol. 70. 2017, pp. 3319–3328 (cit. on p. 135).

[STY17b] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic Attribution for Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. 2017, pp. 3319–3328 (cit. on p. 87).

[Swa+24] Subrat Kumar Swain, Ireshwar Kumar, Guandong Bai, and Dan Dongseoung Kim. “PANDA: Practical Adversarial Attack Against Network Intrusion Detection”. In: *Proceedings of the 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S)*. 2024 (cit. on pp. 122, 124, 125).

[Sym11] Symantec Security Response. *The Duqu Incident: Steganographic Data Exfiltration Techniques*. Technical Report, Symantec Corporation. 2011 (cit. on p. 125).

[Tak+23] Abdulrahman Takiddin, Muhammad Ismail, Rachad Atat, Katherine R Davis, and Erchin Serpedin. “Robust Graph Autoencoder-Based Detection of False Data Injection Attacks Against Data Poisoning in Smart Grids”. In: *IEEE Transactions on Artificial Intelligence* (2023) (cit. on pp. 51, 52).

[TIS22] Abdulrahman Takiddin, Muhammad Ismail, and Erchin Serpedin. “Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids”. In: *IEEE Transactions on Smart Grid* 14.1 (2022), pp. 663–676 (cit. on pp. 51–53).

[TIS21] Abdulrahman Takiddin, Muhammad Ismail, and Erchin Serpedin. “Robust detection of electricity theft against evasion attacks in smart grids”. In: *ICC 2021-IEEE International Conference on Communications*. IEEE. 2021, pp. 1–6 (cit. on pp. 51, 52).

[Tak+20] Abdulrahman Takiddin, Muhammad Ismail, Usman Zafar, and Erchin Serpedin. “Robust electricity theft detection against data poisoning attacks in smart grids”. In: *IEEE Transactions on Smart Grid* 12.3 (2020), pp. 2675–2684 (cit. on pp. 51, 52).

[Tan+24] Heng Chuan Tan, Md Adeeb Hossain, Daisuke Mashima, and Zbigniew Kalbarczyk. “High-fidelity Intrusion Detection Datasets for Smart Grid Cybersecurity Research”. In: *2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE. 2024, pp. 340–346 (cit. on pp. 129, 150).

[Tia+22] Jiwei Tian, Buhong Wang, Jing Li, and Charalambos Konstantinou. “Adversarial attack and defense methods for neural network based state estimation in smart grid”. In: *IET Renewable Power Generation* 16.16 (2022), pp. 3507–3518 (cit. on pp. 50–52, 144).

[Tia+21] Jiwei Tian, Buhong Wang, Jing Li, and Zhen Wang. “Adversarial attacks and defense for CNN based power quality recognition in smart grid”. In: *IEEE Transactions on Network Science and Engineering* 9.2 (2021), pp. 807–819 (cit. on pp. 20, 51–53, 63, 70, 71, 79, 144).

[Tod09] Graham H Todd. “Armed attack in cyberspace: deterring asymmetric warfare with an asymmetric definition”. In: *AFL Rev.* 64 (2009), p. 65 (cit. on p. 32).

[El-+23] Ahmed T El-Toukhy, Mohamed MEA Mahmoud, Atef H Bondok, Mostafa M Fouda, and Maazen Alsabaan. “Countering Evasion Attacks for Smart Grid Reinforcement Learning-based Detectors”. In: *IEEE Access* (2023) (cit. on pp. 51–53).

[Tra+16] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. “Stealing Machine Learning Models via Prediction APIs”. In: *Proceedings of the 25th USENIX Security Symposium (USENIX Security)*. 2016, pp. 601–618 (cit. on p. 125).

[TDZ20] Thanh Cong Truong, Quoc Bao Diep, and Ivan Zelinka. “Artificial intelligence in the cyber domain: Offense and defense”. In: *Symmetry* 12.3 (2020), p. 410 (cit. on p. 32).

[Vos+22] Amirkhosro Vosughi, Ali Tamimi, Alexandra Beatrice King, Subir Majumder, and Anurag K Srivastava. “Cyber–physical vulnerability and resiliency analysis for DER integration: A review, challenges and research needs”. In: *Renewable and Sustainable Energy Reviews* 168 (2022) (cit. on p. 65).

[WD01] David Wagner and Drew Dean. “Intrusion Detection via Static Analysis”. In: *IEEE SP*. 2001 (cit. on p. 23).

[Wan+21] Zhiqiang Wan, Hepeng Li, Hang Shuai, Yan Lindsay Sun, and Haibo He. “Adversarial attack for deep reinforcement learning based demand response”. In: *PESGM’21*. 2021 (cit. on pp. 51, 53).

[WP23] Yu Wang and Bikash Pal. “Destabilizing attack and robust defense for inverter-based microgrids by adversarial deep reinforcement learning”. In: *IEEE Transactions on Smart Grid* (2023) (cit. on pp. 51, 53).

[Wei+24] Hui Wei, Hao Tang, Xuemei Jia, et al. “Physical adversarial attack meets computer vision: A decade survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (2024), pp. 9797–9817 (cit. on p. 84).

[Whi+17] David E Whitehead, Kevin Owens, Dennis Gammel, and Jess Smith. “Ukraine cyber-induced power outage: Analysis and practical mitigation strategies”. In: *2017 70th Annual conference for protective relay engineers (CPRE)*. IEEE. 2017, pp. 1–8 (cit. on p. 32).

[Win+24] Alexandra Wins, Christoph Becker, Sascha Alpers, Lukas Kneis, and Andreas Oberweis. “Enhancing Individual Mobility: A Multistage Personalization Approach for Itinerary Planning in Multimodal Networks.” In: *VEHITS*. 2024 (cit. on p. 99).

[Wu+24] Junying Wu, Yanyan Lu, Zixin Li, et al. “Word-Phrase Fusion Encoding Model for Natural Language Understanding in the Electric Power Field”. In: *Proceedings of the 2024 International Conference on Generative Artificial Intelligence and Information Security*. 2024, pp. 43–49 (cit. on p. 20).

[XY19] Luo Xuan and Li Yongzhong. “Research and implementation of Modbus TCP security enhancement protocol”. In: *Journal of Physics: Conference Series*. Vol. 1213. 5. IOP Publishing. 2019, p. 052058 (cit. on p. 64).

[Yan+17] Wei Yang, Deguang Kong, Tao Xie, and Carl A Gunter. “Malware detection in adversarial settings: Exploiting feature evolutions and confusions in android apps”. In: *Proceedings of the 33rd Annual Computer Security Applications Conference*. 2017, pp. 288–302 (cit. on p. 21).

[YXP19] Dian Ang Yap, Joyce Xu, and Vinay Uday Prabhu. “On detecting adversarial inputs with entropy of saliency maps”. In: *CV-COPS @ CVPR ’19* (2019) (cit. on p. 76).

[Yin+24] Xiao Ying, Chen Zuo, Hongyu Xia, et al. “Development of an Inspection Flying Robot Platform for Lightning Rod Maintenance in Substations”. In: *Proceedings of the 2024 3rd International Symposium on Control Engineering and Robotics*. 2024, pp. 486–490 (cit. on p. 20).

[YAC25] Ying Yuan, Giovanni Apruzzese, and Mauro Conti. “Beyond the west: Revealing and bridging the gap between Western and Chinese phishing website detection”. In: *Computers & Security* 148 (2025), p. 104115 (cit. on p. 34).

[Zen+23] Hui Zeng, Biwei Chen, Rongsong Yang, Chenggang Li, and Anjie Peng. “Towards Undetectable Adversarial Examples: A Steganographic Perspective”. In: *International Conference on Neural Information Processing*. Springer. 2023, pp. 172–183 (cit. on pp. 126, 140).

[Zen+18] Kexiong Curtis Zeng, Shinan Liu, Yuanchao Shu, et al. “All your {GPS} are belong to us: Towards stealthy manipulation of road navigation systems”. In: *27th USENIX security symposium*. 2018 (cit. on p. 100).

[ZQS22] Lanting Zeng, Dawei Qiu, and Mingyang Sun. “Resilience enhancement of multi-agent reinforcement learning-based demand response against adversarial attacks”. In: *Applied Energy* (2022) (cit. on pp. 51, 53).

[ZS22] Guihai Zhang and Biplab Sikdar. “Ensemble and Transfer Adversarial Attack on Smart Grid Demand-Response Mechanisms”. In: *2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE. 2022, pp. 53–58 (cit. on pp. 51–53).

[ZWG19] Liang Zhang, Gang Wang, and Georgios B Giannakis. “Real-time power system state estimation and forecasting via deep unrolled neural networks”. In: *IEEE Transactions on Signal Processing* 67.15 (2019), pp. 4069–4077 (cit. on p. 52).

[Zha+23] Yu Zhang, Chao Huo, Huifeng Bai, and Ganghong Zhang. “Adversarial Defense Based on Mimic Defense and Reinforcement Learning for Power Vision Task in Smart Grid”. In: *ACCES ’23*. 2023 (cit. on pp. 51, 53, 144).

[Zha+24a] Zhenyong Zhang, Mengxiang Liu, Mingyang Sun, et al. “Vulnerability of Machine Learning Approaches Applied in IoT-Based Smart Grid: A Review”. In: *IEEE Internet of Things Journal* (2024) (cit. on p. 1).

[Zha+24b] Bin Zhao, Yi Li, Zhibo Zhang, et al. “Power grid false data injection attack detection method based on S transform and LSTM”. In: *4th International Conference on Internet of Things and Smart City (IoTSC 2024)*. Vol. 13224. SPIE. 2024, pp. 169–175 (cit. on p. 124).

[Zhe+16] Chao Zheng, Shaorong Wang, Yihan Zhang, Pengxiang Zhang, and Yong Zhao. “A robust and automatic recognition system of analog instruments in power system by using computer vision”. In: *Measurement* 92 (2016), pp. 413–420 (cit. on p. 83).

[Zho+24] Hongyu Zhou, Wei Xiang, Qinwei Dong, Weizhi Qi, and Peng Wu. “Research on power grid equipment maintenance based on multi-dimensional inspection image processing”. In: *Proceedings of the 2024 International Conference on Power Electronics and Artificial Intelligence*. 2024, pp. 53–57 (cit. on p. 20).

[Zho+19] Xingyu Zhou, Yi Li, Carlos A Barreto, et al. “Evaluating resilience of grid load predictions under stealthy adversarial attacks”. In: *2019 Resilience Week (RWS)*. Vol. 1. IEEE. 2019, pp. 206–212 (cit. on pp. 51, 52).

[Zhu+23] Yanxu Zhu, Hong Wen, Runhui Zhao, et al. “Research on Data Poisoning Attack against Smart Grid Cyber–Physical System Based on Edge Computing”. In: *Sensors* 23.9 (2023), p. 4509 (cit. on pp. 51, 52).