# An LLM-Based Agentic Pipeline for Generator-Level Adversarial Evaluation of Smart Grid IDSs

**Camilla B. Quincozes[1], Guilherme C. Mundt[1], Ian Rankin[2], Silvio E. Quincozes[1], Paulo S. Severo[1], Daniel Mossé[2]**

[1]Universidade Federal do Pampa (UNIPAMPA) – Alegrete – RS – Brazil

[2]University of Pittsburgh

{camillaborchhardt,guilhermemundt}.aluno@unipampa.edu.br

***Abstract.*** *Intrusion Detection Systems (IDSs) for Smart Grids are often evaluated using fixed datasets, which may hide weaknesses against variations of known attacks. This paper proposes an agentic pipeline for parametric adversarial evaluation of IDSs in IEC-61850 environments. The pipeline integrates the ERENO synthetic dataset generator, a Random Forest-based IDS, and Large Language Model (LLM)-based agents that iteratively modify parameters of Masquerade attack scenarios. Instead of perturbing samples directly, the approach operates at the dataset-generation level by changing attack configuration files. We analyze both traditional metrics (e.g., F1-score and recall) and execution indicators (e.g., skipped iterations, failed executions, and degenerate-variant flags). Our results indicate that the proposed LLM-guided agentic architecture can generate attack variants that decrease the intrusion detection performance (e.g., by increasing false negatives).*

## 1. Introduction

The increasing digitalization of Smart Grids has broadened the use of standardized communication systems for monitoring, automation, and protection of critical infrastructures. In this context, the IEC-61850 standard has become a reference for digital substations by defining data models and communication protocols between Intelligent Electronic Devices (IEDs). Among these protocols, GOOSE (Generic Object-Oriented Substation Event) plays a central role in protection and automation operations, as it enables the transmission of critical messages with low latency in industrial networks [Ustun et al. 2021, Sidhu and Yin 2007].

Despite their operational relevance, the increasing connectivity of these environments also expands the attack surface of digital substations. Attacks against GOOSE messages can compromise system state perception, affect protection decisions, and introduce anomalous behavior in industrial devices. For this reason, Intrusion Detection Systems (IDSs) based on machine learning (ML) have been extensively investigated as mechanisms to support the identification of malicious activities in IEC-61850 and Smart Grid environments [Ustun et al. 2021, de Oliveira et al. 2025, Sahani et al. 2023].

However, the evaluation of IDSs in Smart Grids still faces significant challenges. In many studies [Asimopoulos et al. 2023, Teryak et al. 2023, Mutua et al. 2025], models are trained and evaluated on fixed datasets, in which attack patterns and legitimate traffic remain relatively stable. Although this type of evaluation is useful for measuring initial

performance, it may offer a limited view of IDS robustness against variations of known attacks. A classifier can achieve high performance in a baseline scenario and still be sensitive to small changes in the parameters that define the generation of malicious traffic [Qiao et al. 2026, Mustafa et al. 2025].

The limited availability of real-world data in critical infrastructures reinforces the importance of synthetic dataset generation frameworks. In this context, the ERENO framework enables the generation of realistic traffic for IEC-61850 environments, including legitimate samples and different types of attacks. This makes it possible to investigate not only the classification capability of an IDS in a fixed scenario, but also its behavior under parameterized attack variants generated in a controlled manner.

In parallel, modern *agentic* architectures—characterized by LLM-driven reasoning, tool use, memory, planning capabilities, and autonomous orchestration—have been increasingly explored in automation, planning, and decision-making tasks. In cybersecurity, LLM-based agents have already been explored in defensive applications [Hu et al. 2024], while traditional agent-based models have been investigated in both defensive [Jaffal et al. 2025] and adversarial evaluation [Mutua et al. 2025, Acharya et al. 2026] processes. However, the use of modern agentic architectures to autonomously adjust synthetic industrial traffic generators and iteratively assess IDS robustness in Smart Grids remains underexplored [Zhai et al. 2025, Piccialli et al. 2025].

In this work, we propose an agentic pipeline for evaluating parametric adversarial attacks in IEC 61850 environments. The pipeline integrates ERENO, a Random Forest-based IDS, and an LLM-driven agent that modifies generator-level parameters to create variants of a baseline *Masquerade* attack. Rather than perturbing existing samples or executing attacks on real systems, the method operates on synthetic scenario configurations to assess whether parameterized attack variants can reduce the detection capability of an IDS trained on the baseline scenario.

Our main contribution is a controlled and traceable generator-level evaluation workflow[1] that links LLM-guided configuration changes, synthetic dataset generation, IDS evaluation, and execution indicators such as valid changes, skipped iterations, failed runs, and degenerate variants. Based on this workflow, the study examines whether LLM-based agents can produce applicable *Masquerade* configuration changes in ERENO (RQ1), whether these variants degrade the performance of a baseline-trained Random Forest IDS by inducing evasion effects (RQ2), whether such degradation is mainly associated with false negatives (RQ3), and how different agents compare in output validity, execution behavior, and evasion-oriented adversarial effectiveness (RQ4).

The remainder of this paper is organized as follows. Section 2 presents the background. Section 3 discusses related work. Section 4 describes the methodology adopted for the proposed pipeline. Section 5 presents the results and discussion. Finally, Section 6 concludes the paper and outlines future work.

[1]Source code and replication package: `https://github.com/camillabdt/adversarialagnoagent.git`

## 2. Background

This section briefly reviews the concepts directly related to the proposed generator-level adversarial evaluation pipeline.

### 2.1. IEC-61850 Standard and GOOSE Protocol

The IEC-61850 standard defines communication models and protocol mappings for substation automation systems, enabling information exchange among IEDs [Aftab et al. 2020]. Among its protocols, GOOSE plays a central role in protection and automation because it supports low-latency multicast communication over Ethernet [Ustun et al. 2021, Sidhu and Yin 2007].

This performance-oriented design also introduces cybersecurity challenges. GOOSE prioritizes deterministic behavior and low latency, typically between 2 and 4 ms [Brunner 2008], while providing limited native security mechanisms. As a result, attacks that manipulate GOOSE messages may affect system state perception and protection decisions in digital substations. These characteristics motivate the use of IDSs to detect malicious behavior in IEC-61850 environments [Quincozes et al. 2023].

GOOSE operates on a publish/subscribe model directly at the Ethernet data link layer, bypassing the network and transport layers entirely [Ustun et al. 2021]. When a monitored value changes state — such as a circuit breaker status — the publishing IED immediately transmits a GOOSE frame and retransmits it at exponentially increasing intervals until a steady-state heartbeat period is reached. Each frame carries two sequence counters: *stNum*, which is incremented upon a new state change event, and *sqNum*, which is incremented on each retransmission of the same event. Subscribers use these counters, together with the *timeAllowedToLive* field, to determine message validity and ordering [Sahani et al. 2023, Sidhu and Yin 2007].

The absence of native authentication, encryption, or integrity verification mechanisms in GOOSE [Sahani et al. 2023] exposes substations to a range of cyberattacks. An adversary with access to the process bus can inject fabricated frames, replay legitimate historical messages to cause inappropriate switching events, or impersonate a legitimate IED by spoofing its source MAC address and GOOSE identifiers — a scenario known as a *Masquerade* attack [Quincozes et al. 2023]. In a *Masquerade* fault attack, the adversary broadcasts manipulated GOOSE messages with forged *stNum* values and analog payload variations that simulate fault conditions, potentially causing circuit breakers to operate incorrectly and affecting the physical state of the substation. This attack scenario is the central threat modeled in this work.

### 2.2. Machine Learning-based IDS

IDSs are mechanisms responsible for monitoring activities in networks and computer systems to identify malicious or anomalous behavior [Liao et al. 2013]. In industrial environments and Smart Grids, Machine Learning-based IDSs (ML-IDS) have been widely investigated due to their ability to identify complex attack patterns without relying exclusively on previously known signatures [de Oliveira et al. 2025].

Among the algorithms employed in this context, Random Forest stands out due to its robustness, generalization capability, and strong performance in supervised classification problems involving tabular data [Resende and Drummond 2018]. The algorithm is

based on the construction of multiple independent decision trees, combining their outputs through majority voting. This approach mitigates overfitting problems and improves the model's generalization capability.

In Smart Grids and Industrial Control Systems, ML-based IDSs provide significant advantages compared to traditional signature-based mechanisms. Among these advantages are the ability to detect previously unknown attacks, adaptability to complex traffic patterns, and greater flexibility in identifying behavioral anomalies [Sahani et al. 2023]. Furthermore, supervised models have demonstrated high effectiveness in detecting attacks targeting industrial protocols, especially in scenarios involving IEC-61850 messages and GOOSE traffic [Ustun et al. 2021]. However, the effectiveness of these systems depends directly on the quality, diversity, and representativeness of the datasets used during training, making the generation of realistic data a critical factor for evaluating IDS robustness.

### 2.3. Adversarial Machine Learning

Adversarial Machine Learning (AML) refers to the study of attacks and defensive mechanisms aimed at manipulating machine learning models [Qiao et al. 2026]. In cybersecurity scenarios, adversarial techniques can be employed to reduce the detection capability of supervised classifiers through strategic modifications to training or inference data.

In industrial environments and Smart Grids, recent studies have demonstrated that small modifications to traffic attributes can cause significant degradation in the performance of IDSs [Mustafa et al. 2025]. Nevertheless, most research remains focused on traditional computer network datasets, with still limited investigations involving IEC-61850 industrial protocols and the synthetic generation of adversarial datasets.

### 2.4. Agentic AI

Agentic AI refers to computational architectures in which autonomous or semi-autonomous agents pursue goals through iterative cycles of perception, reasoning, planning, action, and feedback. Instead of operating as isolated predictive models, these systems combine task context, memory, tool use, execution monitoring, and adaptive decision-making to perform complex workflows with limited human intervention [Zhai et al. 2025, Piccialli et al. 2025]. In modern agentic architectures, LLMs may be used as reasoning or interaction components, but the defining aspect is the closed-loop organization of actions, tools and feedback around goal-directed behavior [Alenezi 2026].

In cybersecurity, Agentic AI has been investigated in applications involving offensive automation, vulnerability analysis, support for *penetration testing*, rule generation, and intelligent monitoring [Jaffal et al. 2025]. These applications exploit the ability of agents to interpret contextual information, select actions, invoke external tools, and adapt their strategies according to execution outcomes. This makes agentic architectures particularly relevant for cybersecurity tasks that require iterative exploration, structured decision-making, and continuous evaluation of system responses.

Despite these advances, few studies have investigated Agentic AI in iterative adversarial evaluation workflows for IDSs in industrial environments. In particular, the use of agents to modify attack-generator configurations, produce new synthetic datasets, and assess evasion effects against baseline-trained IDSs remains underexplored. In this

context, this work employs an agentic workflow to modify IEC-61850 attack-generation parameters, observe IDS performance and execution indicators, and iteratively assess how synthetic attack variants affect detection capability.

## 3. Related Work

Works combining Agentic AI, adversarial robustness evaluation, and Smart Grid IDSs are still scarce. Therefore, this section discusses two groups of related studies: (i) works that use LLMs or agent-based approaches in cybersecurity evaluation, and (ii) works that investigate adversarial attacks or robustness of IDSs in Smart Grid and industrial control environments. Table 1 summarizes the related works.

The work by [Hu et al. 2024] proposes an LLM-based agent for the automatic generation and generalization of IDS rules. The methodology integrates the Retrieval-Augmented Generation (RAG) architecture and Chain of Thought (CoT) reasoning, processing data extracted from security reports, malicious payloads, and public threat rules to feed the model. The results demonstrated high efficacy: the new rules achieved 100% precision and an accuracy exceeding 99.6% in attack detection, while generalization raised the accuracy of public defenses from 41.75% to 85.79% against variant attacks, without increasing the false positive rate. As limitations, the study restricts its evaluation to web/common traffic datasets, failing to address industrial Smart Grid protocols, in addition to relying on a robust preliminary extraction of textual features to function properly.

Recently, [Acharya et al. 2026] proposed the ADVIS-G framework, an IDS robust against adversarial attacks in Smart Grids using deep learning. The objective of the study was to investigate defense mechanisms capable of enhancing the resilience of detection models against adversarial attacks. As a contribution, the authors combined adversarial training and autoencoder models to reconstruct compromised samples, thereby increasing the robustness of classifiers applied to Smart Grid communications. The results demonstrated significant improvements in detection metrics, even under FGSM, BIM, and MIM attacks. Nevertheless, the work focuses on defensive strategies and does not investigate the use of generative models as adversarial agents responsible for the iterative modification of synthetic attack configurations.

The research by [Mutua et al. 2025] aims to model high-fidelity adversarial attacks against smart grid IDSs, actively addressing realistic constraints and the true capabilities of an attacker. Operating under a gray-box threat model from an insider perspective, the methodology utilized the FGSM and JSMA algorithms to create adversarial evasions validated against Random Forest, XGBoost, and Naive Bayes techniques using a power system dataset. The results proved a high evasion rate and attested to the transferability of the attacks, causing abrupt performance drops in the defenses. The study, however, focuses on crafting adversarial perturbations directly over samples, rather than modifying the configuration of a synthetic traffic generator.

The work conducted in [Teryak et al. 2023] focuses on developing a machine learning-based IDS for IEC 60870-5-104 networks that is simultaneously capable of withstanding adversarial attacks. Employing a Hierarchical Multilayer Perceptron (HMLP) architecture, the study trained the model with realistic traffic and assessed vulnerabilities by applying FGSM, PGD, and C&W attacks, followed by the use of defensive distillation to mitigate the issue. However, the study is restricted to the IEC 60870-5-104 protocol

and evaluates sample-level adversarial examples rather than generator-level variants.

In [Asimopoulos et al. 2023], the authors investigate the use of artificial intelligence for both defense and attack in smart grid networks, focusing on the IEC 60870-5-104 industrial protocol. The methodology consisted of developing an IDS using classical machine learning algorithms and testing its resilience against adversarial data samples generated by FGSM evasion methods and CTGAN networks. A limitation of the study is that the effectiveness of the adversarial perturbations depends on the adjustment of the noise rate, indicating that the attack models may require specific tuning depending on the expected level of robustness in the network.

**Table 1. Comparison among related works.**

| Reference | Smart Grid/ICS | IDS | AML | LLM | Agentic AI | Generator-level variants |
|---|---|---|---|---|---|---|
| [Hu et al. 2024] | ✗ | ✓ | ✗ | ✓ | ✓ | ✗ |
| [Asimopoulos et al. 2023] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| [Teryak et al. 2023] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| [Mutua et al. 2025] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| [Acharya et al. 2026] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| **This work** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Overall, the literature reveals a dichotomy in Smart Grid security: existing studies employing advanced AI, such as LLMs and agents, focus strictly on defensive mechanisms and rule generation. Conversely, studies investigating adversarial attacks still rely on traditional mathematical perturbations (e.g., FGSM, CTGAN) applied directly to data samples. In contrast, this work bridges this gap by utilizing an agentic AI pipeline from an offensive perspective. Instead of applying blind perturbations to existing samples, our LLM-based agent intelligently modifies the parameters of a synthetic IEC-61850 traffic generator. This approach produces new, semantically cohesive attack variants at the dataset-generation level, evaluating their evasion impact on a fixed baseline-trained IDS.

## 4. Methodology

This section presents the methodology adopted to build and evaluate the proposed agentic pipeline for adversarial IDS evaluation in IEC-61850 environments. The workflow integrates synthetic dataset generation with ERENO, a Random Forest-based IDS, and LLM-based agents that iteratively suggest modifications to the parameters of a *Masquerade* attack. Figure 1 summarizes the overall process.

### 4.1. Proposed Pipeline

The proposed pipeline is organized into two main phases. In the first phase, a baseline *Masquerade* attack configuration is used to generate a synthetic dataset with ERENO. This dataset is then used to train and internally evaluate a Random Forest classifier, which acts as the IDS in the experiments.

In the second phase, an iterative adversarial evaluation loop is executed. At each iteration, the current attack configuration, the most recent IDS performance metrics, and a summary of previous iterations are provided to an LLM-based agent. The agent suggests parameter modifications to generate alternative *Masquerade* variants and assess whether

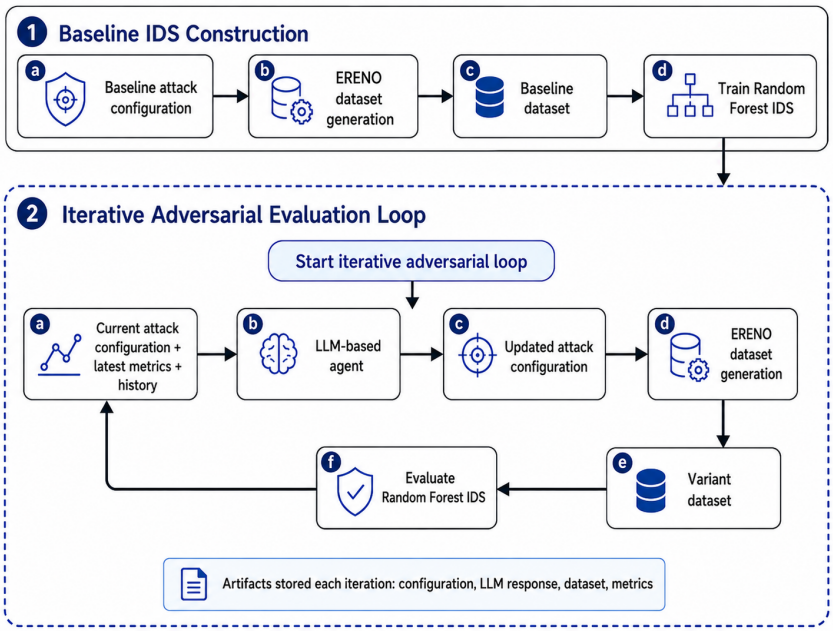


**Figure 1. Overview of the proposed pipeline for baseline IDS construction and iterative adversarial evaluation.**

the IDS trained on the baseline remains robust under configuration changes. When applicable changes are extracted from the LLM response, a new attack configuration is produced and passed to ERENO, which generates a new synthetic dataset. This dataset is then evaluated using the same Random Forest model trained on the baseline dataset. No experiments interact with real substations or execute actual cyberattacks; all runs use synthetic datasets in a controlled environment.

### 4.2. Baseline Scenario and Dataset Generation

The experiments start from a baseline *Masquerade fault* attack configuration represented as a `JSON` file. This file defines the parameters used by ERENO to generate malicious GOOSE traffic, including fault probability, fault duration, circuit breaker status, `sqnum` behavior, analog variation range, and trap-area settings. The baseline configuration is summarized in Table 2. The complete configuration file is stored together with the experimental artifacts.

The datasets used in this work are generated with ERENO, which receives a configuration file describing the attack scenario and produces a labeled CSV dataset containing legitimate and malicious samples. For the baseline scenario, the baseline `JSON` file is copied to the ERENO attack configuration directory and used to generate the initial dataset. In the subsequent adversarial iterations, each updated `JSON` configuration is copied to the same location before ERENO is executed again.

Each generated dataset contains domain-related (electric measures), protocol, temporal, and derived traffic attributes, as well as the class label. The class `normal` represents legitimate traffic, whereas `masquerade_fake_fault` represents malicious traffic associated with the *Masquerade* attack. Since ERENO generates attack samples according to the configured scenario parameters, the number of malicious instances may

**Table 2. Baseline *Masquerade fault* configuration used in the experiments.**

| Parameter | Value | Role |
|---|---|---|
| `attackType` | `masquerade_fault` | Attack scenario generated by ERENO |
| `fault.prob` | 0.6 | Fault occurrence probability |
| `fault.durationMs` | 50–800 ms | Fault duration interval |
| `cbStatus` | 1 | Circuit breaker status |
| `incrementStNumOnFault` | `true` | State number behavior during faults |
| `sqnumMode` | `fast` | Sequence number behavior |
| `ttlMsValues` | [20, 40, 80] | Time-to-live values in GOOSE messages |
| `analog.deltaAbs` | 0.2–0.8 | Analog signal variation interval |
| `trapArea.multiplier` | 1.2–2.5 | Trap-area multiplier interval |
| `trapArea.spikeProb` | 0.5 | Spike probability |

vary across datasets. For this reason, class distributions are recorded for each generated dataset and considered during the analysis.

### 4.3. IDS Training and Evaluation

The IDS used in the experiments is implemented with the Random Forest algorithm. The classifier is trained only once, using the baseline dataset generated from the initial *Masquerade* configuration. The baseline dataset is split into training and testing subsets using a 70/30 ratio, and the internal test provides the reference performance of the IDS before the adversarial iterations begin. After training, the model remains fixed throughout the adversarial loop, so all variant datasets are evaluated against the same classifier.

Before training, the class column is separated from the feature set. Constant columns, timestamp-related fields, direct sequence indicators, and attributes that could trivially encode the dataset-generation process are removed. This step reduces the risk that the classifier learns artifacts of the generator rather than attack behavior. The remaining attributes are used to train the Random Forest model.

The evaluation considers accuracy, precision, recall, and F1-score for the *Masquerade* class. In addition, true positives (**TP**), false positives (**FP**), false negatives (**FN**), and true negatives (**TN**) are recorded. Particular attention is given to recall and false negatives, since false negatives represent malicious samples misclassified as normal traffic.

### 4.4. Agentic Iterative Modification

To explore new attack variants, the LLM-based agent receives the current state and suggests modifications to the attack configuration. At each iteration, the input to the agent includes a fixed base prompt, the current attack configuration in `JSON` format, the most recent IDS performance metrics, and the history of previous iterations. The purpose of the changes is to produce different attack variants for robustness assessment, with particular attention to possible reductions in recall and F1-score for the malicious class.

The base prompt instructs the agent to act as an adversary targeting a digital substation in a controlled academic context, with the explicit goal of suggesting parameter changes that reduce the separability of the *Masquerade* attack from normal traffic — particularly with respect to F1-score. At each iteration, the agent receives the current attack configuration in JSON format, the latest IDS performance metrics, a summarized history

of previous iterations, and the list of editable fields. The agent is also guided by domain hints indicating which features most influence the Random Forest decisions (e.g., cbStatus), and is explicitly instructed to avoid degenerate variants — such as zeroing fault probability or analog deltas — that would render the attack uncharacteristic. The output format is enforced via a dedicated section in which the model must list only field-value pairs in plain text, enabling deterministic parsing by the pipeline.

All agents were instantiated with a temperature of 0.2, using the remaining generation parameters at their default values as defined by the Agno[2] framework. The low temperature value was chosen to encourage structured and consistent outputs, reducing the risk of malformed responses that could not be parsed by the pipeline. A higher temperature setting could potentially increase output diversity and produce more varied attack variants, but at the cost of greater instability in the output format — a trade-off that is left for future investigation.

The LLM output is saved at each iteration and processed by a parser to extract applicable changes. When valid changes are identified, they are applied to the current `JSON` configuration, producing an updated attack configuration for the next dataset generation. When no applicable changes are extracted, or when the extracted changes do not modify the previous configuration, the iteration is recorded as non-applicable and no new dataset is generated for that round.

Each iteration stores the LLM response, the resulting attack configuration, the generated dataset, and the evaluation metrics. The performance log includes the iteration number, evaluation type, dataset path, accuracy, precision, recall, F1-score, support of the malicious class, confusion-matrix values, class label information, and control indicators. The `config_changed` indicator records whether the LLM response effectively changed the attack configuration. The `degenerate_variant` indicator records whether a generated variant was flagged as potentially degenerate according to the execution log.

The automated experiments considered the following LLM models: `groq/compound-mini`, `groq/compound`, `openai/gpt-oss-20b`, `openai/gpt-oss-120b`, `qwen/qwen3-32b`, `llama-3.1-8b-instant`, and `llama-3.3-70b-versatile`.[3] Each model was executed for up to 30 iterations under the same baseline configuration and evaluation procedure.

The selected models cover open-weight LLMs accessible through the Groq and OpenAI-compatible APIs at the time of the experiments, spanning different parameter scales (from 8B to 120B) and provider families (Groq Compound, OpenAI GPT-OSS, Qwen, and Llama). This selection was intended to assess whether agent behavior within the proposed pipeline is consistent across models of different sizes and origins, rather than to benchmark any individual model.

All models were evaluated under the same pipeline, prompt structure, parsing rules, and baseline configuration. Therefore, differences among models should be interpreted as differences in their behavior within this specific agentic workflow, rather than as a general ranking of model capability. In particular, some models may produce syntactically plausible responses that are not directly mapped to valid generator parameters by the

[2]Agno Documentation is available in `https://docs.agno.com`

[3]Model identifiers are reported as returned by the respective provider APIs at the time of the experiments.

parser, or may repeatedly suggest changes that do not alter the current configuration. This distinction is important because the pipeline evaluates not only adversarial effectiveness, but also the practical ability of each agent to produce applicable configuration updates in an automated loop.

## 5. Results and Discussion

This section presents the results obtained from the automated pipeline. The analysis is organized into four parts: (i) baseline IDS performance, (ii) validity of agent-generated configurations, (iii) impact of generated variants on IDS performance, and (iv) discussion of the research questions.

### 5.1. Baseline IDS Performance

The first step of the automated evaluation was to assess the Random Forest IDS on the baseline *Masquerade fault* dataset generated by ERENO. In all automated runs, the baseline IDS achieved perfect performance in the internal test, with precision, recall, and F1-score equal to 1.0 for the *Masquerade* class.

This perfect baseline performance should not be interpreted as evidence of general IDS robustness. Instead, it indicates that the original *Masquerade fault* configuration is highly distinguishable from legitimate traffic under the selected feature set. This result reinforces the need to evaluate the IDS beyond a fixed baseline dataset, using parameterized attack variants generated from modified configurations.

### 5.2. Validity of Agent-Generated Configurations

Before analyzing the impact on IDS performance, we evaluated whether each LLM-based agent was able to generate applicable configuration changes. This step is important because not every LLM response resulted in a valid attack variant. Some iterations produced no applicable changes, repeated the previous configuration, or generated outputs that could not be mapped to existing fields in the attack configuration.

Table 3 summarizes the number of valid changes, skipped iterations, failed iterations, and degenerate variants for each automated run. Iterations marked as skipped were not considered adversarial variants, since no new dataset was generated from them.

**Table 3. Validity of LLM-generated configurations in the automated experiments.**

| Model | Iter. | Valid | Skipped | Failed | Degen. | Status |
|---|---|---|---|---|---|---|
| `groq/compound-mini` | 30 | 15 | 15 | 0 | 0 | Completed |
| `openai/gpt-oss-20b` | 30 | 30 | 0 | 0 | 0 | Completed |
| `groq/compound` | 30 | 28 | 2 | 0 | 0 | Completed |
| `qwen/qwen3-32b`† | 30 | 26 | 2 | 2 | 0 | Partial |
| `llama-3.1-8b-instant` | 30 | 30 | 0 | 0 | 0 | Completed |
| `llama-3.3-70b-versatile` | 30 | 30 | 0 | 0 | 0 | Completed |
| `openai/gpt-oss-120b` | 30 | 1 | 29 | 0 | 0 | Completed |

† Partial execution: two iterations failed and did not produce evaluable variants.

The results show that the effectiveness of the pipeline depends not only on the IDS behavior, but also on the ability of the LLM to produce structured and applicable

outputs. For instance, `openai/gpt-oss-120b` produced only one valid configuration change in 30 iterations, while `openai/gpt-oss-20b`, `llama-3.1-8b-instant`, and `llama-3.3-70b-versatile` produced valid changes in all iterations. Producing valid changes did not, however, necessarily imply stronger adversarial impact, as shown below.

The behavior of `openai/gpt-oss-120b` illustrates this limitation: most responses did not map to applicable updates under the parser rules. Since the same prompt and parser were used for all models, this is reported as an execution characteristic of the pipeline rather than corrected manually.

The `qwen/qwen3-32b` run is reported as partial because two iterations failed and did not produce evaluable variants. Its results are therefore considered indicative rather than definitive.

### 5.3. Impact on IDS Detection Performance

After filtering iterations without configuration changes, we analyzed the impact of the generated variants on IDS performance. For each model, only iterations in which a valid configuration change was applied and a new dataset was effectively generated are considered in this analysis. Table 4 reports the strongest adversarial effect observed for each automated run, considering the lowest F1-score obtained for the *Masquerade* class. Since the goal of the adversarial loop is to expose detection blind spots, the worst-case variant per model — rather than the average across iterations — is the most informative indicator of pipeline effectiveness.

**Table 4. Strongest adversarial effect observed for each model in the automated experiments.**

| Model | Iter. | F1-score | Recall | FN | Attack samples |
|---|---|---|---|---|---|
| `groq/compound-mini` | 8 | 0.9176 | 0.8477 | 4582 | 30084 |
| `openai/gpt-oss-20b` | 23 | 0.9583 | 0.9200 | 2010 | 25128 |
| `groq/compound` | 20 | 0.9765 | 0.9541 | 912 | 19882 |
| `qwen/qwen3-32b`† | 23 | 0.9744 | 0.9501 | 992 | 19886 |
| `llama-3.1-8b-instant` | 24 | 0.9936 | 0.9873 | 322 | 25275 |
| `llama-3.3-70b-versatile` | 27 | 0.9959 | 0.9918 | 144 | 17621 |
| `openai/gpt-oss-120b` | 26 | 0.9999 | 0.9999 | 2 | 35231 |

† Partial execution; results are indicative only.

The strongest degradation in the automated experiments was observed with `groq/compound-mini`, which reduced the F1-score of the *Masquerade* class to 0.9176 and the recall to 0.8477, resulting in 4,582 false negatives. This indicates that a substantial number of malicious samples were classified as normal traffic.

The second strongest effect was obtained with `openai/gpt-oss-20b`, which reduced the F1-score to 0.9583 and the recall to 0.9200, producing 2,010 false negatives. The `groq/compound` and `qwen/qwen3-32b` runs also produced moderate degradation, while the Llama-based runs remained close to perfect performance. The `openai/gpt-oss-120b` run produced only one valid configuration change and almost no degradation, which suggests that following the expected output structure is an important factor in the proposed pipeline.

Figure 2 shows the F1-score evolution across adversarial attack variants for all LLM-based agents. Only variants effectively generated and evaluated are plotted; skipped iterations are omitted. The `qwen/qwen3-32b` curve is shown as a dashed line to indicate partial execution. The strongest degradation appears in specific variants rather than monotonically across the loop, especially for `groq/compound-mini` at v8 and `openai/gpt-oss-20b` at v23.

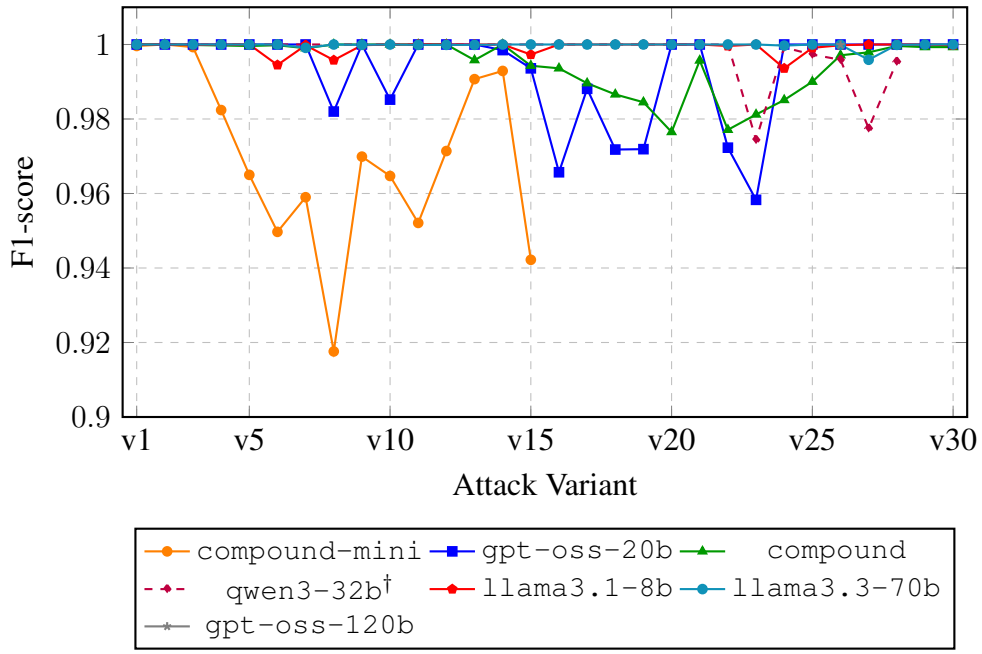


**Figure 2. F1-score evolution across evaluated adversarial variants. Lower values indicate stronger degradation of the baseline-trained IDS; dashed line indicates the partial `qwen3-32b` run.**

We do not perform statistical significance tests because iterations are sequentially dependent within each agentic run. Each iteration may depend on the previous attack configuration, the latest IDS metrics, and the accumulated history provided to the LLM-based agent. Therefore, treating iterations as independent samples for hypothesis testing could lead to misleading conclusions. Instead, we report per-model worst-case degradation, false negatives, execution validity indicators, and the F1-score evolution across the evaluated attack variants.

### 5.4. Qualitative Analysis of False Negatives

Overall, the generated variants affected the IDS mainly through reductions in recall rather than through increases in false positives. In the strongest runs, precision remained equal to 1.0, meaning that samples classified as attacks were correctly identified. However, the recall reduction shows that the IDS failed to detect part of the malicious traffic generated from modified attack configurations. This behavior is particularly relevant in intrusion detection, since false negatives correspond to attacks that bypass the detector. These correspond to false-negative rates of 15.2% and 8.0% for groq/compound-mini and openai/gpt-oss-20b, respectively, in their strongest variants.

The qualitative interpretation of these results is that some generated configurations produced attack datasets that were less distinguishable for the baseline-trained classifier.

However, the effect was not uniform across all agents. Some LLM-based agents produced many valid configuration changes with limited impact on detection performance, whereas others produced fewer or more specific variants that resulted in stronger recall degradation. This reinforces that output validity alone is not sufficient: generated changes must also affect features that are relevant to the classifier decision boundaries.

### 5.5. Discussion

The results indicate that the proposed pipeline can generate and evaluate parameterized variants of the *Masquerade* attack in an automated loop. However, the effectiveness of the approach depends on both the quality of the LLM output and whether the generated configuration changes affect the features used by the Random Forest classifier.

The most relevant degradation in the automated experiments was associated with increased false negatives, as detailed in the qualitative analysis above. This finding indicates that the proposed pipeline is useful for exposing detection blind spots that may not appear when the IDS is evaluated only on a fixed baseline dataset.

At the same time, several generated variants did not reduce IDS performance. This suggests that the selected feature set, especially physical and derived traffic attributes, provides strong separability between legitimate and malicious samples in many configurations. Therefore, the proposed pipeline should not be interpreted as guaranteeing IDS degradation, but rather as a controlled method for probing IDS robustness under parameterized attack variations.

The experiments also highlight limitations of LLM-based agents in this setting. Some models generate repeated, invalid, or non-applicable outputs, while others generate valid changes that do not substantially affect the IDS. For this reason, parsing, validation, and explicit tracking of skipped, failed, and potentially degenerate iterations are necessary components of the pipeline.

Taken together, the results provide a positive answer to RQ1, but with important differences across agents. Most LLM-based agents were able to generate applicable configuration changes for the *Masquerade* attack scenario. However, this ability was not uniform: while `openai/gpt-oss-20b`, `llama-3.1-8b-instant`, and `llama-3.3-70b-versatile` produced valid changes in all iterations, `openai/gpt-oss-120b` produced only one. This indicates that applicability in the proposed pipeline depends not only on the model, but also on its ability to follow the expected output structure and produce changes that can be parsed and applied automatically.

Regarding RQ2 and RQ3, some generated variants reduced the performance of the baseline-trained IDS mainly by increasing false negatives (Tables 3 and 4), with the strongest effect for groq/compound-mini. These findings suggest that the proposed pipeline can reveal detection blind spots that are not visible when the IDS is evaluated only on the original baseline dataset.

Finally, RQ4 highlights that the agents differed not only in adversarial impact, but also in execution behavior. Some models produced many valid configuration changes with limited effect on IDS performance, while others produced fewer or more specific variants that led to stronger recall degradation. Therefore, the results should not be interpreted as a general ranking of LLM capability, but as evidence that different agents

behave differently within this generator-level evaluation workflow. In this context, output validity, skipped iterations, execution failures, and false-negative impact are all relevant dimensions for assessing the usefulness of agentic AI pipelines in adversarial IDS evaluation. We stress that these results show that vulnerable generator configurations exist and that an LLM-driven loop can find them, but not that LLM-guided search is more effective or efficient than simpler parameter-search strategies; establishing that requires the non-LLM baselines discussed in Section 6.

Several factors limit the generality of these results. First, the evaluation relies on a single IDS family, a single generator and a single dominant attack scenario; other classifiers, datasets, or attack types could behave differently. Second, the baseline IDS is almost perfectly separable before adversarial search, so the observed degradation may not transfer to less saturated detectors. Third, the agent operates with a one-step history and receives domain hints about influential features, which may introduce feature-importance leakage in the agent's favor and should be ablated in future work. Fourth, results are reported as per-model worst cases without repeated runs or confidence intervals, since iterations are sequentially dependent within each agentic run; consequently, they establish that vulnerable configurations exist rather than characterizing expected behavior. Finally, we do not define a threat model specifying whether a real attacker could control generator-level parameters, so the study probes IDS robustness under parameterized variation rather than modeling a deployed adversary.

## 6. Conclusion

This paper presented an agentic pipeline for parametric adversarial evaluation of IDSs in IEC-61850 environments. The proposed workflow integrates ERENO, a Random Forest-based IDS, and LLM-based agents that iteratively suggest modifications to the parameters of a *Masquerade fault* attack. Instead of perturbing samples directly, the pipeline operates at the attack-configuration level, generating new synthetic variants through ERENO.

The baseline IDS was perfectly separable on the original scenario, yet the automated loop showed that some LLM-generated variants reduced its performance, mainly through increased false negatives rather than false positives — a critical distinction in IDS evaluation. This effect depended on the LLM used: some models produced applicable changes with impact, while others produced stronger variants or skipped iterations.

Overall, the proposed pipeline provides a controlled and traceable way to probe IDS robustness beyond fixed datasets. Beyond the experimental results, the pipeline itself represents a reusable methodology for generator-level IDS evaluation, potentially applicable to other synthetic traffic frameworks, attack types, and IDS models. Future work includes evaluating these extensions, improving validation of generated configurations, incorporating explainability mechanisms, comparing LLM-guided exploration with non-LLM parameter search baselines, measuring token consumption and wall-clock time per agent iteration, and investigating the effect of generation parameters — such as temperature — on output diversity, validity, and adversarial effectiveness.

## References

Acharya, R., Al Sardy, L., Muhammad, M., and German, R. (2026). Advis-g: An adversarially defended intrusion detection system for smart grids using deep learning.

*KI-Künstliche Intelligenz*, pages 1–23.

Aftab, M. A., Hussain, S. S., Ali, I., and Ustun, T. S. (2020). Iec 61850 based substation automation system: A survey. *International Journal of Electrical Power & Energy Systems*, 120:106008.

Alenezi, M. (2026). From prompt-response to goal-directed systems: The evolution of agentic ai software architecture. *arXiv preprint arXiv:2602.10479*.

Asimopoulos, D. C., Radoglou-Grammatikis, P., Makris, I., Mladenov, V., Psannis, K. E., Goudos, S., and Sarigiannidis, P. (2023). Breaching the defense: Investigating fgsm and ctgan adversarial attacks on iec 60870-5-104 ai-enabled intrusion detection systems. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ARES '23, New York, NY, USA. Association for Computing Machinery.

Brunner, C. (2008). Iec 61850 for power system communication. In *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, pages 1–6. IEEE.

de Oliveira, J. A., dos Santos, A. F. P., and Salles, R. M. (2025). Rnn for intrusion detection in digital substations based on the iec 61850. *Journal of Information Security and Applications*, 94:104197.

Hu, X., Chen, H., Bao, H., Wang, W., Liu, F., Zhou, G., and Yin, P. (2024). A llm-based agent for the automatic generation and generalization of ids rules. In *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1875–1880.

Jaffal, N. O., Alkhanafseh, M., and Mohaisen, D. (2025). Large language models in cybersecurity: A survey of applications, vulnerabilities, and defense techniques. *AI*, 6(9):216.

Liao, H.-J., Richard Lin, C.-H., Lin, Y.-C., and Tung, K.-Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1):16–24.

Mustafa, A., Khan, M. T., Umer, M. A., Masood, Z., and Ahmed, C. M. (2025). Adversarial sample generation for anomaly detection in industrial control systems. In *Proceedings of the 1st Workshop on Modeling and Verification for Secure and Performant Cyber-Physical Systems*, pages 1–7.

Mutua, N. M., Nadjm-Tehrani, S., and Matoušek, P. (2025). Penetrating the power grid: Realistic adversarial attacks on smart grid intrusion detection systems. In Oliva, G., Panzieri, S., Hämmerli, B., Pascucci, F., and Faramondi, L., editors, *Critical Information Infrastructures Security*, pages 249–268, Cham. Springer Nature Switzerland.

Piccialli, F., Chiaro, D., Sarwar, S., Cerciello, D., Qi, P., and Mele, V. (2025). Agentai: A comprehensive survey on autonomous agents in distributed ai for industry 4.0. *Expert Systems with Applications*, 291:128404.

Qiao, Y., Sathyanarayana, N. B., Shi, C., He, Z., Wang, T., and Hou, T. (2026). A survey on adversarial machine learning: Attacks, defenses, real-world applications, and future research directions. *Neurocomputing*, page 132670.

Quincozes, S. E., Albuquerque, C., Passos, D., and Mossé, D. (2023). Ereno: A framework for generating realistic iec–61850 intrusion detection datasets for smart grids. *IEEE Transactions on Dependable and Secure Computing*, 21(4):3851–3865.

Resende, P. A. A. and Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys (CSUR)*, 51(3):1–36.

Sahani, N., Zhu, R., Cho, J.-H., and Liu, C.-C. (2023). Machine learning-based intrusion detection for smart grid computing: A survey. *ACM Trans. Cyber-Phys. Syst.*, 7(2).

Sidhu, T. S. and Yin, Y. (2007). Modelling and simulation for performance evaluation of iec61850-based substation communication systems. *IEEE Transactions on Power Delivery*, 22(3):1482–1489.

Teryak, H., Albaseer, A., Abdallah, M., Al-Kuwari, S., and Qaraqe, M. (2023). Double-edged defense: Thwarting cyber attacks and adversarial machine learning in iec 60870-5-104 smart grids. *IEEE Open Journal of the Industrial Electronics Society*, 4:629–642.

Ustun, T. S., Hussain, S. M. S., Ulutas, A., Onen, A., Roomi, M. M., and Mashima, D. (2021). Machine learning-based intrusion detection for achieving cybersecurity in smart grids using iec 61850 goose messages. *Symmetry*, 13(5).

Zhai, W., Liao, J., Chen, Z., Su, B., and Zhao, X. (2025). A survey of task planning with large language models. *Intelligent Computing*, 4:0124.