electronics

MDPI

Article

A Comparative Study of Large Language Models for Industrial Cyber-Physical Security

J. de Curtò [1,2,3,*], I. de Zarzà [3,4], Juan Carlos Cano [5] and Carlos T. Calafate [5]

[1] Department of Computer Applications in Science & Engineering, BARCELONA Supercomputing Center, 08034 Barcelona, Spain
[2] Escuela Técnica Superior de Ingeniería (ICAI), Universidad Pontificia Comillas, 28015 Madrid, Spain
[3] Estudis d'Informàtica, Multimèdia i Telecomunicació, Universitat Oberta de Catalunya, 08018 Barcelona, Spain; dezarza@uoc.edu
[4] Human Centered AI, Data & Software, LUXEMBOURG Institute of Science and Technology, 4362 Esch-sur-Alzette, Luxembourg
[5] Departamento de Informática de Sistemas y Computadores, Universitat Politècnica de València, 46022 València, Spain; jucano@disca.upv.es (J.C.C.); calafate@disca.upv.es (C.T.C.)
* Correspondence: jdecurto@icai.comillas.edu

Abstract

Intrusion detection in industrial cyber-physical systems is constrained by small labelled-attack corpora and by the subtler signal of physical-process attacks compared with classical IT-network intrusions, motivating renewed interest in foundation-model-based detectors; classical detectors are typically trained per dataset and degrade under the distribution shift that is common in operational technology, where attack repertoires evolve faster than retraining cycles. Two foundation-model families are now plausible candidates: open-source Large Language Models (LLMs) and recent tabular foundation models (TabPFN, TabICL) pre-trained for in-context tabular inference. We compare the two families head-to-head, alongside Random Forest and XGBoost classical anchors, across three established industrial security benchmarks (SWaT, HAI, WUSTL-IIoT-2021) under a controlled multi-seed full-holdout protocol with paired McNemar and cross-seed Mann–Whitney tests. The empirical picture is dataset-dependent rather than universal: tabular foundation models establish a strong, previously unreported baseline that is competitive with or superior to classical anchors on every dataset evaluated, while LLMs are complementary detectors with a specific advantage on schemas that carry process-engineering semantics (such as SWaT's named sensor channels). A per-class analysis on the WUSTL five-class attack taxonomy shows that the two families have structurally different strengths: tabular methods dominate traffic-rich attacks (Denial-of-Service, Reconnaissance), whereas LLMs are competitive on rare attack types (Backdoor, Command Injection). A confidence-gated cascade that escalates only low-confidence tabular decisions to an LLM exceeds either detector alone at a small query budget, and a leave-one-attack-type-out analysis shows that foundation-model detectors generalise to unseen attack families substantially better than the classical anchors. The appropriate detector choice in industrial cyber-physical security is therefore informed by the dataset's feature schema, the attack-type mix, and the operational cost envelope, rather than by a specific performance metric.

Keywords: industrial cyber-physical security; tabular foundation models; TabPFN; TabICL; large language models; SCADA; Industrial Internet of Things

Academic Editor: Emre Tokgoz

Received: 2 June 2026
Revised: 21 June 2026
Accepted: 23 June 2026
Published: 24 June 2026

1. Introduction

Industrial cyber-physical systems integrate sensing, computation, and physical actuation across domains as varied as water treatment, electrical generation, manufacturing, and transportation [1]. The same connectivity that enables remote monitoring and supervisory control also expands the attack surface available to adversaries. Recent incidents, from coordinated cyberattacks on water and energy facilities to ransomware campaigns against manufacturing networks, have promoted intrusion detection on operational technology (OT) and the Industrial Internet of Things (IIoT) from a research curiosity to a first-order security primitive. Designing intrusion detection systems (IDS) that generalise across heterogeneous industrial domains, however, remains an open problem.

Two decades of work on machine-learning-based IDS have produced a mature landscape of supervised and semi-supervised approaches, surveyed in [2–6]. Random Forest, gradient-boosted trees, and deep neural network classifiers are competitive on the standard benchmarks [7–9], particularly when paired with class-balancing techniques [10,11] and modern data augmentation [12]. Two limitations persist. First, these models are typically trained per-dataset and exhibit substantial degradation when the deployment distribution shifts, a common scenario in industrial settings where attack repertoires evolve faster than retraining cycles [13]. Second, performance on OT/ICS-specific benchmarks (water plants, hardware-in-the-loop testbeds) is markedly worse than on IT-network benchmarks, reflecting both the smaller scale of labelled OT corpora and the more subtle physical-process signature of OT attacks.

A conceptual distinction is worth stressing before turning to the foundation-model literature. The bulk of machine-learning IDS work surveyed above operates on network-traffic features: packet and byte counts, flow durations, transport-layer flags, port and protocol fields extracted from IT or IIoT network captures. The OT/ICS detection task targets a substantively different signal type. Supervisory control and data acquisition (SCADA)-instrumented plants such as SWaT, and hardware-in-the-loop testbeds such as HAI, expose the classifier to physical-process channels, including level transmitters, flow transmitters, pressure indicators, motorized valves, and actuator states, sampled at field-bus rates. An attack on a SCADA plant is not a packet pattern; it is a discrepancy between the plant's commanded state and its observed physical trajectory, induced by sensor spoofing, false PLC commands, replayed actuator states, or setpoint manipulation. The detection problem therefore reduces to "is this physical-process trajectory consistent with normal plant operation?" rather than "does this flow resemble attack traffic?". WUSTL-IIoT-2021 is the partial exception that bridges the two regimes, since it captures Modbus/TCP flow features at the IIoT control-network layer; it is also the dataset on which binary classification saturates in our experiments, consistent with the maturity of network-flow-based IDS. SWaT and HAI place foundation-model detectors in a different and less-explored operating regime, where the question is whether LLM-pretrained or tabular-pretrained inductive biases transfer to the classification of physical-process telemetry. The empirical answer, detailed in Section 5, is dataset-dependent and not self-evidently predictable from prior IT-network results.

Foundation models, large pre-trained neural networks deployed without task-specific gradient updates, have, over the past two years, emerged as a candidate for closing this gap. Two distinct families have taken shape. The first is built on large language models (LLMs) [14,15], whose ability to consume tabular or textualised cybersecurity data as prompts has motivated a rapid wave of cybersecurity-oriented applications [16–20], including frameworks for LLM-supported static application security testing [21] and conceptual hybrid LLM/classical IDS pipelines for cyber-physical systems [22,23]. The second family is built on tabular foundation models: neural networks pre-trained on millions of synthetic

tabular tasks to perform in-context inference over previously unseen feature schemas. TabPFN [24,25] establishes this paradigm with a transformer pre-trained on prior-fitted synthetic data, and TabICL [26] extends it to larger sample regimes with a column-then-row tokenisation scheme. Both have demonstrated state-of-the-art accuracy on classical tabular benchmarks [25] but have received markedly less attention than LLMs in security contexts.

Despite the visibility of both families, a systematic head-to-head comparison on OT/ICS intrusion detection is absent from the literature. Existing LLM-based IDS studies focus on enterprise IT networks [23,27] or on hybrid framework design [22] rather than on head-to-head benchmarking against modern tabular alternatives. Conversely, the published evaluations of TabPFN and TabICL have concentrated on classical UCI-style tabular benchmarks [25,26]; their behaviour on industrial cybersecurity workloads, characterised by severe class imbalance, time correlation, and a long tail of low-prevalence attack types, has not been characterised. Practitioners are left without a principled basis for choosing between the two model families.

This paper addresses that gap. Specifically, we make the following contributions:

(1) Tabular foundation models applied to OT/ICS security benchmarks. To the best of our knowledge, no prior study has evaluated TabPFN or TabICL on industrial control system intrusion detection. Where previous evaluations of these models concentrated on classical UCI-style tabular benchmarks [25,26] or IT-network IDS [27,28], we benchmark both on three OT/ICS testbeds, water-treatment SCADA (SWaT), multi-process hardware-in-the-loop (HAI), and Industrial IoT network flows (WUSTL-IIoT-2021), under a full-holdout multi-seed protocol with paired statistical testing.

(2) Head-to-head comparison of open-source LLMs and tabular foundation models. Prior LLM-based IDS studies on industrial and IoT settings [22,23] compared against Random Forest or XGBoost anchors but not against modern tabular foundation models; prior tabular-FM evaluations did not include LLM baselines. We compare four open-source LLMs (Qwen3-235B-A22B, Llama-3.3-70B, Hermes-4-70B, Hermes-4-405B) against TabPFN, TabICL, and two classical baselines (Random Forest and XGBoost) on the same protocol, with cross-seed Mann–Whitney tests and paired McNemar tests to ensure the conclusions are statistically defensible.

(3) Cross-domain operational characterisation. We provide a per-attack-class evaluation on the five-class WUSTL taxonomy, a cost-per-correct-prediction analysis of the LLM family, and a false-alarm-rate/detection-rate Pareto characterisation suitable for guiding deployment choices in industrial cyber-physical security.

(4) Max-context sensitivity analysis. We complement the K-shot headline comparison with a max-context sensitivity analysis (Section 5.8) that quantifies the cost of the K-shot constraint on each tabular method family and confirms that the foundation-model-versus-classical comparison is operative in the data-constrained regime that motivates foundation-model approaches.

(5) Deployment-oriented robustness analyses. We add a deployable confidence-gated tabular-to-LLM cascade that exceeds either standalone detector at ∼6% LLM escalation (Section 6.4), a leave-one-attack-type-out study of generalisation to unseen attack families (Section 5.11), a feature-budget ablation (Section 5.9), and a per-method computational-complexity profile.

The headline findings, reported in detail in Section 5, are that detection performance is strongly dataset-dependent, that tabular foundation models establish a strong new baseline for OT/ICS IDS, and that open-source LLMs are best deployed as complementary detectors rather than as universal replacements. Terminology is shown in Table 1.

Table 1. Terminology as used in this paper.

Term	Definition as Used in this Paper
Foundation model	Large network pre-trained on broad data, deployed without task-specific gradient updates
Open-source LLM	Qwen3-235B-A22B, Llama-3.3-70B, Hermes-4-70B/405B, as zero-gradient in-context classifiers
Tabular foundation model	TabPFN and TabICL, transformers pre-trained for in-context inference over tabular data
Classical anchor	Random Forest and XGBoost reference baselines
OT/ICS-IDS	Intrusion detection on operational-technology/industrial-control-system telemetry
K-shot regime	Headline protocol with *K* in-context examples per class for every method
Max-context regime	Sensitivity protocol where tabular anchors use their native training-budget maximum

The remainder of the paper is organised as follows. Section 2 reviews related work on classical IDS, foundation-model security applications, and tabular foundation models. Section 3 describes the three industrial security datasets and the data preparation protocol. Section 4 details the evaluation methodology, including the multi-seed protocol, the statistical testing framework, and the per-class analysis. Section 5 presents the experimental results. Section 5.7 consolidates the headline findings; Sections 5.8–5.11 report sensitivity and generalisation analyses on the training-data budget, the feature budget, the random-versus-chronological split, and unseen attack families respectively. Section 6 discusses operational implications and limitations. Finally, Section 7 presents the main conclusions.

2. Related Work

This section organises the relevant literature into four threads: classical and deep-learning intrusion detection (Section 2.1); Large Language Models in cybersecurity (Section 2.2); tabular foundation models (Section 2.3); and foundation-model applications to OT/ICS intrusion detection specifically (Section 2.4). Each subsection closes with a gap that this paper addresses.

2.1. Classical and Deep Learning Intrusion Detection

Machine-learning-based intrusion detection has been a mature research area for over two decades. Comprehensive surveys [2–6] document the evolution from signature- and rule-based detectors to supervised classifiers (Decision Trees, Random Forests, Support Vector Machines, multi-layer perceptrons) and, more recently, to deep architectures (Convolutional Neural Networks, Recurrent Neural Networks, autoencoders) trained on the canonical IT-network benchmarks: NSL-KDD [7], UNSW-NB15 [8], and the CIC-IDS family [9]. Reported accuracies on these benchmarks routinely exceed 0.95, but two practical limitations are well-documented. First, performance is highly sensitive to dataset preprocessing and to the class-imbalance handling strategy: synthetic minority over-sampling [10,11] and deep-learning- based augmentation [12] remain active research areas. Second, cross-dataset generalisation is poor; models trained on one benchmark transfer weakly to another, even within the same threat family [13]. Recent work has refined per-dataset performance further through hybrid feature selection and ensemble classifiers [29,30], but the cross-domain question remains open.

Beyond architecture, deep-learning IDS approaches are usefully organised by learning paradigm. Supervised detectors (CNN, RNN/LSTM, and tabular MLP classifiers) learn an explicit attack/normal boundary from labelled traffic and dominate the saturated IT-network benchmarks, but inherit the labelled-data scarcity of OT settings. Semi-supervised and unsupervised detectors (autoencoders, one-class and reconstruction-error models) instead learn a normal-operation manifold and flag deviations from it, which suits the severe class imbalance of industrial telemetry at the cost of weaker attack-type resolution. The same supervised-versus-unsupervised trade-off recurs in deep-learning methods across adjacent engineering domains: supervised multimodal single-image super-resolution with reversible guidance and cyclical knowledge distillation [31] on one side, and unsupervised

spatial–channel multi-scale graph-interaction transfer learning for rotating-machinery fault diagnosis [32] on the other. The foundation-model detectors compared in this paper occupy a third position: pre-trained, gradient-free, in-context classifiers that require neither a per-dataset supervised fit nor an explicit normal-manifold model.

The picture is more limited on OT/ICS-specific corpora [33]. The SWaT (secure water treatment) testbed, the HIL-based augmented (HAI) dataset, and the WUSTL-IIoT-2021 dataset have each been released with companion baselines, but cross-dataset evaluations using modern tabular classifiers (let alone foundation-model-based detectors) are scarce. Koneru and Cho [13] report a substantial performance gap between IT and ICS settings even when the same model is used, attributing the gap to smaller labelled corpora and the subtler physical-process signature of OT attacks. The methodological mismatch this creates, IT-IDS benchmarks saturate, OT-IDS benchmarks expose meaningful method differences, directly motivates the cross-dataset protocol adopted in this paper.

2.2. Large Language Models in Cybersecurity

The introduction of transformer architectures [14] and the demonstration of few-shot capability in large pre-trained models [15] have produced a rapid wave of cybersecurity-oriented LLM applications. Two recent systematic reviews [16,17] together catalogue more than two hundred studies spanning phishing detection, vulnerability discovery, security-operations-centre automation, threat-intelligence summarisation, and adversarial red-teaming. A broader survey of pre-trained foundation models [34] situates this work within the wider trajectory from BERT to modern instruction-tuned LLMs.

Several recent contributions illustrate the breadth of application. DeCusatis et al. [19] evaluate near-term LLMs for cybersecurity scenarios across phishing, social engineering, and log-summarisation; Balogh et al. [18] study generative AI as a support layer for human analysts; and Coppolino et al. [20] deploy an LLM-based pipeline for asset discovery in critical infrastructures. Adjacent applications include LLM-supported static application security testing [21] and the automatic generation of cybersecurity exercise scenarios [35].

For LLM-based intrusion detection specifically, two studies are most relevant. Muhammad et al. [22] propose HyLLM-IDS, a conceptual hybrid framework in which an LLM augments a classical IDS within cyber-physical systems, but stop short of providing an empirical head-to-head comparison against modern tabular alternatives. Li et al. [23] introduce IDS-Agent, an explainable LLM agent for IoT intrusion detection, demonstrating the feasibility of LLM-driven detection but leaving the question of how it compares to a strong tabular baseline open. Across this thread, the central methodological gap is the absence of controlled head-to-head benchmarks against state-of-the-art tabular models; existing comparisons typically use Random Forest or XGBoost as the only classical anchor.

2.3. Tabular Foundation Models

In parallel with the LLM trajectory, a second foundation-model family has emerged for tabular inference. TabNet [36] introduced an attention-based architecture with sequential feature selection and has been applied to a range of tabular workloads. The next conceptual step came with TabLLM [37] and follow-up work on LLM-driven feature engineering for tabular learning [38], which serialises tabular rows into natural-language prompts and exploits pre-trained LLMs as the classifier.

TabPFN [24] departed from this template entirely. Rather than treating a tabular row as language, TabPFN trains a transformer on millions of synthetic tabular tasks drawn from a structural causal model prior, then deploys it as a single-forward-pass in-context classifier on unseen tabular schemas of up to a few thousand rows. The architecture was recently extended in a *Nature* paper [25], showing state-of-the-art accuracy on classical

UCI-style benchmarks while running orders of magnitude faster than tuned gradient-boosted trees. TabICL [26] extends the paradigm to larger sample regimes through a column-then-row tokenisation scheme and a pre-training corpus drawn from realistic tabular distributions, demonstrating competitive or superior accuracy to TabPFN on the larger end of the OpenML-CC18 and TALENT benchmark suites.

What is striking about the published evaluations of both models is their near-exclusive focus on classical, balanced, IT-domain tabular benchmarks. The behaviour of TabPFN and TabICL on severely imbalanced security workloads with rare-attack long tails has not been characterised, and to our knowledge, no published study has applied either model to OT/ICS intrusion detection.

2.4. Foundation Models for OT/ICS Intrusion Detection

Two recent strands converge towards the question this paper addresses. First, on the LLM side, the works of Muhammad et al. [22] and Li et al. [23] establish that foundation-model-based detectors can be designed for industrial and IoT settings, but without controlled comparisons against the strongest tabular alternatives. Second, on the tabular-FM side, prior work [27] reported a first evaluation of TabPFN and few-shot LLM classification on IT-network IDS benchmarks (CIC-IDS-2017, N-BaIoT, CIC-UNSW), finding TabPFN to be a competitive baseline against tuned classical models on that domain. García et al. [28] further extended the framework to report a first evaluation of TabICL on the same benchmarks. Those studies, however, explicitly did not address OT/ICS workloads. The two settings differ at a more fundamental level than the dataset: IT-network IDS classifies flow-level traffic features, whereas the OT/ICS task classifies physical-process telemetry from sensors and actuators on operating critical-infrastructure plants. The class-imbalance profile, the temporal correlation structure of the underlying signals, and the multi-class attack taxonomy differ in turn, but the substantive shift is from network-traffic features to physical-process channels, a regime in which the inductive biases of pre-trained language and tabular models have not previously been characterised.

These threads leave a clearly defined gap. To the best of our knowledge, we are the first to evaluate modern tabular foundation models (TabPFN, TabICL) on the established OT/ICS intrusion-detection benchmarks; to compare open-source LLMs to those tabular foundation models on the same protocol; and to study the operational characteristics that matter for deployment in industrial cyber-physical security: the false-alarm-rate versus detection-rate trade-off, the per-attack-class behaviour on a genuinely multi-class taxonomy, and the cost-per-correct-prediction ratio between model families. The contributions enumerated in Section 1 are designed to close this gap on each axis in turn.

3. Datasets

We evaluate every method on three publicly available industrial security benchmarks that together span the operational diversity relevant to this study: a SCADA-controlled water-treatment plant, a multi-process hardware-in-the-loop testbed, and an Industrial Internet of Things network-flow corpus. Each dataset is summarised in turn, and Section 3.4 describes the preparation pipeline applied uniformly across all three.

3.1. SWaT: Secure Water Treatment

The secure water treatment (SWaT) testbed [39,40] is a six-stage water-purification plant developed at the iTrust Centre for Research in Cyber Security in Singapore. The plant comprises raw-water storage, ultrafiltration, dechlorination, reverse osmosis, and storage stages, instrumented with 51 physical sensors and actuators (24 analog sensors plus 27 actuators) reporting at 1 Hz. The dataset records 11 consecutive days of operation: 7 days

of normal operation followed by 4 days during which 36 attack scenarios were executed against the plant's programmable logic controllers (PLCs), targeting flow, pressure, and level sensors. The original distribution contains approximately 946 k labelled records.

We use the public Kaggle mirror `vishala28/swat-dataset-secure-water-treatment-system`, which ships the dataset in a balanced format suitable for classification benchmarking (50/50 `Normal`/`Attack` prevalence after the mirror maintainer's subsampling). The feature schema is the raw 51-dimensional sensor-actuator vector with the time stamp dropped. We note that the balanced prevalence of the mirror is an artefact of its preparation; the natural prevalence in the original iTrust distribution is approximately 12% attack. We discuss the implications for false-alarm-rate interpretation in Section 6.6.

3.2. HAI: HIL-Based Augmented Industrial Dataset

The HIL-based Augmented Industrial (HAI) dataset [41] was developed by the Korean National Security Research Institute to address the absence of multi-process cyber-physical benchmarks. The testbed integrates four physically interconnected subsystems, a boiler process (GE), a turbine process (Emerson), a water-treatment process (FESTO/Siemens), and a HIL simulator (dSPACE), yielding 79 sensor and actuator channels sampled at 1 Hz. The dataset spans approximately 600 k records across one week of operation with 38 attack scenarios, each targeting individual subsystems and combinations thereof. Attack prevalence in the natural distribution is approximately 3%, reflecting realistic industrial baseline ratios.

We use the public Kaggle mirror `icsdataset/hai-security-dataset`, which preserves the natural prevalence and the full 79-dimensional feature schema. The multi-process character of HAI makes it the most operationally heterogeneous of the three datasets evaluated here and the most challenging in terms of cross-method ranking, as the results in Section 5 will show.

3.3. WUSTL-IIoT-2021

The WUSTL-IIoT-2021 dataset [42] was released by Washington University in St. Louis and reflects a different operational regime: instead of physical-process signals, it captures network traffic between PLCs, human–machine interfaces (HMIs), and field devices on a Modbus/TCP industrial backbone. The dataset comprises approximately 1.19 M flow records summarising 53 h of operation, each row representing an Argus-style flow with 41 flow-level features (packet and byte counts, transport-layer flags, TCP TTLs, port and protocol fields, flow rate and load). The natural attack prevalence is approximately 7%, matching the literature characterisation [42].

WUSTL-IIoT-2021 is unique among the three datasets in our study in that it ships a five-class attack taxonomy: Normal, Denial-of-Service (DoS), Reconnaissance, Command Injection, and Backdoor. We use the Kaggle mirror `annaamalaiu/wustl-iiot-2021-dataset`, which is a thin redistribution of the original release.

The mirror requires two pieces of attention. First, the WUSTL authors explicitly warn that six metadata columns, `StartTime`, `LastTime`, `SrcAddr`, `DstAddr`, `sIpId`, `dIpId`, are unique to individual attacks and would let a classifier exploit the address structure rather than the network behaviour; these are dropped before any model sees the data. Second, the Kaggle mirror ships two candidate label columns: a categorical `Traffic` column containing the multi-class attack-type string and a numeric `Target` column carrying the binary indicator. The preprocessing pipeline auto-resolves which column is binary by inspecting the value distribution (Section 3.4).

3.4. Data Preparation and Preprocessing

A unified preparation pipeline is applied across the three datasets so that any per-dataset performance difference reflects intrinsic data characteristics rather than methodological choices.

Leakage column removal.

For WUSTL-IIoT-2021, the six leakage columns flagged by the dataset authors are dropped automatically. For SWaT and HAI, time-stamp columns are dropped to prevent the classifier from memorising temporal positions of attack windows. Source-file provenance columns introduced by Kaggle redistribution are likewise removed.

Label resolution.

On WUSTL, the binary label column is auto-detected by inspecting which of the candidate columns (`Target`, `Traffic`, `Label`) contains numerically-valid 0/1 values for at least 99% of sampled rows; the categorical column is reserved for the multi-class evaluation. On SWaT and HAI, the binary label is taken from the canonical `Normal/Attack` indicator shipped by each Kaggle mirror. Constant feature columns and columns with >50% missing values are dropped during a basic numeric clean.

Feature selection.

To bound prompt length for the LLM and to equalise the feature budget across all method families, each dataset is reduced to its $K_{\text{feat}} = 12$ most informative columns prior to any model fitting. Feature importance is scored by mutual information with the label (`mutual_info_classif`, computed on a class-balanced subsample for tractability), and the top 12 columns are retained. The identical 12-feature schema is presented to every method, the four open-source LLMs, the two tabular foundation models, and the two classical anchors, so that no method enjoys a feature-budget advantage. For SWaT the selected channels are `AIT201`, `AIT501`, `AIT402`, `LIT301`, `AIT502`, `PIT501`, `PIT503`, `PIT502`, `LIT401`, `FIT503`, `AIT203`, and `FIT401`; the selected channels for HAI and WUSTL are listed in the repository. The retained SWaT channels remain semantically named physical-process sensors (analyser, level, pressure, and flow transmitters), which are relevant to the feature-semantics interpretation in Section 6.1.

Class subsampling and holdout construction.

To equalise compute and prompt costs across datasets, each dataset is subsampled to at most 50,000 rows per class, yielding approximately 100,000 rows of total processed data per dataset. A stratified random 80%/20% split produces the train and test partitions; the split is performed independently of the timestamp axis, so train and test rows are drawn from the entire span of plant operation rather than from disjoint time windows. The choice of a random rather than a chronological split is a deliberate methodological trade-off, discussed in detail in the next paragraph. The multi-seed protocol described in Section 4 draws repeated test-subsets from the 20% holdout to obtain bootstrap-style cross-seed variance estimates. For the LLM evaluation, an additional stratified subsample of $n = 6000$ rows is drawn from the holdout under the natural class distribution; this is the LLM's effective test set. The four anchors (Random Forest, XGBoost, TabPFN, TabICL) are evaluated on the full holdout (approximately 99,980 rows for SWaT and WUSTL, and 62,010 rows for HAI). The LLM-anchor accuracy gap is then corrected for the size difference via paired McNemar testing on the identical $n = 6000$ subsample, as detailed in Section 4.

Random vs. chronological splitting.

SWaT and HAI are time-indexed datasets sampled at 1 Hz from running industrial processes, and an attack-class label persists across the duration of each scripted attack window (tens of seconds to several minutes). A random stratified split therefore places test samples within the temporal neighbourhood of training samples; if a detector exploits short-term temporal continuity of the underlying physical process, this neighbourhood structure could, in principle, produce an optimistic accuracy estimate relative to a deployment where the detector is asked to classify previously-unseen attack windows. We make the random-split choice deliberately, for three reasons. First, the present study is a controlled comparison of foundation-model families on the same protocol, not an absolute benchmark of any single method; both the LLM and the tabular anchors are subject to the same potential temporal-neighbourhood bias, so cross-family relative comparisons are minimally affected. Second, none of the eight methods evaluated in this study makes explicit use of temporal context: the LLM prompt is a permutation of feature blocks, TabPFN and TabICL are order-invariant in-context classifiers, and the Random Forest is fed each sample independently. The temporal-continuity exploit would therefore have to operate through coincidental nearest-neighbour structure in feature space rather than through a modelled time dimension. Third, the random-split protocol matches the evaluation convention adopted by the published SWaT, HAI, and WUSTL baselines [40–42] as well as the prior IT-network IDS work in our group [27,28], preserving comparability with this prior literature. A chronological split, in which the train partition covers an earlier time interval and the test partition a strictly later one, would address the neighbourhood concern at the cost of conflating the method-comparison question with the question of how well each method generalises across attack repertoires that evolved over the recording window. The chronological-split robustness check reported in Section 5.10 quantifies the effect of the random-vs-chronological choice on the headline findings; cross-campaign generalisation, in which training and test partitions correspond to distinct attack campaigns with potentially non-overlapping attack repertoires, remains a future-work direction discussed in Section 6.7.

Multi-class data availability.

A practical limitation of the three Kaggle mirrors deserves explicit acknowledgement. The SWaT and HAI Kaggle redistributions carry only the binary `Normal/Attack` indicator; the per-attack-type labels present in the original iTrust and NSRI distributions were not re-uploaded. As a result, only WUSTL-IIoT-2021 supports a genuine multi-class evaluation in our protocol. The multi-class runs reported for SWaT and HAI in Section 5, therefore, reduce to their binary equivalents, while the five-class WUSTL evaluation provides the only attack-taxonomy-resolved comparison in the paper.

Summary.

After preprocessing, the three datasets present complementary operational regimes: SWaT, a balanced and saturated physical-signal setting; HAI, a natural-prevalence multi-process setting with substantial cross-method variance; and WUSTL-IIoT-2021, a heavily imbalanced flow-level setting with both saturated binary accuracy and a non-trivial five-class structure. Table 2 summarises the resulting characteristics across the three datasets after preparation.

Table 2. Summary of the three OT/ICS datasets after preparation. The native feature count is the schema shipped by the Kaggle mirror; the used count ($K_{\text{feat}} = 12$ on every dataset) is the mutual-information-selected subset that every method actually sees, as described in Section 3.4. n_{anchor} is the tabular-anchor test set (full holdout); n_{LLM} the LLM stratified subsample. Prevalence is on the Kaggle mirror, with the natural-distribution value in parentheses.

Dataset	Domain	Features (Native)	Features (Used)	n_{anchor}	n_{LLM}	Attack Prevalence
SWaT	Water-treatment SCADA	51 sensor/actuator	12 (MI-selected)	99,980 †	6000	50% (natural: ∼12%)
HAI	Multi-process HIL	79 sensor/actuator	12 (MI-selected)	62,010	6000	natural (∼3%)
WUSTL	IIoT Modbus/TCP flows	41 flow features	12 (MI-selected)	99,980 †	6000	natural (∼7%)

† The matching SWaT and WUSTL anchor-holdout sizes (99,980) follow from the per-class subsampling cap of 50,000 rows per class described in Section 3.4: both datasets saturate the cap, yielding approximately 100,000 total rows.

To complement the quantitative results, Figure 1 shows two-dimensional t-SNE and UMAP projections of the standardised $K_{\text{feat}} = 12$ feature space for each dataset, coloured by class. The geometry mirrors the measured difficulty: SWaT and especially WUSTL show well-separated attack structures, whereas HAI attacks overlap the normal manifold, consistent with its lower detection scores across all methods (Section 5.1).

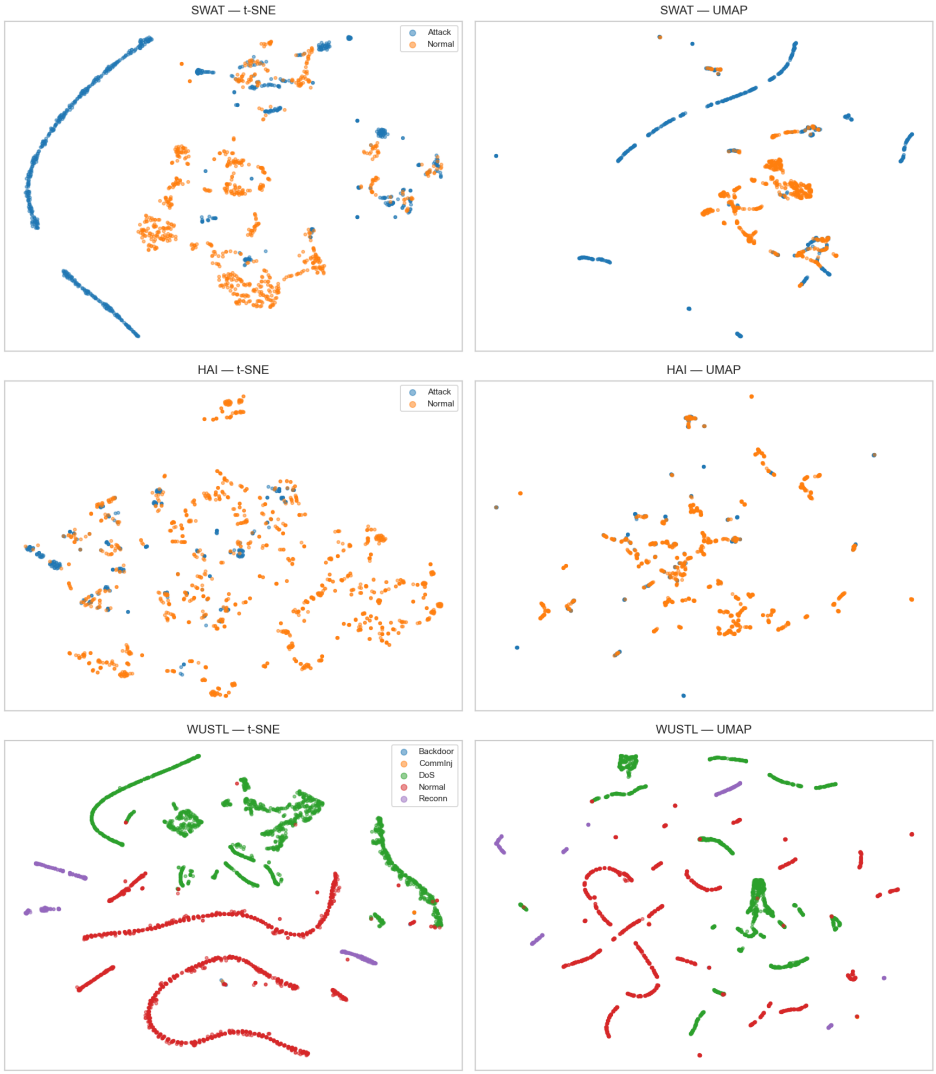

Figure 1. t-SNE (**left**) and UMAP (**right**) projections of the standardised $K_{\text{feat}} = 12$ feature space for SWaT, HAI, and WUSTL (rows), coloured by class (binary for SWaT/HAI; five-class for WUSTL). Class separability in the projections tracks the quantitative detection difficulty of each dataset.

4. Methodology

This section specifies the evaluation protocol applied uniformly across all three datasets. Section 4.1 enumerates the methods under comparison; Sections 4.2 and 4.3 document the LLM and tabular-anchor configurations; Section 4.4 describes the multi-seed holdout protocol; Section 4.5 the metrics; and Section 4.6 the paired and cross-seed statistical tests used to support every quantitative claim. The pipeline is shown in Figure 2.

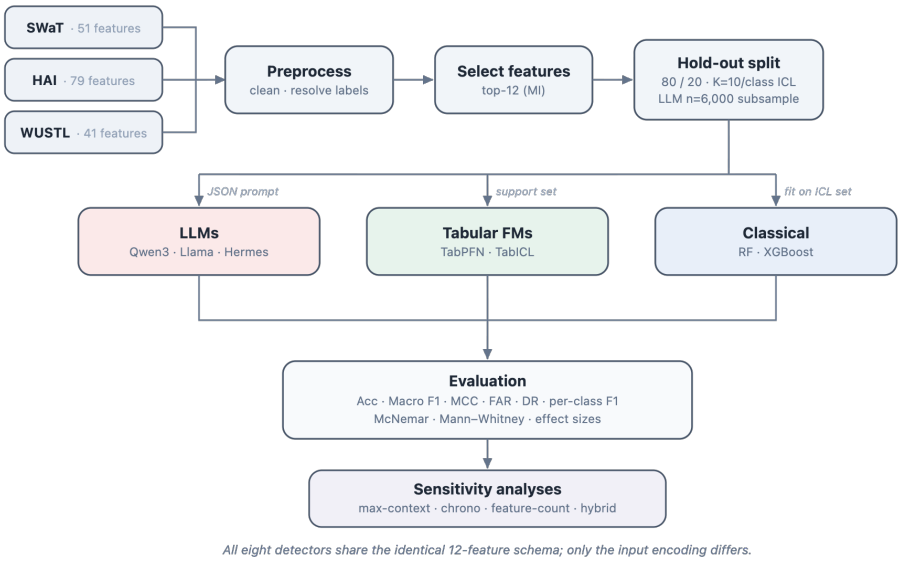

Figure 2. Unified evaluation pipeline. Raw OT/ICS telemetry is cleaned and label-normalised, reduced to a common feature budget by mutual-information ranking, and split into a balanced in-context (support) set and a stratified test set. The same inputs are presented to three detector families—classical anchors (Random Forest, XGBoost), tabular foundation models (TabPFN, TabICL), and open-source LLMs—each with its native input encoding, and scored under one evaluation and statistical-testing protocol.

4.1. Methods Evaluated

The benchmark compares eight methods drawn from three families. The first family comprises four open-source large language models accessed through the Nebius AI Studio inference API: Qwen3-235B-A22B (the designated primary LLM, an instruction-tuned Mixture-of-Experts model with 22 B active parameters), Llama-3.3-70B (Meta's instruction-tuned 70 B dense model), Hermes-4-70B and Hermes-4-405B (Nous Research's instruction-tuned variants). The second family comprises two tabular foundation models: TabPFN [24,25], the prior-fitted-network transformer accessed through the official `tabpfn-client` cloud API, and TabICL [26], the in-context learning model deployed locally at the v1.1 checkpoint `tabicl-classifier-v1.1-20250506`. The third family comprises two classical baselines: RandomForest (the primary classical anchor, scikit-learn implementation with 200 trees and otherwise default hyperparameters) and XGBoost (a gradient-boosted control whose role and configuration are discussed in Section 4.3, "Choice of classical baselines").

4.2. LLM Inference Protocol

Each LLM is deployed as a zero-gradient, in-context binary or multi-class classifier. The prompt structure follows a role-instructed convention described next; gradient-based fine-tuning is not performed on any LLM in this study.

Role-instructed system message.

Each dataset is associated with a domain-specific system message that describes the testbed to the LLM in natural language. For SWaT, the system message identifies the LLM as "a senior process-control engineer monitoring a six-stage water-treatment plant"; for HAI, "a control-room operator monitoring a hardware-in-the-loop industrial process"; and for WUSTL, "a network and ICS cybersecurity analyst monitoring an Industrial IoT testbed running Modbus/TCP traffic". The system message lists the feature naming convention (e.g., LIT for level transmitters in SWaT, TotPkts for total packet counts in WUSTL) and the allowed output labels. The same template is used across all four LLMs to avoid prompt-engineering confounds. Listing 1 reproduces a representative SWaT binary prompt; Listing 2 reproduces a representative WUSTL five-class prompt.

Listing 1. Representative SWaT binary prompt ($K_{\text{shot}} = 2$ shown for brevity; the protocol uses $K_{\text{shot}} = 10$). The listing is schematic: the deployed prompt serialises the 12 mutual-information-selected channels (Section 3.4) as a JSON object per row, e.g., `{"AIT201":177.77, ...}` → `Normal`, rather than the human-readable form shown here.

```
[system]
You are a senior process-control engineer monitoring a six-stage
water-treatment plant. The˜plant is instrumented with 51 channels:
- LIT-xxx: level transmitters (water level in tanks, units: mm).
- FIT-xxx: flow transmitters (water flow rate, units: m˜3/h).
- AIT-xxx: analyzer transmitters (chemistry, e.g.,˜pH, conductivity).
- MV-xxx, P-xxx, UV-xxx: motorized valves, pumps, UV lamps (state: 0/1/2).
Classify each row as one of {Normal, Attack}. Respond with one word˜only.

[user]
Example 1 (label: Normal):
LIT-101=521.3, FIT-101=2.42, MV-101=2, P-101=1, .., AIT-503=7.89
→ Normal

Example 2 (label: Attack):
LIT-101=812.7, FIT-101=2.41, MV-101=2, P-101=1, .., AIT-503=7.91
→ Attack

Query:
LIT-101=655.2, FIT-101=2.40, MV-101=2, P-101=1, .., AIT-503=7.88
→
```

In-context examples.

The system message is followed by K_{shot} labelled examples per class drawn from the train partition, presented as alternating `<features>` → `<label>` blocks. The default value is $K_{\text{shot}} = 10$, balanced across classes; an ablation (Section 5, E2) sweeps this value across $\{5, 10, 25, 50, 100\}$.

Query and output parsing.

Each test sample is presented as a final feature block, and the LLM is instructed to respond with exactly one label string. The temperature is set to 0 (deterministic decoding), where the platform permits it; maximum output tokens are capped at 8, which is comfortably above the longest label string. A simple deterministic parser extracts the first valid label from the LLM response; samples for which no valid label can be parsed are counted as *Attack* in the binary setting (the conservative default for IDS) and as the most common attack class in the multi-class setting. The parse rate is reported alongside accuracy and is >0.99 on every combination of LLM, dataset, and seed in our runs.

Listing 2. Representative WUSTL-IIoT-2021 five-class prompt ($K_{\text{shot}} = 2$ per class shown for brevity). As in Listing 1, the listing is schematic: the deployed prompt serialises the 12 selected flow features (Section 3.4) as a JSON object per row rather than the human-readable form shown here.

```
[system]
You are a network and ICS cybersecurity analyst monitoring an
Industrial IoT testbed running Modbus/TCP traffic. Each row is
a network flow summarised by 41 features:
- TotPkts, TotBytes: total packet/byte counts in the flow.
- SrcBytes, DstBytes: bytes from source/destination.
- Rate, Load: flow rate and load.
- sTtl, dTtl: source/destination TCP TTLs.
- Sport, Dport, Proto: ports and transport protocol.
- Other categorical and flag fields.
Classify each flow as one of:
  Normal, DoS, Reconnaissance, CommandInjection, Backdoor.
Respond with exactly one~label.

[user]
Example 1 (label: Normal):
TotPkts=14, TotBytes=842, Rate=12.4, Sport=502, Dport=4711, ..
→ Normal

Example 2 (label: DoS):
TotPkts=8731, TotBytes=512040, Rate=4128.1, Sport=502, Dport=4712, ..
→ DoS

Example 3 (label: Reconnaissance):
TotPkts=22, TotBytes=1408, Rate=4.1, Sport=1024, Dport=502, ..
→ Reconnaissance

Example 4 (label: CommandInjection):
TotPkts=6, TotBytes=312, Rate=2.0, Sport=49152, Dport=502, ..
→ CommandInjection

Example 5 (label: Backdoor):
TotPkts=3, TotBytes=180, Rate=0.8, Sport=44321, Dport=502, ..
→ Backdoor

Query:
TotPkts=15, TotBytes=920, Rate=11.8, Sport=502, Dport=4711, ..
→
```

4.3. Tabular Anchor Configuration

Random Forest.

The scikit-learn `RandomForestClassifier` with 200 trees, square-root feature subsampling, and balanced class weights. Hyperparameters were not tuned per-dataset; the goal is to assess foundation-model behaviour against a competent default rather than a per-dataset-tuned ceiling. In the headline E7 protocol Random Forest is fit on the same $K_{\text{shot}} = 10$-per-class ICL set as every other anchor (Section 4.3, "Choice of classical baselines"); the full-training-partition regime is reported in Section 5.8.

TabPFN.

We use the `tabpfn-client` cloud API. In the headline E7 protocol TabPFN is given the same $K_{\text{shot}} = 10$-per-class in-context support set as Random Forest and every other anchor (Section 4.4), and performs in-context classification in a single forward pass without gradient updates; the full holdout is used at inference. The full-training-budget regime,

in which TabPFN is given up to 10,000 stratified support rows (its v2 context limit), is reported separately in the max-context sweep of Section 5.8.

TabICL.

We use TabICL v1.1 checkpoint with the published default configuration: $n_{\text{estimators}} = 16$, normalisation methods `["none","power"]`, Latin-square feature shuffling, shifted class shuffling, outlier threshold 4.0, softmax temperature 0.9, and logit averaging across estimators. The model is loaded from the `tabicl-classifier-v1.1-20250506` checkpoint and run on CPU; GPU acceleration is not required at the holdout sizes in this study.

Choice of classical baselines.

Random Forest was originally selected as the primary classical anchor for three reasons: (1) it is the most-cited classical baseline in the OT/ICS-IDS literature against which SWaT, HAI, and WUSTL-IIoT-2021 are typically reported, providing direct comparability with prior work; (2) it has low variance across seeds and minimal hyperparameter sensitivity, which is desirable for a reference point in a paired-comparison study; and (3) its bootstrap-based ensemble produces stable estimates even at the K-shot training-set sizes used by the foundation-model anchors. To address the concern that a single classical anchor risks understating the performance of modern classical methods, we additionally evaluate XGBoost [43] as a second classical anchor. XGBoost is the canonical gradient-boosted tree implementation and the standard "modern classical" reference in recent tabular benchmarks. Both anchors are configured with 200 trees, balanced class weighting, and the same K-shot ICL training set ($K_{\text{shot}} = 10$ examples per class) as every other anchor; XGBoost additionally uses `max_depth=6`, `learning_rate=0.1`, and `eval_metric="logloss"` (or `"merror"` for multi-class), which are the library defaults. As reported in Section 5.1, XGBoost trails Random Forest on every dataset under the K-shot protocol, which is consistent with the known sensitivity of gradient-boosted methods to extremely small-sample training sets [43] and validates the choice of Random Forest as the primary classical reference point for this regime; XGBoost recovers to near-ceiling performance once the training budget is lifted, as the max-context analysis in Section 5.8 shows.

Reproducibility summary.

Table 3 consolidates the implementation versions, model checkpoints, and inference parameters used throughout this study. The full prompt templates (the listings in Section 4.2 are abridged), per-method seed loops, and statistical-testing utilities are available alongside a pinned environment specification in the public repository linked in the Data Availability statement.

4.4. Evaluation Protocol

The same multi-seed, holdout-based protocol is applied to every method and dataset.

Holdout construction.

After the preprocessing described in Section 3.4, the full processed corpus is partitioned into a training partition and a held-out evaluation partition. In the headline E7 protocol, all four anchors (Random Forest, XGBoost, TabPFN, TabICL) are fit on the $K_{\text{shot}} = 10$-per-class in-context support set drawn from the training partition, identical to the LLM's ICL set, and not the full training partition; the full-training-budget regime is reported separately in the max-context sweep of Section 5.8. Held-out partition sizes are $n = 99{,}980$ for SWaT, $n = 62{,}010$ for HAI, and $n = 99{,}980$ for WUSTL; the matching SWaT and WUSTL sizes are a consequence of the 50,000 rows-per-class subsampling cap described in Section 3.4, which both balanced (SWaT) and naturally-imbalanced (WUSTL)

datasets saturate at approximately 100,000 total rows. The HAI natural prevalence of ∼3% attack yields a smaller after-cap total of ∼62,010 rows.

Table 3. Reproducibility summary: library versions, model checkpoints, and inference parameters. Seeds {42, 43, 44} in E7, max-context, and chronological-split; {42, 43, 44, 45, 46} in R1 and R2.

Item	Value
Python/environment	3.12 on Google Colab; NVIDIA A100 for orchestration, LLM inference offloaded to Nebius AI Studio API, TabICL on CPU
scikit-learn	1.6.1
xgboost	3.2.0
tabpfn-client	cloud API (`tabpfn-client`)
tabicl	checkpoint `tabicl-classifier-v1.1-20250506`
openai (Nebius client)	2.37.0
numpy/pandas/scipy	2.0.2/2.2.2/1.16.3
Nebius AI Studio model IDs	
Qwen3-235B-A22B	`Qwen/Qwen3-235B-A22B-Instruct-2507`
Llama-3.3-70B	`meta-llama/Llama-3.3-70B-Instruct`
Hermes-4-70B	`NousResearch/Hermes-4-Llama-3.1-70B`
Hermes-4-405B	`NousResearch/Hermes-4-Llama-3.1-405B`
LLM decoding	temperature 0, top_p 1, max_tokens 8, $n_{\text{vote}} = 1$; first-valid-label parse; unparseable → `Attack` (binary)/most common attack class (multi-class)
K_{shot}	10 per class (primary); {5, 10, 25, 50, 100} in the E2 sweep
Random Forest	`n_estimators=200, max_features="sqrt", class_weight="balanced"`, otherwise scikit-learn defaults
XGBoost	`n_estimators=200, max_depth=6, learning_rate=0.1, eval_metric="logloss"/"mlogloss"`, balanced class weighting
TabPFN	`TabPFNClassifier` cloud API; $K_{\text{shot}} = 10$-per-class ICL support set in the headline E7 protocol, up to 10,000 stratified support rows (v2 cap) in the max-context sweep
TabICL	v1.1 checkpoint, $n_{\text{estimators}} = 16$, `norm_methods=["none","power"]`, Latin-square feature shuffling, shifted class shuffling, outlier threshold 4.0, softmax temperature 0.9, logit averaging
Code repository	https://github.com/drdecurto/fm-security (accessed on 2 June 2026)

Tabular anchors: full holdout.

The four anchors (Random Forest, XGBoost, TabPFN, TabICL), each fit on the $K_{\text{shot}} = 10$-per-class ICL set as described above, are evaluated on the full holdout partition. Holdout sizes are $n = 99{,}980$ for SWaT, $n = 62{,}010$ for HAI, and $n = 99{,}980$ for WUSTL, yielding tight within-run Wilson confidence intervals on every reported metric.

LLM: stratified-natural subsample.

LLM inference cost scales linearly with the number of test samples. To make the LLM evaluation operationally feasible without sacrificing the class-imbalance structure of the holdout, each LLM is evaluated on a $n_{\text{LLM}} = 6000$-row stratified subsample drawn from the holdout under the holdout's empirical (i.e., natural for HAI and WUSTL; balanced for the SWaT Kaggle mirror) class distribution. The LLM-versus-anchor significance comparison reported in Section 5 is performed on the identical $n = 6000$ samples (the anchors are re-evaluated on the same subsample for the paired test).

Multi-seed repetition.

The protocol is repeated across $S = 3$ random seeds for the primary E7 evaluation in Section 5. The 5-seed multi-LLM ablation (R1) and the WUSTL per-class analysis (R2) draw

on $S = 5$ seeds, where the slightly larger seed budget is needed to power non-parametric tests against the Random Forest baseline.

Two-level comparison protocol.

The asymmetric evaluation budgets above support two distinct, mutually consistent comparisons that are reported in turn in Section 5 and are worth separating explicitly. At *Level 1*, each method is reported on its native evaluation budget: tabular anchors on the full holdout ($n = 99{,}980$ for SWaT and WUSTL, $n = 62{,}010$ for HAI), LLMs on the $n_{\text{LLM}} = 6000$ stratified subsample. Because the LLM subsample is drawn from the same holdout under the same empirical class distribution, the LLM and anchor numbers in Table 4 are unbiased estimates of each method's performance on a common underlying population; the budget difference manifests only as a wider within-run Wilson confidence interval on the LLM rows. At *Level 2*, the LLM and every anchor are re-evaluated on the identical $n = 6000$ samples and a paired McNemar test is computed on the matched correct/incorrect pattern (Section 5.2, Table 5). This eliminates the subsample-vs-full-holdout asymmetry of Level 1 and provides the strict pairwise comparison that underwrites every reported p-value. The two levels answer different operational questions: "what accuracy does each method achieve at its deployed inference budget?" and "is the LLM significantly better or worse than each anchor on the same rows?", and they yield qualitatively consistent rankings on every dataset evaluated in this study.

4.5. Metrics

We report Accuracy, Macro F1, Matthews Correlation Coefficient (MCC), False Alarm Rate (FAR), and Detection Rate (DR) on every method-dataset combination. For the multi-class evaluation on WUSTL, per-class F1 is reported with bootstrap 95% confidence intervals computed across the five seeds. For threshold-independent reporting on the tabular anchors, we additionally compute AUROC and AUPRC on the full holdout from the `predict_proba` output; the LLMs are omitted from AUROC reporting as they do not provide calibrated probabilities, following the OT/ICS evaluation convention [40–42].

For the multi-seed runs we report the mean and standard deviation across seeds, and the cross-seed minimum and maximum. Within-seed Wilson 95% confidence intervals are computed for the binary accuracy and the FAR and DR rates following the standard binomial-proportion formula.

4.6. Statistical Testing

Three layers of statistical testing support the comparative claims in Section 5.

Paired McNemar tests.

For every (LLM, anchor, dataset) triple, we conduct a paired McNemar test on the identical $n = 6000$ LLM subsample. Let b denote the number of samples on which the LLM is correct and the anchor is wrong, and c the reverse. The McNemar test statistic is computed under the standard exact-binomial formulation (no continuity correction), reported as a two-sided p-value. This is the primary test for the within-seed LLM-versus-anchor comparison and underpins the McNemar heat-map figure in Section 5.

Cross-seed Mann–Whitney tests.

The R1 multi-LLM evaluation uses the non-parametric Mann–Whitney U test on the five seed-level macro-F1 values per LLM against the five seed-level macro-F1 values of Random Forest, performed independently per dataset. This is the appropriate test for distributions with $n = 5$ per arm where the normality assumption of a t-test is unjustified, and it underpins the cross-LLM significance analysis in Section 5.3.

Per-class significance.

For the WUSTL multi-class per-class analysis (R2), per-class F1 deltas between each LLM and Random Forest are tested with Mann–Whitney across 5 seeds, with $p < 0.05$ marked as significant in the per-class delta analysis of Section 5.4. No multiple-testing correction is applied; we treat each per-class test as a separate hypothesis and report the raw p-values, with the explicit caveat in Section 6.6 that the per-class deltas should be read descriptively rather than as confirmatory tests of a pre-registered hypothesis.

Effect size reporting.

Where significance is reported, we accompany every p-value with the corresponding effect size: Δaccuracy (raw difference) for McNemar, ΔmacroF1 (difference of seed means) for Mann–Whitney, and the per-class F1 gap for the WUSTL per-class analysis. This is intended to discourage the reader from interpreting a small p-value on a large n as operationally meaningful when the effect size is negligible.

5. Results

This section reports the empirical findings. Section 5.1 presents the headline binary-classification results across the three datasets. Section 5.2 reports paired McNemar tests of the LLM against each anchor on the identical $n = 6000$ subsample. Section 5.3 reports the cross-seed Mann–Whitney comparison of every LLM against the Random Forest baseline. Section 5.4 reports the genuine multi-class evaluation on WUSTL-IIoT-2021. Section 5.5 characterises the operational trade-offs (false-alarm-rate against detection-rate, and cost-per-correct-prediction against macro-F1). Section 5.6 collects two robustness analyses: in-context-example scaling and per-seed dispersion. Section 5.7 consolidates the headline findings; Sections 5.8–5.11 report sensitivity and generalisation analyses on the training-data budget, the feature budget, the random-versus-chronological split, and unseen attack families respectively.

5.1. Headline Binary Comparison Across Datasets

Table 4 reports mean and standard deviation across three random seeds (E7 protocol). Figure 3 visualises the same data as a four-panel forest plot, one panel per metric, with SWaT, HAI, and WUSTL points stacked within each method row.

Table 4. Cross-dataset E7 multi-seed binary evaluation (mean ± std across 3 seeds). Anchors on the full holdout, LLM on a stratified $n_{\text{LLM}} = 6000$ subsample; best per dataset on Acc, Macro F1, MCC bolded.

Dataset	Model	Acc.	Macro F1	MCC	FAR	DR
HAI	RandomForest	0.707 ± 0.014	0.657 ± 0.011	0.413 ± 0.014	0.326 ± 0.019	0.842 ± 0.012
HAI	XGBoost	0.659 ± 0.000	0.607 ± 0.000	0.314 ± 0.000	0.364 ± 0.000	0.759 ± 0.000
HAI	TabPFN	0.683 ± 0.006	0.641 ± 0.006	0.406 ± 0.010	0.363 ± 0.006	0.875 ± 0.006
HAI	TabICL	**0.733 ± 0.019**	0.678 ± 0.014	**0.432 ± 0.013**	0.288 ± 0.029	0.822 ± 0.028
HAI	Qwen3-235B-A22B	0.690 ± 0.003	0.633 ± 0.004	0.350 ± 0.012	0.328 ± 0.003	0.764 ± 0.016
SWaT	RandomForest	0.827 ± 0.001	0.822 ± 0.001	0.695 ± 0.002	0.003 ± 0.002	0.657 ± 0.003
SWaT	XGBoost	0.824 ± 0.000	0.819 ± 0.000	0.684 ± 0.000	0.015 ± 0.000	0.662 ± 0.000
SWaT	TabPFN	0.805 ± 0.000	0.803 ± 0.000	0.627 ± 0.001	0.080 ± 0.002	0.691 ± 0.001
SWaT	TabICL	0.818 ± 0.001	0.813 ± 0.001	0.671 ± 0.003	0.022 ± 0.003	0.659 ± 0.003
SWaT	Qwen3-235B-A22B	**0.836 ± 0.003**	0.833 ± 0.004	**0.700 ± 0.005**	0.026 ± 0.003	0.699 ± 0.008
WUSTL	RandomForest	0.985 ± 0.001	0.985 ± 0.001	0.970 ± 0.002	0.023 ± 0.002	0.993 ± 0.001
WUSTL	XGBoost	0.894 ± 0.000	0.893 ± 0.000	0.806 ± 0.000	0.001 ± 0.000	0.790 ± 0.000
WUSTL	TabPFN	**0.988 ± 0.000**	0.988 ± 0.000	**0.976 ± 0.000**	0.022 ± 0.000	0.998 ± 0.000
WUSTL	TabICL	0.985 ± 0.000	0.985 ± 0.000	0.971 ± 0.001	0.028 ± 0.000	0.999 ± 0.001
WUSTL	Qwen3-235B-A22B	0.986 ± 0.000	0.986 ± 0.000	0.972 ± 0.001	0.026 ± 0.000	0.998 ± 0.000

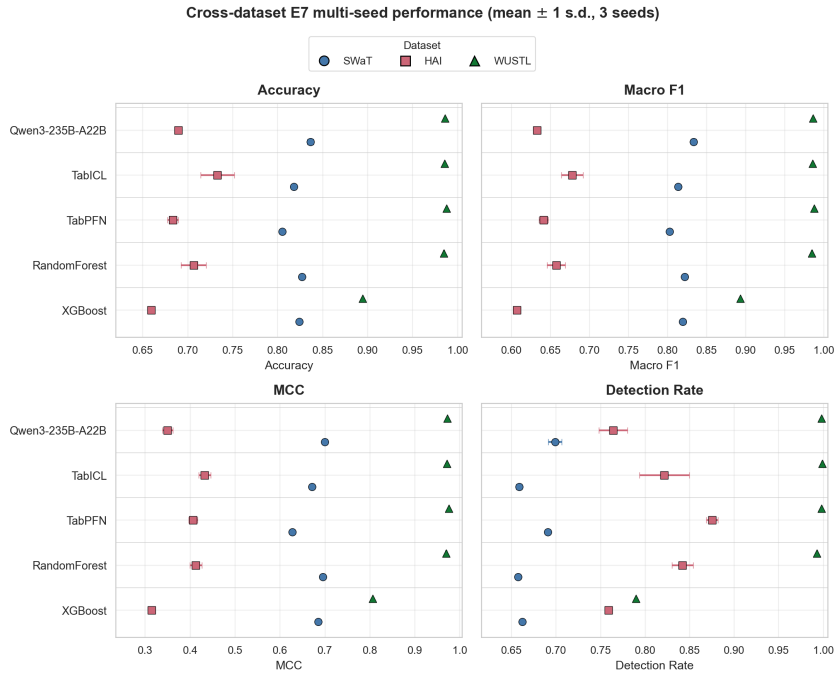

Figure 3. Headline cross-dataset performance of the five detector families, shown as a 2×2 panel (Accuracy, Macro F1, MCC, Detection Rate) for legibility. Markers are the mean over three seeds and horizontal bars are ± 1 standard deviation on the held-out E7 test split; the false-alarm rate retains its dedicated Pareto figure. The dataset-dependent ranking is visible directly: the open-source LLM leads on SWaT, the tabular foundation models lead on HAI and WUSTL, and the classical anchors remain competitive only where attacks are traffic-dense.

The cross-dataset story is the central finding, and the table makes it visible at a glance: the bold cells migrate across model families between dataset blocks. SWaT goes to the LLM, HAI to TabICL, and the WUSTL binary block is a saturation regime in which all four methods agree to within a percentage point of MCC. The gap between best and second on each dataset also tells its own story: SWaT is a narrow LLM-over-RandomForest win (under a percentage point of MCC), HAI is a substantial TabICL-over-LLM lead (more than eight percentage points, with the LLM finishing last among the four methods), and WUSTL is a genuine four-way tie at the ceiling. The implication is not that one method is universally best, but that the operating regime—feature schema, attack prevalence, signature strength—determines which foundation-model family transfers.

This dataset-dependent ordering is the empirical signature of a deeper phenomenon. SWaT, with its physical-signal feature schema and its balanced Kaggle-mirror prevalence, plausibly rewards the LLM's ability to reason over named sensor channels (`LIT`, `FIT`, `P` prefixes) in natural language; we treat this as an interpretive hypothesis rather than an established mechanism, since the present protocol does not include a controlled feature-naming ablation. HAI, with 79 heterogeneous channels from four interconnected subsystems and a natural 3% attack prevalence, rewards the tabular foundation models that excel at exactly this kind of high-dimensional small-anomaly detection. WUSTL, with simple Modbus/TCP flow features and a clean attack signature, saturates every method.

The XGBoost row in Table 4 merits a brief comment. Under the K-shot protocol adopted here, XGBoost trails Random Forest on every dataset, with the gap most pronounced on WUSTL binary (ΔMCC $= -0.164$) and HAI (ΔMCC $= -0.099$). This is a property of the operating regime rather than a general statement about gradient boosting: at $K_{\text{shot}} = 10$ examples per class, sequential tree construction has very few rows from which to fit successive residual stages, while Random Forest's bootstrap ensemble produces

stable averages even at this sample size. With abundant labelled data XGBoost is fully competitive with Random Forest, as the max-context sensitivity analysis in Section 5.8 confirms: there XGBoost reaches MCC $\in [0.944, 1.000]$ across all datasets and is the strongest Backdoor-recall method at max-context. The K-shot result therefore validates Random Forest as the appropriate classical reference point for the data-constrained regime that motivates foundation-model approaches, without implying that gradient boosting is a weaker method in general.

5.2. Paired Significance: LLM Versus Each Anchor

To control for between-sample variance, we evaluate the primary LLM (Qwen3-235B-A22B) and each anchor on the identical $n = 6000$ stratified subsample and conduct a paired McNemar test on the binary correct/incorrect pattern. Table 5 reports the results; Figure 4 renders them as a heat-map of the signed log-significance score

$$S(\Delta, p) \;=\; \mathrm{sign}(\Delta) \cdot \big[-\log_{10}(p)\big], \tag{1}$$

where $\Delta = \Delta\mathrm{acc}$ is the paired accuracy difference (LLM minus anchor) and p is the McNemar two-sided p-value. Positive S indicates the LLM wins; negative S indicates the anchor wins; the magnitude is proportional to the significance.

Table 5. Paired McNemar tests, LLM versus each anchor, on the identical $n = 6000$ E7 subsample. b/c are LLM-only/anchor-only correct counts; $p < 0.05$ marked *.

Dataset	LLM	Anchor	Δacc	*b* (LLM Only)	*c* (Anc Only)	*p*
HAI	Qwen3-235B-A22B	RandomForest	−0.036	253	472	5.66×10^{-16} *
HAI	Qwen3-235B-A22B	TabICL	−0.072	262	696	1.80×10^{-44} *
HAI	Qwen3-235B-A22B	TabPFN	−0.001	428	432	0.919
SWaT	Qwen3-235B-A22B	RandomForest	+0.008	137	87	1.06×10^{-3} *
SWaT	Qwen3-235B-A22B	TabICL	+0.017	228	123	2.84×10^{-8} *
SWaT	Qwen3-235B-A22B	TabPFN	+0.032	364	171	1.03×10^{-16} *
WUSTL	Qwen3-235B-A22B	RandomForest	+0.003	18	2	4.02×10^{-4} *
WUSTL	Qwen3-235B-A22B	TabICL	+0.001	15	8	0.210
WUSTL	Qwen3-235B-A22B	TabPFN	+0.001	20	14	0.391

The pattern is striking and consistent with Section 5.1. Three observations stand out beyond the single-cell direction of each test. First, on SWaT all three paired tests cross the significance threshold in favour of the LLM, with the effect size growing as the anchor weakens (smallest Δacc against RandomForest, largest against TabPFN), indicating that the LLM's SWaT advantage is consistent across anchor families rather than specific to one tabular method. Second, on HAI the picture is the mirror image: the LLM is significantly beaten by both RandomForest and TabICL, and the magnitude of the TabICL gap is the largest single effect in the study at $p < 10^{-44}$; only TabPFN ties the LLM, and does so essentially because both methods fail in similar ways on HAI's small-anomaly regime (see Section 6.2). Third, on WUSTL only the LLM-versus-RandomForest test crosses the significance threshold, and even there, the effect size is small enough that the saturation already observed in the headline table accounts for it; the LLM and the two tabular foundation models are statistically indistinguishable on this benchmark. The three datasets therefore exhibit three different operational regimes: LLM dominance (SWaT), tabular dominance (HAI), and saturation (WUSTL).

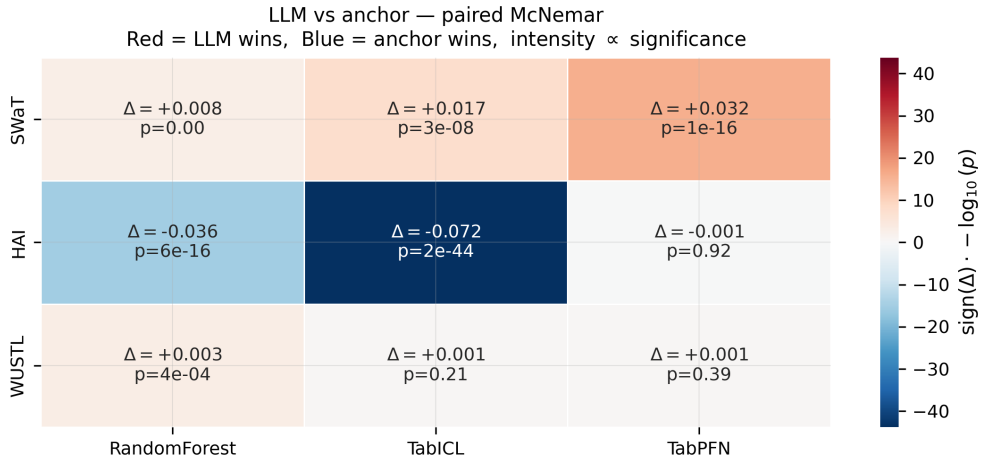

Figure 4. Paired McNemar tests of Qwen3-235B-A22B against each anchor on the identical $n = 6000$ subsample. Cell colour encodes the signed log-significance score $S(\Delta, p)$ of Equation (1): red cells indicate the LLM wins, blue cells the anchor wins, and intensity is proportional to significance. Each cell is labelled with Δacc and the McNemar p-value.

5.3. Cross-LLM Significance Against the Baseline

The single-LLM headline of Section 5.1 could in principle be specific to Qwen3-235B-A22B rather than indicative of the LLM family. To test this, the R1 protocol evaluates all four LLMs against RandomForest across $S = 5$ seeds, applying a Mann–Whitney U test to the seed-level macro-F1 distributions.

Figure 5 reports the result as a four-by-three heat-map (LLM × dataset), with each cell carrying the sign of the delta versus Random Forest, the Mann–Whitney p-value, and the effect size. On SWaT, all four LLMs sit at positive deltas; Qwen3-235B-A22B crosses the $p < 0.05$ threshold (ΔmacroF1 $= +0.062$, $p = 0.032$), with the other three positive but not significant. On HAI, the picture is uniformly unfavourable: every LLM has a negative delta, with Hermes-4-405B ($\Delta = -0.147$, $p = 0.008$) and Hermes-4-70B ($\Delta = -0.324$, $p = 0.008$) crossing the significance threshold; the remaining two LLMs lose by smaller margins that do not pass the $n = 5$ Mann–Whitney threshold. On WUSTL the saturation pattern from Sections 5.1 and 5.2 reappears: every delta is within ± 0.01 of zero and no p-value approaches significance.

The implication is operationally important: the LLM family does not uniformly beat or lose to the tabular foundation models. The best LLM on SWaT does outperform RandomForest with statistical confidence, but no LLM beats Random Forest on HAI, and the picture is a near-tie on saturated WUSTL.

5.4. Multi-Class Evaluation on WUSTL-IIoT-2021

As discussed in Section 3.4, WUSTL-IIoT-2021 is the only dataset in our study that supports a genuine five-class attack taxonomy (Backdoor, CommInj, DoS, Normal, Reconn). Table 6 reports multi-seed accuracy, macro F1, and MCC; Table 7 reports per-class F1 with bootstrap 95% confidence intervals across five seeds.

The headline numbers show TabPFN to be the strongest detector at macroF1 $= 0.924 \pm 0.003$, ahead of RandomForest (0.883 ± 0.003), TabICL (0.862 ± 0.004), and Qwen3-235B-A22B (0.805 ± 0.011). The all-LLM family ranges from 0.785 (Hermes-4-70B) to 0.842 (Qwen3-235B-A22B), trailing every tabular anchor by 4 to 14 percentage points on macro F1.

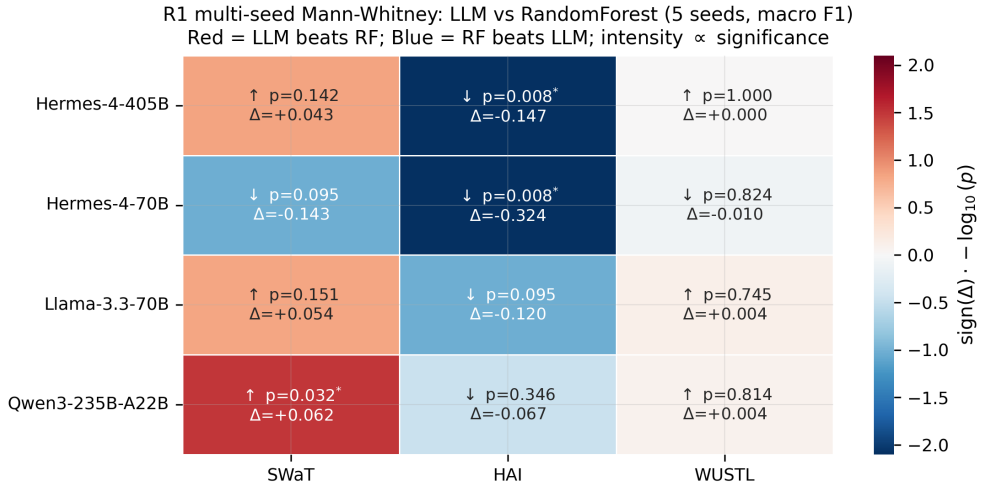

Figure 5. Cross-seed Mann–Whitney comparison of each LLM against the RandomForest baseline (R1 protocol, $S = 5$ seeds per cell). Cells encode the signed log-significance score $S(\Delta, p)$ of Equation (1), with $\Delta = \Delta$macroF1 (LLM minus Random Forest) and p the Mann–Whitney two-sided p-value. Both Δ and p are shown numerically in each cell; the asterisk marks $p < 0.05$.

Table 6. Genuine multi-class evaluation on WUSTL-IIoT-2021 (5 classes: Backdoor, CommInj, DoS, Normal, Reconn). Mean ± std across 3 seeds.

Model	Kind	Accuracy	Macro F1	MCC
TabPFN	Anchor	0.992 ± 0.000	0.924 ± 0.003	0.985 ± 0.000
RandomForest	Anchor	0.988 ± 0.001	0.883 ± 0.003	0.979 ± 0.001
TabICL	Anchor	0.986 ± 0.001	0.862 ± 0.004	0.976 ± 0.002
Qwen3-235B-A22B	LLM	0.981 ± 0.004	0.805 ± 0.011	0.967 ± 0.007

Table 7. Per-class F1 on WUSTL-IIoT-2021 multi-class (bootstrap 95% CIs across 5 seeds; macro is the unweighted class mean). Best per column bolded. XGBoost is shown as point estimates (zero cross-seed std at $K_{\text{shot}} = 10$).

Model	Backdoor	Comminj	Dos	Normal	Reconn	Macro
Hermes-4-405B	0.766 [0.718, 0.799]	0.871 [0.834, 0.899]	0.747 [0.700, 0.791]	0.841 [0.777, 0.900]	0.763 [0.653, 0.874]	0.798
Hermes-4-70B	0.791 [0.764, 0.825]	0.860 [0.828, 0.897]	0.694 [0.633, 0.769]	0.874 [0.829, 0.925]	0.704 [0.625, 0.818]	0.785
Llama-3.3-70B	0.823 [0.786, 0.866]	0.882 [0.853, 0.911]	0.819 [0.742, 0.886]	0.870 [0.813, 0.920]	0.748 [0.606, 0.889]	0.828
Qwen3-235B-A22B	**0.824 [0.778, 0.868]**	0.872 [0.835, 0.896]	0.819 [0.752, 0.898]	0.922 [0.880, 0.965]	0.774 [0.673, 0.889]	0.842
XGBoost	0.650	0.280	0.924	0.917	0.993	0.753
RandomForest	0.812 [0.761, 0.860]	**0.891 [0.868, 0.908]**	**0.974 [0.955, 0.992]**	**0.938 [0.903, 0.967]**	**0.908 [0.859, 0.957]**	**0.904**

The per-class breakdown explains the gap. The radar plot (Figure 6) makes the structure visually unmistakable: RandomForest forms a uniformly strong polygon across all five classes, while every LLM has a pointed shape with vertices that collapse toward the centre on DoS and Reconn. Quantitatively, Random Forest scores 0.974 on DoS, while the LLMs cluster at 0.694 to 0.819, a deficit of 15 to 28 percentage points. The same pattern holds on Reconn: Random Forest at 0.908, LLMs at 0.704 to 0.774. On rarer attack types (Backdoor, CommInj), the LLMs are competitive: Qwen3 matches Random Forest on Backdoor (0.824 vs. 0.812) and is within 2 percentage points on CommInj (0.872 vs. 0.891).

The XGBoost per-class profile on this benchmark deserves a comment. Under the K-shot protocol, XGBoost at $K_{\text{shot}} = 10$ tracks the LLM family on the high-support classes (DoS F1 = 0.924, Normal = 0.917, Reconn = 0.993) but degrades on the rare classes (Backdoor F1 = 0.650, CommInj F1 = 0.280); the macro F1 of 0.753 is the lowest among the evaluated anchors in this regime, 15.1 percentage points below Random Forest. As in the binary case, this is a K-shot effect rather than a general property of gradient boosting: with effectively 50 training rows distributed across 5 classes, sequential tree construction has too few

examples per class to recover the rare-class boundary, whereas Random Forest's bootstrap aggregation provides enough averaging to do so. When the training budget is lifted, the picture inverts decisively: at max-context (Section 5.8) XGBoost reaches F1 $\geq$ 0.977 on every WUSTL class, including the two rare ones, and the rare-class precision–recall analysis there shows XGBoost achieving perfect recall on both Backdoor and CommInj. The choice of Random Forest as the primary classical anchor therefore reflects K-shot stability rather than a verdict on gradient boosting's underlying capability.

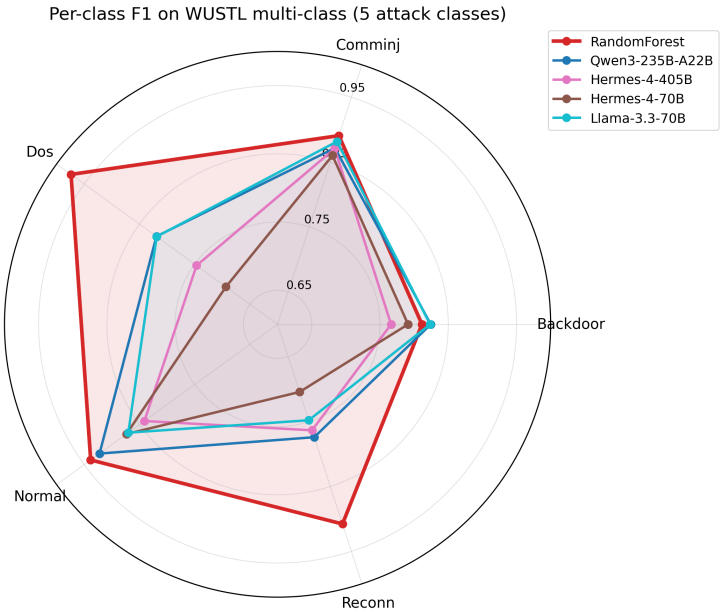

Figure 6. Per-class F1 on WUSTL-IIoT-2021 multi-class (five classes, $S = 5$ seeds). RandomForest forms a uniformly strong polygon; every evaluated LLM is asymmetric with weak vertices at DoS and Reconn.

Figure 7 renders the per-class deltas explicitly, with Mann–Whitney significance stars across the five seeds. The ΔF1 against Random Forest is significantly negative for three of four LLMs on DoS (the exception being Hermes-4-70B, whose p-value sits just above the threshold) and for Hermes-4-405B on Reconn. The smaller deltas on Backdoor, CommInj, and Normal do not pass the significance threshold.

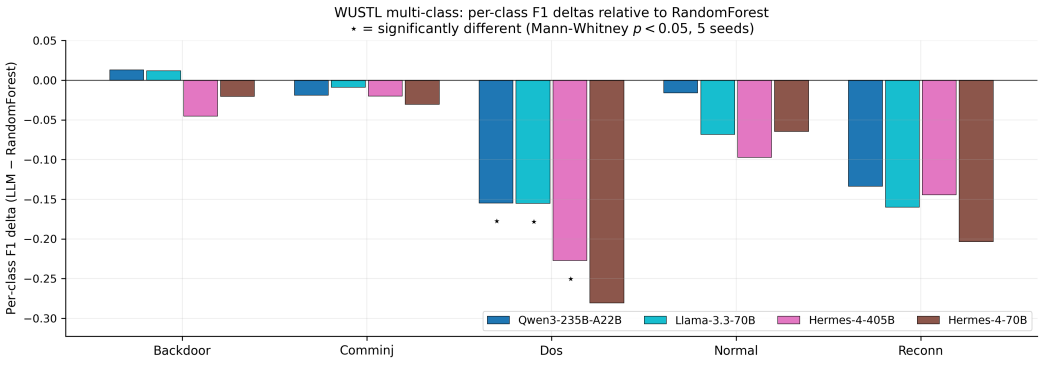

Figure 7. Per-class F1 deltas of each LLM relative to Random Forest on WUSTL-IIoT-2021 multi-class. Significance stars mark Mann–Whitney $p < 0.05$ across five seeds.

5.5. Operational Characterisation

Two operational quantities matter for deployment in industrial cyber-physical security: the false-alarm-rate/detection-rate trade-off (operators care about both) and the inference cost (procurement teams care about both). Figures 8 and 9 report these respectively.

The FAR/DR Pareto plot (Figure 8) characterises each method as a single point in (FAR, DR) space; the Pareto frontier is the lower-right envelope. On SWaT, Qwen3-235B-A22B sits at the upper-left of the frontier (FAR = 0.026, DR = 0.699); RandomForest sits at the lower-left (FAR = 0.003, DR = 0.657); TabPFN sits far off the frontier (FAR = 0.080). On HAI, the LLM moves to the dominated region: TabPFN occupies the upper-right of the frontier (DR = 0.875), while Qwen3 has both higher FAR and lower DR than the entire anchor cluster. WUSTL saturates as before, with all methods within 0.5 percentage points of each other in DR.

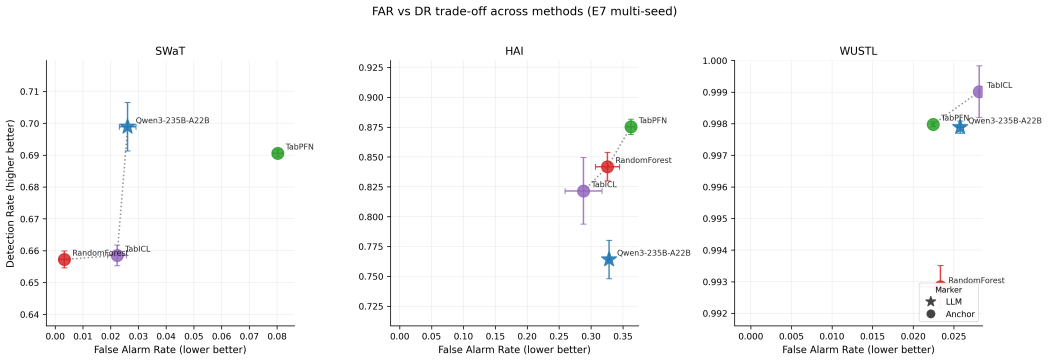

Figure 8. False-alarm-rate versus detection-rate Pareto frontier per dataset. Star marker = LLM, circle = anchor; colour encodes method (red = RandomForest, green = TabPFN, purple = TabICL, blue = Qwen3-235B-A22B), consistent across panels. Error bars are ±1 standard deviation across three seeds.

The cost-per-correct-prediction analysis (Figure 9) reflects the LLM family's heterogeneity. Llama-3.3-70B occupies the lower-left of the Pareto frontier on SWaT (≈8.6×10^{-4} USD per correct sample at macroF1 = 0.836); Hermes-4-405B is dominated everywhere. On WUSTL the macro-F1 saturates at ≈0.99 for every LLM, making cost-per-correct the only meaningful distinction: the smaller Llama-3.3-70B and Hermes-4-70B models are roughly 2× cheaper than Qwen3-235B-A22B and approaching an order of magnitude cheaper than Hermes-4-405B for equivalent accuracy.

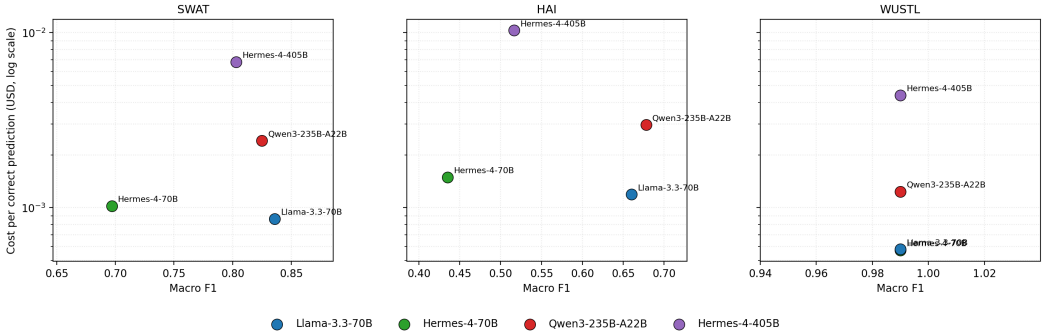

Figure 9. Cost-per-correct-prediction (USD, log scale) versus macro F1 for each LLM, per dataset. Lower-right is better. On WUSTL the macro F1 saturates at ≈0.99 for every LLM; the panel highlights the order-of-magnitude cost differences within the LLM family.

Scope and time validity of the cost analysis.

The cost-per-correct figures reported above are operational estimates under the specific experimental conditions of this study rather than general conclusions. They use the per-token Nebius AI Studio public-tariff snapshot captured in May 2026, summarised in Table 8, and include only inference token charges at the rates listed; they exclude the one-off compute associated with data preprocessing, prompt construction, and response parsing,

which are negligible at the per-sample scale of this study but may become non-negligible in production deployments where preprocessing pipelines dominate. The figures are vendor- and configuration-specific in three further respects: (1) the same models served by an alternative provider, on user-owned hardware, or via batch-discounted APIs would yield materially different per-correct costs; (2) the $n_{\text{vote}} = 1$ greedy-decoding regime used here is the cheapest LLM configuration, and any self-consistency or chain-of-thought variant would scale cost roughly linearly with the vote budget; (3) inference pricing for open-source LLMs has historically fallen by roughly an order of magnitude per year for a given parameter budget, so the absolute USD figures should be read as a fixed-point reference for the May 2026 snapshot rather than as a forward-looking deployment forecast. The Random Forest, TabPFN, and TabICL anchors are excluded from the cost analysis because they run on user-owned compute and incur no per-call API charges under the present protocol; an apples-to-apples cost comparison would require pricing the GPU-hours or CPU-hours consumed by those methods on a chosen cloud SKU, which is itself a vendor- and date-specific exercise. The relative cost ratios between LLMs in the same parameter-size class (e.g., Llama-3.3-70B versus Hermes-4-70B at the same 70 B parameter count) are more stable across pricing revisions than the absolute USD numbers, and the deployment recommendation in Section 6.5 rests on those relative ratios rather than on the absolute cost levels.

Table 8. Per-million-token Nebius AI Studio public-tariff snapshot (May 2026) used for the cost-per-correct analysis. The anchors are excluded (no per-call API charges).

Model	Input (USD/10^6 tok)	Output (USD/10^6 tok)
Llama-3.3-70B	0.13	0.40
Qwen3-235B-A22B	0.20	0.60
Hermes-4-70B	0.13	0.40
Hermes-4-405B	1.00	3.00

Table 9 summarises the computational profile of each detector. The contrast spans several orders of magnitude. The classical anchors and the tabular foundation models are sub-megabyte models with sub-second-per-thousand inference on CPU; the open-source LLMs are served and require on the order of 10^1–10^2 TFLOP per query, with API latency 13–45$\times$ that of the tabular foundation models and four orders of magnitude above the tree anchors. Note that Qwen3-235B-A22B activates only 22 B of its 235 B parameters per token, giving it the lowest per-query FLOPs among the LLMs while retaining the strongest unseen-family generalisation (Section 5.11).

5.6. Robustness Analyses

Two additional analyses confirm that the headline findings are not artefacts of the primary protocol.

The E2 ablation sweeps the number of in-context examples per class $K_{\text{shot}} \in \{5, 10, 25, 50, 100\}$ and reports macro F1 (Figure 10). The picture is consistent across datasets: every method saturates by $K = 50$, with the steepest gain between $K = 5$ and $K = 25$. On HAI the LLM (Qwen3-235B-A22B) is below all three anchors at every K, confirming that the HAI under-performance seen in Section 5.1 is not a low-shot artefact. On SWaT and WUSTL the methods are within noise of each other at $K \geq 25$.

Table 9. Computational profile of each detector family. Parameter counts and FLOPs are not directly comparable across paradigms: tree ensembles are non-parametric, so for the tabular methods we report serialised size, fit and inference wall-clock, and peak memory (measured locally on CPU); for the served LLMs, we report total/active parameters, an approximate $2\,N_{\text{active}}\,\tau$ FLOPs-per-query estimate, and measured API latency. Inference is seconds per 1000 queries (LLMs evaluated with parallel API calls).

Method	Family	Params (Tot./Act.)	Size	Fit (s)	Infer. (s/1k)	FLOPs/Query
RandomForest	tabular	non-param.	0.14 MB	0.56	0.003	negligible
XGBoost	tabular	non-param.	0.16 MB	0.03	<0.001	negligible
TabPFN	tabular	non-param.	0.003 MB	0.42	1.15	negligible
TabICL	tabular	non-param.	0.015 MB	0.26	3.13	negligible
Llama-3.3-70B	LLM	70B/70B	served	0 †	52	∼58.8 TFLOP
Qwen3-235B-A22B	LLM	235B/22B	served	0 †	40	∼18.5 TFLOP
Hermes-4-70B	LLM	70B/70B	served	0 †	32	∼58.8 TFLOP
Hermes-4-405B	LLM	405B/405B	served	0 †	49	∼340.2 TFLOP

† LLMs are used training-free (in-context), so there is no per-dataset fit; they are accessed as a hosted API, hence "served" for on-disk size and peak memory.

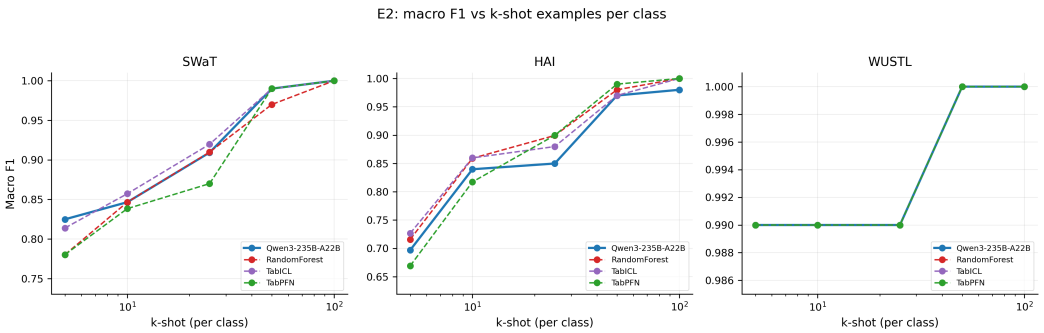

Figure 10. Macro F1 as a function of in-context-example budget per class $K_{\text{shot}} \in \{5, 10, 25, 50, 100\}$, per dataset (E2 protocol).

Figure 11 reports the per-seed MCC values from the E7 protocol as a strip plot with horizontal mean bars and ±1 standard deviation whiskers. Within-method seed-to-seed variance is small relative to between-method differences: the gap between the worst- and best-method means on a given dataset is consistently larger than the within-method spread across seeds, supporting the claim that the cross-method rankings reported above are not driven by seed noise.

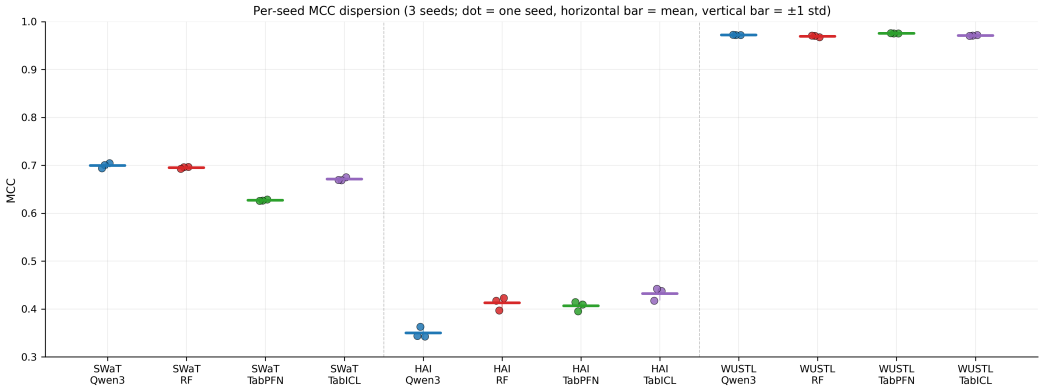

Figure 11. Per-seed MCC dispersion (E7 protocol, three seeds). Each dot is one seed; the horizontal bar marks the mean and the vertical whisker ±1 standard deviation. Cross-method differences exceed within-method seed variance on every dataset.

5.7. Summary of Findings

The empirical picture across the six subsections is consistent:

- **No single method is universally best.** Qwen3-235B-A22B wins SWaT with statistical confidence; TabICL wins HAI; TabPFN wins WUSTL at saturation.
- **LLMs lose to tabular foundation models on HAI by a substantial margin.** The -0.072 accuracy deficit against TabICL on HAI is the largest single-dataset effect in the study and is significant at $p < 10^{-44}$.
- **The WUSTL multi-class deficit is class-specific.** LLMs are competitive on rare attacks (Backdoor, CommInj) and substantially weaker on traffic-rich attacks (DoS, Reconn), losing 13 to 28 percentage points of per-class F1 to Random Forest.
- **Cross-LLM variability matters for cost-per-correct.** Llama-3.3-70B is on the cost/accuracy Pareto frontier on SWaT; Hermes-4-405B is dominated everywhere.

Sections 5.8–5.11 then stress-test these findings under larger training budgets, feature-budget variation, chronological splitting, and unseen attack families; the deployment implications are discussed in Section 6.

5.8. Max-Context Sensitivity Analysis

The K-shot protocol used in Sections 5.1–5.4 is the only operating regime that all eight method families (Hermes-4-70B, Hermes-4-405B, Llama-3.3-70B, Qwen3-235B-A22B, TabPFN, TabICL, RandomForest, XGBoost) can support uniformly, because open-source LLMs have hard context-window limits. The four tabular anchors, however, can ingest substantially more training data at their native operating regimes: TabPFN-v2 supports up to ∼10,000 in-context support rows [25], TabICL supports ∼50,000 or more [26], and classical tree-based methods (RandomForest, XGBoost) scale to arbitrary sample sizes. To probe whether the K-shot ranking transfers to regimes where the tabular methods are not data-constrained, we re-evaluate the four tabular anchors at each method's native training-data budget while keeping the LLM at K-shot (its structural ceiling).

Protocol.

For each dataset, the processed corpus (after preprocessing per Section 3.4 and the 50,000-per-class cap) is partitioned by a stratified 80%/20% split into a training pool and a held-out evaluation set. Two sweeps are run on the same held-out evaluation set so that only the training budget varies between regimes: a K-shot sweep ($K_{\text{shot}} = 10$ examples per class drawn from the training pool, matching the headline protocol) and a max-context sweep at each method's native budget (Table 10). Per-seed variance is reported across three seeds (42, 43, 44). The K-shot MCC values in Table 11 differ from those in Tables 4 and 6 because the two protocols draw the K-shot ICL examples from different populations (the full corpus for the headline protocol vs the 80% training pool for the sensitivity protocol) and use different-sized holdouts (∼100,000 rows vs ∼20,000 rows). The methods and metrics are identical; only the K-shot example selection and the holdout size differ. The largest discrepancies are on HAI, where the natural ∼3% attack prevalence makes the K-shot draw highly sensitive to the specific attack examples chosen, and TabPFN and TabICL pick up an additional 0.12–0.19 MCC under the sensitivity protocol's draw. On other (dataset, method) combinations, the discrepancies are smaller (typically below 0.10 MCC) but non-negligible in a few cells. The two tables are therefore reporting the same methods under matched-but-not-identical K-shot protocols; the gain-from-K-shot-to-max-context delta reported in Table 11 is a within-protocol comparison and is the quantity of interest for the sensitivity analysis.

Table 10. Per-method training budget in the max-context sensitivity sweep. The training pool is the 80% stratified partition of the processed corpus; classical methods use the full pool, tabular foundation models use a stratified subsample at the published native limit. The LLM is not re-run because $K_{\text{shot}} = 10$ is its structural ceiling.

Method	Training Budget
RandomForest	Full training pool (∼80,000 SWaT, ∼50,000 HAI, ∼80,000 WUSTL)
XGBoost	Full training pool (same as RandomForest)
TabPFN	10,000 stratified support rows (v2 native limit)
TabICL	50,000 stratified support rows (native large-context support)

Findings.

Table 11 reports the per-method MCC at both regimes, with the delta computed as $\Delta\text{MCC} = \text{MCC}_{\text{max-ctx}} - \text{MCC}_{\text{K-shot}}$.

Table 11. Max-context sensitivity (3 seeds, mean ± std). ΔMCC is the K-shot-to-max-context gain per (dataset, method). Several rows have zero cross-seed std because the configurations are deterministic.

Dataset	Method	n_{train} (K-Shot)	MCC (K-Shot)	n_{train} (Max-Ctx)	MCC (Max-Ctx)	ΔMCC
HAI	RandomForest	20	0.446 ± 0.024	49,624	0.967 ± 0.000	+0.521
HAI	XGBoost	20	0.313 ± 0.000	49,624	0.944 ± 0.000	+0.632
HAI	TabPFN	20	0.595 ± 0.000	10,000	0.958 ± 0.000	+0.363
HAI	TabICL	20	0.556 ± 0.006	49,624	0.959 ± 0.003	+0.403
SWaT	RandomForest	20	0.649 ± 0.002	80,000	0.998 ± 0.000	+0.350
SWaT	XGBoost	20	0.582 ± 0.000	80,000	0.997 ± 0.000	+0.415
SWaT	TabPFN	20	0.650 ± 0.000	10,000	0.994 ± 0.000	+0.344
SWaT	TabICL	20	0.661 ± 0.000	50,000	0.997 ± 0.000	+0.336
WUSTL (binary)	RandomForest	20	0.970 ± 0.001	80,000	1.000 ± 0.000	+0.029
WUSTL (binary)	XGBoost	20	0.828 ± 0.000	80,000	0.999 ± 0.000	+0.171
WUSTL (binary)	TabPFN	20	0.980 ± 0.000	10,000	0.999 ± 0.000	+0.019
WUSTL (binary)	TabICL	20	0.971 ± 0.002	50,000	1.000 ± 0.000	+0.029
WUSTL (mc)	RandomForest	50	0.909 ± 0.005	86,969	1.000 ± 0.000	+0.090
WUSTL (mc)	XGBoost	50	0.813 ± 0.000	86,969	0.999 ± 0.000	+0.187
WUSTL (mc)	TabPFN	50	0.898 ± 0.000	10,000	0.999 ± 0.000	+0.101
WUSTL (mc)	TabICL	50	0.955 ± 0.005	50,000	1.000 ± 0.000	+0.045

Three observations follow.

First, all sixteen (dataset, method) cells saturate at max-context, with MCC $\in [0.944, 1.000]$ without exception. The K-shot ranking does not transfer to the max-context regime: the methods that were dispersed at K-shot (HAI MCC ranging from 0.31 to 0.60 across the four anchors) converge to indistinguishability when given their native training budget. The foundation-model-versus-classical question is therefore operative only in the K-shot regime that motivates foundation-model approaches in the first place; with abundant labelled data, every classifier saturates and the question becomes moot in the empirical sense.

Second, HAI is the dataset most strongly affected by the K-shot constraint. The apparent difficulty of HAI in Table 4 (where the best K-shot method reaches MCC ≈ 0.43) is almost entirely a K-shot artefact: every tabular method reaches MCC > 0.94 at its native budget. This re-scopes the HAI narrative in Section 5.1: HAI is not intrinsically difficult, but is the dataset where the K-shot constraint imposes the largest performance penalty. By contrast, WUSTL binary is already near-saturation at K-shot for three of the four methods (ΔMCC ≤ 0.029 for RandomForest, TabPFN, TabICL), reflecting that WUSTL features are sufficiently discriminative that twenty labelled examples already suffice.

Third, the per-class rare-class collapse seen at K-shot in Table 12 disappears entirely at max-context. Every method's F1 on Backdoor jumps from below 0.30 at K-shot to above 0.91 at max-context, and every method's F1 on CommInj jumps from below 0.71 at K-shot to

above 0.98. The K-shot rare-class behaviour is therefore a diagnostic of method robustness to label scarcity rather than of intrinsic algorithmic capability.

Table 12. Per-class F1 on WUSTL-IIoT-2021 multi-class, K-shot vs. max-context (3 seeds, mean).

Regime	Method	Backdoor	CommInj	DoS	Normal	Reconn
K-shot	RandomForest	0.049	0.708	0.949	0.985	0.981
K-shot	XGBoost	0.273	0.120	0.883	0.932	0.810
K-shot	TabPFN	0.091	0.233	0.937	0.989	0.900
K-shot	TabICL	0.121	0.622	0.983	0.985	0.981
Max-context	RandomForest	0.968	0.991	1.000	1.000	1.000
Max-context	XGBoost	0.977	1.000	1.000	1.000	1.000
Max-context	TabPFN	0.914	0.981	1.000	0.999	1.000
Max-context	TabICL	0.962	1.000	1.000	1.000	1.000

Rare-class drill-down.

The F1 figures in Table 12 mask a sharper compositional story when precision and recall are separated. Table 13 reports per-class precision and recall at both regimes for the two rare classes (Backdoor, ∼42 test rows; CommInj, ∼52 test rows). The three majority classes (DoS, Normal, Reconn) saturate at recall ≥ 0.99 at both regimes for all methods and are omitted from the table.

Three observations follow. First, the K-shot rare-class detection advantage is distributed across method families rather than concentrated in the tabular foundation models: TabPFN ties with XGBoost for the best Backdoor recall (0.810), RandomForest leads on CommInj recall (0.962), and TabICL trails on Backdoor (0.643). The K-shot ranking is therefore better characterised as "*TabPFN and gradient boosting both excel on Backdoor; bootstrapped ensembles (RF, TabICL) excel on CommInj*" than as a simple tabular-FM win on rare classes.

Second, the K-shot rare-class recall ranking reverses at max-context. XGBoost achieves perfect Backdoor recall (1.000) and TabICL reaches 0.992; TabPFN, despite being the K-shot recall leader on Backdoor, drops to the lowest Backdoor recall of any anchor (0.881). On CommInj, RandomForest, XGBoost, and TabICL all reach recall 1.000, with TabPFN again the only method below saturation (0.981). The reversal is consistent with the general "K-shot rankings do not transfer" finding of Table 11: methods that exploit small-sample priors (TabPFN especially) do so at the cost of an asymptotic plateau, while methods that scale with the training budget (RandomForest, XGBoost, TabICL at its 50,000-row context) keep improving as the support grows.

Third, TabPFN's max-context ceiling is best read as a budget artefact rather than an algorithmic weakness. The TabPFN result reflects in-context inference over a 10,000-row stratified support set, the v2 native maximum [25]. At that support size, the rare-class supports themselves are correspondingly truncated: with the natural prevalence preserved by stratified subsampling, TabPFN sees only ∼6 CommInj support rows and ∼5 Backdoor rows out of the 10,000. The other anchors, operating on training pools 5–9× larger, see proportionally more rare-class examples and accordingly resolve them more completely. The rare-class gap on TabPFN at max-context, therefore, reflects a property of the v2 context-size budget on heavily imbalanced datasets, rather than a limitation of in-context tabular learning per se; subsequent generations of TabPFN with larger context support, or balanced rather than stratified subsampling, would be expected to close this gap.

Implication for the paper's central claim.

The sensitivity analysis sharpens rather than weakens the paper's framing. The K-shot regime is precisely the deployment scenario where foundation-model methods are useful in practice: large-scale labelled attack data is structurally scarce in OT/ICS environments, and the labelled examples that do exist are typically limited to a handful per attack type for novel campaigns.

Table 13. Rare-class precision and recall on WUSTL-IIoT-2021 multi-class, K-shot vs. max-context (3 seeds, mean). Majority classes (recall ≥ 0.99) are omitted; leader per metric within each regime block bolded.

		Backdoor		CommInj	
Regime	**Method**	**Precision**	**Recall**	**Precision**	**Recall**
K-shot	RandomForest	0.025	0.643	**0.560**	**0.962**
K-shot	XGBoost	**0.164**	**0.810**	0.064	0.923
K-shot	TabPFN	0.048	**0.810**	0.133	0.942
K-shot	TabICL	0.067	0.643	0.470	0.923
Max-context	RandomForest	**0.976**	0.960	0.981	**1.000**
Max-context	XGBoost	0.955	**1.000**	**1.000**	**1.000**
Max-context	TabPFN	0.949	0.881	0.981	0.981
Max-context	TabICL	0.933	0.992	**1.000**	**1.000**

5.9. Sensitivity to the Number of Selected Features

The headline protocol fixes the mutual-information feature budget at $K_{\text{feat}} = 12$ (Section 3.4). To verify this choice does not advantage any family, we sweep $K_{\text{feat}} \in \{6, 8, 10, 12, 16, 20, \text{all}\}$ for every detector on all three datasets (Figure 12; mean MCC at $K_{\text{feat}} = 12$ in Table 14). Three observations follow. First, the ranking of detector families is stable across K_{feat} on every dataset, so the paper's conclusions are not an artefact of the feature budget. Second, $K_{\text{feat}} = 12$ is a near-optimal compromise for the foundation models and the LLM: on SWaT and WUSTL, each is within $\leq$0.01 MCC of its own best K_{feat}, and the tabular foundation models are remarkably flat in K_{feat}. Third, the main sensitivities are confined to the classical anchors and to HAI: XGBoost benefits from a larger budget on WUSTL (+0.13 MCC from $K_{\text{feat}} = 12$ to $K_{\text{feat}} = 20$), and on HAI, where the discriminative signal is spread across more channels, all methods improve with additional features (e.g., Random Forest +0.05, TabPFN +0.09 beyond $K_{\text{feat}} = 12$). We accordingly retain $K_{\text{feat}} = 12$ for the cross-method headline and note HAI as the dataset where a larger budget is beneficial. This is consistent with the max-context finding of Section 5.8: HAI is the dataset most penalised by the data-constrained regime, whether the constraint is on examples or on features.

Table 14. Mean MCC at the headline budget $K_{\text{feat}} = 12$ (3 seeds). Rankings are stable across the full sweep (Figure 12); best per row bolded. The largest gap to a method's own best-K_{feat} is XGBoost on WUSTL.

Dataset	**RandomForest**	**XGBoost**	**TabPFN**	**TabICL**	**Qwen3-235B-A22B**
SWaT	0.696	0.651	0.682	0.682	**0.724**
HAI	0.531	0.459	0.532	**0.609**	0.373
WUSTL	0.971	0.845	**0.977**	0.973	0.973

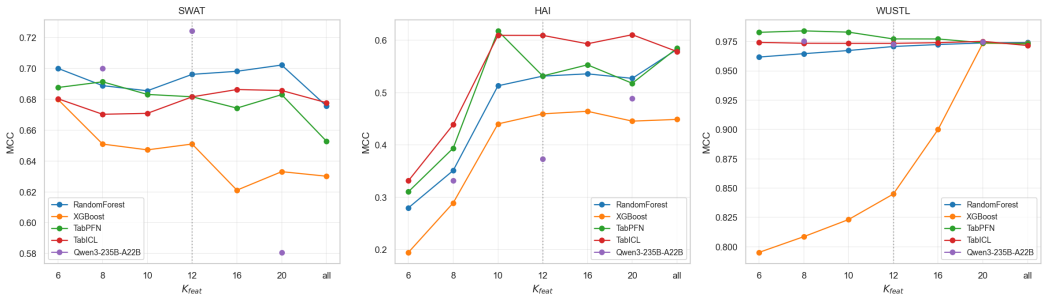

Figure 12. Feature-count ablation: mean MCC versus the number of mutual-information-selected features K_{feat}, one panel per dataset, with the headline $K_{\text{feat}} = 12$ marked. Tabular foundation models are nearly flat in K_{feat}; XGBoost is the most K_{feat}-sensitive anchor; HAI benefits from larger budgets across all methods.

5.10. Chronological-Split Sensitivity Analysis

The evaluation protocol of Section 4 uses a stratified random 80%/20% split, with the methodological rationale given in Section 3.4. To quantify the effect of this choice on the headline findings, we re-run the E7 protocol on SWaT and HAI under a chronological 80%/20% split: the first 80% of timestamps form the training pool (from which the $K_{\text{shot}} = 10$ ICL examples per class are drawn) and the last 20% form the held-out test set. The K-shot ICL budget and every method configuration are identical to those of Section 5.1; only the split varies. WUSTL-IIoT-2021 is not re-run because its flow records are temporally independent units (each row is a discrete network flow rather than a sample of an evolving physical state), so a chronological split on WUSTL collapses to a random split in expectation.

Table 15. Chronological-split binary E7 evaluation (3 seeds, mean ± std). Anchor holdouts $n = 20{,}000$ (SWaT), $n = 12{,}406$ (HAI); LLM on $n = 6000$. Best per dataset on Acc, Macro F1, MCC bolded (HAI also FAR/DR). Compare with Table 4.

Dataset	Model	Acc.	Macro F1	MCC	FAR	DR
SWaT	TabPFN	**0.989 ± 0.003**	**0.989 ± 0.003**	**0.978 ± 0.007**	0.012 ± 0.006	0.990 ± 0.002
SWaT	Qwen3-235B-A22B	0.986 ± 0.001	0.986 ± 0.001	0.972 ± 0.003	0.015 ± 0.003	0.987 ± 0.002
SWaT	RandomForest	0.944 ± 0.016	0.944 ± 0.016	0.893 ± 0.028	0.106 ± 0.033	0.995 ± 0.002
SWaT	TabICL	0.914 ± 0.022	0.913 ± 0.023	0.841 ± 0.039	0.000 ± 0.000	0.828 ± 0.044
HAI	RandomForest	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**0.000 ± 0.000**	**1.000 ± 0.000**
HAI	TabPFN	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**0.000 ± 0.000**	**1.000 ± 0.000**
HAI	Qwen3-235B-A22B	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**0.000 ± 0.000**	**1.000 ± 0.000**
HAI	TabICL	0.997 ± 0.003	0.995 ± 0.005	0.989 ± 0.009	0.004 ± 0.004	1.000 ± 0.000

Three observations follow from Table 15.

First, the direction of the chronological-versus-random effect is the opposite of what the deployment-drift framing would anticipate. On both SWaT and HAI, every method achieves substantially higher MCC under chronological splitting than under random splitting. On SWaT the MCC range moves from $[0.63, 0.71]$ (random, Table 4) to $[0.84, 0.98]$ (chronological); on HAI from $[0.35, 0.43]$ to $[0.99, 1.00]$. The chronological holdouts on these two testbeds are therefore easier test sets than the random holdouts, not harder ones. The mechanism is the scripted-attack structure of SCADA testbeds: each attack scenario contributes tens to hundreds of seconds of physically-correlated samples, and the chronologically-late holdout in each case contains repeated instances of attack scenarios whose dynamics are already represented (under a stratified-by-class draw) in the K-shot ICL examples drawn from the earlier training window. The chronological protocol thus models within-deployment-run detection, where the detector encounters the same scenario types it was prompted on, rather than cross-campaign generalisation, where it would face novel attack repertoires.

Second, the cross-method ranking changes between protocols. On SWaT, TabPFN moves from the lowest-MCC anchor under random splitting (MCC = 0.627) to the highest under chronological splitting (MCC = 0.978); the paired McNemar test on the identical $n = 6000$ subsample confirms a small but significant TabPFN advantage over Qwen3-235B-A22B ($\Delta\text{acc} = -0.006$, $p = 3.6 \times 10^{-3}$ in the LLM-versus-anchor direction). Qwen3-235B-A22B remains the strongest LLM and continues to beat RandomForest ($\Delta\text{acc} = +0.059$, $p < 10^{-58}$) and TabICL ($\Delta\text{acc} = +0.041$, $p < 10^{-31}$) under chronological splitting; the SWaT LLM-over-classical finding is therefore preserved, but the tabular-foundation-model winner identity flips. On HAI, three methods saturate at MCC = 1.0 across all three seeds, and the random-split TabICL lead is not recoverable from the chronological data; the ranking question collapses.

Third, R1-style multi-LLM analysis under chronological splitting sharpens rather than overturns the random-split LLM-family picture. On HAI chronological, two LLMs (Llama-

3.3-70B and Qwen3-235B-A22B) are statistically indistinguishable from RandomForest at the seed-level Mann–Whitney test ($p = 0.91$ and $p = 0.67$ respectively), while the two Hermes variants are significantly worse ($p = 0.012$ each). This corroborates the random-split finding that the LLM family is heterogeneous on HAI and that the distinction between strong and weak LLMs matters more than the binary LLM-vs-tabular split.

The implication for the paper's central claim is two-sided. The qualitative conclusion that no single method dominates across SWaT, HAI, and WUSTL is preserved under chronological splitting, and the conclusion that the LLM family is competitive with tabular foundation models on SWaT is reinforced. The specific random-split dataset-level winners on HAI (TabICL) and SWaT (Qwen3-235B-A22B), however, do not transfer to the chronological protocol unchanged. We read this as evidence that the choice of best detector depends on the assumed deployment regime (within-run detection vs cross-campaign generalisation) at least as much as on the algorithmic family, and that both protocols should be reported when feasible.

5.11. Generalisation to Unseen Attack Families

The chronological holdout of Section 5.10 measures within-run detection: the late window repeats attack scenarios already present earlier in the same scripted capture. The deployment-relevant question raised by the random-versus-chronological discussion is different: how a detector behaves on an attack family it has never observed at training time. To probe this directly we run a leave-one-attack-type-out protocol on WUSTL-IIoT-2021, the only dataset with a genuine multi-class taxonomy. For each of the four attack families in turn, that family is removed entirely from the in-context support set; the model is then tested on Normal traffic plus the held-out family, and we report the recall on the unseen family (its detection rate), averaged over three seeds. This is a within-WUSTL probe of unseen-repertoire generalisation; a genuine cross-campaign evaluation across distinct recording campaigns remains future work (Section 6.7).

The contrast with the within-run picture is stark (Table 16, Figure 13). The classical anchors, which were competitive within-run, degrade sharply on unseen families: XGBoost detects only 17.8% of unseen-family instances on average and misses the family almost entirely (recall < 0.20) in 9 of 12 family-by-seed runs, and the Random Forest reaches only 45.4%. The tabular foundation models and the LLM generalise far better, TabPFN 64.1%, TabICL 79.2%, and the LLM 79.5% mean unseen-family recall, with the LLM never suffering a catastrophic miss (0/12). The same ordering holds on the held-out-family MCC (XGBoost 0.153, RandomForest 0.479, TabPFN 0.631, TabICL 0.740, LLM 0.750) at comparable false-alarm rates (all within 0.017–0.098). The gap between detector families, therefore, widens in the unseen-repertoire regime relative to the within-run results, which is the regime that matters when novel campaigns appear in the field. This substantiates the caution, already noted in Section 5.10, that the dataset-level winners reported under random splitting should be read as within-run estimates.

Table 16. Generalisation to unseen attack families on WUSTL-IIoT-2021: recall on the held-out family (mean over 3 seeds) under leave-one-attack-type-out. Higher is better; the bottom row is the mean across the four held-out families; best per row bolded. Foundation models and the LLM retain detection of families unseen at training time, whereas the classical anchors collapse.

Held-Out Family	RandomForest	XGBoost	TabPFN	TabICL	Qwen3-235B-A22B
Backdoor	0.668	0.064	0.664	0.640	**0.893**
CommInj	0.183	0.145	**0.970**	0.964	0.716
DoS	0.587	0.228	0.403	**0.754**	0.595
Reconn	0.380	0.275	0.528	0.809	**0.978**
Mean	0.454	0.178	0.641	0.792	**0.795**

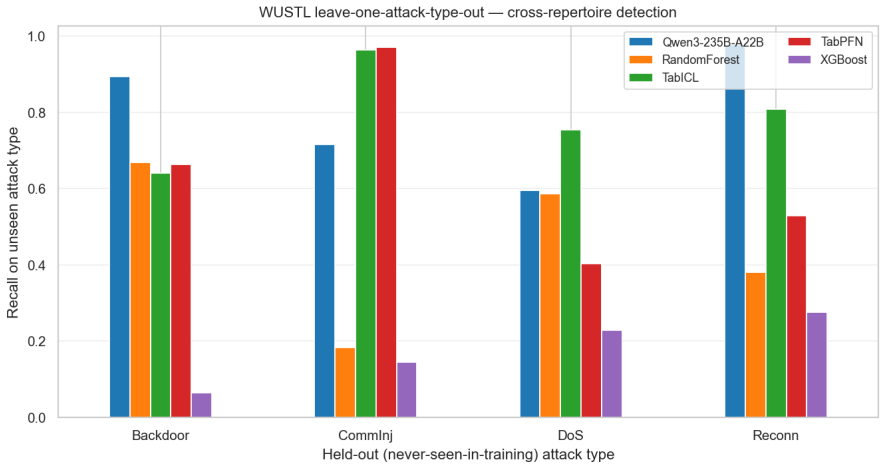

Figure 13. Leave-one-attack-type-out recall on WUSTL-IIoT-2021 per held-out family and detector. Classical anchors (especially XGBoost) frequently fail to detect a family they never saw in training, while the tabular foundation models and the LLM retain substantial recall. Mean over three seeds.

6. Discussion

The results in Section 5 establish that no single method is universally best across the three OT/ICS intrusion-detection benchmarks evaluated. This section interprets that finding. Section 6.1 reads the dataset-dependence pattern through the lens of feature schema, prevalence, and attack type. Sections 6.2 and 6.3 examine the two strongest divergences between LLMs and tabular foundation models, and Section 6.4 turns the resulting complementarity into a deployable confidence-gated hybrid. Section 6.5 translates the empirical picture into deployment recommendations. Section 6.6 surfaces the limitations that scope these recommendations, and Section 6.7 outlines the follow-on research directions opened by this study.

6.1. Interpreting the Dataset-Dependence Pattern

Three structural differences align with the observed method ranking. SWaT presents a 51-dimensional schema with semantically named physical-process sensors (`LIT`, `FIT`, `P` prefixes) and balanced 50% prevalence: the LLM can interpret `LIT-101 = 0.42` as a low water level rather than an opaque numerical fact and apply commonsense process knowledge. HAI presents 79 heterogeneous channels with numerical identifiers and natural 3% prevalence: the LLM's semantic advantage is defeated and tabular foundation models, with pre-training distributions that mirror the class-imbalance regime, become preferable. WUSTL-IIoT-2021 presents low-redundancy flow features with a strong attack signature, saturating every modern classifier and reducing cross-method differences to sample-level noise.

6.2. Why Tabular Foundation Models Dominate HAI

The HAI result, TabICL at MCC $= 0.432$ versus Qwen3-235B-A22B at MCC $= 0.350$, with the LLM significantly beaten on a paired McNemar test at $p = 1.8 \times 10^{-44}$, is the largest cross-family effect in the study and deserves careful unpacking. Three factors plausibly compound. First, the natural 3% attack prevalence on HAI means that the LLM's stratified in-context-example budget of 10 per class draws non-attack examples from a distribution 30$\times$ broader than its attack examples; the LLM consequently sees a biased view of the deployment-time noise floor relative to TabICL, which trains on the full 80% training partition. Second, the HAI feature schema is built from numerical channel identifiers without semantic names, defeating the LLM's most distinctive advantage relative to a purely numerical classifier. Third, TabICL's specific design for larger sample regimes

is well-matched to HAI's holdout of 62,010 samples and to the cross-subsystem feature correlations that the column-then-row tokenisation appears to capture effectively.

The in-context-example scaling analysis in Section 5.6 confirms that this gap is not an artefact of the $K_{\text{shot}} = 10$ choice. Even at $K = 100$, the LLM remains below the anchor cluster on HAI. The implication is that for the class of OT/ICS workloads exemplified by HAI, multi-subsystem, naturally imbalanced, with numerically-coded channels, a modern tabular foundation model is the appropriate default and an LLM-based detector should be treated as a supplementary signal rather than as a replacement.

The size of the TabICL-over-LLM gap is statistically robust but does not imply that HAI is solved: even the strongest K-shot method reaches only MCC ≈ 0.43. As Section 5.8 shows, this headroom closes given abundant labelled data, but in the data-constrained regime operative for OT/ICS deployment, no evaluated method, LLM or tabular, attains deployment-grade accuracy on HAI-like multi-process testbeds.

6.3. Why LLMs Trail on Traffic-Rich Attacks

The WUSTL multi-class per-class analysis (Figure 7) shows a structured deficit: every evaluated LLM is 13–28 percentage points weaker than Random Forest on DoS and Reconn, while being statistically indistinguishable on Backdoor and CommInj. This is a class-specific phenomenon and points to a specific weakness of the LLM-as-classifier paradigm.

DoS and Reconn attacks present a distinctive statistical signature in the WUSTL flow-feature schema: high packet counts (`TotPkts`), inflated source-to-destination byte ratios (`SrcBytes/DstBytes`), elevated flow rates, and characteristic flag patterns. A Random Forest, which is, after all, an ensemble of threshold-and-split decisions over numerical features, is structurally well-suited to learning these density signatures from training data. An LLM, presented with the same numerical features as part of a prompt-formatted in-context example, must implicitly infer the same thresholding logic from natural-language reasoning over the example distribution. The pre-training of an LLM does not specifically reward this skill, and the experimental result here is consistent with the intuition that LLMs are operationally weak at recognising statistical density patterns relative to specialised classifiers.

By contrast, Backdoor and CommInj attacks present subtler signatures that are not concentrated in obvious traffic-density features. On these attack types the LLM's ability to perform pattern matching over the entire feature vector, including categorical fields such as port numbers, transport protocols, and flag combinations, becomes competitive with the Random Forest's threshold-based decomposition. This explanation is consistent with the observation that the within-LLM-family variance on Backdoor and CommInj (± 0.06 F1 across the four LLMs) is smaller than the variance on DoS and Reconn (± 0.13 F1).

6.4. From Diagnosis to Deployment: Confidence-Gated Hybrid Detection

The complementary error profiles established above, classical ensembles excel on traffic-dense attacks through threshold-split density estimation, whereas the LLM is competitive on rare, categorical attack patterns, suggesting a hybrid that routes each decision to the better-suited detector. Crucially, such a router must be deployable: it cannot consult the ground-truth label. We therefore evaluate two label-free policies on the WUSTL five-class task, with the Random Forest as the primary on-device detector and the open-source LLM (Qwen3-235B-A22B) as the escalation target, averaged over three seeds.

Predicted-class router. The LLM is queried only when the tabular model predicts a rare class (Backdoor, CommInj, Reconn). This raises the F1 on the hardest class, Backdoor, from 0.845 (Random Forest alone) to 0.855 while leaving the overall macro F1 essentially unchanged (0.938) and querying the LLM on only 27.4% of samples.

Confidence-gated cascade. The LLM is queried only when the tabular model's maximum class probability falls below a threshold τ; sweeping τ traces the accuracy–cost frontier (Figure 14). At $\tau = 0.5$ the cascade escalates just 6.3% of decisions yet improves the mean macro F1 from 0.938 (Random Forest) and 0.934 (LLM alone) to **0.945**, the MCC from 0.939/0.934 to **0.946**, and the Backdoor F1 to **0.868**. The cascade thus dominates either standalone detector at a small, tunable query budget, turning the diagnostic finding of Section 6.3 into a concrete deployment recipe: keep a cheap tabular model on the wire and escalate only its low-confidence decisions to a foundation model.

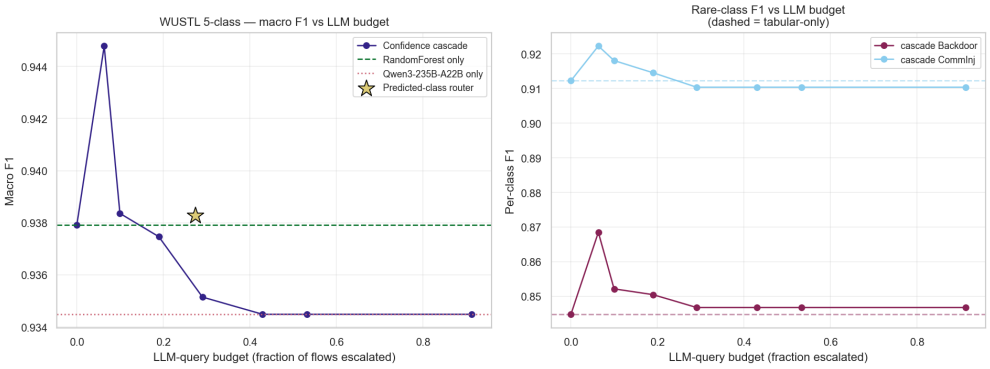

Figure 14. Confidence-gated cascade on WUSTL (5-class). Sweeping the escalation threshold τ trades the fraction of decisions sent to the LLM (x-axis) against macro F1 (y-axis). $\tau = 0$ is the Random Forest alone and large τ approaches the LLM alone; the knee near $\tau = 0.5$ ($\sim$6% escalation) exceeds both standalone detectors. Mean over three seeds.

6.5. Deployment Recommendations

The empirical picture above suggests the following deployment patterns for industrial cyber-physical security teams.

Default to a tabular foundation model.

TabPFN and TabICL offer competitive or superior performance on every dataset evaluated, are cheap to deploy (CPU inference, no API fees), and do not require labelled prompt-format conversion. For new OT/ICS deployments, the appropriate default detector is a tabular foundation model rather than an LLM or a hand-engineered classical classifier.

Add an LLM-based detector where semantic feature interpretation helps.

Deployments where the feature schema carries explicit process-engineering semantics (such as the SWaT-style `LIT`, `FIT`, `P` channels) can benefit from an LLM as a complementary detector. The R1 result demonstrates that Qwen3-235B-A22B beats Random Forest on SWaT with statistical confidence; the cost analysis in Section 5.5 shows that Llama-3.3-70B provides this advantage at substantially lower cost-per-correct prediction than the larger LLMs evaluated, ranging from roughly 3$\times$ cheaper than Qwen3-235B-A22B to approaching an order of magnitude cheaper than Hermes-4-405B under the pricing snapshot of Table 8.

Do not rely on an LLM alone for traffic-rich attacks.

On the WUSTL multi-class evaluation, the per-class deficit on DoS and Reconn is large and consistent across LLMs. A defence-in-depth deployment should layer an LLM detector behind, not in place of, a classical flow-feature classifier for these attack classes.

Escalate only low-confidence decisions.

Where an LLM detector is added, the confidence-gated cascade of Section 6.4 captures most of its benefit at a fraction of the cost: escalating only the tabular model's

low-confidence decisions (∼6% of samples at $\tau = 0.5$) exceeded both standalone detectors on the WUSTL five-class task. This keeps a cheap tabular model on the wire and reserves the LLM's per-query cost (Table 9) for the decisions that need it.

Read FAR and DR together, not just accuracy.

The Pareto frontier in Figure 8 shows that two methods with near-identical accuracy can occupy substantially different operating points. Security Operations Centre (SOC) teams should specify the tolerable false-alarm rate first and pick the highest-detection-rate method that meets it, rather than the highest-MCC method overall.

6.6. Limitations

Several limitations scope the empirical claims of this study.

Public-mirror data artefacts.

The three datasets used in this study are accessed through their public Kaggle mirrors rather than through their original distributions. This choice was made for reproducibility (the iTrust SWaT distribution is restricted-access and was not available to us at the time of writing; the HAI distribution is hosted on a Korean institutional repository with locale-specific access patterns), but it carries methodological consequences that scope the interpretation of every result reported in Section 5. First, the SWaT Kaggle mirror is rebalanced to 50/50 `Normal/Attack` prevalence, against a natural iTrust prevalence of approximately 12% attack; reported SWaT accuracies therefore overstate the operational accuracy a detector would achieve on the natural distribution, and the SWaT false-alarm rate cannot be directly compared with studies that use the original distribution. Second, the SWaT and HAI mirrors strip the per-attack-type labels present in the iTrust and NSRI distributions, which is why the only genuine multi-class evaluation in this paper is the one on WUSTL-IIoT-2021. The conclusions about per-class behaviour, specifically the LLM deficit on traffic-rich attacks and the LLM competitiveness on rare attacks, are therefore demonstrated on WUSTL alone and should not be assumed to transfer to per-class breakdowns on SWaT or HAI without a follow-on evaluation on those datasets' original distributions. The natural-distribution evaluation on the original iTrust and NSRI releases identified as future work in Section 6.7 would close this gap directly.

LLM calibration.

Open-source LLMs of the type evaluated here do not provide calibrated class-probability outputs; the only LLM output is the predicted label. As a consequence, the LLMs are excluded from the AUROC/AUPRC comparison reported in Section 5.5 and cannot be operated at non-default decision thresholds. This is a structural limitation of the LLM-as-classifier paradigm that the present study does not attempt to overcome through token-probability proxies or chain-of-thought confidence estimates.

Multi-testing correction.

The McNemar tests in Section 5.2 comprise $3 \times 3 = 9$ comparisons, and the per-class WUSTL test in Section 5.4 comprises $4 \times 5 = 20$ comparisons. No multiple-testing correction (Bonferroni, Holm–Bonferroni, Benjamini–Hochberg) is applied. We report the raw p-values, with the explicit interpretation that significant findings should be treated as supporting evidence within the broader cross-dataset, multi-seed protocol rather than as confirmatory tests against a strict family-wise error-rate threshold.

Seed budget.

The primary E7 protocol uses $S = 3$ seeds per cell; the R1 and R2 protocols use $S = 5$. The $S = 3$ budget is sufficient to expose between-method effects of the size observed (cross-method differences exceed within-method standard deviations on every dataset, as Section 5.6 shows), but a larger seed budget would tighten the cross-seed confidence intervals further.

LLM model selection.

The four LLMs evaluated here are open-source, accessible via a single inference API, and span the 70 B–405 B parameter range. The study does not evaluate proprietary commercial LLMs (e.g., GPT-4 family, Claude family, Gemini family) for cost and reproducibility reasons, nor does it cover smaller models below 70 B parameters. The conclusions should be read as applying to the modern open-source frontier rather than to the LLM family at large.

Random splitting and the chronological-split robustness check.

The primary protocol uses a stratified random 80%/20% split rather than a strictly chronological one; the rationale is given in Section 3.4. Section 5.10 re-runs the E7 protocol on SWaT and HAI under a chronological split and finds that, on these scripted-attack testbeds, the late-window holdout is in fact easier than the random one, because it repeats attack scenarios already represented in the earlier-window ICL examples; the qualitative cross-method conclusions are preserved, though the SWaT and HAI dataset-level winners change. We therefore read the random-split numbers as estimates of within-run detection accuracy. The leave-one-attack-type-out probe of Section 5.11 provides a first measure of generalisation to attack families unseen at training time within WUSTL; genuine cross-campaign generalisation across distinct recording campaigns remains future work (Section 6.7).

No real-world operational validation.

All three datasets used in this study are released laboratory testbeds (SWaT at iTrust, HAI at NSRI, WUSTL-IIoT-2021 at Washington University) running scripted attack scenarios over recording windows of days to weeks. The detection task as formulated here is offline classification of pre-recorded telemetry rather than live deployment inside a Security Operations Centre. Several operational concerns are therefore outside the scope of the present evaluation: concept drift under long-running deployment, integration latency with SCADA control loops and SIEM platforms, analyst-in-the-loop workflow design, adversarial robustness against attackers aware of the detector, explainability requirements for regulated industries, and the ground-truth labelling pipeline for novel attacks observed in production. A field study on an operating critical-infrastructure plant remains the natural validation step beyond the laboratory benchmarks reported here.

K-shot constraint as a deliberate design choice.

The E7 protocol fixes $K_{\text{shot}} = 10$ in-context examples per class to enable uniform comparison across all eight method families, since open-source LLMs have hard context-window limits. Section 5.8 quantifies the cost of this constraint and shows that all four tabular anchors saturate (MCC > 0.94) at their native training budgets, so the headline ranking characterises the data-constrained regime that motivates foundation-model approaches rather than the asymptotic limit; the latter is reported in Table 11.

6.7. Future Work

The findings of this study point to the following follow-on directions.

- Learned attack-type routing. The confidence-gated cascade of Section 6.4 escalates low-confidence tabular decisions to the LLM using only a scalar confidence threshold. A natural extension is a learned routing head, a lightweight pre-classifier over the same features that predicts which detector family will handle each sample best, which could exploit the per-attack-type complementarity (Section 5.4) more directly than a single global threshold.
- Soft-probability recovery for LLM detectors. Open-source LLMs return only a predicted label, leaving them out of the AUROC and AUPRC comparisons of Section 5.5. Recent calibration techniques that read token-level log-probabilities of the predicted-label token would recover a soft-probability proxy and enable threshold-tunable deployment.
- Cross-dataset transfer. Training on SWaT and deploying on HAI (or vice versa) would address the long-standing OT/ICS-IDS generalisation question on a modern foundation-model substrate, and would clarify whether the dataset-dependence pattern reported here is itself transferable.
- Natural-distribution evaluation on the original iTrust and NSRI releases. Re-running the protocol on the original SWaT and HAI distributions (rather than their Kaggle mirrors) would close the dataset-artefact loop flagged in Section 6.6 and would also expose the multi-class behaviour of the tabular foundation models on those two testbeds.
- Cross-campaign generalisation. The chronological-split analysis (Section 5.10) probes within-run detection, and the leave-one-attack-type-out analysis (Section 5.11) gives a first within-WUSTL measure of generalisation to attack families unseen at training time. Neither is a genuine cross-campaign evaluation, in which the training and test partitions correspond to distinct recording campaigns with potentially non-overlapping attack repertoires; that would require either the original iTrust/NSRI distributions with their campaign-level metadata or purpose-built multi-campaign OT/ICS corpora, and is a natural follow-on study.
- Feature-naming ablation on SWaT. The LLM advantage on SWaT (Section 5.1) is consistent with the hypothesis that semantically named sensor channels (`LIT`, `FIT`, `P` prefixes) let the LLM draw on process-engineering priors from pre-training, but the present protocol does not directly test this. A controlled comparison between the original SWaT prompt and an anonymised variant in which feature names are replaced with neutral identifiers (`F1`, `F2`, …, `F51`) under a generic binary-classifier system message would isolate the contribution of feature semantics to the LLM's SWaT advantage and clarify whether the effect operates primarily through detection rate or through false-alarm rate.

7. Conclusions

This paper has presented a systematic head-to-head comparison of open-source LLMs and modern tabular foundation models (TabPFN, TabICL) on three established OT/ICS intrusion-detection benchmarks (SWaT, HAI, WUSTL-IIoT-2021), under a controlled multi-seed full-holdout protocol with paired McNemar and cross-seed Mann–Whitney testing, a five-class attack-taxonomy analysis on WUSTL-IIoT-2021, a cost-per-correct-prediction analysis, and a false-alarm-rate/detection-rate Pareto characterisation.

The empirical picture is unambiguously dataset-dependent. Qwen3-235B-A22B significantly outperforms every tabular anchor on SWaT; TabICL is the strongest detector on HAI by a substantial and statistically significant margin; TabPFN is the strongest detector on the WUSTL-IIoT-2021 five-class taxonomy. The per-class WUSTL analysis exposes a structural weakness of the LLM family on traffic-rich attacks (Denial-of-Service, Reconnaissance), with competitiveness preserved on rare attack types (Backdoor, Command Injection).

The central message of this study is straightforward. Tabular foundation models, TabPFN and TabICL in particular, provide a highly competitive baseline for industrial cybersecurity tasks framed as tabular classification, and should be the default starting point on new OT/ICS deployments. Open-source LLMs do not generally supersede these models, but they remain valuable in specific scenarios, notably when the feature schema carries process-engineering semantics that the LLM can interpret in natural language, or when the attack-type mix favours the LLM's pattern-matching strengths over the tabular model's density-estimation strengths. The appropriate detector choice in industrial cyber-physical security is therefore a function of the operational domain, the underlying signal type (physical-process telemetry versus network flow), the cost and latency envelope of the target system, and the attack-type profile the detector is expected to face, rather than of a single performance metric reported on a single benchmark.

Author Contributions: Conceptualization, J.d.C., I.d.Z. and J.C.C.; data curation, I.d.Z. and J.d.C.; formal analysis, J.d.C. and I.d.Z.; funding acquisition, J.d.C. and I.d.Z.; investigation, I.d.Z. and J.d.C.; methodology, J.d.C. and I.d.Z.; software, J.d.C. and I.d.Z.; supervision, J.d.C. and I.d.Z.; validation, J.d.C., I.d.Z. and C.T.C.; visualization, J.d.C. and I.d.Z.; writing—original draft, J.d.C. and I.d.Z.; writing—review and editing, J.d.C., I.d.Z., J.C.C. and C.T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the LUXEMBOURG Institute of Science and Technology through the projects 'ADIALab-MAST' and 'LLMs4EU' (Grant Agreement No 101198470) and the BARCELONA Supercomputing Center through the project 'TIFON' (File number MIG-20232039).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All processed data, intermediate CSV summaries, and per-method holdout predictions supporting the reported results are contained within the manuscript and its supplementary artifacts. The full implementation of the benchmarking framework, data-preprocessing modules, model adapters for the four open-source LLMs (Qwen3-235B-A22B, Llama-3.3-70B, Hermes-4-70B, Hermes-4-405B) and the four tabular anchors (RandomForest, XGBoost, TabPFN, TabICL), the multi-seed evaluation pipelines, the paired McNemar and cross-seed Mann–Whitney statistical-testing utilities, and full reproduction instructions, is publicly available at https://github.com/drdecurto/fm-security (accessed on 2 June 2026).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AUPRC	Area Under the Precision–Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
DoS	Denial-of-Service
DR	Detection Rate
FAR	False Alarm Rate
HAI	HIL-based Augmented Industrial Dataset
HIL	Hardware-in-the-Loop
HMI	Human–Machine Interface
ICL	In-Context Learning
ICS	Industrial Control System
IDS	Intrusion Detection System

IIoT	Industrial Internet of Things
LLM	Large Language Model
MCC	Matthews Correlation Coefficient
OT	Operational Technology
PLC	Programmable Logic Controller
SCADA	Supervisory Control and Data Acquisition
SOC	Security Operations Centre
SWaT	Secure Water Treatment
TabICL	Tabular In-Context Learning
TabPFN	Tabular Prior-Fitted Network
WUSTL	Washington University in St. Louis IIoT-2021 Dataset
XGBoost	eXtreme Gradient Boosting

References

1. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201. [CrossRef]
2. Liao, H.J.; Lin, C.H.R.; Lin, Y.C.; Tung, K.Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* **2013**, *36*, 16–24. [CrossRef]
3. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1153–1176. [CrossRef]
4. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
5. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
6. Lansky, J.; Ali, S.; Mohammadi, M.; Majeed, M.K.; Karim, S.H.T.; Rashidi, S.; Hosseinzadeh, M.; Rahmani, A.M. Deep learning-based intrusion detection systems: A systematic review. *IEEE Access* **2021**, *9*, 101574–101599. [CrossRef]
7. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*; IEEE: New York, NY, USA, 2009; pp. 1–6.
8. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS)*; IEEE: New York, NY, USA, 2015; pp. 1–6.
9. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* **2018**, *1*, 108–116. [CrossRef]
10. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
11. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
12. Mohammad, R.; Saeed, F.; Almazroi, A.A.; Alsubaei, F.S.; Almazroi, A.A. Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach. *Systems* **2024**, *12*, 79. [CrossRef]
13. Koneru, S.S.; Cho, J. Bridging the Gap: A Comparative Analysis of ICS and IT Datasets for IDS Evaluation. In *Proceedings of the 2024 2nd International Conference on Foundation and Large Language Models (FLLM)*; IEEE: New York, NY, USA, 2024; pp. 300–304.
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*; NeurIPS; Curran Associates: Red Hook, NY, USA, 2017; Volume 30, pp. 5998–6008.
15. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In *Proceedings of the Advances in Neural Information Processing Systems*; NeurIPS; Curran Associates: Red Hook, NY, USA, 2020; Volume 33, pp. 1877–1901.
16. Hasanov, I.; Virtanen, S.; Hakkala, A.; Isoaho, J. Application of Large Language Models in Cybersecurity: A Systematic Literature Review. *IEEE Access* **2024**, *12*, 176751–176778. [CrossRef]
17. Zhang, J.; Bu, H.; Wen, H.; Liu, Y.; Fei, H.; Xi, R.; Li, L.; Yang, Y.; Zhu, H.; Meng, D. When llms meet cybersecurity: A systematic literature review. *Cybersecurity* **2025**, *8*, 1–41. [CrossRef]
18. Balogh, S.; Mlyncek, M.; Vranák, O.; Zajac, P. Using Generative AI Models to Support Cybersecurity Analysts. *Electronics* **2024**, *13*, 4718. [CrossRef]

19. DeCusatis, C.; Tomo, R.; Singh, A.; Khoury, E.; Masone, A. Cybersecurity Applications of Near-Term Large Language Models. *Electronics* **2025**, *14*, 2704. [CrossRef]
20. Coppolino, L.; Iannaccone, A.; Nardone, R.; Petruolo, A. Asset Discovery in Critical Infrastructures: An LLM-Based Approach. *Electronics* **2025**, *14*, 3267. [CrossRef]
21. Keltek, M.; Hu, R.; Sani, M.F.; Li, Z. LSAST: Enhancing Cybersecurity Through LLM-Supported Static Application Security Testing. In *Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 166–179.
22. Muhammad, M.; Shaaban, A.M.; German, R.; Al Sardy, L. HyLLM-IDS: A Conceptual Hybrid LLM-Assisted Intrusion Detection Framework for Cyber-Physical Systems. In *Proceedings of the International Conference on Computer Safety, Reliability, and Security*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 129–142.
23. Li, Y.; Xiang, Z.; Bastian, N.D.; Song, D.; Li, B. IDS-Agent: An LLM Agent for Explainable Intrusion Detection in IoT Networks. In *Proceedings of the NeurIPS 2024 Workshop on Open-World Agents*; NeurIPS; Curran Associates: Red Hook, NY, USA, 2024.
24. Hollmann, N.; Müller, S.; Eggensperger, K.; Hutter, F. TabPFN: A transformer that solves small tabular classification problems in a second. In Proceedings of the International Conference on Learning Representations 2023, Kigali, Rwanda, 1–5 May 2023.
25. Hollmann, N.; Müller, S.; Purucker, L.; Krishnakumar, A.; Körfer, M.; Hoo, S.B.; Schirrmeister, R.T.; Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature* **2025**, *637*, 319–326. [CrossRef] [PubMed]
26. Qu, J.; Holzmüller, D.; Varoquaux, G.; Morvan, M.L. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. *arXiv* **2025**, arXiv:2502.05564.
27. García, P.; de Curtò, J.; de Zarzà, I. Foundation Models for Tabular Intrusion Detection: Evaluating TabPFN and LLM Few-Shot Classification on IoT Network Security. In *Proceedings of the 2025 3rd International Conference on Foundation and Large Language Models (FLLM)*; IEEE: New York, NY, USA, 2025.
28. García, P.; de Curtò, J.; de Zarzà, I.; Cano, J.C.; Calafate, C.T. Foundation Models for Cybersecurity: A Comprehensive Multi-Modal Evaluation of TabPFN and TabICL for Tabular Intrusion Detection. *Electronics* **2025**, *14*, 3792. [CrossRef]
29. Hossain, M.A.; Islam, M.S. Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: A promising solution for robust cybersecurity. *Meas. Sens.* **2024**, *32*, 101037. [CrossRef]
30. Lai, T.; Farid, F.; Bello, A.; Sabrina, F. Ensemble learning based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis. *Cybersecurity* **2024**, *7*, 44. [CrossRef]
31. Yan, J.; Wang, Q.; Cheng, Y.; Su, Z.; Zhang, F.; Zhong, M.; Liu, L.; Jin, B.; Zhang, W. Optimized single-image super-resolution reconstruction: A multimodal approach based on reversible guidance and cyclical knowledge distillation. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108496. [CrossRef]
32. Wang, X.; Jiang, H.; Dong, Y.; Mu, M. Spatial-channel collaborative multi-scale graph interaction deep transfer learning for unsupervised rotating machinery fault diagnosis. *Eng. Appl. Artif. Intell.* **2026**, *176*, 114691. [CrossRef]
33. Ismail, S.; Dandan, S.; Qushou, A. Intrusion detection in IoT and IIoT: Comparing lightweight machine learning techniques using TON_IoT, WUSTL-IIOT-2021, and EdgeIIoTset datasets. *IEEE Access* **2025**, *13*, 73468–73485. [CrossRef]
34. Zhou, C.; Li, Q.; Li, C.; Yu, J.; Liu, Y.; Wang, G.; Zhang, K.; Ji, C.; Yan, Q.; He, L.; et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *Int. J. Mach. Learn. Cybern.* **2025**, *16*, 9851–9915.
35. Yamin, M.M.; Hashmi, E.; Ullah, M.; Katt, B. Applications of llms for generating cyber security exercise scenarios. *IEEE Access* **2024**, *12*, 143806–143822. [CrossRef]
36. Arik, S.Ö.; Pfister, T. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2021; Volume 35, pp. 6679–6687.
37. Hegselmann, S.; Buendia, A.; Lang, H.; Agrawal, M.; Jiang, X.; Sontag, D. Tabllm: Few-shot classification of tabular data with large language models. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Valencia, Spain, 25–27 April 2023; pp. 5549–5581.
38. Han, S.; Yoon, J.; Arik, S.O.; Pfister, T. Large language models can automatically engineer features for few-shot tabular learning. *arXiv* **2024**, arXiv:2404.09491.
39. Mathur, A.P.; Tippenhauer, N.O. SWaT: A water treatment testbed for research and training on ICS security. In *Proceedings of the 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater)*; IEEE: New York, NY, USA, 2016; pp. 31–36.
40. Goh, J.; Adepu, S.; Junejo, K.N.; Mathur, A. A dataset to support research in the design of secure water treatment systems. In *Proceedings of the International Conference on Critical Information Infrastructures Security*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 88–99.
41. Shin, H.K.; Lee, W.; Yun, J.H.; Kim, H. {HAI} 1.0:{HIL-based} augmented {ICS} security dataset. In *Proceedings of the 13Th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*; USENIX Association: Berkeley, CA, USA, 2020.

42. Zolanvari, M. *WUSTL-IIoT-2021: Industrial IoT Cybersecurity Dataset*; IEEE-DataPort: New York, NY, USA, 2021. [CrossRef]
43. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp. 785–794.