Applied Expert Systems and Knowledge Management

www.aeskm.org
eISSN: 3153–3825

Machine Learning Expert Framework for Cyberattack Detection in the International Electrotechnical Commission 61850 Substations

Md. Taukir Ahmed[1]*

[1] Institute of Information and Communication Technology, Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh

ARTICLE INFO

Article history:
Received 5 September 2026
Received in revised form 22 October 2026
Accepted 1 November 2026
Available online 4 November 2026

Keywords:
Cyber-Physical Expert Systems; Smart Grid Security; Intrusion Detection Frameworks Machine Learning Classification; IEC 61850 Substation Protection

ABSTRACT

Digital communication expansion in electrical substations introduces significant cybersecurity vulnerabilities into critical power system protection functions. This study presents an intelligent expert decision-support framework utilizing conventional machine learning algorithms and stacking ensembles to classify physical power faults alongside distinct cyberattack vectors within International Electrotechnical Commission (IEC) 61850 - based digital substations. Evaluating a publicly available 52-feature digital substation dataset, an initial integrity assessment revealed event overlaps between official splits, prompting the implementation of stratified five-fold cross-validation for robust model validation. Gradient Boosting demonstrated superior diagnostic performance, achieving 87.20% accuracy, a 0.791 macro averaged F1-score, and an 0.809 Matthews correlation coefficient. Feature ablation established that protocol-level and consistency features yield higher predictive value than raw electrical measurements. While effective at identifying physical faults and masquerade attacks, reduced recall on replay and injection threats highlights specific diagnostic limitations. Ultimately, this research provides an actionable applied framework emphasizing data integrity and class wise evaluation for critical infrastructure cybersecurity systems.

1. Introduction

The digitalization of electrical substations has transformed the way protection, monitoring, and control devices communicate within modern power systems. The IEC 61850 standard supports communication among intelligent electronic devices (IEDs), merging units, protection relays, and circuit breakers. In particular, Generic Object-Oriented Substation Event (GOOSE) messages support the rapid exchange of event and protection-related information, while Sampled Values (SV) transmit digitized current and voltage measurements. Although these communication mechanisms improve interoperability and automation, they also expose substation operations to cybersecurity threats. Attacks on communication networks can compromise message integrity, interfere with control decisions, and potentially affect the operation of physical equipment [1,2,3,4]. Digital substations are

* *Corresponding author.*
E-mail address: taukirahmed.vu@gmail.com
https://doi.org/10.59543/pbvwpx90

cyber-physical environments in which network activity and physical electrical processes are closely connected. An attacker may impersonate a legitimate device, replay previously captured messages, inject falsified information, or manipulate control-related communication. Such attacks can make a malicious message appear legitimate or cause a protection system to receive information that does not represent the actual state of the electrical network. Research on smart-grid cybersecurity has examined these threats and discussed defense mechanisms involving communication security, anomaly detection, and intelligent classification [2,3,4,5]. Consequently, intrusion detection is an important part of protecting digital-substation communication and supporting reliable power system operation. Machine learning (ML) provides a data-driven approach to identifying abnormal patterns in smart-grid communication. A trained classifier can analyze message counters, inter-message timing, device states, consistency indicators, and electrical measurements to distinguish normal events from faults and cyberattacks. Previous studies have investigated ML-based security mechanisms for smart grids, including approaches for detecting false-data injection and other malicious activities. Comparative studies have also considered tree-based classifiers, support vector machines, and neural-network-based methods for identifying abnormal behavior [5,6,7]. However, model performance depends not only on the selected algorithm but also on the quality, diversity, and independence of the data used for training and evaluation. Several conventional classifiers are suitable for evaluating this problem. Decision Trees provide rule-based classification, while Random Forest and Extra Trees combine multiple trees to model nonlinear relationships. Gradient Boosting builds successive learners to improve predictive performance. Logistic Regression provides a linear baseline, whereas Support Vector Machines (SVMs) and k-nearest neighbors (k-NN) offer alternative approaches based on decision boundaries and local similarity. Ensemble methods can combine predictions from several classifiers, but their effectiveness depends on the quality and diversity of their component models. Previous smart-grid intrusion detection research has demonstrated the potential of tree-based and other ML methods, while broader reviews have highlighted continuing challenges related to model robustness, computational cost, and practical deployment [5,8]. A particularly important challenge is the reliability of the evaluation dataset. High classification accuracy does not necessarily demonstrate that a model can recognize previously unseen attacks. When training and test sets contain duplicated events or closely related records, a model may exploit information shared between the two sets rather than learn patterns that generalize to new observations. Dataset construction and validation are therefore important considerations in smart-grid intrusion detection research. Alani and Baker examined the characteristics of smart-grid intrusion detection datasets and discussed their use in ML-based security research [9]. More broadly, research on ML-based smart-grid intrusion detection has identified data generation, evaluation procedures, and testbed characteristics as important factors affecting the interpretation of reported performance [8,9]. Class imbalance presents another challenge in multiclass intrusion detection. Normal events and frequently occurring attack types may dominate a dataset, while less frequent classes receive limited representation. Under these conditions, overall accuracy can hide poor detection of security-relevant events. Macro-averaged precision, recall, and F1-score provide a more balanced assessment because each class contributes equally to the final metric. The Matthews correlation coefficient (MCC), confusion matrices, receiver operating characteristic (ROC) curves, and precision–recall analysis provide additional information about classification quality and class-specific errors. These measures are especially important when replay and injection attacks are difficult to distinguish from normal traffic or from other attack classes [10]. Recent research has continued to investigate intelligent security mechanisms for digital substations and smart grids. In particular, research on IEC 61850-based substations has examined recurrent neural networks and other

learning-based methods for detecting attacks involving GOOSE communication. Broader reviews have also discussed the opportunities and limitations of ML-based cybersecurity, including dataset limitations, model robustness, and the challenges of deploying detection systems in operational environments [11]. Nevertheless, reliable evaluation requires more than comparing headline accuracy values. It also requires checking data integrity, reporting class-wise performance, and examining which features contribute to the predictions. Feature-ablation analysis can help determine whether detection depends primarily on protocol behavior, consistency flags, or physical electrical measurements. Protocol counters and timing information may reveal abnormal communication, while consistency indicators can identify disagreement between commands and the reported physical state. Electrical measurements may provide complementary evidence, but their contribution depends on the attack scenarios represented in the dataset. Comparing different feature subsets can therefore clarify which information supports classification and whether a model relies on signals that may be specific to the testbed. This type of analysis is particularly relevant when the objective is to develop efficient and reliable detection methods rather than maximize accuracy on a single dataset.

Against this background, this study evaluates seven conventional ML classifiers and a stacking ensemble for multiclass intrusion detection in an IEC 61850 digital-substation testbed. The dataset contains five classes: Normal, Fault, Masquerade, Replay, and Injection. Before model training, the official training and test files are examined for missing values, duplicated records, class distribution, and event-level overlap. The analysis reveals that both files contain the same underlying events, with differences limited to four derived protocol features. Accordingly, the official split is retained for comparison, while stratified five-fold cross-validation is used as the primary evaluation protocol. This approach provides a more cautious assessment of the models' ability to classify events that are excluded from the corresponding training fold.

The main contributions of this study are summarized as follows:

1. The relationship between the official training and test files is examined to identify event overlap that may produce overly optimistic performance estimates.
2. Seven conventional ML classifiers and a stacking ensemble are evaluated using the official split and stratified five-fold cross-validation.
3. Accuracy, macro-averaged precision, recall and F1-score, MCC, confusion matrices, and ranking metrics are used to assess overall performance and weaknesses in detecting individual attack classes.
4. Protocol features, consistency and relay flags, and electrical measurements are evaluated in different combinations to identify the feature groups that contribute most to classification.
5. The study examines whether stacking improves on the best individual classifier and discusses the dataset and evaluation limitations that must be addressed before applying the findings to operational substations.

2. Methodology

This section describes the dataset, the integrity checks that shaped the evaluation design, the exploratory analysis, the seven base classifiers, the stacking ensemble, and the evaluation protocols. The overall workflow is shown in Fig.1.

2.1 Testbed and Data Source

This study uses a publicly available Kaggle dataset derived from an IEC 61850 digital substation testbed. The testbed, illustrated in Fig.2, represents a substation communication environment in which protection related measurements and control messages are exchanged over a switched ring network. It comprises four Merging Units (MU1–MU4), a Protection Intelligent Electronic Device (IED), an Intrusion Detection System (IDS), an attacker node, and a circuit breaker (CB). The Merging Units digitize current and voltage measurements from four transformers and transmit the resulting data as Sampled Values (SV). The Protection IED processes the SV streams and communicates through Generic Object-Oriented Substation Event (GOOSE) messages to support protection operations. Meanwhile, the IDS monitors network traffic, and the attacker node represents a potential source of malicious communication within the same network. The circuit breaker receives GOOSE trip commands from the protection system.

Each dataset record corresponds to a monitored event and is assigned to one of five classes: Normal, Fault, Masquerade, Replay, or Injection. These classes represent normal operating conditions, physical faults, and three cyberattack scenarios, respectively. This combination of physical and cyber events provides a basis for investigating whether machine learning models can distinguish legitimate substation operation from fault conditions and malicious network activity. The dataset is therefore used to evaluate the classification performance of the proposed detection framework under different operating and attack scenarios.

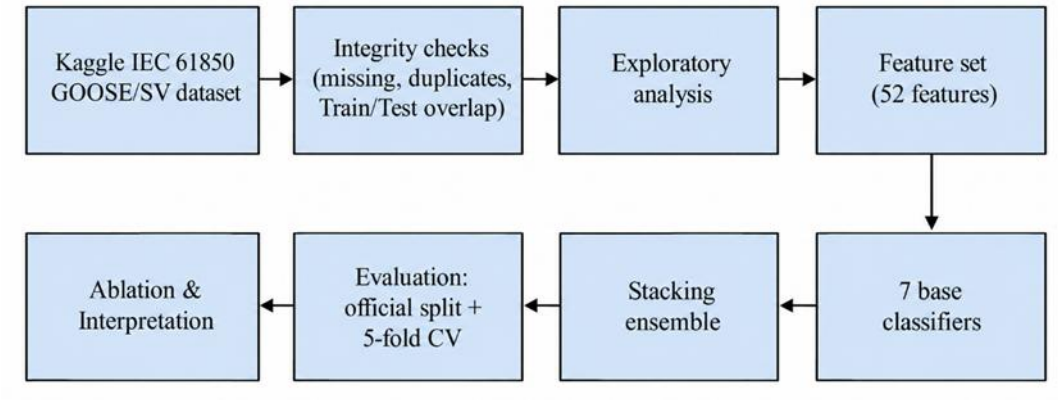


Fig. 1. Overall workflow of the study.

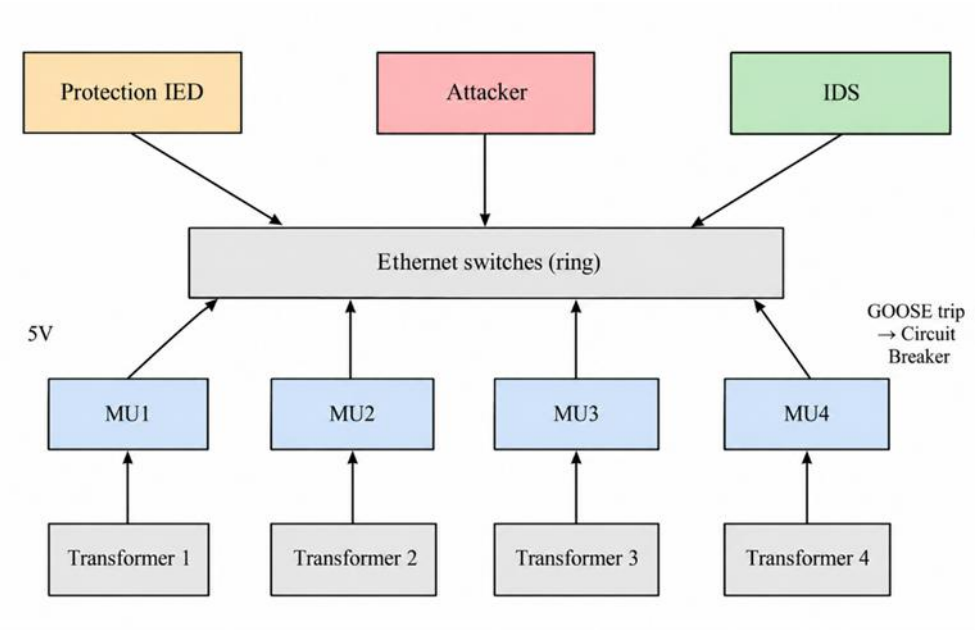


Fig. 2. Schematic of the IEC 61850 digital substation testbed

2.2 Dataset Description

The dataset consists of two files, Train.csv and Test.csv, each containing 1,094 instances and 54 columns. Of these columns, 52 represent predictive features, one records the timestamp (time), and one contains the target class label. Initial data inspection identified no missing values or duplicate rows in either file. The 52 predictive features were organized into four groups according to their functional roles in the substation monitoring environment, as summarized in Table 1. Fig.5 presents the number of features in each group. The timestamp was removed before modelling so that models could not exploit the order in which experiments were run. The class distribution, reported in Table 2 and illustrated in Fig.3 and 4, indicates a moderate imbalance among the five categories. Normal and Masquerade events together make up 78% of the data, while Fault, the smallest class, has only 41 events. Class balance is identical in the two files.

Table 1
Feature groups used for modelling.

Feature group	Number	Example features	Description
GOOSE protocol	7	sqNum, stnum, state_cb, sqDiff, stDiff, timeLastMsg, recentChange	Message counters, breaker state, sequence differences, and message timing
Consistency and relay flags	11	MU1Cs–MU4Cs, Cs, consistency, any_relay, MU1Log–MU4Log	Consistency checks across MUs and relay logic indicators
Aggregate electrical	2	threePhaseVSum, threePhaseCSum	Aggregate three-phase voltage and current measurements
Per-MU measurements	32	threePhaseVMU1–4, voltage/current phase angles, IED4_iA–iC	Electrical measurements reported by individual MUs and the Protection IED
Total	52	—	Predictive features used for classification

Table 2
Class distribution of the dataset

Class	Events	Share (%)	Description
Normal	455	41.6	Legitimate operation
Masquerade	397	36.3	Messages associated with an impersonated device
Injection	101	9.2	Falsified values inserted into the data stream
Replay	100	9.1	Previously captured messages retransmitted
Fault	41	3.7	Physical short-circuit event
Total	1,094	100.0*	—

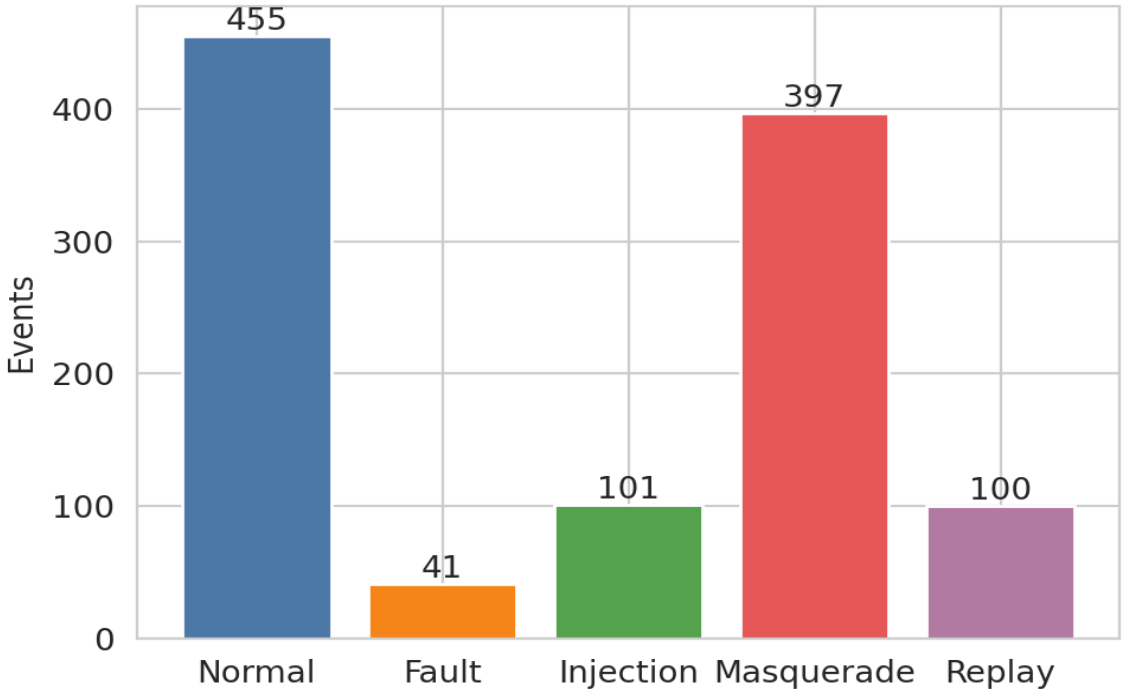


Fig. 3. Number of events per class.

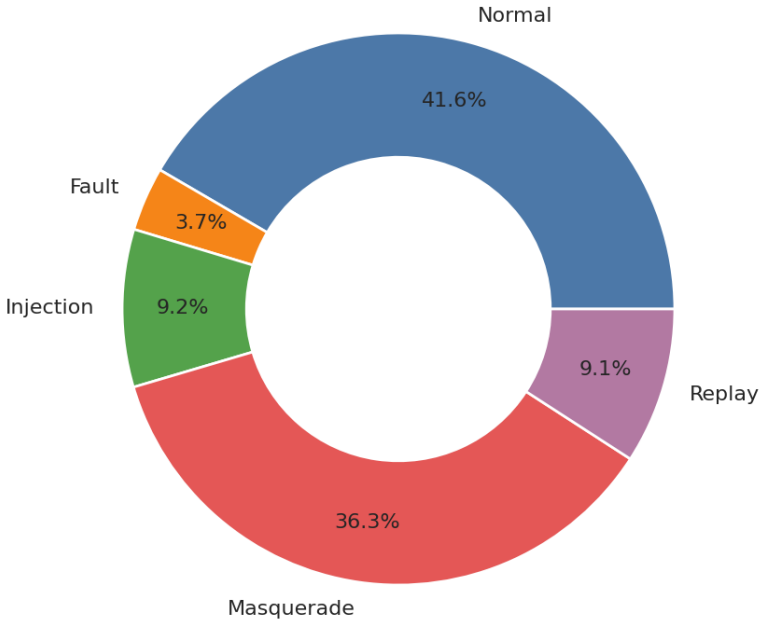


Fig. 4. Class shares in the dataset.

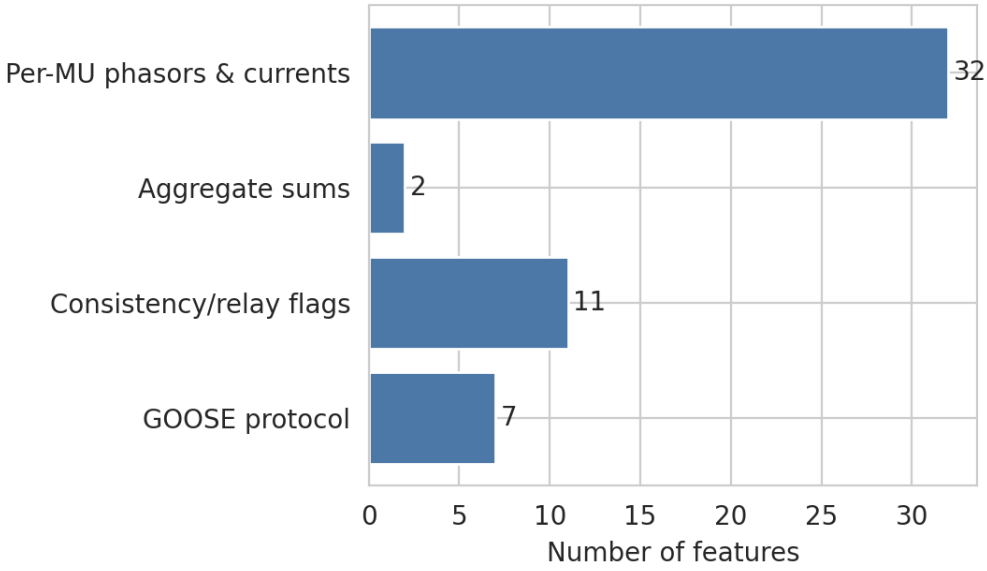


Fig. 5. Number of features in each feature group.

2.3 Dataset Description

Before model training, Train.csv and Test.csv were compared to assess their independence. Both files contain the same 1,094 timestamps and class labels. After aligning records by timestamp, 49 of the 53 non-label features were identical, while four protocol features differed: sqDiff (65%), stDiff

(68%), timeLastMsg (69%), and recentChange (56%), as shown in Fig. 6. This overlap indicates that Test.csv is not an independent test set.

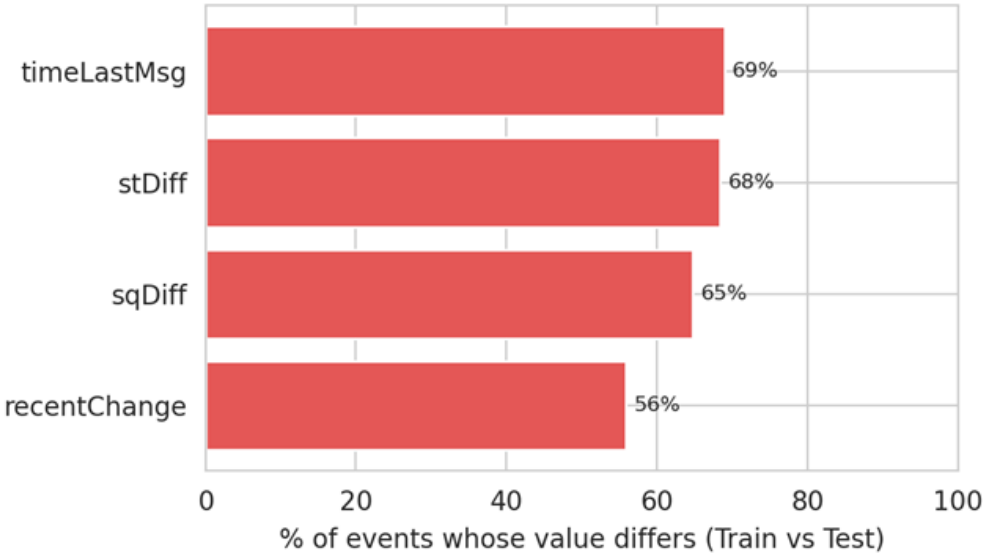


Fig. 6. Percentage of events with different values between Train.csv and Test.csv for the four recomputed protocol features.

Two evaluation protocols were considered. Protocol A follows the original train test split, while Protocol B uses stratified five-fold cross-validation on Train.csv with a fixed random seed of 42. In each fold, four subsets are used for training and the remaining subset for validation, ensuring that each event is evaluated once without being included in that fold's training data Fig. 7. The two protocols and their roles in model evaluation are summarized in Fig. 8. Protocol B provides the primary basis for evaluating model performance. However, the absence of experimental run identifiers means that related events may appear in different folds, potentially leading to optimistic estimates.

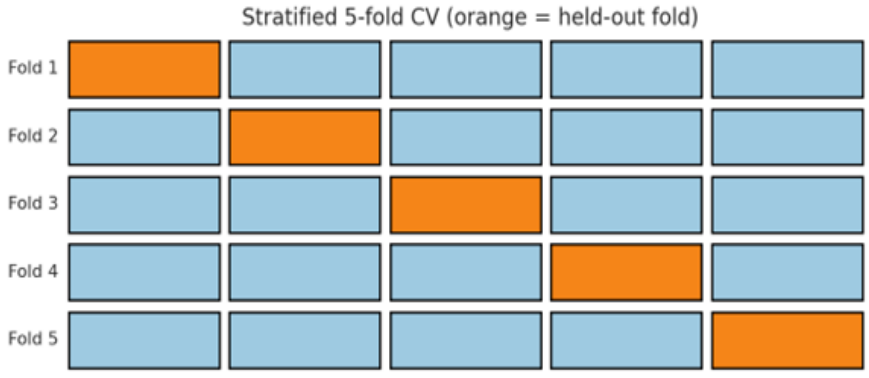


Fig. 7. Stratified five-fold cross-validation procedure used for model evaluation.

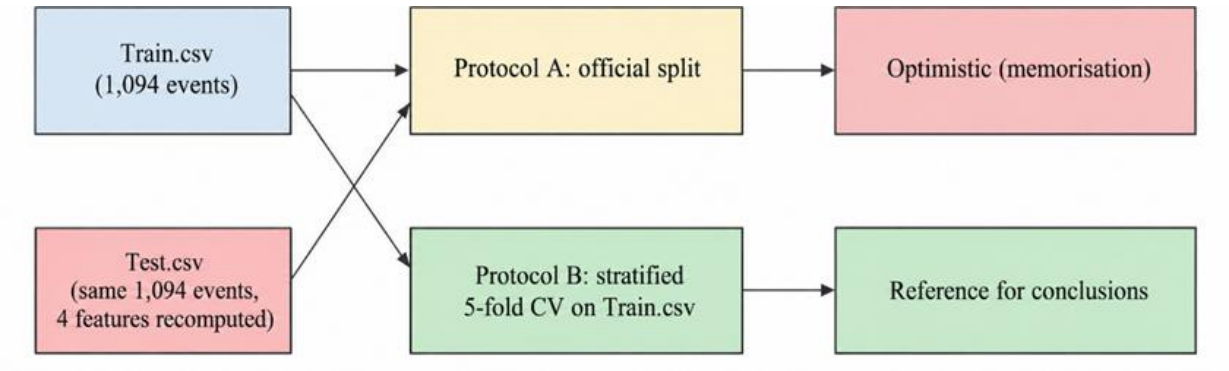


Fig. 8. Overview of the two evaluation protocols.

2.4 Exploratory Data Analysis

Exploratory data analysis was conducted to examine differences in electrical and communication features across the five classes. Fig. 9 shows that the circuit-breaker state (state_cb) is 0 for all Normal events and 1 for all Fault events. It is also 1 in 91% of Masquerade events but only 12% of Replay events.

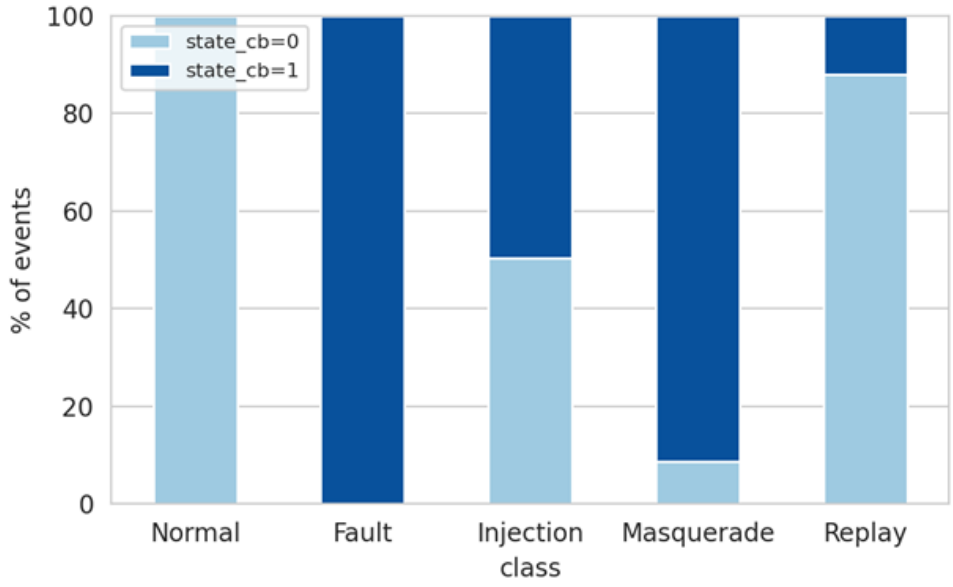


Fig. 9. Distribution of circuit-breaker state (state_cb) across the five event classes.

Fig. 10 shows that the consistency flag is 1 for 98% of Normal events but only 6% of Masquerade events, while 84% of Replay events remain consistent. These patterns suggest that Masquerade events often exhibit inconsistencies between breaker commands and electrical measurements, whereas Replay events may resemble normal traffic. The inter-message timing feature (timeLastMsg) is strongly right-skewed. As shown in Fig. 11, its median is 619 for Normal events and ranges from 228 to 311 for the other classes. Fifteen observations with non-positive values are excluded from the logarithmic-scale plot. Fig. 12 presents the Pearson correlation heatmap for selected protocol,

consistency, and aggregate electrical features. Most features show weak linear correlations, although notable relationships exist among state_cb, consistency, and recentChange. This finding motivates the evaluation of non-linear classifiers alongside conventional models.

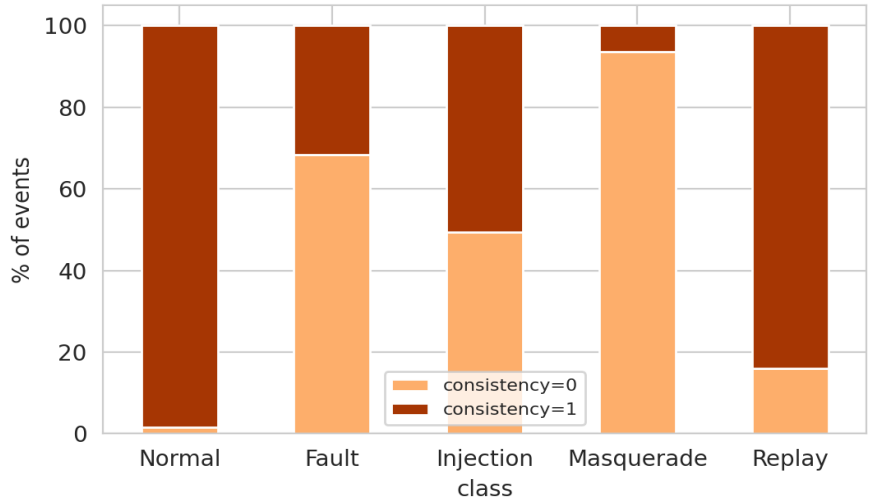


Fig.10. Distribution of the consistency flag across the five event classes.

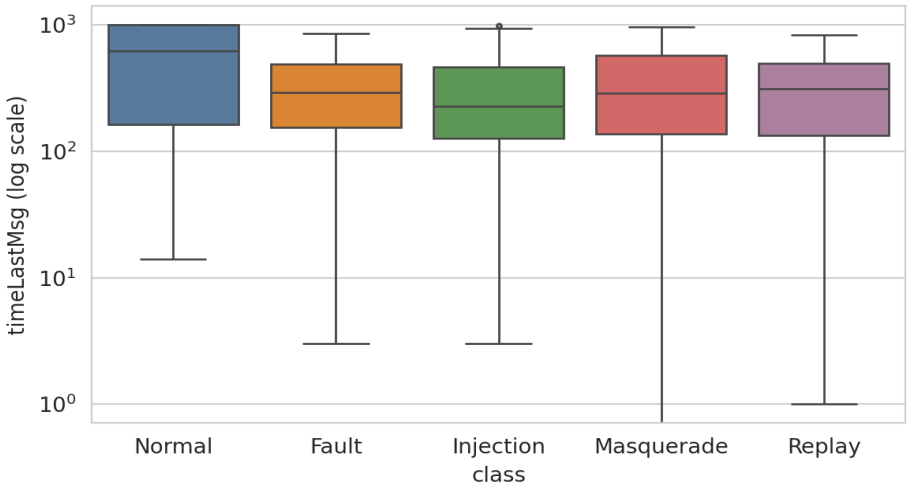


Fig. 11. Class-wise distribution of timeLastMsg on a logarithmic scale; 15 non-positive observations are omitted.

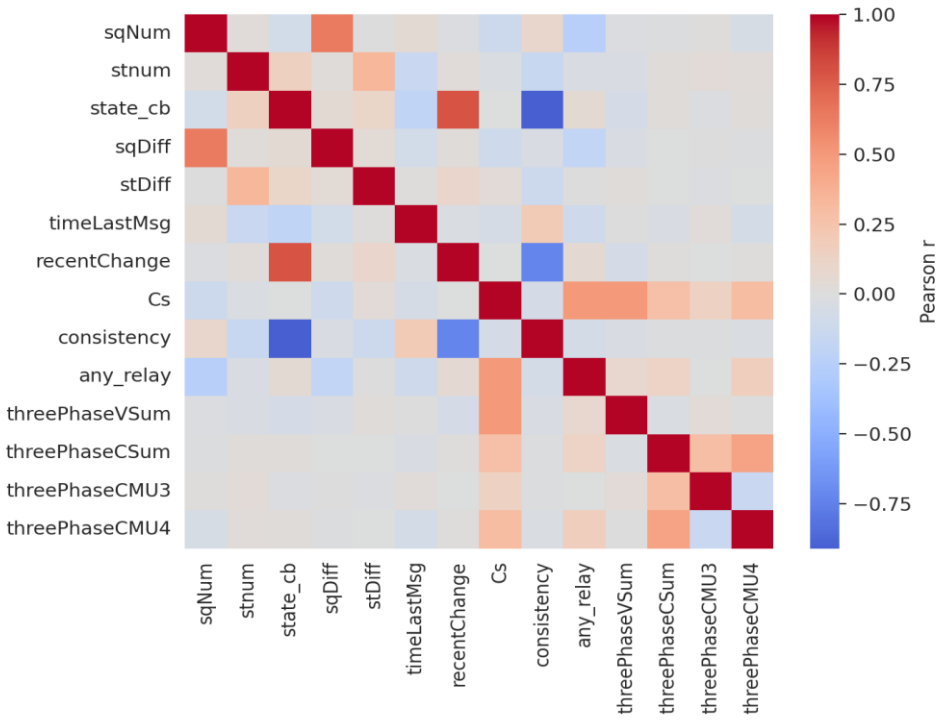


Fig. 12. Pearson correlation heatmap of selected protocol, consistency, and aggregate electrical features.

Fig. 13 shows the PCA projection of standardized features, while Fig. 14 presents the corresponding t-SNE projection. Both visualizations indicate relatively compact clusters for Normal and Masquerade events, whereas Injection and Replay events overlap with other classes, suggesting that distinguishing these categories may be challenging.

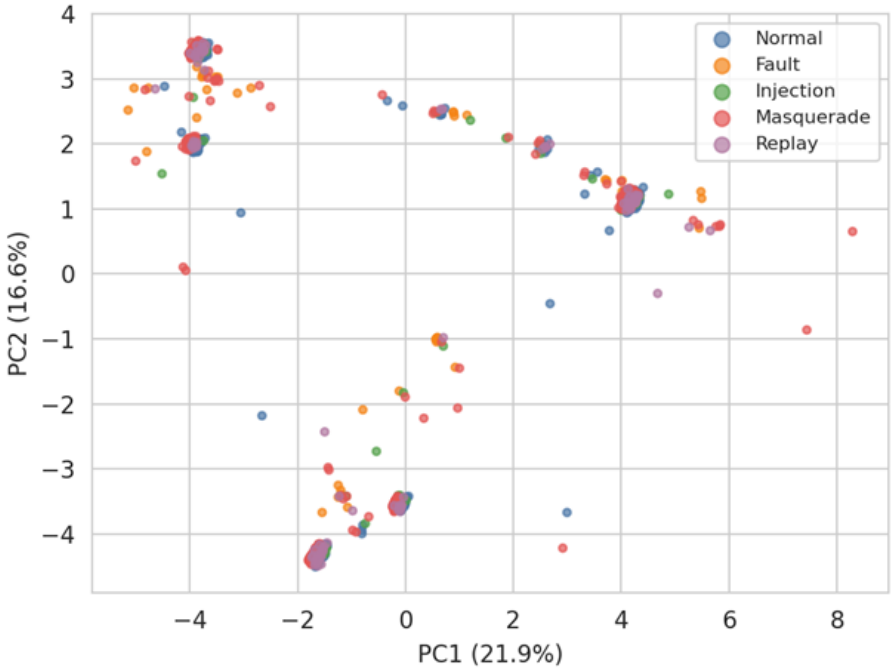


Fig. 13. PCA projection of standardized features, with observations colored by class.

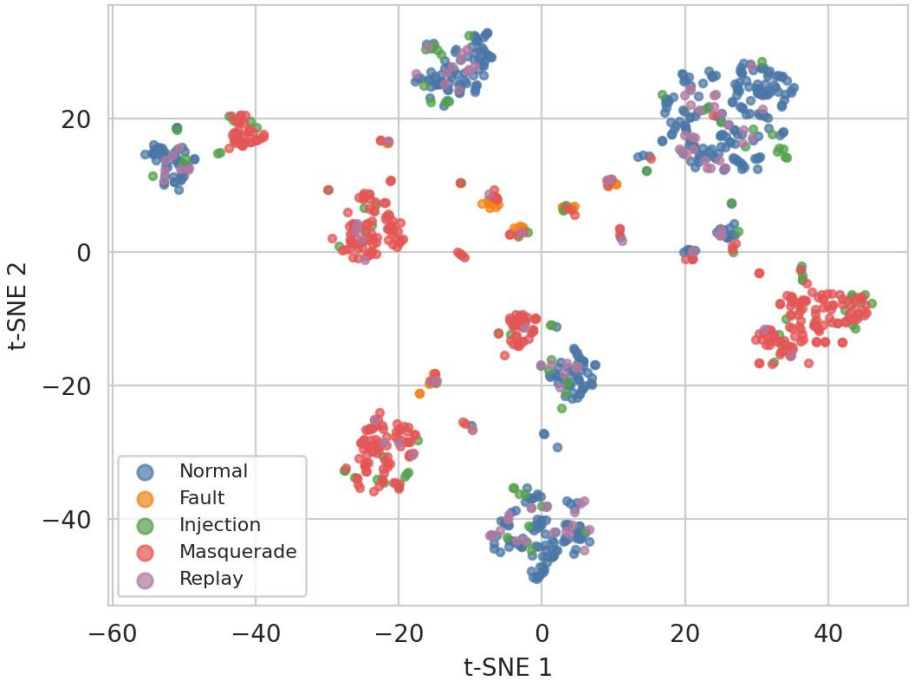


Fig. 14. t-SNE projection of standardized features, with observations colored by class.

2.5 Preprocessing

All 52 features were used to train the machine learning models. Since the dataset contained no missing values, no imputation was required. Logistic Regression, Support Vector Machine (SVM), and k-Nearest Neighbors (k-NN) were trained using standardized features. Standardization was performed within each training fold to prevent data leakage during cross-validation. Tree-based models used the original feature values because they generally do not require feature scaling. No resampling was applied to balance the classes. Instead, macro-averaged evaluation metrics were used to assess model performance across all five classes, giving equal importance to each class.

2.6 Base Classifiers

Seven machine learning classifiers were selected for comparison: Decision Tree, Random Forest, Extra Trees, Gradient Boosting, Logistic Regression, Support Vector Machine (SVM) with an RBF kernel, and k-Nearest Neighbors (k-NN). These models represent different learning approaches, including individual decision trees, tree ensembles, linear classification, kernel-based learning, and instance-based classification. Most hyperparameters were kept at their default settings to establish a baseline without extensive tuning on the relatively small dataset. The main settings used for each model are summarized in Table 3. All models were implemented using scikit-learn version 1.8.0.

Table 3
Configuration of the base classifiers and stacking ensemble

Model	Key settings	Scaling
Decision Tree	Gini criterion, unlimited depth, random seed 42	No
Random Forest	300 trees, Gini criterion, bootstrap sampling, random seed 42	No
Extra Trees	300 randomized trees, random seed 42	No
Gradient Boosting	100 boosting stages, maximum tree depth of 3, learning rate of 0.1, random seed 42	No
Logistic Regression	L2 regularization, C = 1, maximum 3,000 iterations	Yes
SVM (RBF)	C = 1, gamma = scale, probability estimates enabled	Yes
k-NN	5 neighbors, Euclidean distance	Yes
Stacking Ensemble	Seven base classifiers, Logistic Regression meta-learner (C = 10), internal 3-fold cross-validation	Per base model

2.7 Stacking Ensemble

A stacking ensemble was developed to examine whether combining the predictions of the seven base classifiers could improve detection performance. Fig. 15 illustrates the overall structure of the ensemble. Each base classifier generates class probabilities through three-fold cross-validation on the training data. The resulting probabilities are combined into 35 features, representing seven classifiers and five event classes. These features are then used to train a Logistic Regression meta-learner. During prediction, the base classifiers are fitted again using the complete training set. Their predicted class probabilities are then passed to the trained meta-learner to obtain the final predictions. Under Protocol B, the entire stacking procedure is repeated within each outer cross-validation fold. This ensures that the meta-learner is trained without access to the corresponding validation events.

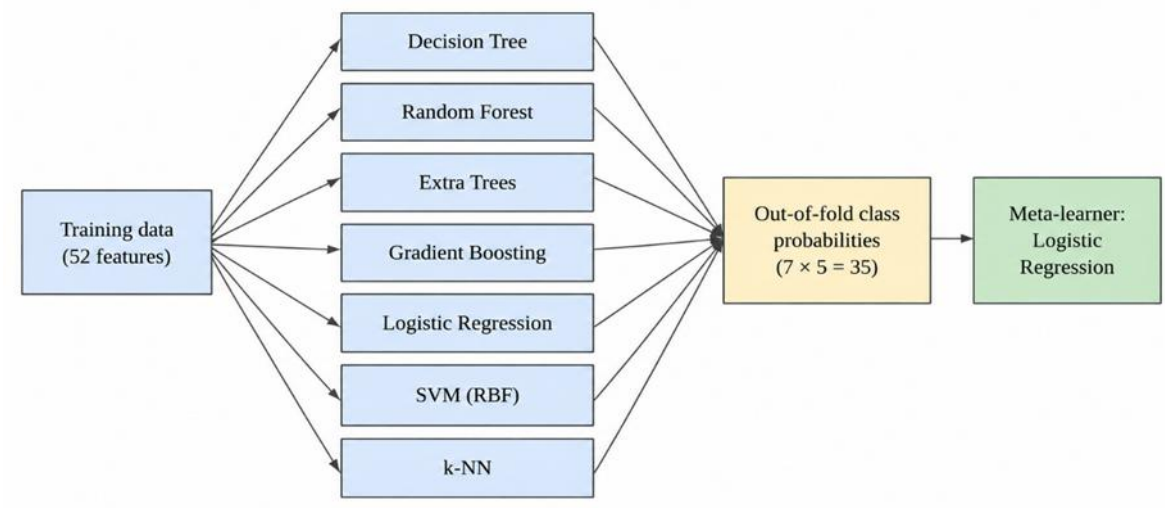


Fig. 15. Architecture of the stacking ensemble, showing seven base classifiers generating out-of-fold class probabilities for the Logistic Regression meta learner.

2.8 Evaluation Metrics and Ablation Design

The models were evaluated using accuracy, macro-averaged precision, recall, and F1-score, together with the multiclass Matthews correlation coefficient (MCC). Macro-averaged metrics were used to give equal importance to all five classes, including those with fewer samples. The evaluation also included per-class precision, recall, and F1-score, confusion matrices, one-versus-rest ROC and precision–recall curves, and a binary classification analysis to distinguish attack events from normal events. Training time, prediction time, and learning curves were examined to provide further insight into model performance.

To understand how different feature groups influenced detection, an ablation study was performed using Gradient Boosting and cross-validation. Six feature sets were compared: all 52 features, the seven protocol features, the 18 protocol and flag features, the 11 flag features, the 34 electrical features, and the 48 features remaining after removing the four columns that differed between Train.csv and Test.csv. This analysis helped examine the contribution of protocol information, consistency flags, and electrical measurements to classification performance. Feature importance was also extracted from a Gradient Boosting model trained on the full dataset. A fixed random seed of 42 was used throughout the experiments to support reproducibility. All analyses were carried out in Python using scikit-learn, pandas, NumPy, and Matplotlib.

3. Results and Discussion

3.1 Comparison of the seven base models and the stacking ensemble

Table 4 and Fig. 16, Fig.17 and Fig. 18 compare the seven base models and the stacking ensemble under the official train-test split and stratified five-fold cross-validation. On the official split, Random Forest achieved the highest accuracy (99.27%), followed by Extra Trees (99.18%) and Gradient Boosting (98.63%). However, their cross-validated accuracies decreased to 82.45%, 81.17%, and 87.20%, respectively. The largest decreases occurred for Extra Trees (18.01 percentage points) and Random Forest (16.82 points), suggesting that the official split may overestimate performance because of overlapping events.

Table 4
Performance comparison of the base models and stacking ensemble.

Model	Official accuracy (%)	Official F1	CV accuracy (%)	CV precision	CV recall	CV F1	CV MCC
Decision Tree	94.15	0.922	80.35	0.712	0.717	0.714	0.709
Random Forest	99.27	0.990	82.45	0.753	0.636	0.631	0.738
Extra Trees	99.18	0.990	81.17	0.690	0.631	0.617	0.717
Gradient Boosting	98.63	0.980	87.20	0.828	0.775	0.791	0.809
Logistic Regression	85.10	0.690	78.52	0.596	0.568	0.558	0.674
SVM (RBF)	82.36	0.588	79.98	0.598	0.558	0.540	0.700
k-NN	85.37	0.704	77.51	0.587	0.580	0.547	0.660
Stacking Ensemble	91.41	0.833	83.27	0.756	0.673	0.690	0.748

Gradient Boosting performed best under cross-validation, achieving 87.20% accuracy, a macro-F1 score of 0.791, and an MCC of 0.809. The stacking ensemble achieved 83.27% accuracy and a

macro-F1 score of 0.690, so it did not outperform its strongest base model. Its lower performance may reflect limited data for training the meta-learner, weak predictions from some base models, and the strong performance of Gradient Boosting. The ensemble also required 17.6 s to train, compared with 4.0 s for Gradient Boosting Fig.19. Therefore, Gradient Boosting was the more suitable model for this dataset.

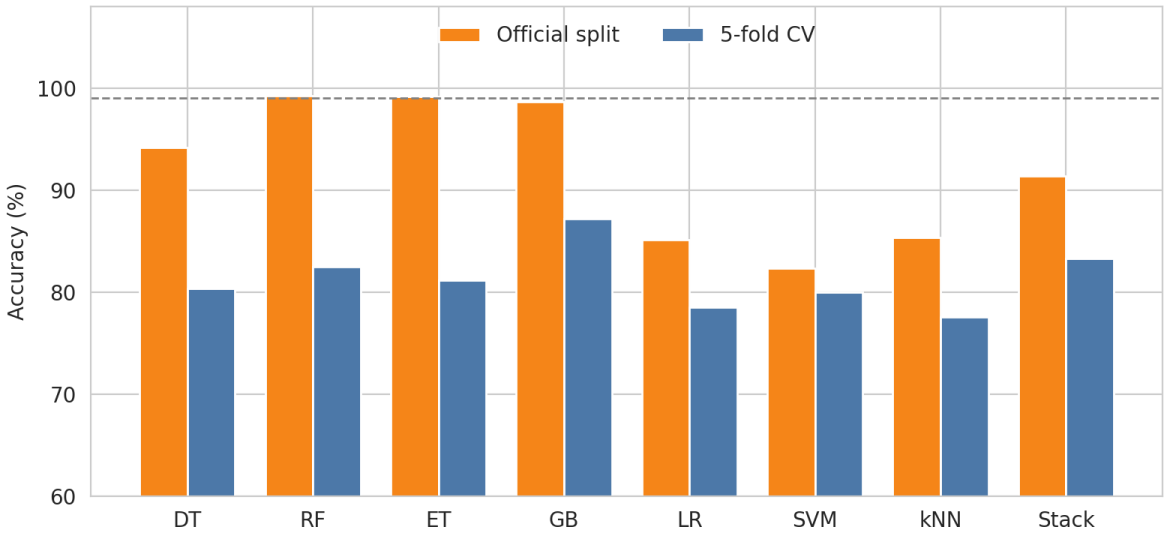


Fig.16. Accuracy of each model under the official split and under 5-fold cross-validation. The dashed line marks 99%.

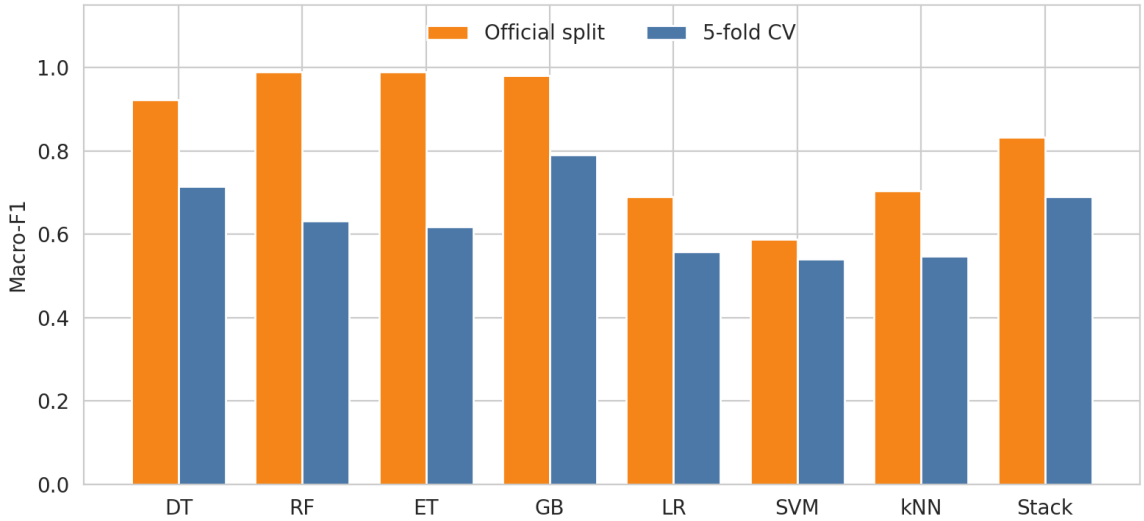


Fig.17. Macro-F1 of each model under the official split and under 5-fold cross-validation.



Fig.18. Accuracy lost when moving from the official split to cross-validation. Models that memories rows lose the most.

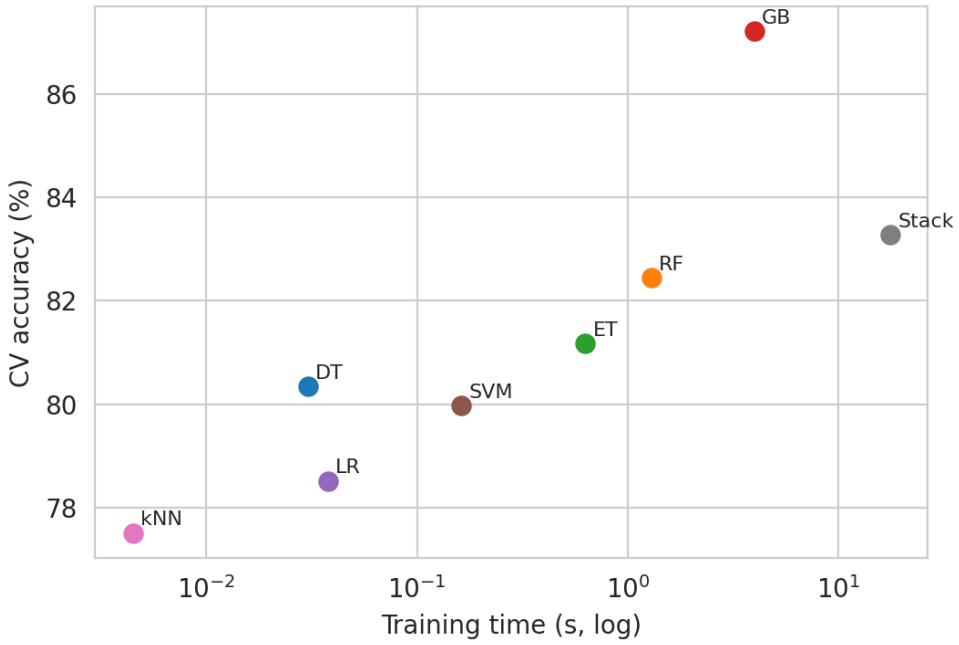


Fig.19. Training time against cross-validated accuracy (logarithmic time axis).

3.2 Per-Class Classification Performance

Table 5 and Fig.20 ,Fig.21 and Fig. 22 show the class-wise performance of Gradient Boosting. The model achieved strong results for Fault (F1 = 0.952), Masquerade (0.936), and Normal (0.915) events. However, Injection and Replay were more difficult to classify, with F1 scores of 0.624 and 0.526, respectively.

Table 5,
Per-class performance of Gradient Boosting under five-fold cross-validation.

Class	Precision	Recall	F1-score	Support
Normal	0.882	0.952	0.915	455
Fault	0.930	0.976	0.952	41
Injection	0.682	0.574	0.624	101
Masquerade	0.912	0.962	0.936	397
Replay	0.732	0.410	0.526	100

The confusion matrix shows that 37 of 100 Replay events and 20 of 101 Injection events were classified as Normal. Another 21 Injection events were misclassified as Masquerade. These errors are important because attacks classified as Normal may remain undetected. The stacking ensemble also showed a substantial difference in Replay recall between the official split (93%) and cross-validation (34%), suggesting possible overlap between the training and test records. However, this explanation requires confirmation through direct record-level comparison. Injection recall remained low at 19.8% under both protocols.

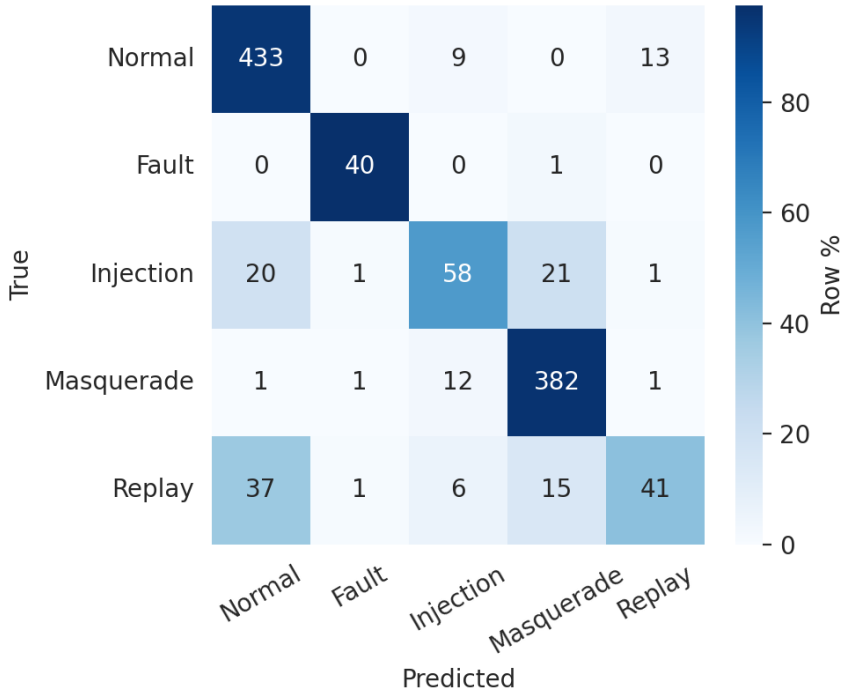


Fig.20. Confusion matrix of Gradient Boosting under 5-fold cross-validation.

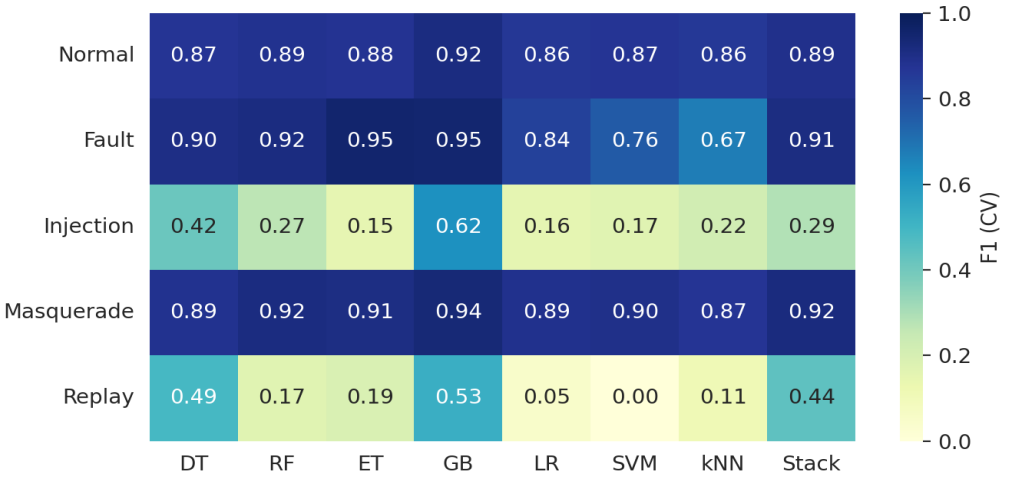


Fig.21. Per-class F1 of every model under 5-fold cross-validation.

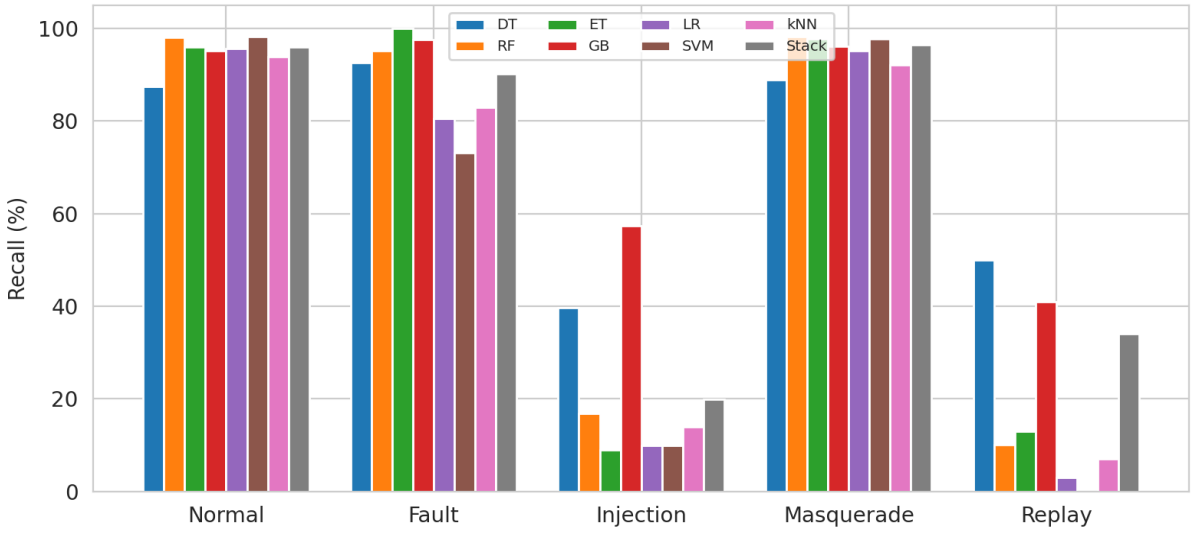


Fig.22. Per-class recall of every model under 5-fold cross-validation.

The two confusion matrices for the stacking ensemble make the effect of leakage visible. On the official split Fig. 23, Replay recall is 93% (93 of 100 events), whereas under cross-validation in Fig. 24. it is 34%. The Replay events in the official test file were already in the training data, so the model had seen them. Injection recall, in contrast, is 19.8% under both protocols, which suggests that the ensemble cannot separate Injection from Masquerade on these features even when it has seen the events.

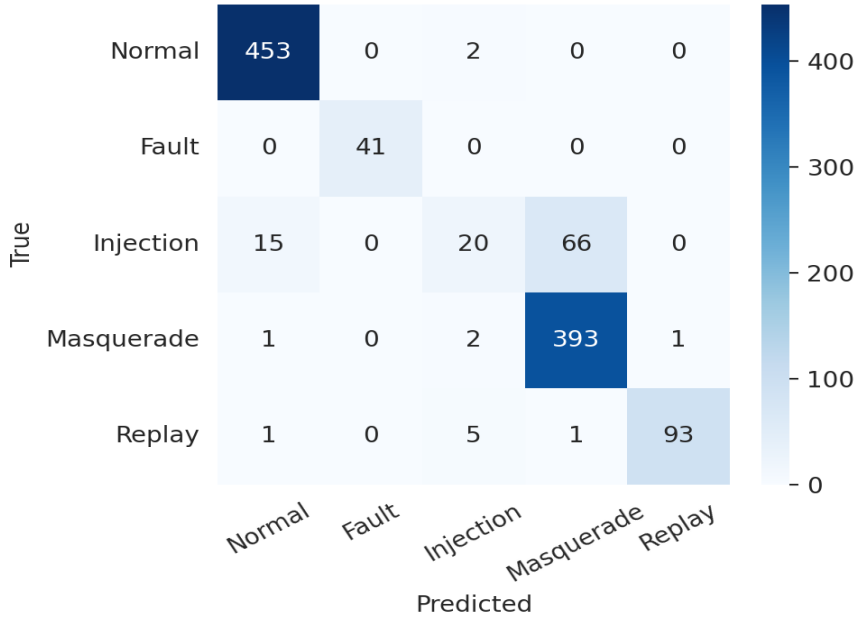


Fig.23. Confusion matrix of the stacking ensemble on the official split (counts).

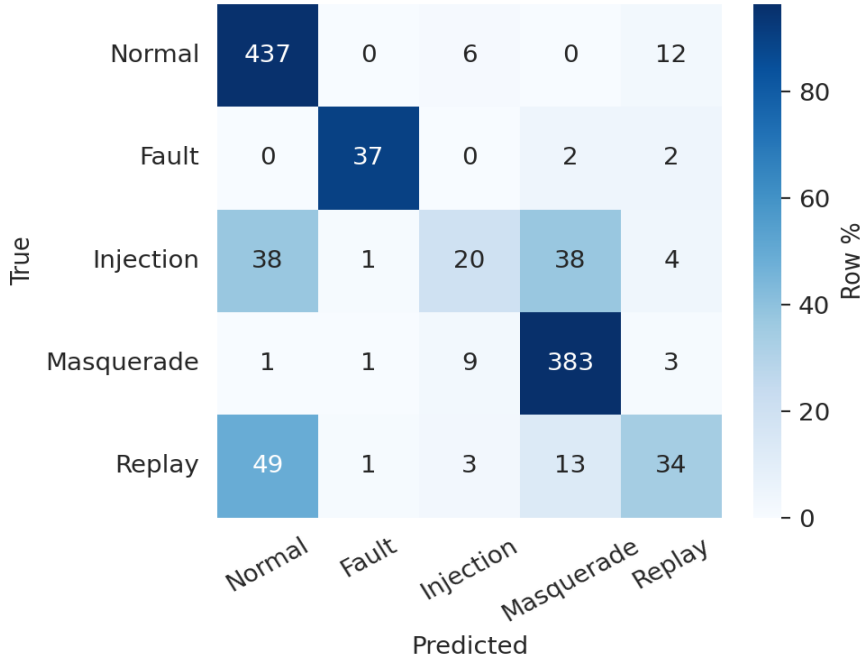


Fig.24. Confusion matrix of the stacking ensemble under 5-fold cross-validation.

3.3 Ranking quality and confidence

Table 6 presents the ROC AUC and average precision of the stacking ensemble under five-fold cross-validation. The model achieved high ROC AUC values for Fault (0.997), Masquerade (0.971), and Normal (0.963), while Replay and Injection achieved 0.847 and 0.794, respectively.

Table 6
ROC AUC and average precision of the stacking ensemble.

Class	ROC AUC	Average precision
Normal	0.963	0.925
Fault	0.997	0.970
Injection	0.794	0.350
Masquerade	0.971	0.915
Replay	0.847	0.487

The precision and recall results in Fig.25 and Fig.26 show that average precision was particularly low for Injection (0.350) and Replay (0.487), compared with Fault (0.970). This indicates that the model struggles to identify these attack classes reliably. Fig.27 also shows that some incorrect predictions had maximum probabilities between 0.85 and 1.0, meaning that high confidence did not always indicate a correct prediction.

For binary attack-versus-Normal classification, Gradient Boosting achieved 91.41% accuracy, 0.945 precision, 0.906 recall, and 0.925 F1-score. Random Forest achieved 89.03% accuracy and 0.859 recall. Thus, approximately 9.4% of attacks were missed by Gradient Boosting and 14.1% by Random Forest. These results suggest that binary classification simplifies the task but does not eliminate missed attacks.

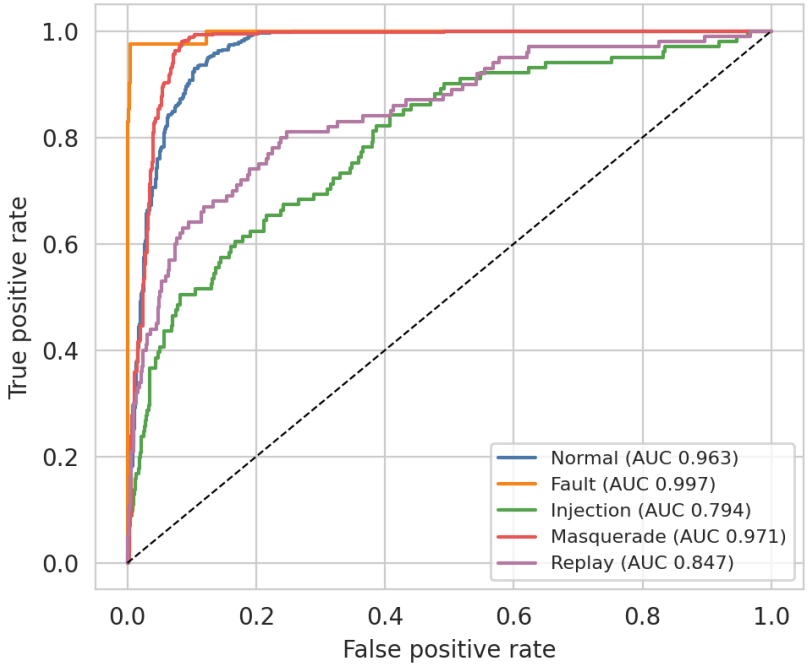


Fig.25. One-vs-rest ROC curves of the stacking ensemble under 5-fold cross-validation.

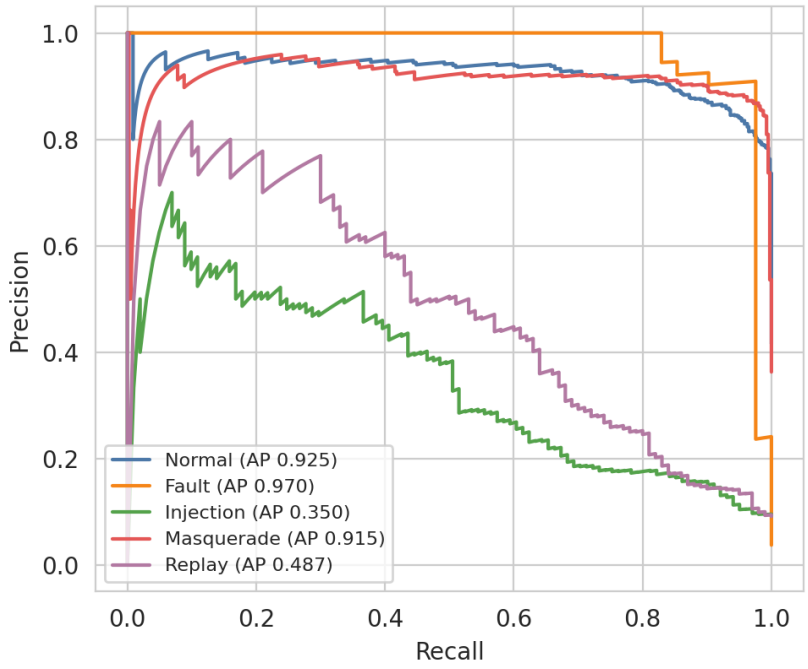


Fig.26. One-vs-rest precision-recall curves of the stacking ensemble under 5-fold cross-validation.

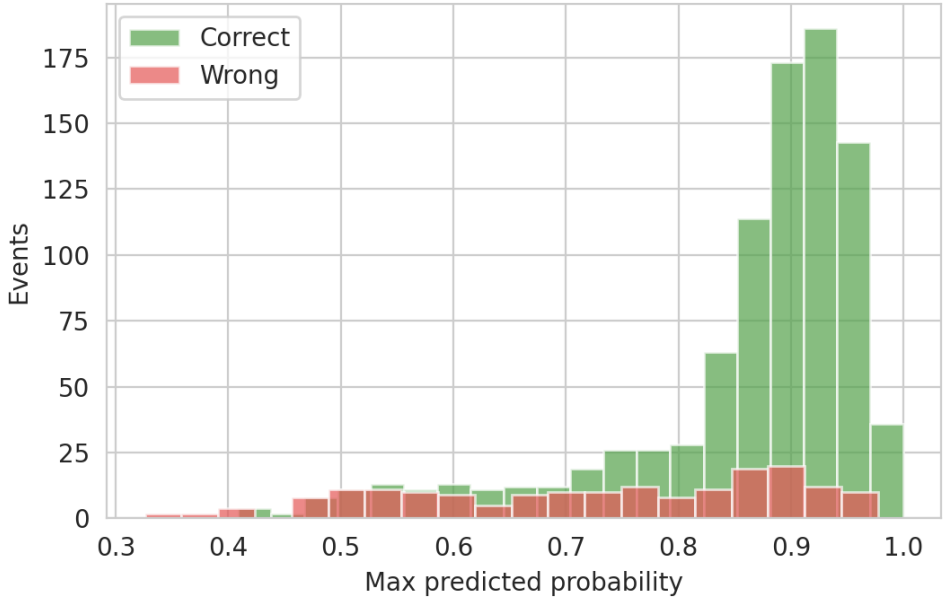


Fig.27. Distribution of the maximum predicted class probability for correct and wrong predictions of the stacking ensemble.

3.4 Feature Importance and Ablation Analysis

Fig.28 shows that Gradient Boosting relied mainly on protocol-related and consistency features. The five most important features were state_cb (0.271), consistency (0.232), stDiff (0.150), any_relay (0.098), and timeLastMsg (0.060), accounting for approximately 81% of total feature importance. Electrical features contributed substantially less to the predictions.

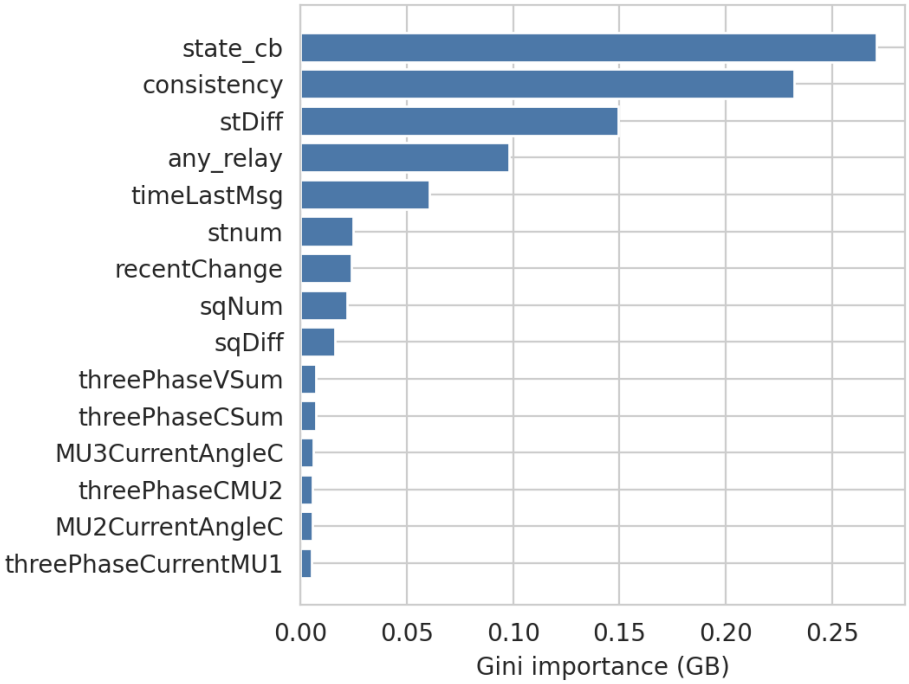


Fig.28. The fifteen most important features of Gradient Boosting.

Table 7 and Fig.29 support this finding. The seven protocol features achieved 84.28% accuracy, while combining protocol and flag features increased accuracy to 87.84% and macro-F1 to 0.806. These results slightly exceeded those of the complete 52-feature set, which achieved 87.20% accuracy and 0.791 macro-F1. Electrical features alone achieved only 42.41% accuracy and a macro-F1 score of 0.290, indicating limited discrimination among the five classes.

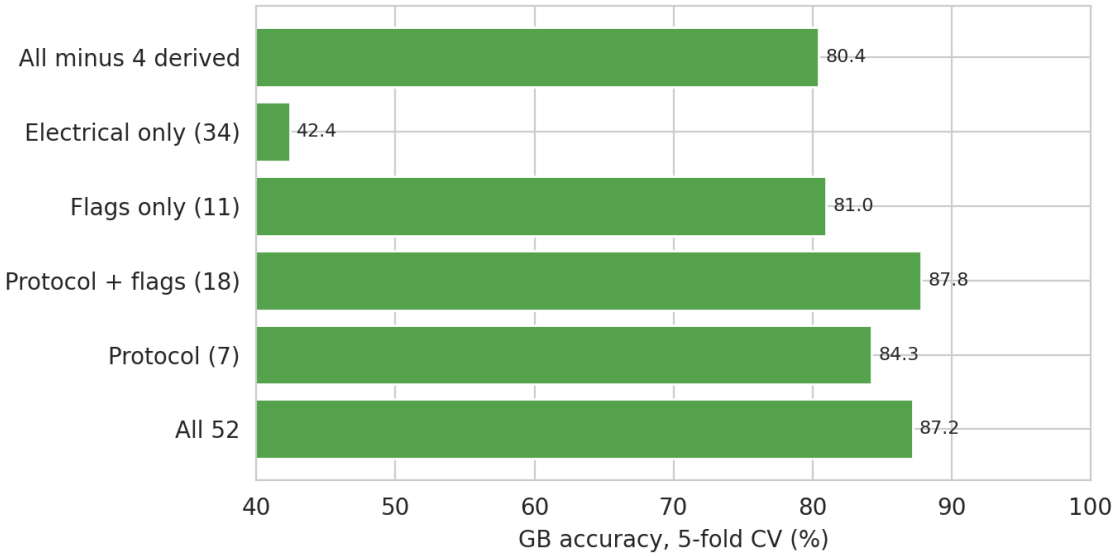


Fig.29. Accuracy of Gradient Boosting using different feature subsets under 5-fold cross-validation.

Table 7.
Feature-subset ablation results for Gradient Boosting.

Feature subset	Features	Accuracy (%)	Macro-F1
All features	52	87.20	0.791
Protocol	7	84.28	0.722
Protocol + flags	18	87.84	0.806
Flags only	11	80.99	0.541
Electrical only	34	42.41	0.290
All except four derived features	48	80.44	0.610

Removing four derived features reduced accuracy to 80.44% and macro-F1 to 0.610, suggesting that these features contain useful classification information. Overall, protocol behavior, timing, counters, and consistency were more informative than the electrical measurements in this dataset.

3.5 Learning curve and limitations

The learning curve in Fig.30 shows that training accuracy remained close to 100%, while validation accuracy increased from approximately 22% with the smallest training subset to 87% with the full training folds. Since validation performance had not clearly plateaued, additional training data may improve the model. The difference between training and validation accuracy also suggests possible overfitting.

Four limitations should be considered. First, the dataset is relatively small, particularly for Fault events, which contain only 41 samples; one event changes recall by approximately 2.44 percentage points. Second, related events from the same attack run may appear in different cross-validation folds, potentially producing optimistic estimates. Group-based cross-validation using attack-run identifiers would help address this issue. Third, default hyperparameters and the absence of class-imbalance correction may limit performance. Hyperparameter tuning, class weighting, and resampling could be investigated using leakage-aware evaluation. Fourth, all observations came from a single testbed, limiting confidence in generalization to other substations.

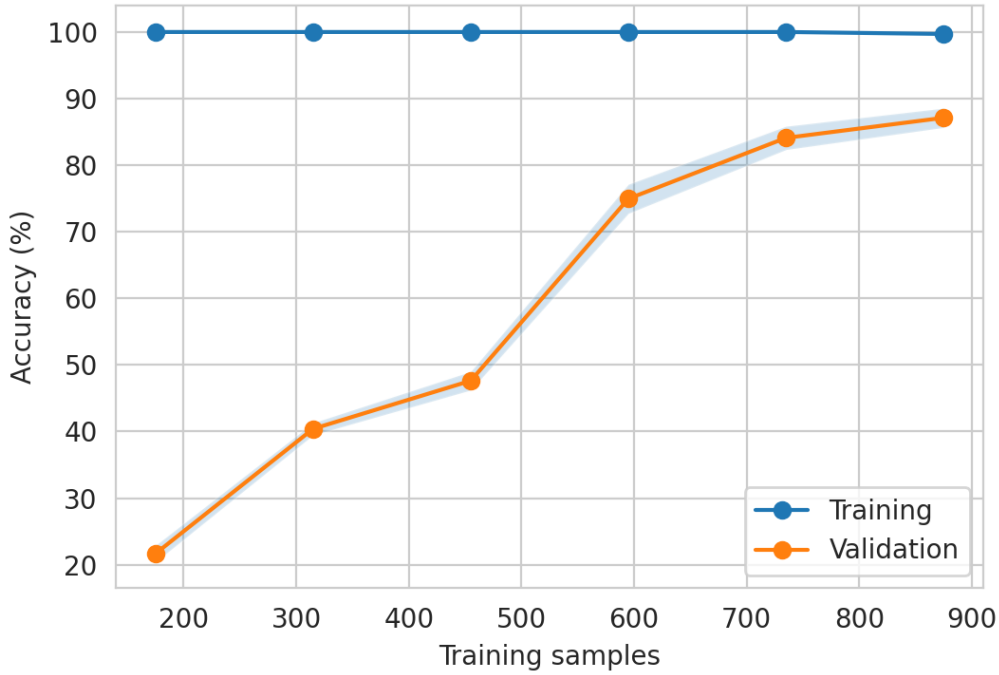


Fig.30. Learning curve of Gradient Boosting under five-fold cross-validation.

3.6 Summary of findings

This study compared seven conventional machine learning models and a stacking ensemble for classifying Normal, Fault, Injection, Masquerade, and Replay events in an IEC 61850 substation cybersecurity dataset. Although Random Forest and Extra Trees achieved accuracies above 99% on the official split, their cross-validated accuracies decreased substantially. Gradient Boosting performed best under five-fold cross-validation, achieving 87.20% accuracy, a macro-F1 score of 0.791, and an MCC of 0.809. The stacking ensemble did not improve on this performance and required more training time. Class-wise results showed reliable detection of Fault and Masquerade events but weaker detection of Replay and Injection attacks. Feature analysis identified protocol and flag features as the main sources of predictive information, while electrical features alone provided limited discrimination. These findings are specific to the evaluated dataset and testbed. Improving detection of unseen attacks will require more diverse labelled data, attack-run identifiers, temporal information, leakage-aware validation, and testing on independent testbeds. The results also show that high overall accuracy alone is insufficient to establish reliable cybersecurity detection.

4. Conclusions

This study compared seven machine learning classifiers and a stacking ensemble for detecting normal events, physical faults, masquerade attacks, replay attacks, and injection attacks in IEC 61850-based digital substations. The results highlight the importance of checking dataset integrity before evaluating model performance, as the official training and test files showed substantial overlap in the underlying events. Under stratified five-fold cross-validation, Gradient Boosting achieved the best overall performance, with 87.20% accuracy, a macro-F1 score of 0.791, and an MCC of 0.809. However, the model struggled to identify replay and injection attacks compared with physical faults

and masquerade attacks. The feature-ablation results also showed that protocol-related and flag features contributed more to classification performance than electrical features alone. The stacking ensemble did not improve on Gradient Boosting and required more training time, indicating that a more complex model does not always provide better results.

Ultimately, this research provides an actionable applied framework emphasizing data integrity and class wise evaluation for critical infrastructure cybersecurity systems, aligning with a broader body of recent studies that leverage advanced machine learning and intelligent decision frameworks to enhance resilience, forecasting, and efficiency across modern smart grid and critical infrastructure domains [12-20].

This study was limited by the small dataset, the low number of fault events, and the use of a single testbed. The lack of attack-run identifiers also prevented group-based cross-validation, so some similarities between training and validation events may remain. Future work should evaluate the proposed approach using larger datasets, independent testbeds, and more diverse attack scenarios. Further investigation of class weighting, feature selection, and temporal information may help improve the detection of replay and injection attacks. Overall, the findings show that careful dataset preparation and reliable evaluation are essential for developing machine learning-based intrusion detection systems for digital substations.

Author Contributions

Conceptualization, Md. Taukir Ahmed; methodology, Md. Taukir Ahmed; software, Md. Taukir Ahmed; validation, Md. Taukir Ahmed; formal analysis, Md. Taukir Ahmed; investigation, Md. Taukir Ahmed; data curation, Md. Taukir Ahmed; writing-original draft preparation, Md. Taukir Ahmed; writing-review and editing, Md. Taukir Ahmed; visualization, Md. Taukir Ahmed. The author has read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Data Availability Statement

The dataset used in this study is publicly available on Kaggle and contains data from an IEC 61850 digital substation testbed.

Conflicts of Interest

The author declares no conflict of interest.

Acknowledgement

The authors thank the researchers who made the IEC 61850 digital substation dataset publicly available. Their work provided the data used to develop and evaluate the machine learning models in this study.

References

[1] Kim, Y., Hakak, S., & Ghorbani, A. (2023). Smart grid security: Attacks and defence techniques. IET Smart Grid, 6(2), 103–123. https://doi.org/10.1049/stg2.12090

[2] Ding, J., Qammar, A., Zhang, Z., Karim, A., & Ning, H. (2022). Cyber threats to smart grids: Review, taxonomy, potential solutions, and future directions. Energies, 15(18), Article 6799. https://doi.org/10.3390/en15186799

[3] Ghiasi, M., Niknam, T., Wang, Z., Mehrandezh, M., Dehghani, M., & Ghadimi, N. (2023). A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future. Electric Power Systems Research, 215, Article 108975. https://doi.org/10.1016/j.epsr.2022.108975

[4] Ibrahim, N., & Kashef, R. (2025). Exploring the emerging role of large language models in smart grid cybersecurity: A survey of attacks, detection mechanisms, and mitigation strategies. Frontiers in Energy Research, 13, Article 1531655. https://doi.org/10.3389/fenrg.2025.1531655

[5] Majidi, S. H., Hadayeghparast, S., & Karimipour, H. (2022). FDI attack detection using extra trees algorithm and deep learning algorithm-autoencoder in smart grid. International Journal of Critical Infrastructure Protection, 37, Article 100508. https://doi.org/10.1016/j.ijcip.2022.100508

[6] Yu, T., Da, K., Wang, Z., Ling, Y., Li, X., Bin, D., & Yang, C. (2022). An advanced accurate intrusion detection system for smart grid cybersecurity based on evolving machine learning. Frontiers in Energy Research, 10, Article 903370. https://doi.org/10.3389/fenrg.2022.903370

[7] Berghout, T., Benbouzid, M., & Muyeen, S. M. (2022). Machine learning for cybersecurity in smart grids: A comprehensive review-based study on methods, solutions, and prospects. International Journal of Critical Infrastructure Protection, 38, Article 100547. https://doi.org/10.1016/j.ijcip.2022.100547

[8] Sahani, N., Zhu, R., Cho, J.-H., & Liu, C.-C. (2023). Machine learning-based intrusion detection for smart grid computing: A survey. ACM Transactions on Cyber-Physical Systems, 7(2). https://doi.org/10.1145/3578366

[9] Alani, M. M., & Baker, T. (2023). A survey of smart grid intrusion detection datasets. In Proceedings of the Workshop of the 19th International Conference on Intelligent Environments (IE2023) (Vol. 32, pp. 5–13). IOS Press. https://doi.org/10.3233/AISE230004

[10] Mazhar, T., Irfan, H. M., Khan, S., Haq, I., Ullah, I., Iqbal, M., & Hamam, H. (2023). Analysis of cyber security attacks and its solutions for the smart grid using machine learning and blockchain methods. Future Internet, 15(2), Article 83. https://doi.org/10.3390/fi15020083

[11] Hamdi, N. (2025). Enhancing cybersecurity in smart grid: A review of machine learning approaches. Telecommunication Systems, 88(2), 1–23. https://doi.org/10.1007/s11235-025-01308-9

[12] Şan, Y., & Şenyiğit, E. (2026). Smart Forecasting and Clustering of Renewable Energy Capabilities for Smart Grid Planning by Using Machine Learning Model. **Journal of Urban Intelligence and Smart Systems**, 1, 28-44. https://juiss.org/index.php/juiss/article/view/317

[13] Demir, A. T. (2026). Explainable Machine Learning-Guided Virtual Screening and Molecular Docking for Identification of Novel FYN Kinase Inhibitors. **Intelligent Systems Research and Applications Journal**, 2, 340-363.

[14] Ghaly, A. (2026). Governing the Intelligent University: An Empirical Analysis of Artificial Intelligence (AI) Integration Challenges in Smart City Knowledge Systems. Applied Expert Systems and Knowledge Management, 1, 32-50. https://aeskm.org/index.php/aeskm/article/view/302

[15] Kayiran, H. F. (2026). AI-Driven Predictive Composite Infrastructure for Energy-Efficient Smart Buildings. Journal of Urban Intelligence and Smart Systems, 1, 45-57. https://juiss.org/index.php/juiss/article/view/319

[16] Kayiran, H. F. (2026). Explainable Physics-Informed Deep Learning Framework for Thermoelastic Analysis of Rotating Ti-6Al-4V Disks. Applied Expert Systems and Knowledge Management, 1, 1-15. https://aeskm.org/index.php/aeskm/article/view/292

[17] Sahoo, L., & Ghosh, S. L. (2026). A Reinforcement Learning-Based Framework for Multi-Criteria Decision-Making. *Intelligent Systems Research and Applications Journal*, *2*, 129-145. https://doi.org/10.59543/ydm6xr12

[18] Jiang, W., Wang, Z., & Jamil, D. (2026). Artificial Intelligence-Driven Integration of Terrestrial and Non-Terrestrial Networks: A Perspective. *Journal of Intelligent Decision Making and Granular Computing*, *2*(1), 59-75. https://doi.org/10.31181/jidmgc21202634

[19] A Machine Learning–Based Long–Short Decision Framework for Hourly EUR/USD Forecasting under Strict Temporal Alignment. (2026). *Journal of Expert Systems and Sustainable Development*, *2*(1), 41-57. https://doi.org/10.65069/jessd21202610

[20] Badi, I., & Bouraima, M. B. (2026). Fuzzy SWARA and Its Application to Prioritize the Artificial Intelligence-Based SWOT Factors for Achieving SDG 2 (Zero Hunger). *Journal of Intelligent Decision Making and Granular Computing*, *2*(1), 186-196. https://doi.org/10.31181/jidmgc21202641