Journal of Information & Systems Management

DLINE JOURNALS

Print ISSN: 0976-4143
Online ISSN: 0976-4151

JISR 2026: 17 (4)
https://doi.org/10.6025/jisr/2026/17/4/222-236

Evaluating Cyberattack Classification Models on Network Traffic: A Chronological Holdout Study

Hajar Ait Lamkademe
University of Hassan II of Casablanca
Morocco
hajar.aitlamkademe-etu@etu.univh2c.ma

ABSTRACT

The application of machine learning (ML) to Network Intrusion Detection Systems (NIDS) is frequently undermined by methodological flaws, including temporal data leakage, target leakage, and spurious correlations. This study presents a rigorously conservative evaluation of cyberattack classification using a strict chronological 80/20 holdout design on a 40,000-record multi-class dataset comprising DDoS, Malware, and Intrusion events. To isolate genuine predictive signal and prevent leakage, post-detection alerts and high cardinality identifiers were deliberately excluded, retaining only pre-decision traffic and contextual features. Multinomial logistic regression and Extra Trees classifiers were benchmarked against a majority-class baseline. Results indicate near chance discrimination capability: the best performing model (Extra Trees) achieved an accuracy of 34.0%, a macro-F1 of 0.333, and a macro ROC AUC of 0.503, with a Matthews Correlation Coefficient near zero. Complementary statistical tests further confirmed the absence of significant associations between the retained features and attack labels. These findings suggest that the dataset exhibits synthetic like structural properties or weak label-feature dependencies, failing to support reliable chronological generalization. Ultimately, this negative result underscores the critical need for documented label provenance, richer sequential telemetry, and temporally aware evaluation frameworks to validate the operational utility of ML-based cybersecurity models.

Keywords: Network Intrusion Detection, Chronological Evaluation, Target Leakage, Machine Learning, Cybersecurity Datasets, Extra Trees

Received: 12 March 2026, Revised 8 July 2026, Accepted 25 July 2026

Copyright: **DLINE**

1. Introduction

Cybersecurity remains a major concern as networks, software, hardware, and communication architectures continue to evolve. The expansion of technologies such as the Internet of Things (IoT) has contributed to

increasingly complex and sophisticated cyberattacks, making malicious activity more difficult to identify reliably [1]. Network intrusion detection systems (NIDS) are an important component of network security because they monitor traffic for potentially malicious behaviour and help protect systems against attacks [2]. They also support the identification of unauthorized interference that has not been prevented by other security mechanisms [3].

An intrusion detection system (IDS) monitors activity within a computing environment including network traffic, system logs, and operating system calls to determine whether observed behaviour is legitimate or indicative of an attack [4]. IDSs are commonly divided into network based intrusion detection systems (NIDS) and host based intrusion detection systems (HIDS). NIDS analyse feature vectors summarising network traffic over specified intervals, whereas HIDS monitor activity on individual hosts [5]. These complementary approaches provide different perspectives on potentially malicious behaviour.

2. Background

2.1 Intrusion Detection Approaches and Machine Learning

Intrusion detection methods are generally grouped into signature based and anomaly based approaches. Signature based detection, also referred to as misuse detection, compares observed activity with predefined attack patterns. Although signature matching can maintain relatively low false alarm rates for known threats [6], its effectiveness depends on the availability and quality of signatures. Flexible signature based NIDS have therefore also been explored [7, 8]. Operational tools such as Snort and Suricata illustrate the use of signature-oriented NIDS; signature based detection can identify known attacks with few false positives [9]. However, these systems are limited when signatures for emerging, sophisticated, or zero day attacks are unavailable [10].

Anomaly based detection instead develops heuristic rules or statistical models from patterns of normal and abnormal activity, then uses these patterns to classify observed behaviour as benign or malicious. Because it is not restricted to matching previously specified signatures, anomaly detection has the potential to identify previously unseen attacks. Artificial intelligence (AI) and machine learning (ML) methods can further support this approach. The need to automate attack detection has motivated the application of ML algorithms to network security [11]. Such algorithms can learn attack related behaviours and construct nonlinear classification models that distinguish malicious traffic from normal traffic [12]. Research has consequently investigated supervised learning algorithms, ensemble methods, and deep learning architectures for classifying network traffic into multiple intrusion categories [13].

Candidate methods include neural networks and conventional classifiers such as Random Forests (RF), k-nearest neighbours (kNN), and Support Vector Machines (SVM). Their evaluation, however, depends on access to substantial and realistic datasets that represent network conditions and attack behaviours. The lack of reliable datasets has been identified as a persistent obstacle to intrusion detection research [14].

2.2 Practical Challenges in Learning-Based Intrusion Detection

Despite the promise of ML and deep learning based NIDS, practical deployment remains challenging. Models may require large numbers of malicious traffic examples and may need to be retrained when new attack types emerge. These requirements can constrain their ability to adapt promptly to changing threats in real world environments. In response to this problem, adaptive NIDS approaches have been proposed to classify newly encountered attacks and deliver timely predictions [15].

Another difficulty is the gap between statistical anomaly detection and operational security requirements. A model that performs well on a dataset may not necessarily be effective in a live adversarial environment, where false positives impose substantial costs and background traffic changes over time [16, 17]. Reviews of security focused ML research have also identified methodological problems including sampling bias, data snooping, and spurious correlations that can undermine reproducibility and lead to overly optimistic conclusions [18, 19]. These concerns make careful evaluation design essential when comparing cyberattack classification models.

2.3 Class Imbalance and Strategies for Mitigation

Class imbalance is a major challenge in ML-based NIDS because normal traffic may outnumber attack traffic by several hundred times [20, 21]. Consequently, a classifier can achieve apparently strong overall performance while failing to detect less frequent attack classes. This is particularly problematic when minority attacks are important to identify, despite their limited representation in the training data.

Existing approaches to class imbalance can broadly be divided into data level and algorithm level strategies. At the data level, the Synthetic Minority Oversampling Technique (SMOTE) generates artificial minority class examples through interpolation between existing minority observations and can reduce learning bias while improving classification performance [22]. However, in high-dimensional network traffic data, synthetic interpolation may introduce bias or overfitting. SMOTE-Tomek combines SMOTE with Tomek links in an effort to improve generalisation and address class overlap [23]. Nevertheless, such methods do not necessarily preserve the temporal structure and dependencies that characterise network traffic.

At the algorithm level, cost sensitive learning assigns greater weight to minority class examples during model training [24]. These approaches may require additional optimisation and can remain unreliable when very few examples are available for a class, such as Web attacks in CICIDS2017. More broadly, synthetic oversampling may be unsuitable for high dimensional traffic data containing complex nonlinear patterns [25]. Network flows are shaped by temporal and protocol specific dependencies that simple interpolation cannot fully represent; synthetic generation may therefore produce unrealistic combinations of flows, packet timings, and traffic features [26] [Krawczyk, B]. Accordingly, imbalance mitigation strategies should be assessed not only by aggregate performance but also by their effect on minority class detection and the realism of generated data. [27].

2.4 Research Motivation

Prior work has examined supervised ML methods for intrusion detection in industrial communication environments. For example, Asyzauri presented a comparative evaluation of supervised ML models for intrusion detection in IEC-104 communication using a laboratory supervisory control and data acquisition (SCADA) testbed and labelled datasets obtained from packet captures. This line of research highlights the importance of evaluating classifiers using traffic drawn from defined operational settings.

Taken together, the literature identifies several connected issues: signature based systems are constrained by the availability of known attack patterns; learning based systems must adapt to evolving attack behaviour; network traffic datasets frequently exhibit severe class imbalance; and evaluation results can be distorted by sampling bias, data leakage, or spurious relationships. These issues motivate rigorous comparisons of cyberattack classification models using evaluation procedures that reflect the temporal ordering of network observations. A chronological holdout design is relevant in this context because it separates training observations from later observations in time, providing a more deployment oriented test of whether learned patterns generalise to subsequent traffic.

2.5 Scope and Evaluation Considerations

The literature reviewed here supports the need for an evaluation framework that considers both classification effectiveness and the limitations of the underlying data. In particular, comparisons should account for class-wise performance rather than relying exclusively on overall accuracy, given the imbalance between benign and malicious traffic. They should also guard against sampling bias, data snooping, and spurious correlations, and should ensure that any preprocessing or resampling is applied without allowing information from the chronological test period to influence model training.

The supplied source material establishes the motivation for a chronological holdout study but does not specify the dataset, the complete set of classification models, the experimental protocol, quantitative results, or the evaluation metrics used in the study. These details should therefore be provided in the methods and results sections before the manuscript is presented as a complete empirical paper.

2.6 Research Statement

Cybersecurity remains a critical challenge as network infrastructures, software systems, and communication architectures continue to evolve. The proliferation of Internet of Things (IoT) devices and distributed services has expanded the attack surface, producing increasingly sophisticated and polymorphic threats that are difficult to identify reliably using traditional signature based mechanisms alone. Network intrusion detection systems (NIDS) form a core component of modern defence in depth strategies by continuously monitoring traffic for potentially malicious behaviour and supporting the detection of unauthorised activity that has bypassed preventive controls.

Machine learning (ML) approaches have been widely proposed as a means of overcoming the limitations of static signature matching. Supervised classifiers, ensemble methods, and deep learning architectures can, in principle, learn complex decision boundaries that separate benign traffic from multiple attack categories. In practice, however, the reliability of published performance claims is frequently undermined by methodological shortcomings: non chronological data splits that permit temporal leakage, the inclusion of post detection or label adjacent features, severe class imbalance that is masked by aggregate accuracy, and the use of datasets whose provenance, sampling frame, and label generation process remain undocumented.

This study therefore adopts a deliberately conservative evaluation design. Using a 40,000-record multi-class cyberattack dataset spanning January 2020 to October 2023, we conduct a strict chronological 80/20 holdout that trains models exclusively on earlier observations and evaluates them exclusively on later observations. Only pre-decision traffic and contextual features ports, packet length, protocol, packet type, traffic type, network segment, log source, and timestamp derived temporal components are retained as predictors. Post-event alert fields, high cardinality identifiers, and potential leakage variables are excluded. Two representative supervised models (multinomial logistic regression and Extra Trees) are compared against a majority-class baseline using a comprehensive suite of metrics that emphasise class wise performance and calibration under temporal shift.

The central research claim is that, under these constraints, the selected feature set yields essentially no reliable discrimination among DDoS, Malware, and Intrusion labels on later records. The strongest model achieves only marginal improvement over chance level performance, demonstrating that chronological generalisation cannot be assumed from the present data and feature configuration. The negative finding is itself informative: it underscores the necessity of documented label provenance, richer sequential telemetry, and evaluation designs that respect temporal ordering before claims of operational utility can be sustained.

2.7 Research Objectives

The study is guided by the following specific objectives:

1. To quantify the chronological generalisation performance of two supervised multi-class classifiers (multinomial logistic regression and Extra Trees) when trained on the first 80 % of temporally ordered records and evaluated on the final 20 % of records.

2. To isolate the predictive contribution of pre-decision traffic and context features by systematically excluding post detection, high cardinality, and potential leakage variables.

3. To characterise class wise performance (precision, recall, F1) and overall discrimination (macro-F1, Matthews correlation coefficient, macro one vs rest ROC AUC) relative to a majority class baseline.

4. To examine the statistical association between individual numeric and categorical predictors and attack type using non parametric tests and effect size measures, thereby assessing whether observable feature differences exist independent of model fitting.

5. To document data quality characteristics missingness patterns, cardinality, uniqueness of identifiers, and temporal coverage that condition the interpretability of any classification result.

2.8 Research Questions

RQ1.To what extent do multinomial logistic regression and Extra Trees, trained solely on pre-decision traffic and context features from earlier records, discriminate among DDoS, Malware, and Intrusion labels on chronologically later records?

RQ2. How do the resulting macro-F1, balanced accuracy, MCC, and macro ROC-AUC values compare with those of a majority class baseline under the same chronological split?

RQ3. Do individual numeric predictors (packet length, anomaly score, source port, destination port) exhibit statistically detectable differences across attack classes, and do categorical predictors show non-negligible association with attack type after multiple testing correction?

RQ4. What data quality and feature availability constraints limit the strength of any inference that can be drawn about operational detection capability from the present dataset?

2.9 Hypotheses

Given the synthetic looking structure of the dataset (unique source and destination IP addresses in every row, unique payload strings, near constant values in several security log fields) and the deliberate exclusion of post event features, the following directional hypotheses are advanced:

A. H1: Macro-F1 of both fitted models on the chronological holdout will not exceed 0.40, indicating limited class-wise discrimination beyond chance.

B. H2: Macro one-vs-rest ROC-AUC will be statistically indistinguishable from 0.50 for both models, consistent with absence of ranking discrimination.

C. H3: Kruskal–Wallis tests on numeric predictors will fail to reject the null of identical distributions across attack classes at $\alpha = 0.05$.

D. H4: After Benjamini–Hochberg false-discovery rate adjustment, no categorical association with attack type will retain statistical significance at FDR = 0.05.

3. Research Design

3.1 Overall Design

The study employs a non experimental, observational, comparative design. The unit of analysis is an individual network traffic record labelled with one of three attack types (DDoS, Malware, Intrusion). No experimental manipulation of traffic or labels is performed; all inference rests on the natural temporal ordering of the supplied observations.

The design is deliberately chronological rather than randomised. Records are sorted by parsed timestamp; the first 80 % (n = 32,000) constitute the training window, and the final 20 % (n = 8,000) constitute the independent test window. Training ends at 2023-01-06 10:11:14; testing begins at 2023-01-06 11:25:22. This temporal separation prevents any form of future information leakage and provides a more deployment-oriented test of whether patterns learned from earlier traffic generalise to subsequent traffic.

3.2 Sampling Frame and Dataset

The source material is a single CSV file (cybersecurity_attacks.csv) containing 40,000 rows and 25 columns, extracted from a supplied archive. Timestamps span 2020-01-01 00:43:27 to 2023-10-11 19:34:23. No exact duplicate rows were detected. Attack-class frequencies are nearly balanced:

• DDoS: 13,428 records (33.57 %)

• Malware: 13,307 records (33.27 %)

• Intrusion: 13,265 records (33.16 %)

Training-set class counts are DDoS = 10,692, Malware = 10,681, Intrusion = 10,627. Test set counts are DDoS = 2,736, Intrusion = 2,638, Malware = 2,626. The near balance removes the necessity for aggressive resampling strategies and allows macro-averaged metrics to be interpreted without severe class weight distortion.

3.3 Feature Selection and Leakage Control

A formal target leakage review was conducted prior to model fitting. The following fields were excluded from the primary predictor set because they may encode the label directly or are generated after detection or response actions:

• Attack Signature, Malware Indicators, Alerts/Warnings, Action Taken, Severity Level, Firewall Logs, IDS/IPS Alerts

In addition, high-cardinality or unique identifiers that encourage memorisation rather than generalisable learning were excluded:

• Source IP, Destination IP, Payload Data, User Information, Device Information, Geolocation, Proxy Information
The anomaly score was also excluded from the primary classifier because its provenance and availability at decision time were undocumented. The retained primary predictors are therefore:

• Source port, destination port, packet length

• Timestamp-derived hour, day-of-week, month, and year

• Protocol, packet type, traffic type, network segment, log source

3.4 Model Specification

Two supervised multi-class models were selected to represent complementary modelling paradigms:

Multinomial logistic regression with standardised numeric features and one hot encoding of low cardinality categorical features. This linear model provides a transparent baseline that assumes additive log odds contributions.

Extra Trees (Extremely Randomized Trees) with the same feature set and one hot encoding. The ensemble introduces additional randomisation in split selection relative to Random Forests and is known to be robust to noisy or weakly informative features.

Both models were embedded in scikit-learn pipelines that fit all preprocessing steps (standardisation, one-hot encoding) exclusively on the training fold, thereby preventing any information from the chronological test period from influencing transformations.

3.5 Evaluation Metrics

Because the three classes are nearly balanced, both overall and class-wise metrics are informative. The following suite was computed on the fixed chronological test predictions:

• Accuracy and balanced accuracy

• Macro-averaged precision, recall, and F1-score

• Weighted F1-score

• Matthews correlation coefficient (MCC)

• Macro one-vs-rest ROC-AUC

Macro-F1 confidence intervals were obtained by 1,000 bootstrap resamples of the fixed test predictions. These intervals capture test sample uncertainty conditional on the fitted models and the chosen split; they do not incorporate variability arising from retraining or alternative time boundaries.

4. Methodology

4.1 Data Preparation Pipeline

The following sequential steps were applied:

1. Time stamp parsing**.** All records were successfully parsed; the resulting datetime objects defined the chronological order.

2. Temporal sorting and split**.** Records were sorted ascending by timestamp. The first 32,000 observations formed the training set; the remaining 8,000 formed the test set.

3. Feature engineering. Hour, day of week, month, and year were extracted from the timestamp. No other derived features were constructed.

4. Missing value handling. Missingness was concentrated in post event fields that were already excluded. Retained predictors exhibited negligible missing rates and required no imputation for the primary models.

5. Pipeline construction. Numeric features were standardised (zero mean, unit variance) and categorical features were one hot encoded inside model specific pipelines fitted only on training data.

4.2 Statistical Association Analysis

Prior to classification modelling, exploratory association tests were performed on the full dataset (or the training portion where appropriate) to characterise univariate relationships:

Numeric predictors**.** Kruskal Wallis rank sum tests compared the distributions of packet length, anomaly score, source port, and destination port across the three attack classes.

Categorical predictors**.** Chi-square tests of independence and Cramér's V effect-size measures were computed for protocol, packet type, traffic type, network segment, log source, severity, action taken, and attack signature. Benjamini Hochberg false discovery rate adjustment was applied across the eight tests to control the expected proportion of false discoveries.

4.3 Model Training and Inference

Both models were trained once on the complete chronological training window. No hyperparameter search that used test set information was performed. Default or literature standard hyperparameter settings were retained to emphasise the contribution of the feature set rather than exhaustive tuning. Predictions on the test set were obtained in a single forward pass; predicted class labels and, where available, class probability estimates were retained for metric computation and bootstrap resampling.

4.4 Permutation Importance

To quantify the contribution of individual features to holdout performance, permutation importance was computed for the Extra Trees model on the chronological test set. Each feature column was randomly shuffled

(preserving the marginal distribution), and the resulting change in macro-F1 was recorded. The procedure was repeated across multiple permutations; the mean decrease in macro-F1 constitutes the importance estimate. Values near zero indicate that the feature contributed little measurable signal under the chronological evaluation.

4.5 Reproducibility Measures

All preprocessing, model fitting, and metric computation steps were implemented in Python using scikit learn and standard scientific libraries. Random seeds were fixed where stochasticity was present (Extra Trees split selection, bootstrap resampling). The chronological split boundary is fully determined by the sorted time stamps, rendering the train test partition deterministic once the 80/20 proportion is fixed.

4.6 Dataset Characteristics and Data Quality

Missingness was heavily concentrated in five fields: Alerts/Warnings (50.17%), IDS/IPS Alerts (50.13%), Malware Indicators (50.00%), Firewall Logs (49.90%), and Proxy Information (49.63%). The first four of these fields each contained only a single distinct non missing value when present, indicating that they function as sparse binary presence indicators rather than multi valued descriptive attributes. All 40,000 source IP addresses and all 40,000 destination IP addresses were unique; Payload Data was likewise unique in every row. User Information and Device Information exhibited high cardinality. These structural properties are incompatible with typical real world network traffic, which normally contains repeated entities and flow level dependencies, and they constrain the strength of any claim that models have learned generalisable attack signatures.

5. Results

This section details the empirical findings of the chronological holdout evaluation, beginning with an examination of the dataset's descriptive characteristics and the distribution of the retained numeric predictors. Before assessing the predictive performance of the machine learning models, it is essential to establish a baseline understanding of the feature space and the inherent separability of the classes. Table 1 provides a comprehensive summary of the three attack categories (DDoS, Malware, and Intrusion), presenting the record counts alongside the mean and standard deviation of key numeric variables, including packet length, anomaly score, source port, and destination port. This initial descriptive overview highlights the central tendencies and variances within each class, setting the stage for the subsequent formal statistical testing and multiclass classification results.

Attack Type n-	packet_ length_ mean	packet_ length_ sd	anomaly_ mean	anomaly _ sd	source_ port_ mean	destination_ port_ mean
DDoS	13428785.3487489	416.6854121	50.2350633	28.85746457	32944.89693	33207.36394
Intrusion	13265780.9243875	418.2099362	49.9780294	28.81540244	32987.20822	33227.40588
Malware	13307778.0479447	413.2211274	50.12579394	28.88931482	32979.24889	33017.56399

Table 1. Class Summaries

5.1 Classification performance on later records

The majority class baseline achieved accuracy=0.342, balanced accuracy=0.333, and macro-F1=0.170. Logistic Regression achieved accuracy=0.333, balanced accuracy=0.333, macro-F1=0.302 (95% bootstrap CI 0.292–0.312), and macro one vs rest ROC-AUC=0.498. Extra Trees achieved accuracy=0.340, balanced

accuracy = 0.339, macro F1=0.333 (95% boot strap CI 0.323–0.344), and macro ROC-AUC=0.503. Extra Trees was the strongest of the two fitted models by macro F1, but its improvement over the baseline was small, and its ROC-AUC was essentially 0.5.

For Extra Trees, class level F1 was 0.400 for DDoS, 0.270 for Intrusion, and 0.329 for Malware. DDoS recall was 0.457, whereas Intrusion recall was only 0.232. The confusion matrix showed substantial cross class errors. Logistic Regression produced macro F1=0.302 and MCC approximately zero (-0.001); Extra Trees produced MCC=0.009. These values indicate that the models extracted little useful separation from the selected predictors under chronological evaluation.

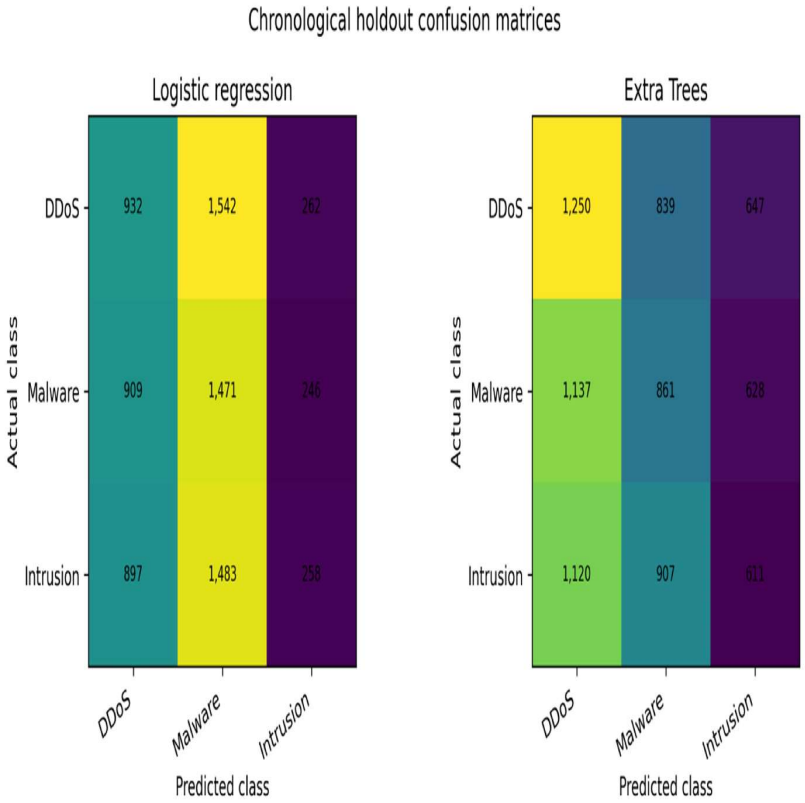

Figure 1. Confusion matrices for Logistic Regression and Extra Trees on the chronological hold out
Rows are actual classes and columns are predicted classes

5.2 Descriptive and inferential patterns

Mean packet length was similar across DDoS (785.35), Malware (778.05), and Intrusion (780.92). The Kruskal–Wallis test did not indicate a statistically detectable difference across classes (H=2.097, p=0.351). Anomaly-score means were also very similar (50.235, 50.126, and 49.978, respectively; H=0.536, p=0.765). Source-port and destination port distributions did not differ significantly by attack class (p=0.980 and p=0.591, respectively).

Across the eight categorical association tests, no association remained statistically significant after Benjamini–Hochberg adjustment. Packet Type had an unadjusted p-value of approximately 0.050 but a negligible Cramér's V=0.012 and adjusted p=0.400. Protocol, traffic type, network segment, log source, severity, action taken, and attack signature likewise showed negligible association with Attack Type in these tabulations. Statistical significance should not be interpreted as operational usefulness; effect sizes here were very small.

5.3 Permutation importance

On the test set, packet type, traffic type, log source, destination port, and day of week had the largest positive

permutation importance estimates for Extra Trees. However, the absolute changes in macro-F1 were small (the largest mean decrease was approximately 0.010 for packet type), and several features had estimates near or below zero. This is consistent with a weak predictive signal rather than a robust set of attack discriminators.

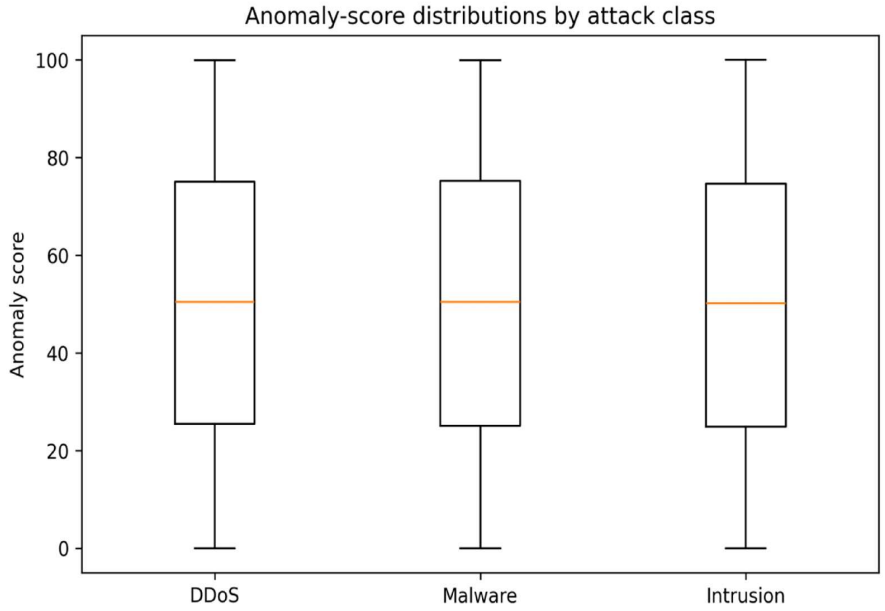

Figure 2. Anomaly-score distributions by attack class. Overlapping distributions are consistent with the non-significant Kruskal–Wallis result

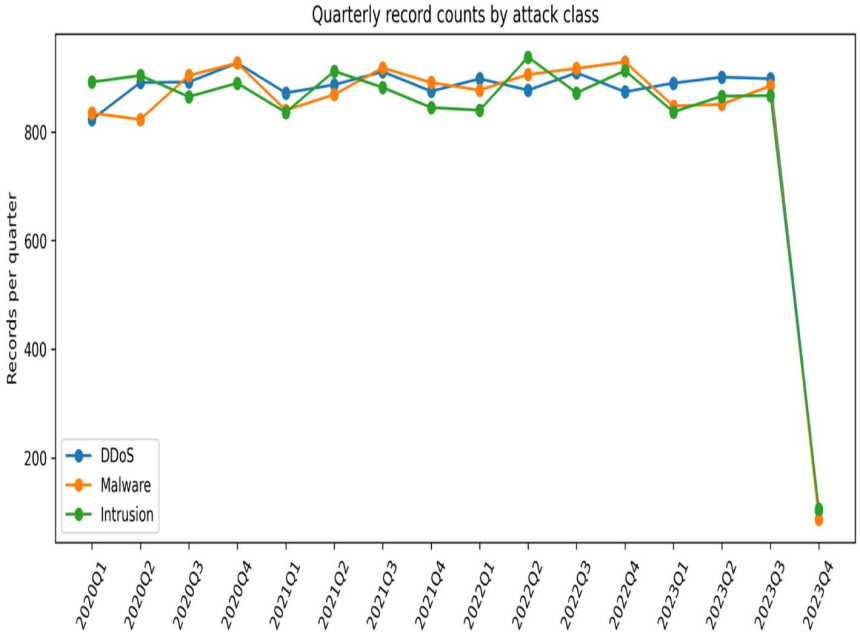

Figure 3. Quarterly record counts by attack class. Counts describe the supplied dataset and should not be interpreted as real-world attack incidence without evidence about data collection and sampling

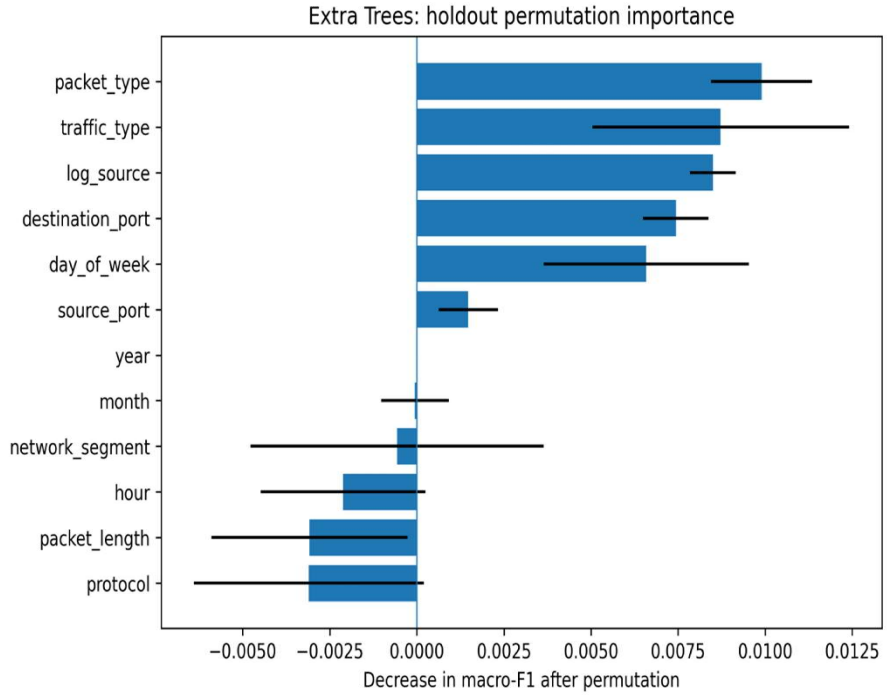

Figure 4. Extra Trees permutation importance on the chronological holdout. Values near zero indicate little measurable contribution to holdout macro-F1

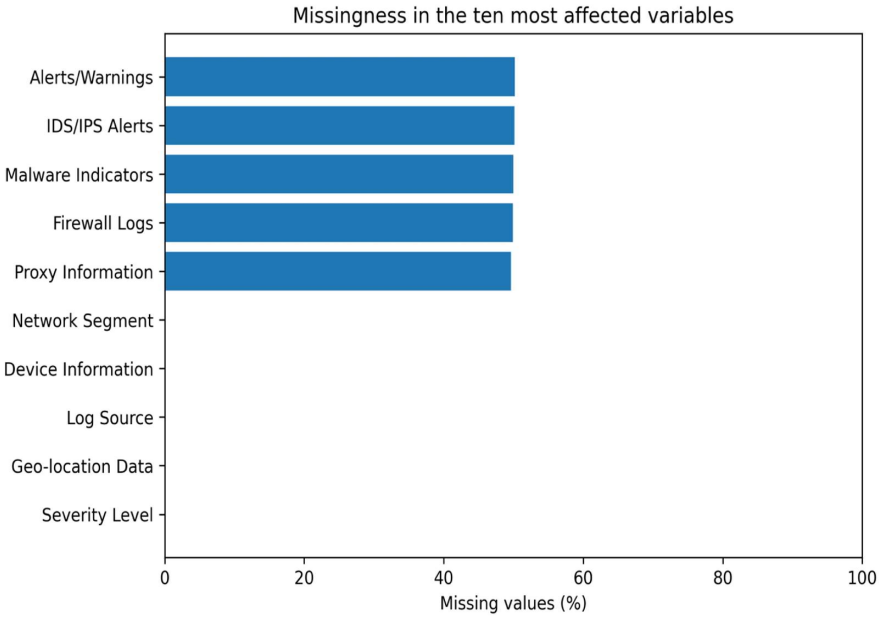

Figure 5. Missingness in most affected variables

Figure 5 displays the proportion of missing values for the variables with the highest rates of incompleteness in the 40,000-record cybersecurity_attacks dataset. Missingness is heavily concentrated in five fields: Alerts/ Warnings (20,067 missing; 50.17 %), IDS/IPS Alerts (20,050; 50.13 %), Malware Indicators (20,000; 50.00 %), Firewall Logs (19,961; 49.90 %), and Proxy Information (19,851; 49.63 %).

The first four of these fields each contain only a single distinct non missing value when present. This pattern indicates that they function as sparse binary presence indicators (i.e., the value is recorded only when an alert, indicator, or log entry exists) rather than as multi valued descriptive attributes. Proxy Information shows a similar high missing rate but is additionally high cardinality when observed.

The extreme and systematic missingness has two important consequences for modelling. First, these fields cannot be treated as ordinary continuous or multi category predictors without introducing substantial imputation artefacts or selection bias. Second, several of them (Alerts/Warnings, IDS/IPS Alerts, Malware Indicators, Firewall Logs) are post detection or label adjacent and were therefore excluded from the primary feature set to avoid leakage. Their near -50 % missing rates further underscore that any analysis relying on them would be evaluating a post event reconstruction task rather than a realistic pre decision detection problem. The figure therefore visually confirms both the data quality limitation and the deliberate feature-selection decision described in the methods.

6. Discussion

The central empirical finding is negative yet practically informative: under a strict chronological 80/20 holdout, the selected pre decision traffic and context features (source/destination ports, packet length, protocol, packet type, traffic type, network segment, log source, and timestamp derived temporal components) provide essentially no reliable discrimination among the three attack classes DDoS, Malware, and Intrusion. Extra Trees, the stronger of the two models examined, reached only 34.0 % accuracy, a macro F1 of 0.333 (95 % bootstrap CI 0.323–0.344), and a macro one vs rest ROC-AUC of 0.503. These figures are only marginally above the majority class baseline and yield a Matthews correlation coefficient near zero (0.009). Class wise performance was uneven, with DDoS achieving a higher F1 (0.400) than Intrusion (0.270), yet the overall pattern remains one of weak separation and substantial cross class confusion.

Several complementary lines of evidence support this conclusion. Numeric variables (packet length, anomaly score, source and destination ports) showed no statistically detectable differences across attack classes by Kruskal–Wallis tests. After Benjamini–Hochberg adjustment, none of the eight categorical association tests retained significance, and all effect sizes (Cramér's V) were negligible. Permutation importance on the holdout set similarly produced only tiny absolute changes in macro-F1 (largest mean decrease » 0.010). Collectively, these results indicate that the observed features contain little usable signal for chronological generalisation.

Three non mutually exclusive explanations are consistent with the data. First, the supplied labels may be only weakly (or spuriously) related to the recorded traffic fields. Second, the dataset exhibits synthetic looking structure unique source and destination IP addresses in every row, unique payload strings, and near constant values in several security log fields which is incompatible with real network traffic that normally contains repeated entities and flow level dependencies. Third, many potentially informative fields are either post-event metadata or high cardinality identifiers that encourage memorisation rather than generalisable learning; their deliberate exclusion was necessary to avoid evaluating label reconstruction rather than detection. The anomaly score distributions likewise show almost complete overlap across classes, reinforcing that this variable cannot be treated as a validated threat indicator without further documentation of its generation process.

These findings align with broader methodological concerns in the intrusion detection literature: class imbalance (here largely mitigated by the near balanced labels), the risk of data leakage and sampling artefacts, and the gap between statistical performance on a static dataset and operational utility under temporal drift. The chronological holdout design was chosen precisely to expose such gaps; the near chance results demonstrate that the gap is large for the feature set examined.

7. Limitations and reproducibility

- The archive contains only a README and one CSV; no data dictionary, collection protocol, provenance, or label generation method was provided.

• Chronological holdout tests later timestamps, but the dataset does not establish that records represent operationally independent traffic or a real production deployment.

• Time stamp derived features can capture dataset generation or sampling artifacts. Their contribution should be interpreted cautiously.

• The 1,000-resample bootstrap intervals condition on the fixed holdout and fitted predictions; they do not capture variability from retraining, alternate time boundaries, or external datasets.

• The analysis is observational. Association tests do not establish causality, and recorded response actions do not prove mitigation effectiveness.

• The primary model intentionally excludes high cardinality identifiers, payload text, and post detection fields. Different feature availability assumptions would constitute a different research task and must be evaluated separately.

8. Conclusion

This study conducted a rigorously conservative evaluation of machine learning based cyberattack classification, employing a strict chronological 80/20 holdout design on a 40,000 record multi-class dataset. By deliberately excluding post detection alerts, high-cardinality identifiers, and potential leakage variables, the research isolated the genuine predictive signal of pre-decision traffic and contextual features. The central finding is fundamentally negative yet methodologically critical: the retained features provide essentially no reliable discrimination among DDoS, Malware, and Intrusion events on chronologically later records.

The best-performing model, Extra Trees, achieved an accuracy of 34.0%, a macro-F1 of 0.333 (95% bootstrap CI 0.323–0.344), and a macro ROC-AUC of 0.503. These metrics are statistically indistinguishable from a majority-class baseline, indicating near-chance discrimination. This lack of predictive power was further corroborated by complementary statistical analyses; Kruskal–Wallis tests and chi-square association tests with Benjamini Hochberg adjustments confirmed the absence of significant univariate relationships between the retained predictors and the attack labels, while permutation importance scores on the holdout set remained negligible.

These results are heavily conditioned by the structural properties of the dataset, which exhibits synthetic-like characteristics such as universally unique IP addresses and payloads alongside systematic, near-50% missingness in critical sparse security-log fields. Consequently, the study fails to demonstrate reliable chronological generalization, exposing a critical gap between the theoretical promise of ML-based NIDS and the practical realities of dataset provenance, feature utility, and temporal drift.

Ultimately, this negative result underscores a vital methodological imperative for cybersecurity research: high benchmark accuracy on static, randomized splits is insufficient without rigorous temporal evaluation and transparent data provenance. The findings demonstrate that claims of operational utility cannot be sustained without addressing data leakage, the synthetic nature of training environments, and the reliance on post event metadata. Future work must prioritize the use of rich, validated sequential telemetry that preserves real world entity dependencies, coupled with strictly chronological evaluation frameworks. Until models can demonstrably generalize across temporal shifts using authentic, well documented operational data, their effectiveness in real world intrusion detection remains unsupported.

References

[1] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525–41550.

[2] Diana, L., Dini, P., Paolini, D. (2025). Overview on intrusion detection systems for computer networking security. *Computers, 14*, Article 87.

[3] Hajj, S., El Sibai, R., Bou Abdo, J., Demerjian, J., Makhoul, A., Guyeux, C. (2021). Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets. *Transactions on Emerging Telecommunications Technologies, 32*, e4240.

[4] Debar, H., Dacier, M., Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer Networks, 31*, 805–822.

[5] Rashid, A., Siddique, M. J., Ahmed, S. M. (2020). Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system. In *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)* (p. 1–9). IEEE.

[6] He, H., Sun, X., He, H., Zhao, G., He, L., Ren, J. (2019). A novel multimodal sequential approach based on multi-view features for network intrusion detection. *IEEE Access, 7*, 183207–183221.

[7] Papamartzivanos, D., Gómez Mármol, F., Kambourakis, G. (2019). Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE Access, 7*, 13546–13560.

[8] Hu, Z., Li, Z., Wu, J. (2008). A novel network intrusion detection system (NIDS) based on signatures search of data mining. In *Proceedings of the First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008)* (p. 10–16). IEEE.

[9] Qutqut, M. H., Ahmed, A., Taqi, M. K., Abimanyu, J., Ajes, E. T., Alhaj, F. (2026). A comparative evaluation of Snort and Suricata for detecting data exfiltration tunnels in cloud environments. *Journal of Cybersecurity and Privacy, 6*, Article 17.

[10] Masoodi, F. S. (2026). Intrusion detection in the age of deep learning: An introduction. In *Deep learning intrusion detection: Techniques and applications* (p. 1–23).

[11] Zhen, L., Kamarudin, N. H., Kok, V. J., Qamar, F. (2025). Anomaly detection model in network security situational awareness based on machine learning: Limitation, techniques, and future trends. *IEEE Access, 13*, 126084–126129.

[12] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies, 32*, e4150.

[13] Kocher, G., Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: Recent developments and challenges. *Soft Computing, 25*, 9731–9763.

[14] Ma³owidzki, M., Berezinski, P., Mazur, M. (2015). Network intrusion detection: Half a kingdom for a good dataset. In *Proceedings of the NATO STO SAS-139 Workshop*. https://pdfs.semanticscholar.org/b39e/0f1568d8668d00e4a8bfe1494b5a32a17e17.pdf

[15] Di Monda, D., Montieri, A., Persico, V., Voria, P., De Ieso, M., & Pescapè, A. (2024). Few-shot class-incremental learning for network intrusion detection systems. *IEEE Open Journal of the Communications Society, 5*, 6736–6757. https://doi.org/10.1109/OJCOMS.2024.3481895.

[16] Roy, U. S., Minu, M. J. (2026). Unmasking shortcut learning in IoT intrusion detection: A forensic, multi-paradigm evaluation of feature dependence and data leakage (arXiv:2609.28725). arXiv. https://arxiv.org/abs/2609.28725

[17] Sommer, R., Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy (S&P)* (pp. 305–316). IEEE.

[18] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, R., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In *31st USENIX Security Symposium (USENIX Security 22)* (p. 3971–3988). USENIX Association.

[19] Apruzzese, G., Laskov, P., de Oca, E. M., Malliet, W., Radosz, £. B., Conti, M. (2023). The role of machine learning in cybersecurity. *Digital Threats: Research and Practice, 4*(1), 1–38.

[20] Zhang, Y., Muniyandi, R. C., Qamar, F. (2025). A review of deep learning applications in intrusion detection systems: Overcoming challenges in spatiotemporal feature extraction and data imbalance. *Applied Sciences, 15*, Article 1552.

[21] Hussain, M. Z., Hanapi, Z. M., Abdullah, A., Hussin, M., Ninggal, M. I. H. (2025). Hybrid-CNNTree: A convolutional neural network and decision tree fusion model for wormhole attack detection. *IEEE Access, 13*, 186811–186833.

[22] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73*, 220–239.

[23] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

[24] Batista, G. E., Prati, R. C., Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter, 6*, 20–29.

[25] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting, and hybridbased approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*, 463–484.

[26] Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence, 5*, 221–232.

[27] Asyzauri, S., Affandi, A., Mukti, P. H., Murdani, A., Mukti, F. S., & Hamid, A. (2026). Comparative study of machine learning models for intrusion detection in SCADA communication based on IEC 60870-5-104. In *2026 International Seminar on Intelligent Technology and Its Applications (ISITIA)* (p. 430–435). IEEE. https://doi.org/10.1109/ISITIA71267.2026.11642359.