Wadi Alshatti University Journal of
Pure and Applied Sciences

مجلة جامعة وادي الشاطئ
للعلوم البحتة والتطبيقية

Volume 4, No. 2, July-December 2026 Online ISSN: 3006-0877 المجلد 4، الاصدار 2، يوليو- ديسمبر 2026

RESEAECH ARTICLE **COMPUTER NETWORKS AND CYBERSECURITY**

Feature-Selected Machine Learning for 5G Intrusion Detection: Leakage-Aware Evaluation and Sensitivity to Sequence Metadata in 5G-NIDD

Abraheem Sulayman Alsaedi Abraheem[1,*], Ali Saed Riheel Arhoumah[2], Hamza Saleh Ali Fokla[3], Majeda M. Abdosalam Basheer[3]

[1]Libyan Center for Studies & Researches in Environmental Science & Technology, Brak Alshati, Libya
[2]Department of Electrical and Electronic Technologies, Higher Institute of Sciences and Technology, Tamzawah Alshati, Libya
[3]Electrical and Electronic Engineering Department, Faculty of Engineering, Wadi Alshatti University, Brak Alshati, Libya

ARTICLE HISTORY
Received 18 May 2026
Revised 30 June 2026
Accepted 12 July 2026
Online 18 July 2026

KEYWORDS
5G-NIDD;
Intrusion detection;
Feature selection;
Data leakage;
Random Forest;
XGBoost;
Acquisition metadata

ABSTRACT

Fifth-generation networks present a multifaceted and software-centric attack surface, thereby necessitating a systematic approach to reproducible intrusion-detection evaluation through traffic data gathered from authentic 5G environments. This investigation examines binary intrusion detection utilizing the comprehensive 5G-NIDD Combined.csv dataset acquired from its official IEEE DataPort distribution, adhering to a standardized leakage-aware methodology. The original dataset comprised 1,215,890 records and 52 columns. Following the elimination of 21 exact duplicate entries, 1,215,869 observations were retained. A stratified 70:30 partition was executed at the record level prior to preprocessing, with all data-dependent procedures being exclusively calibrated on the pertinent training dataset. The application of zero-variance filtering, Pearson-correlation reduction, and ANOVA F-test ranking condensed the 46 usable predictors into a consistent Top-10 representation. Random Forest, XGBoost, and a scalable random Fourier feature support-vector machine (RFF-SVM) approximation of an RBF-kernel classifier were assessed employing an untouched holdout partition in conjunction with five-fold stratified cross-validation, incorporating fold-specific preprocessing and feature selection. XGBoost and Random Forest attained Top-10 holdout accuracies of 99.9748% and 99.9745%, respectively, with no statistically significant disparity between their paired errors subsequent to Holm correction. The full, correlation-reduced, and Top-10 representations yielded virtually identical outcomes for both ensemble models, suggesting that the compact feature set maintained rather than enhanced random-split performance. Nevertheless, Seq and Offset emerged as the two highest-ranked predictors. The exclusion of these predictors resulted in a decline in holdout accuracy to 71.18% for XGBoost, 71.16% for Random Forest, and 73.09% for RFF-SVM, with a similar trend noted during cross-validation. These results illustrate that nearly flawless record-level outcomes are heavily contingent upon acquisition-order metadata and should not be construed as indicative of cross-session or cross-environment generalization. Prior to operational implementation, session-aware, base-station-aware, and external validation are imperative.

التعلم الآلي المعتمد على اختيار الخصائص لكشف التسلل في شبكات الجيل الخامس: تقييم مضاد لتسرب البيانات وتحليل الحساسية للبيانات الوصفية التسلسلية في NIDD-G5

إبراهيم سليمان الساعدي إبراهيم[1,*]، علي سعيد رحيل أرحومة[2]، حمزة صالح علي فوكلا[3]، ماجدة م. عبدالسلام بشير[3]

الملخص

تُشكّل شبكات الجيل الخامس سطح هجوم متعدد الأبعاد وقائمًا بدرجة كبيرة على البرمجيات، مما يستلزم اتباع منهجية منظّمة وقابلة لإعادة الإنتاج لتقييم أنظمة كشف التسلل، بالاعتماد على بيانات حركة مرور جُمعت من بيئات جيل خامس حقيقية. تبحث هذه الدراسة في الكشف الثنائي عن التسلل باستخدام ملف Combined.csv الكامل من مجموعة بيانات NIDD-G5، الذي تم الحصول عليه من توزيعه الرسمي عبر منصة IEEE DataPort، وذلك وفق منهجية موحّدة تراعي منع تسرّب البيانات. اشتملت مجموعة البيانات الأصلية على 1,215,890 سجلًا و52 عمودًا. وبعد حذف 21 سجلًا مكررًا تكرارًا تامًا، تم الاحتفاظ بـ 1,215,869 مشاهدة. وأُجري تقسيم طبقي بنسبة 70:30 على مستوى السجلات قبل تنفيذ المعالجة المسبقة، مع ضبط جميع الإجراءات المعتمدة على البيانات حصريًا باستخدام مجموعة التدريب ذات الصلة. أدى تطبيق مرشح التباين الصفري، وتقليل الخصائص استنادًا إلى ارتباط بيرسون، وترتيب الخصائص باستخدام اختبار test-ANOVA F إلى تقليص عدد المتنبئات القابلة للاستخدام من 46 متنبئًا إلى تمثيل ثابت يضم أفضل عشر خصائص Top-10. وتم تقييم نماذج Random Forest وBoostXG، إلى جانب تقريب قابل للتوسع لآلة

الكلمات المفتاحية

شبكات الجيل الخامس
كشف التسلل
اختيار الخصائص
تسرب البيانات
الغابة العشوائية
XGBoost
البيانات الوصفية التسلسلية

*Corresponding author
https://doi.org/10.63318/waujpasv4i2_14

المتجهات الداعمة باستخدام خصائص فوريه العشوائية SVM-RFF لمحاكاة مصنف ذي نواة دالة الأساس الشعاعي RBF، وذلك باستخدام مجموعة اختبار نهائية لم تُستخدم أثناء التطوير، إلى جانب التحقق المتقاطع الطبقي بخمس طيات، مع إعادة تطبيق المعالجة المسبقة واختيار الخصائص بصورة مستقلة داخل كل طية. حقق نموذجا XGBoost وRandom Forest، باستخدام أفضل عشر خصائص، دقة على مجموعة الاختبار بلغت 99.9748% و99.9745% على التوالي، دون وجود فرق ذي دلالة إحصائية بين أخطائهما المزدوجة بعد تطبيق تصحيح mHol. كما حققت التمثيلات الكاملة، والمخفضة بالارتباط، والمقتصرة على أفضل عشر خصائص نتائج متطابقة تقريبًا لكلا نموذجي التجميع، مما يشير إلى أن مجموعة الخصائص المدمجة حافظت على أداء التقسيم العشوائي بدلًا من تحسينه. ومع ذلك، ظهرت الخاصيتان Seq وOffset بوصفهما أعلى المتنبئات ترتيبًا. وقد أدى استبعادهما إلى انخفاض دقة مجموعة الاختبار إلى 71.18% لنموذج XGBoost، و71.16% لنموذج Random Forest، و73.09% لنموذج RFF-SVM، مع ملاحظة اتجاه مماثل أثناء التحقق المتقاطع. توضح هذه النتائج أن الأداء شبه المثالي المحقق على مستوى السجلات يعتمد بدرجة كبيرة على البيانات الوصفية المرتبطة بترتيب جمع البيانات، ولذلك لا ينبغي تفسيره على أنه دليل على القدرة على التعميم عبر جلسات أو بيئات مختلفة. وقبل التطبيق التشغيلي، يصبح من الضروري إجراء تحقق يراعي الجلسات ومحطات القاعدة، إلى جانب التحقق الخارجي

Introduction

Fifth-generation mobile systems have been systematically engineered to facilitate enhanced mobile broadband, massive machine-type communications, and ultra-reliable low-latency communications, thereby enabling a plethora of services including industrial automation, interconnected transportation, intelligent healthcare, and expansive Internet of Things implementations [1]. The 5G core architecture adopts a service-oriented paradigm and accommodates modular network functionalities, slicing, virtualization, and edge computing integration [2].

This inherent flexibility amplifies the quantity of software-driven components, interfaces, and trust relationships necessitating protection. The European Union Agency for Cybersecurity (ENISA) delineates threats across radio access networks, core networks, virtualization environments, cloud infrastructures, and management assets [3]. The security architecture established by the 3rd Generation Partnership Project (3GPP) delineates protocols for authentication, authorization, privacy, and inter-network safeguarding pertinent to 5G systems [4]. Furthermore, the National Institute of Standards and Technology (NIST) has disseminated principles of 5G security design tailored for both commercial and private implementations [5].

Intrusion detection serves as a vital adjunct to preventive security measures by discerning malevolent or anomalous activities within surveilled traffic. Denning's seminal model laid the groundwork for the behavioral framework underlying anomaly detection [6]. Modern machine-learning-based intrusion detection systems extend this concept to accommodate high-volume flow data; however, their effectiveness is influenced by dimensionality and class imbalance [7].

The historical dearth of publicly accessible native 5G traffic has constrained the ability to conduct reproducible evaluations. The peer-reviewed 5G-NIDD descriptor elucidates a comprehensively labeled dataset generated within a functional 5G testing environment [8], while the original investigation delineates benign traffic patterns, denial-of-service attack simulations, and scanning scenarios, alongside providing a baseline for machine-learning experiments [9].

Recent studies report that 5G-NIDD can achieve very high performance across a wide range of models. Studies comparing classifiers designed to control data leakage stress that data-dependent preprocessing must be kept separate from the evaluation dataset [10]. Work on edge-computing applications has combined feature pruning with chronological or session-disjoint validation methods [11], while studies on feature selection in 5G core traffic show that using a smaller set of predictors can greatly reduce computational cost [12]. Other studies have examined visualization and dimensionality reduction [13], deep-learning methods [14], data-augmentation techniques [15], robustness to adversarial attacks [16], automated model generation [17], conditional neural architectures [18], temporal interpretability [19], and conventional supervised-learning methods [20].

Despite these strong benchmark results, their meaning depends heavily on the way the experiment is designed. If preprocessing or feature selection is carried out before the evaluation split, information from the test dataset may unintentionally affect model development. In addition, random record-level partitioning may place observations from the same data-collection process in both the training and evaluation subsets. This may allow models to learn patterns associated with the collection procedure rather than attack behavior that can generalize to other settings. Therefore, all data-dependent operations, including feature selection and parameter tuning, should be fitted only on the relevant training data and repeated within each cross-validation fold when evaluating the full modeling pipeline [21]. Cawley and Talbot further showed that overfitting the model-selection criterion can introduce substantial selection bias into performance evaluation [22].

This investigation introduces a leakage-aware machine-learning framework tailored for binary intrusion detection, utilizing the 5G-NIDD dataset. The primary contributions of this research are delineated as follows:

1. An exhaustive examination of the official IEEE DataPort Combined.csv file was executed, encompassing the elimination of exact duplicates, meticulous file-hash documentation, and the implementation of a split-before-processing protocol.
2. Three distinct feature representations were analyzed under consistent experimental conditions: the comprehensive usable set comprising 46 predictors, the correlation-reduced subset containing 34 predictors, and the Top-10 feature subset identified through ANOVA ranking.
3. The entire preprocessing and feature-selection pipeline was rigorously assessed employing five-fold stratified cross-validation. Procedures such as imputation, categorical encoding, zero-variance filtering, correlation reduction, and ANOVA ranking were systematically reiterated within each training fold.
4. A focused ablation analysis was conducted by omitting Seq and Offset to ascertain the extent to which the docum-

ented record-level performance was contingent upon sequence-related acquisition metadata.

5. The evaluation encompassed paired McNemar tests augmented with Holm correction, an analysis of feature-selection stability, quantification of false-positive and missed-attack incidents, and controlled comparisons of two-thread timing.

Related Work

5G-NIDD intrusion-detection studies

Siriwardhana et al. disseminated a peer-reviewed exposition concerning the 5G-NIDD framework [8], which builds upon a prior dataset investigation that elucidated the testbed, traffic-generation methodology, attack scenarios, and baseline assessments [9]. These contributions established a benchmark based on traffic collected from a functional 5G test network, rather than relying solely on legacy datasets or purely simulated environments.

Bouke and Abdullah compared thirteen machine-learning classifiers using a leakage-aware protocol in which the dataset was split before preprocessing and feature selection, thereby preventing test-set information from influencing model development [10]. Subsequently, Patnaik et al. underscored the importance of chronological and session-discontinuous validation, feature pruning, and lightweight assessment specifically tailored for 5G network slices [11]. Lassoued et al. compared four conventional supervised classifiers on a 5G intrusion-detection dataset described in their study as 5G-NIDD, reporting particularly high performance for KNN [20].

Ghani et al. amalgamated PCA, t-SNE, UMAP, mutual information, and synthetic balancing prior to the evaluation of six classifiers [13]. Moubayed introduced a comprehensive exploratory and deep-learning pipeline, reporting binary-detection and attack-identification performance exceeding99.5% [14]. Across several intrusion-detection datasets, Mohammad et al. examined data-augmentation strategies combined with deep-learning architectures [15]. Baldini adopted a different approach by combining Extremely

Randomized Trees with Infinite Feature Selection and evaluating the resulting model against adversarial manipulation [16]. Another line of research has focused on automating model development and improving the interpretation of model decisions. Yang and Shami developed an AutoML framework that integrates data preprocessing, feature engineering, hyperparameter optimization, and ensemble learning within a single pipeline [17]. Ilias et al. merged convolutional encoders with a sparsely gated mixture-of-experts architecture and evaluated both the 5G-NIDD and the NANCY testbed datasets [18]. Lau et al. proposed a temporal detection mechanism accompanied by feature- and time-level elucidations [19]. More recently, Osman et al. benchmarked quantized autoencoders, compact LSTMs, and lightweight Transformers for real-time anomaly detection on the RT-IoT2022 dataset, explicitly considering F1-score, latency, model size, and energy consumption under edge-device constraints [23].

Feature selection and evaluation reliability

Kim et al. evaluated filter- and wrapper-based feature-selection techniques for detecting IoT DDoS attacks in a 5G core network. Their results showed that using fewer carefully selected features lowered the computational burden without noticeably reducing detection performance [12]. Kasongo and Sun used XGBoost-based importance on UNSW-NB15 before comparing multiple classifiers [24], and Maseer et al. showed that performance on CICIDS2017 varies with the selected algorithms, parameter configurations, evaluation criteria, and the highly imbalanced distribution of attack classes [25].

Feature selection is itself a model-development step. Varma and Simon showed that performing feature selection or parameter tuning outside the relevant cross-validation loop can bias error estimates [21]. Cawley and Talbot further demonstrated that overfitting during model selection can produce selection bias in subsequent performance evaluation [22]. These findings motivate the split-before-processing and fold-specific selection protocol used here.

As shown in Table 1, previous studies have addressed dataset construction, classifier comparison, dimensionality reduction, deep learning, machine modeling, resistance to malicious attacks, and interpretability. However, the combined effect of feature set reduction, leakage control at each layer of cross-validation, and reliance on metadata associated with the data collection sequence has not been adequately isolated within a single experimental framework. This study aims to address this gap.

Research gap

This study addresses a research gap by exploring two interrelated questions:

1. Can a condensed representation of the top ten features maintain the predictive performance of larger feature sets under cross-validation protocols and leakage control?
2. To what extent do sequencing and offset contribute to the near-perfect results obtained when randomly partitioning data at the record level?

By combining a comparison of controlled feature sets, preprocessing of each fold, feature selection, paired statistical testing, and removal of targeted metadata, this study differentiates the performance of the condensed model from that based on acquisition order information.

Materials and Methods

Dataset and data audit

The investigation employed the entirety of the Combined.csv file procured from the official IEEE DataPort repository pertaining to the 5G-NIDD dataset. This dataset was initially generated on 2 December 2022, and the version utilized in the current experiments was last revised on 1 January 2026. The SHA-256 hash of the file is fa36f80859585f474504ca69eb951a079b35145537976c86b00aae3aab46ee59. The unprocessed file encompassed a total of 1,215,890 records distributed across 52 columns. An exact duplicate identification procedure was conducted across all substantive columns, while the exported CSV index was excluded from consideration. A total of 21 duplicate records were deleted, resulting in a remaining dataset of 1,215,869 observations. All identified duplicates were classified within the benign category. No cases of artificial over- or under-sampling were used in this analysis.

The distribution of classes subsequent to the removal of exact duplicates is delineated in Table 2. Figure 1 encapsulates the comprehensive leakage-aware experimental workflow, whereas Figure 2 offers a consolidated overview of the record-level and feature-space reductions instituted throughout the analysis.

Split-before-processing protocol

The sanitized records were divided utilizing a stratified 70:30 record-level allocation with random_state set to 42. The development subset consisted of 851,108 records (334,401

Table 1: Comparison of representative intrusion-detection studies related to 5G-NIDD

Study	Dataset	Approach	Main emphasis	Relevant limitation
Siriwardhana et al. [8] and Samarakoon et al. [9]	5G-NIDD	Dataset descriptor and baseline machine learning	Creation and documentation of a native 5G intrusion-detection benchmark	Primarily dataset description and baseline evaluation; limited model and feature optimisation
Bouke and Abdullah [10]	5G-NIDD	Comparison of 13 machine-learning classifiers	Leakage-aware classifier comparison	The independent effect of feature-set size and acquisition metadata was not isolated
Patnaik et al. [11]	5G-NIDD	Feature pruning and lightweight machine learning	Chronological and session-discontinuous validation for edge-oriented deployment	Different classifier configurations and deployment-oriented objectives
Kim et al. [12]	5G core intrusion dataset	Filter- and wrapper-based feature selection	Reduction of computational cost while retaining detection performance	Did not examine sequence-related metadata in 5G-NIDD
Ghani et al. [13]	5G-NIDD	PCA, mutual information, visualisation methods, and six classifiers	Dimensionality reduction and visual exploration	Multiple preprocessing and balancing choices were combined, making individual effects difficult to isolate
Moubayed [14]	5G-NIDD	Exploratory data analysis and three deep-learning models	Binary and multiclass deep-learning detection	Focused mainly on deep architectures rather than compact classical models
Mohammad et al. [15]	Four intrusion-detection datasets	Data augmentation and deep learning	Cross-dataset analysis of augmentation strategies	Not focused on compact classical machine-learning pipelines or 5G-NIDD metadata
Baldini [16]	5G-NIDD	Extremely Randomized Trees and Infinite Feature Selection	Robustness against adversarial manipulation	Addressed a different threat model and classifier setting
Yang and Shami [17]	CICIDS2017 and 5G-NIDD	Automated machine-learning ensemble	Automated preprocessing, tuning, and model development	Used a more complex automated framework and did not isolate sequence-related attributes
Ilias et al. [18]	5G-NIDD and NANCY	Convolutional neural network and sparse mixture of experts	Neural-model evaluation on two 5G intrusion-detection datasets	High architectural complexity and substantial dependence on deep-learning design choices
Lau et al. [19]	5G-NIDD	BiLSTM, attention mechanisms, and explainable AI	Temporal detection and interpretability	Required sequence construction and focused on deep temporal architectures
Lassoued et al. [20]	5G dataset described by the authors as 5G-NIDD	Conventional supervised machine-learning classifiers	Evaluation of conventional supervised classifiers	The reported dataset configuration differs from the original IEEE DataPort 5G-NIDD release. Did not jointly isolate feature-set compactness, fold-specific selection, and sequence-metadata sensitivity
Present study	5G-NIDD	Correlation reduction, ANOVA ranking, Random Forest, XGBoost, and RFF-SVM	Compact feature representation, untouched holdout evaluation, fold-specific five-fold cross-validation, paired statistical testing, metadata ablation, and controlled timing	Binary classification on one dataset; grouped session-level and external validation remain future work

Table 2: Distribution after exact-duplicate removal

Class / attack category	Records
Benign	477,716
UDP Flood	457,340
HTTP Flood	140,812
Slow-rate DoS	73,124
TCP Connect Scan	20,052
SYN Scan	20,043
UDP Scan	15,906
SYN Flood	9,721
ICMP Flood	1,155
Total	1,215,869

benign and 516,707 malicious), whereas the unaltered holdout comprised 364,761 records (143,315 benign and 221,446 malicious). The holdout was not employed for fitting imputation, encoding, variance filtering, correlation reduction, or ANOVA ranking.

The exported index (Unnamed: 0), Attack Type, and Attack Tool were omitted from the predictor matrix; Label was utilized as the binary target variable. This yielded 48 potential predictors. Numeric missing values were imputed using medians derived from the development dataset.Missing Missing categorical values were imputed with the most frequent value from the development set. Variables Proto, sDSb, dDSb, Cause, and State were encoded employing an OrdinalEncoder fitted on the development data; previously unseen categories were assigned a value of −1. Ordinal encoding was applied to generate a deterministic numerical representation while ensuring that information from the holdout partition did not influence the encoding procedure. The assigned integer values were interpreted as computational identifiers rather than as indicators of a significant ordinal relationship among categories. None of the ordinally encoded categorical variables were included in the final development-selected Top-10 feature set. A zero-variance filter, fitted on the development data, eliminated sVid and dVid, resulting in 46 usable predictors.

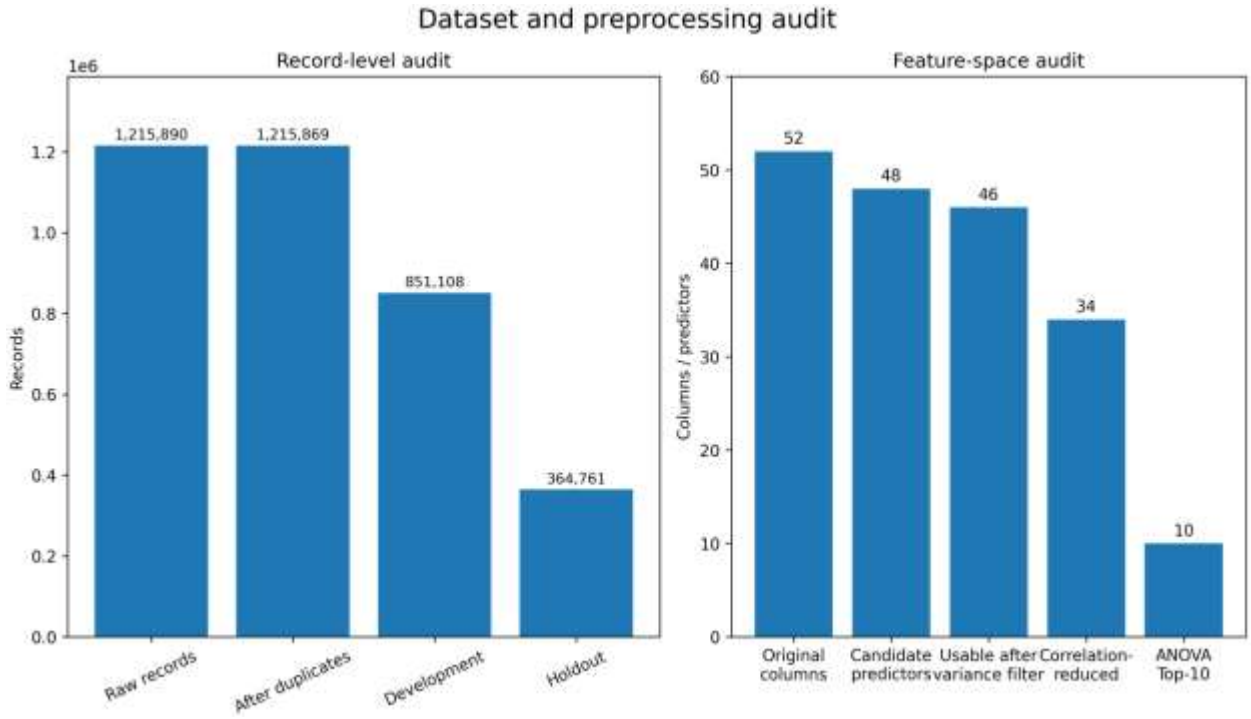

Figure 1: Unified leakage-aware experimental workflow

Correlation reduction and ANOVA ranking

Absolute Pearson correlations were computed on the development partition only. When a pair exceeded $|r| > 0.90$, the variable with the stronger absolute correlation with the binary target was retained using a deterministic greedy procedure. Twelve redundant variables were removed, leaving 34 predictors. This filter preserves original variables rather than constructing latent components. The predictor pairs exceeding the predefined correlation threshold are presented in Figure 3.

The remaining predictors were ranked by the one-way ANOVA F statistic. For two classes, the statistic compares between-class variance with within-class variance for each predictor. The ten highest-scoring features were retained. During cross-validation, all preprocessing and selection operations were re-estimated independently in each training fold. The resulting ANOVA ranking is illustrated in Figure 4, while the selected features and their corresponding F-scores and p-values are reported in Table 3.

Table 3: Development-selected Top-10 features

Rank	Feature	F-score	p-value
1	Seq	328,676.32	<1e-300
2	Offset	222,989.93	<1e-300
3	sTtl	190,379.11	<1e-300
4	AckDat	80,629.67	<1e-300
5	TcpRtt	34,316.03	<1e-300
6	sMeanPktSz	26,834.54	<1e-300
7	sHops	24,458.95	<1e-300
8	Dur	23,312.37	<1e-300
9	dTtl	14,620.84	<1e-300
10	SrcBytes	12,674.50	<1e-300

Classifiers

Random Forest used 100 decision trees with the Gini impurity criterion, a maximum tree depth of 20, a minimum of two samples per leaf, square-root feature sampling, bootstrap sampling, and balanced class weights. The implementation used random_state = 42 and was restricted to two worker threads. The algorithm is based on the ensemble method introduced by Breiman [26].

XGBoost used the binary objective with 100 boosting trees, a learning rate of 0.3, a maximum tree depth of 6, and histogram-based tree construction. Class weighting was implemented using scale_pos_weight, calculated as the ratio of benign to malicious records in the corresponding training partition. For the final development partition, this value was 334,401/516,707 = 0.647177. The implementation used random_state = 42 and was restricted to two worker threads. XGBoost is described by Chen and Guestrin [27].

The nonlinear support-vector baseline was implemented as an RFF-SVM rather than an exact RBF-SVC. Inputs were standardized using a StandardScaler fitted exclusively on the corresponding training data and were then transformed into 300 random Fourier features using an RBFSampler with $\gamma = 1/p$, where p denotes the number of input features.

Classification was performed using an averaged SGDClassifier with hinge loss, L2 regularization, $\alpha = 1 \times 10^{-5}$, and an optimal learning-rate schedule. Training was conducted incrementally using partial_fit over five shuffled epochs with batches of 20,000 records and balanced sample weights calculated from the corresponding training partition. The holdout implementation used random_state = 42, while fold-specific seeds were used during cross-validation. This approach provides a scalable approximation to an RBF-kernel support-vector classifier by combining the support-vector framework [28] with the random-feature approximation proposed by Rahimi and Recht [29]. The implementation used scikit-learn [30].

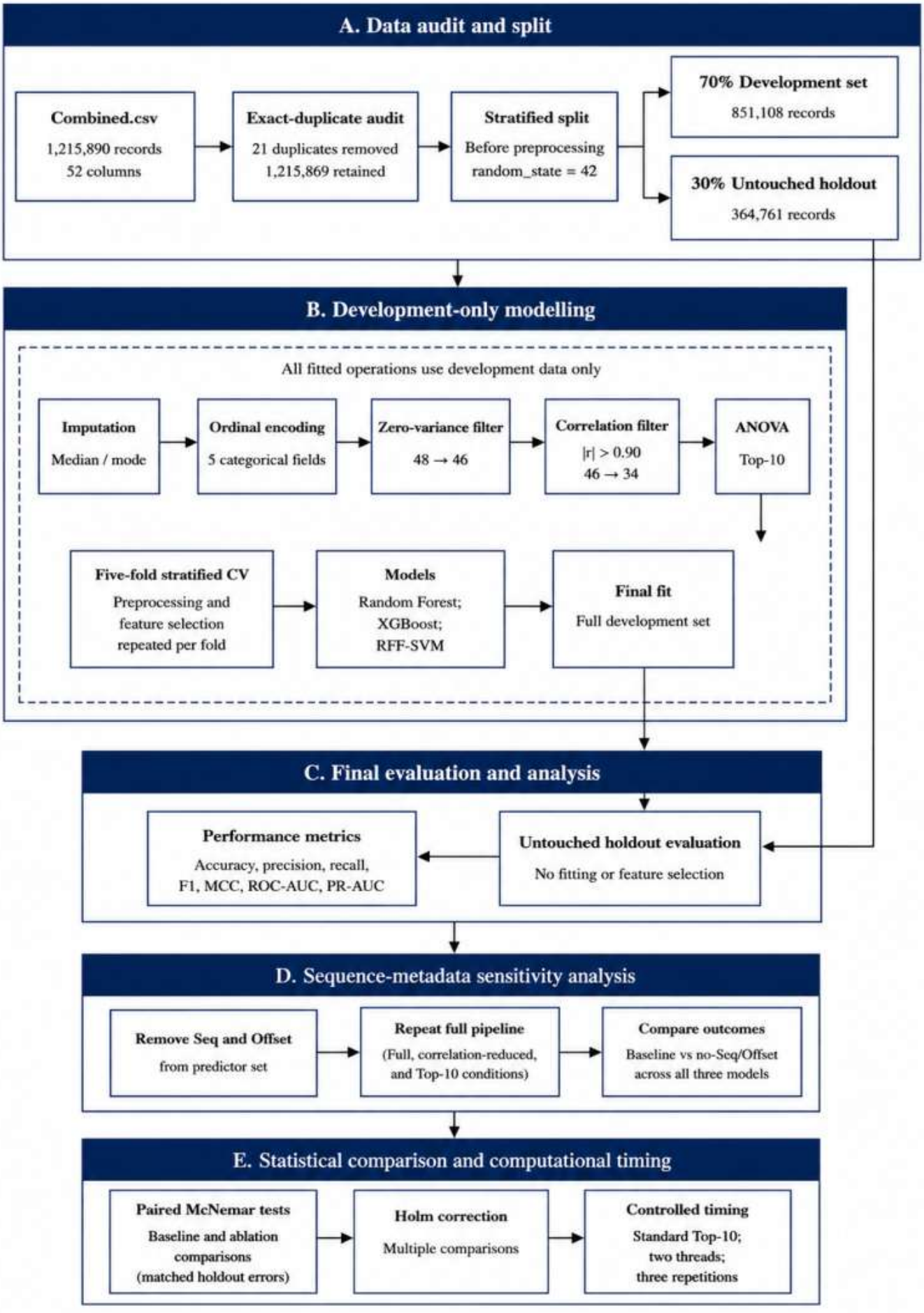

Figure 2: Record-level and feature-space audit

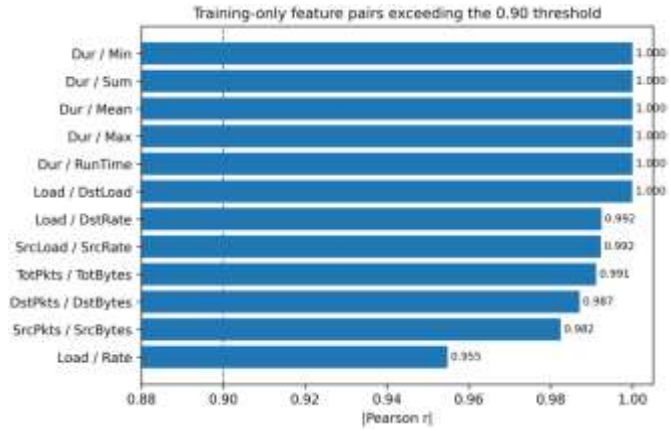

Figure 3: Highly correlated predictor pairs identified in the development partition

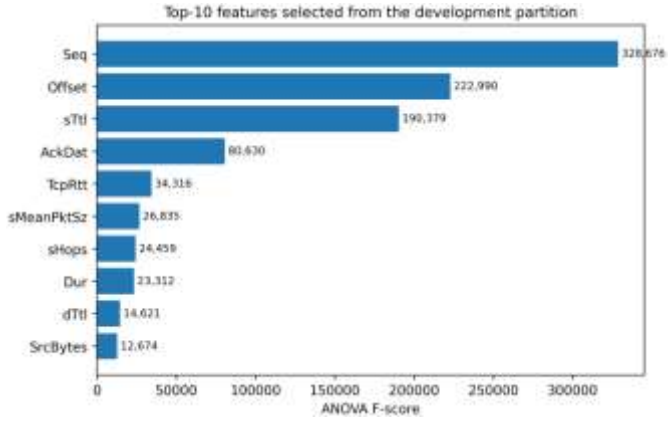

Figure 4: Top-10 ANOVA-ranked features selected from the development partition

Evaluation design

Each model underwent assessment under six distinctive feature configurations: encompassing all 46 viable predictors, 34 predictors with reduced correlation, and the Top-10 predictors; additionally, each configuration was reiterated post-exclusion of Seq and Offset prior to correlation reduction and ANOVA ranking. This methodological approach effectively disentangles the influence of dimensionality reduction from the responsiveness to sequence-related metadata. A five-fold stratified cross-validation was executed on the 70% development subset employing shuffled folds with random_state set to 42. Within each fold, processes such as imputation, encoding, zero-variance filtering, correlation reduction, and ANOVA ranking were recalibrated exclusively utilizing the training data from that specific fold. Consequently, the procedure constitutes a leakage-controlled, fold-specific cross-validation rather than a nested hyperparameter optimization.

Metrics, paired tests, and timing

The performance metrics reported include accuracy, precision, recall, F1-score, specificity, false-positive rate, false-negative rate, ROC-AUC, precision–recall AUC, and Matthews correlation coefficient. Throughout the computation of all classification metrics, the malicious class was consistently designated as the positive class. ROC analysis adheres to the framework established by Fawcett [31]; precision, recall, and associated metrics are based on Powers [32]; the MCC is aligned with Matthews [33]; and precision–recall analysis is grounded in the work of Saito and Rehmsmeier [34]. Five-fold results are summarized as the mean and sample standard deviation across folds. Stratified cross-validation was used to preserve approximately the same class proportions in each fold, following the general cross-validation framework described by Kohavi [35]. The assessment of pairwise differences in holdout errors was conducted using the precise McNemar test for matched binary outcomes [36]. The Holm procedure [37] was employed to control for family-wise error across the predetermined pairwise comparisons. Computational timing was regulated using the Top-10 representation with a fixed two-thread limit and three independent repetitions utilizing random seeds 42, 43, and 44. Both Random Forest and XGBoost were configured to operate with two worker threads, while threadpool constraints were imposed on the underlying numerical libraries across all models. Dataset loading, common data preprocessing, feature selection, and output-file writing were excluded from the measurements. For RFF-SVM, training time included training-data standardization and five-epoch incremental classifier fitting, while prediction time included standardization, random Fourier feature transformation, and classification. All experiments were conducted on a computer running the 64-bit Microsoft Windows operating system, equipped with a Intel Core i7-10610U processor operating at 1.80 GHz and 8 GB of RAM. The reported values are intended for controlled implementation-level comparison and should not be interpreted as universal operational or deployment latency As reported in Table 4, Random Forest and XGBoost produced practically equivalent performance across the full, correlation-reduced, and Top-10 representations. The Top-10 representation therefore preserved the record-level holdout performance of the larger feature sets; the results do not support a claim that the ten-feature representation materially improved predictive accuracy. The corresponding accuracy and F1-score comparisons are presented in Figures 5 and 6 respectively.

Table 4: Holdout comparison of full, correlation-reduced, and Top-10 representations

Feature set	Model	No. of predictors	Accuracy	F1	MCC
Full usable	Random Forest	46	99.9734%	99.9781%	0.999443
Full usable	XGBoost	46	99.9756%	99.9799%	0.999489
Full usable	RFF-SVM	46	99.4470%	99.5439%	0.988424
Correlation-reduced	Random Forest	34	99.9723%	99.9772%	0.999420
Correlation-reduced	XGBoost	34	99.9751%	99.9794%	0.999477
Correlation-reduced	RFF-SVM	34	99.5575%	99.6351%	0.990736
Top-10	Random Forest	10	99.9745%	99.9790%	0.999466
Top-10	XGBoost	10	99.9748%	99.9792%	0.999471
Top-10	RFF-SVM	10	99.4939%	99.5832%	0.989392

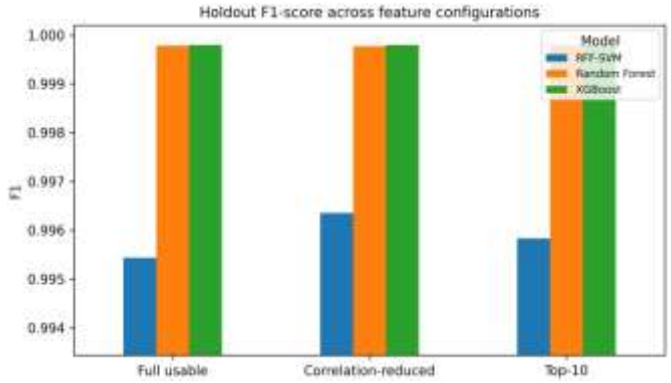

Figure 5: Holdout accuracy across full, correlation-reduced, and Top-10 representations

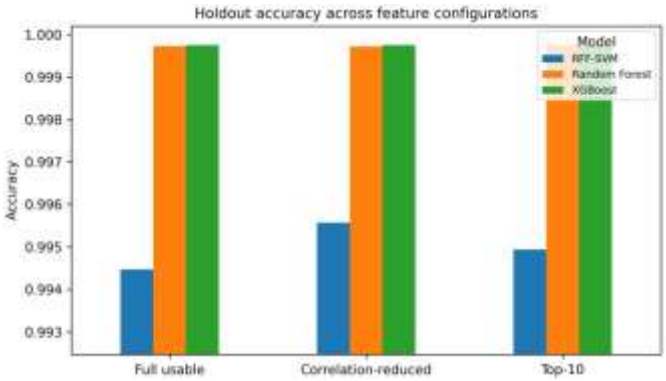

Figure 6: Holdout F1-score across feature representations

Detailed Top-10 holdout performance

Table 5 presents the detailed Top-10 holdout metrics, while Table 6 reports the corresponding confusion-matrix counts. XGBoost achieved the highest accuracy and produced only seven false-positive predictions, whereas Random Forest achieved marginally higher recall and missed 78 malicious records compared with 85 for XGBoost. RFF-SVM remained highly accurate under the record-level split but generated substantially more errors than the tree-based ensembles, as further illustrated by the confusion matrices in Figure 7.

Sensitivity to Seq and Offset

As shown in Table 7, removing Seq and Offset substantially reduced holdout performance across all three feature representations. For Random Forest and XGBoost, the full usable and correlation-reduced representations produced nearly identical results, with accuracies of approximately 72.87–72.89% and recall values of approximately 59.42%. Their newly selected Top-10 representations achieved lower accuracies of approximately 71.16–71.18% and recall values of approximately 52.53%, although their MCC values were slightly higher. RFF-SVM exhibited a different pattern, with the Top-10 representation achieving its highest post-ablation accuracy and recall of 73.0870% and 75.2346%, respectively, while retaining a lower MCC than the tree-based models. Overall, the results indicate that dependence on Seq and Offset was not limited to the compact feature representation. The Top-10 changes in accuracy, recall, and MCC are presented in Figures 8, 9, and 10, respectively, and the associated confusion matrices are shown in Figure 11.

Table 6: Top-10 holdout confusion-matrix counts

Model	TN	FP	FN	TP
Random Forest	143300	15	78	221368
XGBoost	143308	7	85	221361
RFF-SVM	142392	923	923	220523

Leakage-controlled five-fold cross-validation

As reported in Table 8, fold-specific cross-validation reproduced the principal pattern observed on the independent holdout partition. With Seq and Offset retained, Random Forest and XGBoost exceeded 99.96% mean accuracy. After their removal, mean accuracy decreased to approximately 71–72%. The corresponding mean accuracy and F1-score results are presented in Figures 12 and 13. Feature-selection stability across the five folds is summarized in Figure 14: the standard Top-10 set was selected in every fold, whereas the no-Seq/Offset condition retained nine features in all folds, with dMeanPktSz selected in four folds and Cause in one fold.

Table 5: Top-10 holdout performance

Model	Accuracy	Precision	Recall	F1	Specificity	ROC-AUC	PR-AUC	MCC
Random Forest	99.9745%	99.9932%	99.9648%	99.9790%	99.9895%	0.999995	0.999997	0.999466
XGBoost	99.9748%	99.9968%	99.9616%	99.9792%	99.9951%	0.999999	0.999999	0.999471
RFF-SVM	99.4939%	99.5832%	99.5832%	99.5832%	99.3560%	0.998917	0.999006	0.989392

Table 7: Untouched holdout performance across feature representations after removing Seq and Offset

Feature representation	Model	No. of predictors	Accuracy	Recall	F1-score	MCC
Full usable without Seq/Offset	Random Forest	44	72.8691%	59.4285%	72.6748%	0.5324
Full usable without Seq/Offset	XGBoost	44	72.8894%	59.4199%	72.6867%	0.5331
Full usable without Seq/Offset	RFF-SVM	44	69.1812%	53.4577%	67.8055%	0.4805
Correlation-reduced without Seq/Offset	Random Forest	32	72.8694%	59.4262%	72.6742%	0.5324
Correlation-reduced without Seq/Offset	XGBoost	32	72.8899%	59.4208%	72.6874%	0.5331
Correlation-reduced without Seq/Offset	RFF-SVM	32	69.2412%	54.3202%	68.1962%	0.4743
Top-10 without Seq/Offset	Random Forest	10	71.1644%	52.5365%	68.8686%	0.5499
Top-10 without Seq/Offset	XGBoost	10	71.1787%	52.5315%	68.8771%	0.5504
Top-10 without Seq/Offset	RFF-SVM	10	73.0870%	75.2346%	77.2431%	0.4447

Table 8: Fold-specific five-fold cross-validation results

Condition	Model	Accuracy	Precision	Recall	F1	MCC
Top-10	Random Forest	99.9671% ± 0.0070%	99.9830% ± 0.0051%	99.9628% ± 0.0088%	99.9729% ± 0.0058%	0.9993 ± 0.0001
Top-10	XGBoost	99.9704% ± 0.0061%	99.9936% ± 0.0021%	99.9576% ± 0.0090%	99.9756% ± 0.0051%	0.9994 ± 0.0001
Top-10	RFF-SVM	99.3457% ± 0.1029%	99.4081% ± 0.1661%	99.5150% ± 0.0328%	99.4615% ± 0.0843%	0.9863 ± 0.0022
Top-10 without Seq/Offset	Random Forest	71.4442% ± 0.8695%	98.6687% ± 2.8335%	53.8053% ± 3.2378%	69.5366% ± 1.8391%	0.5457 ± 0.0063
Top-10 without Seq/Offset	XGBoost	71.4604% ± 0.8770%	98.7343% ± 2.8125%	53.7941% ± 3.2310%	69.5444% ± 1.8426%	0.5463 ± 0.0061
Top-10 without Seq/Offset	RFF-SVM	71.7803% ± 1.6170%	81.7767% ± 5.9595%	70.0575% ± 9.2715%	74.8187% ± 3.9618%	0.4423 ± 0.0236

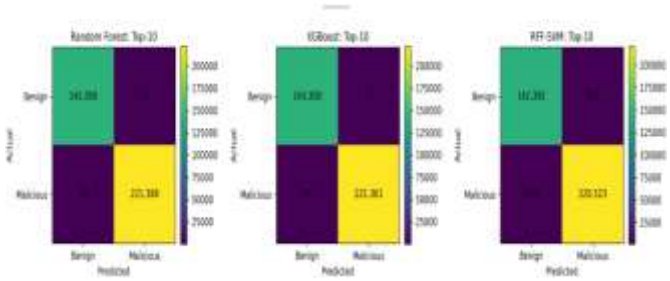

Figure 7: Confusion matrices for the Top-10 holdout condition

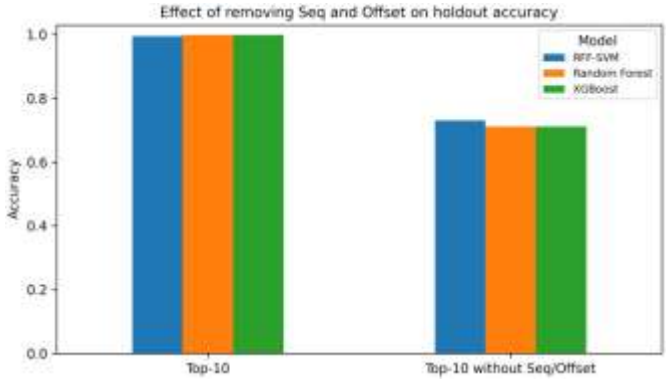

Figure 8: Holdout accuracy before and after removing Seq and Offset

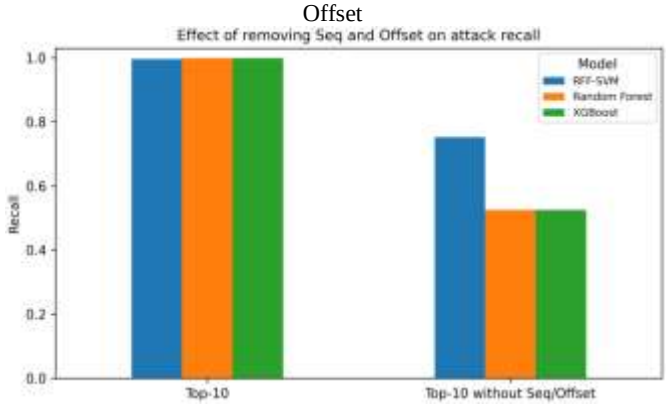

Figure 9: Holdout recall before and after removing Seq and Offset.

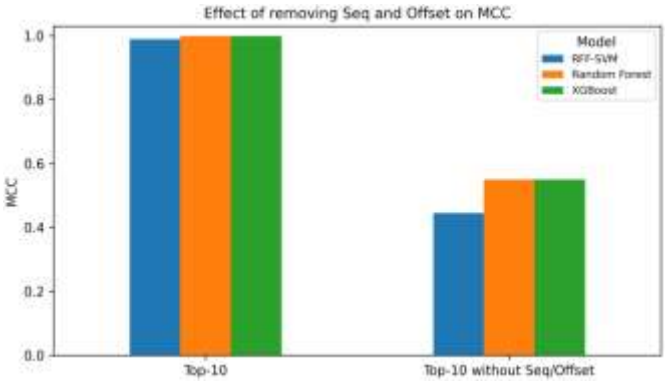

Figure 10: Holdout MCC before and after removing Seq and Offset

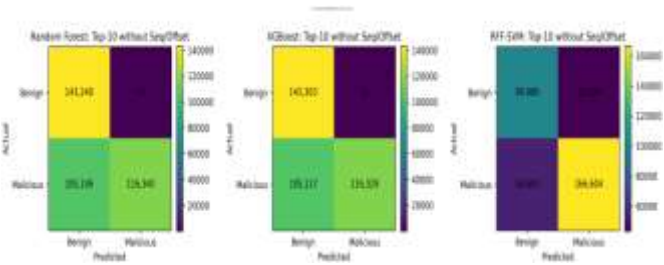

Figure 11: Confusion matrices for Top-10 without Seq/Offset

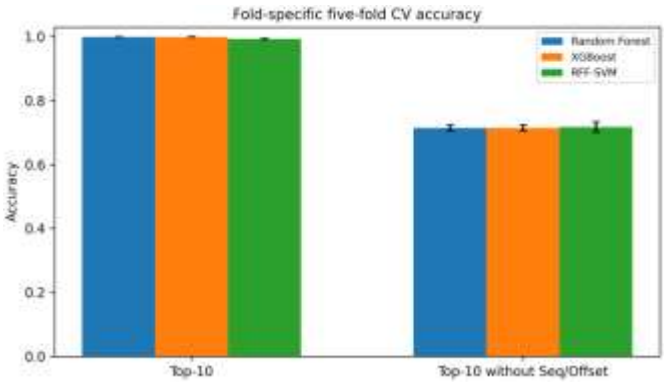

Figure 12: Cross-validation accuracy (mean ± standard deviation).

Paired statistical tests As shown in Table 9, Random Forest and XGBoost did not differ significantly under the standard Top-10 condition (Holm-adjusted p = 1.0). Both tree ensembles significantly outperformed RFF-SVM under that condition. For all three models, the standard Top-10 predictions differed significantly from the corresponding no-Seq/Offset predictions ($p < 0.001$).

Controlled computational timing

As reported in Table 10, XGBoost was the fastest model,

Table 9: Exact paired McNemar tests

Comparison	A correct / B wrong	A wrong / B correct	Holm-adjusted p-value
top10: RF vs XGBoost	12	13	p = 1.000
top10: RF vs RFF-SVM	1766	13	p < 0.001
top10: XGBoost vs RFF-SVM	1755	1	p < 0.001
top10_no_seq_offset: RF vs XGBoost	13	65	p < 0.001
top10_no_seq_offset: RF vs RFF-SVM	56443	63456	p < 0.001
top10_no_seq_offset: XGBoost vs RFF-SVM	56432	63393	p < 0.001
RF: Top-10 vs Top-10 without Seq/Offset	105104	16	p < 0.001
XGBoost: Top-10 vs Top-10 without Seq/Offset	105043	6	p < 0.001
RFF-SVM: Top-10 vs Top-10 without Seq/Offset	97235	913	p < 0.001

Table 10: Controlled two-thread timing over three repetitions

Model	Training time (s)	Prediction time (s)	Prediction latency per 1,000 records (ms)
Random Forest	47.434 ± 1.276	0.631 ± 0.028	1.731 ± 0.077
XGBoost	2.740 ± 0.166	0.203 ± 0.009	0.558 ± 0.024
RFF-SVM	6.631 ± 0.112	0.306 ± 0.015	0.840 ± 0.041

with mean training and prediction times of 2.740 s and 0.203 s, respectively. RFF-SVM trained in 6.631 s and predicted in 0.306 s, whereas Random Forest required 47.434 s for training and 0.631 s for prediction. The controlled training- and prediction-time comparisons are illustrated in Figures 15 and 16, respectively. These measurements are implementation-specific and exclude dataset loading, preprocessing, and feature selection.

Discussion

Compact feature representation

The comparison of the full usable, correlation-reduced, and Top-10 representations indicates that a substantial portion of the predictive efficacy of Random Forest and XGBoost can be preserved by employing merely ten features. The contraction of the input from 46 usable predictors to the Top-10 representations did not yield a meaningful enhancement in accuracy; however, it likewise resulted in no considerable detriment to performance within the record-level split. Consequently, the primary merit of the compact representation lies not in heightened predictive accuracy, but rather in the capacity to achieve nearly equivalent outcomes with a markedly reduced input space.

This finding implies that the original feature set encompasses a significant degree of redundancy. The correlation filter successfully diminished the predictor count from 46 to 34; nevertheless, the performance of the two tree-based models remained virtually unchanged. A subsequent reduction to ten ANOVA-ranked features engendered a similarly negligible alteration. This trend corroborates previous research demonstrating that meticulously selected features can truncate the dimensions of an intrusion-detection model without markedly influencing its predictive performance [12]. The current findings amplify that observation by evaluating the three feature representations under uniform training partitions, model configurations, and assessment conditions.

The cross-validation outcomes further substantiate the stability of the compact representation. The standard Top-10 features were consistently chosen in each fold, suggesting that their ranking was not influenced by a singular development split. Both Random Forest and XGBoost sustained mean cross-validation accuracies surpassing 99.96%, closely aligning with their holdout results. This concordance between the holdout and fold-specific assessments indicates that the compact representation demonstrated stability within the record-level evaluation framework.

The results were not uniform across all classifiers. RFF-SVM exhibited high accuracy; however, its results fluctuated marginally across the full, correlation-reduced, and Top-10 representations. Optimal performance was achieved with the correlation-reduced set rather than the Top-10 set. This observation suggests that the impact of feature reduction may be contingent upon the specific learning algorithm employed. Tree-based ensembles possess the capability to directly model nonlinear relationships and feature interactions, whereas the RFF-SVM depends on an approximate kernel representation and may exhibit differential responses when the available input dimensions are curtailed.

A more compact feature set can also confer advantages from an implementation standpoint. It diminishes the number of variables that necessitate storage, inspection, and provision to the trained model. Furthermore, it may facilitate subsequent interpretation and monitoring processes. Nonetheless, the timing experiment assessed model training and prediction subsequent to the completion of feature selection. The findings, therefore, do not explicitly illustrate the extent of end-to-end computational cost mitigated by the reduction of the feature set. The practical efficiency of the Top-10 representation should be evaluated independently within a comprehensive deployment pipeline.

Sensitivity to sequence-related metadata

The paramount discovery of this investigation is the pronounced reliance of the results on Seq and Offset. These two variables attained the highest ANOVA F-scores by a considerable margin. According to the Argus documentation, Seq denotes the sequence number ascribed to a record, while Offset signifies its location within a file or stream [38].

They consequently elucidate facets of acquisition sequence and retention rather than the network dynamics of a singular data flow. The elimination of these two variables precipitated a substantial deterioration in performance across all three classifiers. This decrement was noted not only under the Top-10 condition but also within the entirety of usable and correlation-reduced representations. The reliance on Seq and Offset was thus not engendered by the selection of a compact feature set. Rather, it was evident throughout the feature space and influenced the overarching record-level classification endeavor. For Random Forest and XGBoost, the complete and correlation-reduced conditions, devoid of Seq and Offset, yielded accuracies approaching 73%, whereas the newly curated Top-10 conditions exhibited marginally lower results. Their recall values also declined substantially. Concurrently, precision maintained a high level, particularly during cross-validation. This juxtaposition of elevated precision and diminished recall indicates that the tree-based models adopted a more conservative stance following the removal of the sequence-related variables.

When these models predicted an attack, they were predominantly accurate; however, they overlooked a considerable fraction of malicious records.

RFF-SVM presented a divergent trend. Its Top-10 condition, subsequent to the removal of Seq and Offset, attained superior recall and F1-score compared to the two tree ensembles, albeit its MCC remained inferior. The confusion matrices illustrate that this augmented recall was coupled with an increase in false-positive predictions. This distinction is critical, as accuracy in isolation does not encapsulate the post-ablation performance of the models. Random Forest and XGBoost prioritized precision, while RFF-SVM identified a greater number of attacks at the expense of an increased false-alarm rate.

The significant alteration following ablation implies that the random partitioning enabled the models to capitalize on patterns associated with the sequence in which the records were acquired or stored. One plausible explanation is that benign and attack traffic were generated during disparate timeframes or campaigns, thereby permitting sequence position to serve as an indirect marker of class affiliation. The existing results bolster this interpretation; however, they do not substantiate that Seq and Offset signify intentional target leakage. Verifying the precise mechanism would necessitate comprehensive insights into the original collection sessions and the temporal arrangement of the dataset.

The analogous performance decline was evident in both the holdout evaluation and five-fold cross-validation. This consistency indicates that the effect was not a consequence of an anomalous holdout sample. Nevertheless, both methodologies employed random record-level partitioning from the identical dataset. They may consequently replicate the same acquisition-related structure across their training and validation subsets. The concordance between the two evaluations should be construed as internal consistency rather than as evidence of cross-session or cross-environment generalizability.

These findings elucidate the rationale behind the attainment of near-perfect results on 5G-NIDD under random record-level evaluation. Prior investigations have similarly reported exceptionally high performance on this dataset utilizing classical machine learning and deep learning models [10], [14]. Such outcomes remain valuable as benchmark assessments, yet their operational significance is contingent upon the methodology of data partitioning. Research employing chronological or session-disjoint evaluation has already underscored the necessity of segregating related collection periods during validation [11].

The markedly different results reported by Ilias et al. across the 5G-NIDD and NANCY datasets suggest that benchmark performance may be strongly dataset-dependent [18].

A significant implication for practical applications is that forthcoming research utilizing 5G-NIDD should specify the inclusion of sequence- and file-position variables. Presenting outcomes both with and without these variables would enhance the transparency of model comparisons. In cases where session or base-station identifiers are accessible, employing grouped or chronological validation would yield a more rigorous examination of whether the identified patterns accurately reflect attack behaviors rather than characteristics inherent to the acquisition process.

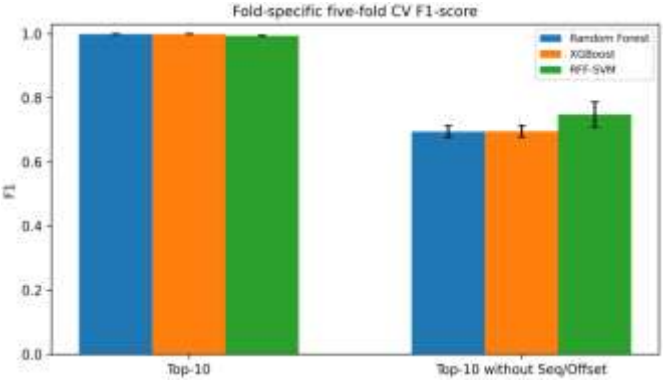

Figure 13: Cross-validation F1-score (mean ± standard deviation).

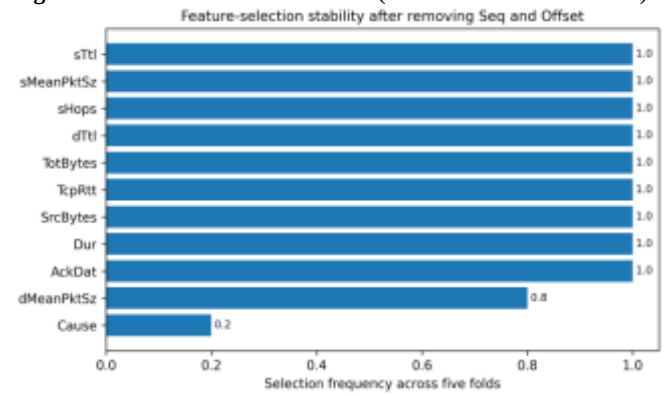

Figure 14: Fold-level frequency of selected features.

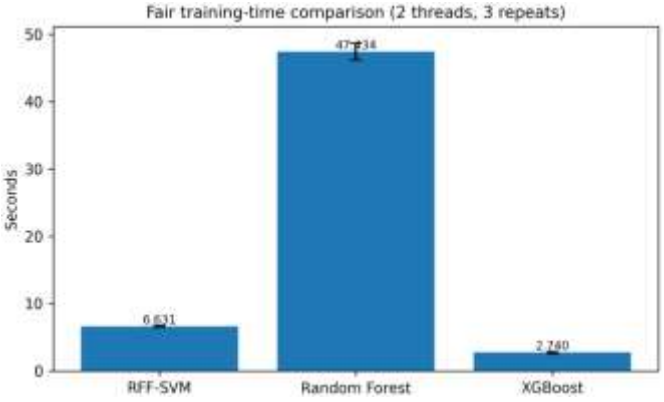

Figure 15: Controlled two-thread training time over three repetitions

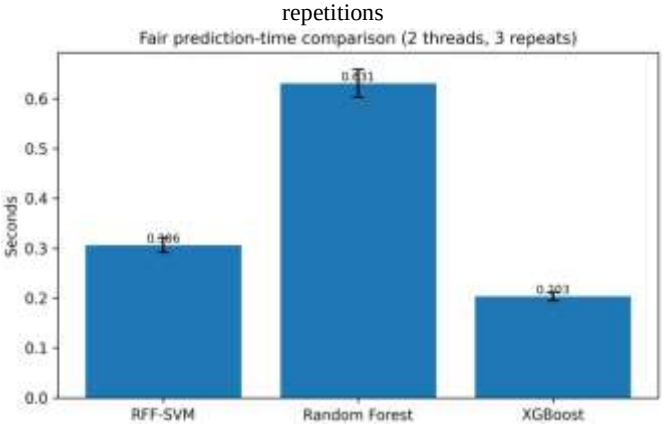

Figure 16: Controlled two-thread prediction time over three repetitions

Classifier trade-offs and computational implications

Within the standard Top-10 framework, both Random Forest and XGBoost yielded nearly indistinguishable overall performance metrics. Their respective accuracies, F1-scores, ROC-AUC values, PR-AUC values, and MCC values exhibited only marginal variations at minimal decimal points. The application of the paired McNemar test substantiated that their prediction errors were not statistically different post-Holm adjustment. Consequently, the slight numerical superiority of XGBoost in accuracy should not be construed as indicative of its general superiority over Random Forest.

The insights gleaned from the confusion-matrix results offer a more substantive foundation for comparing the two models. XGBoost generated seven false positives, as opposed to 15 for Random Forest. Conversely, Random Forest failed to identify 78 malicious records, while XGBoost overlooked 85. Although the disparity is modest, it underscores the operational trade-off between false alarms and unrecognized attacks. In contexts where extraneous alerts incur significant costs, XGBoost may be favored. Alternatively, if the repercussions of failing to detect an attack are deemed more severe, the marginally enhanced recall of Random Forest may be more pertinent.

This analysis also elucidates the necessity for multiple performance metrics. Although accuracy was notably high for both models, accuracy alone did not elucidate the specific type of error each model was predisposed to commit. Metrics such as precision, recall, specificity, MCC, and confusion-matrix counts provide a more comprehensive perspective on their operational behavior. Consequently, the selection of a model for an operational intrusion-detection system should be guided by the anticipated costs associated with false positives and false negatives, rather than solely by the maximization of accuracy.

Under the standard Top-10 condition, RFF-SVM was evidently less accurate than the tree-based ensembles, and paired tests corroborated that the observed difference was statistically significant. Nevertheless, RFF-SVM ought to be regarded as a scalable nonlinear baseline rather than a precise RBF-SVC. The incorporation of random Fourier features and incremental SGD training facilitated the model's application to over one million records without incurring the computational burden typical of training a conventional kernel SVM. Its findings indicate that this approximation remained competitive within the standard record-level task, despite yielding a higher error rate than both Random Forest and XGBoost.

Upon the removal of Seq and Offset, the ranking of the models became less unequivocal. RFF-SVM attained the highest recall and F1-score within the Top-10 condition, while Random Forest and XGBoost demonstrated superior MCC values and substantially higher precision. This alteration implies that model comparison is highly contingent upon the feature condition and the chosen evaluation metric. A model that may initially appear inferior under the original benchmark could present a different and potentially advantageous balance when sequence-related information is not accessible.

The temporal assessment introduces an additional pragmatic dimension to the comparative analysis. Within the regulated dual-thread environment, XGBoost achieved the most abbreviated mean durations for both training and prediction processes. RFF-SVM emerged as the second-most expeditious implementation, whereas Random Forest necessitated substantially longer training times. These findings favor XGBoost in scenarios necessitating frequent retraining or expeditious model updates. Nevertheless, these outcomes pertain specifically to the implementations utilized, the fixed hyperparameters, the software versions, and the hardware configurations employed in this investigation. They ought not to be construed as generalized estimates of latency within a deployed 5G intrusion-detection framework.

The temporal comparison further omitted considerations related to data loading, shared preprocessing, feature selection, and output generation. Consequently, it delineates the computational expenses associated with the fitting and application of the chosen classifiers, rather than encompassing the total expenses of the entire intrusion-detection pipeline. In a practical implementation, feature extraction, traffic aggregation, and communication overhead could constitute a considerable segment of the overall processing duration. Related IoT monitoring research has likewise demonstrated the value of combining feature extraction, machine learning, and edge-oriented deployment considerations when designing practical real-time detection frameworks [39].

In summation, the findings yield two interrelated conclusions. Primarily, a concise Top-10 representation is adequate for maintaining the near-optimal performance of both Random Forest and XGBoost within the established random record-level benchmark. Secondly, this benchmark performance is significantly affected by Seq and Offset. Thus, the principal contribution of this study lies not in the identification of yet another near-optimal classifier, but in the elucidation that such performance must be contextualized concerning the metadata and validation framework employed to derive it.

Limitations and Threats to Validity

This investigation is subject to multiple limitations that warrant consideration when interpreting the findings. Initially, the experiments were conducted utilizing a singular dataset and were exclusively concentrated on binary classification. While the holdout set was not employed during the processes of model training or feature selection, it was generated through a random record-level partitioning of the same dataset. Consequently, the outcomes reported predominantly reflect performance within the confines of the identical data-collection environment. They do not substantiate that the models would exhibit equivalent efficacy on traffic gathered from disparate sessions, base stations, attack campaigns, or operational 5G networks.

Moreover, the ablation analysis targeted Seq and Offset, as these were identified as the two highest-ranking features intrinsically linked to record order and file position. The removal of these features resulted in a significant decline in performance, thereby indicating that the original random-split results were heavily reliant on acquisition-related information. Nevertheless, this does not validate that these variables signify intentional target leakage. It further does not assure that all collection-specific patterns have been eradicated, as other features may still encapsulate temporal or session-related characteristics. Hence, the results derived post-removal of Seq and Offset ought to be regarded as a sensitivity analysis rather than a definitive assessment of performance in an alternate environment. Furthermore, the study employed fixed classifier parameters to ensure that the feature representations were evaluated under consistent conditions. Comprehensive hyperparameter tuning and nested

cross-validation were not conducted. As a result, the reported findings are reflective of the chosen model configurations and should not be construed as indicative of the optimal performance that each algorithm could potentially achieve. Additionally, the RFF-SVM model was conceptualized as a scalable approximation of an RBF-kernel SVM and is not synonymous with an exact RBF-SVC.

The computational timing outcomes also exhibit limited generalizability. These were acquired within a singular controlled virtual environment, constrained by a two-thread limit and comprising merely three iterations. The timing measurements excluded dataset loading, shared preprocessing, feature selection, and file writing. While these results are beneficial for comparing the implementations utilized in this study, they should not be regarded as general estimates of deployment latency.

Future research endeavors should assess the models using session-disjoint, base-station-disjoint, and chronological splits. Additionally, testing on independently collected 5G traffic and other datasets is imperative.

Further investigations may also contemplate multiclass attack classification, an expansive hyperparameter evaluation, and deployment testing on representative edge or network-security platforms.

Conclusion

This study examined feature-selected machine learning for binary intrusion detection using the complete 5G-NIDD Combined.csv file under a leakage-aware, split-before-processing protocol. Following the removal of 21 exact duplicate records, the analysis was conducted on 1,215,869 observations. The dataset was divided into development and untouched holdout partitions before any data-dependent operation was applied. Imputation, categorical encoding, zero-variance filtering, correlation reduction, and ANOVA ranking were fitted using development data only and were repeated separately within each cross-validation fold. This design allowed the full usable, correlation-reduced, and Top-10 feature representations to be compared under consistent evaluation conditions.

The results showed that reducing the feature space from 46 usable predictors to ten ANOVA-ranked features caused no meaningful loss of performance for Random Forest and XGBoost under the conventional random record-level split. On the untouched holdout, XGBoost achieved an accuracy of 99.9748%, while Random Forest achieved 99.9745%. Their paired prediction errors were not statistically different after Holm correction. XGBoost generated the fewest false positives and recorded the shortest training and prediction times in the controlled timing experiment. Random Forest, however, achieved slightly higher recall and missed fewer malicious records. RFF-SVM produced somewhat lower performance under the standard Top-10 condition, but it remained a computationally scalable nonlinear alternative for a dataset containing more than one million records.

The main finding emerged from the analysis of Seq and Offset. These variables received the two highest ANOVA scores, although they describe sequence order and file position rather than the network behaviour of an individual flow. Their removal reduced Top-10 holdout accuracy to 71.1644% for Random Forest, 71.1787% for XGBoost, and 73.0870% for RFF-SVM. A similar decline was observed during fold-specific cross-validation and across the larger feature representations. This indicates that the dependence on sequence-related metadata was not introduced by selecting a compact Top-10 set. Instead, it was already present in the broader record-level classification problem.

The outcomes subsequent to the ablation process also altered the relative performance of the classifiers. While both Random Forest and XGBoost maintained a commendable level of precision, they exhibited a considerable deficiency in identifying a substantial segment of malicious records. In contrast, RFF-SVM achieved superior recall and F1-score, albeit at the expense of an increased incidence of false-positive predictions and a diminished Matthews Correlation Coefficient (MCC). These discrepancies illustrate that the ranking of models is contingent not merely upon overall accuracy, but also upon the predictors at hand and the operational costs attributed to false positives and overlooked attacks. In summation, the judicious selection of compact features can uphold performance metrics while simplifying the input representation when evaluating 5G-NIDD through a random record-level benchmark. Nonetheless, the nearly flawless outcomes attained under this framework should not be misconstrued as proof of generalizability across diverse collection sessions, base stations, attack campaigns, or operational 5G environments. Accordingly, the primary contribution of this study lies not in the identification of yet another near-flawless classifier, but rather in the elucidation that the reported performance must be contextualized with respect to the metadata preserved within the predictor set and the partitioning strategies employed during evaluation. Subsequent investigations ought to emphasize chronological, session-disjoint, and base-station-disjoint methodologies for evaluation. It is also imperative to conduct tests on independently gathered or live 5G traffic to ascertain whether the identified patterns are transferable beyond the original collection context. Further scholarly inquiry may seek to broaden the framework to encompass multiclass attack identification, extensive hyperparameter analysis, cross-dataset transferability, and deployment-oriented assessments on representative edge and network-security infrastructures.

Author Contributions: All authors made substantial intellectual contributions to the work, reviewed the manuscript, and approved the final version for publication.

Funding: This research received no external funding.

Data Availability Statement: The dataset examined in this research is the 5G-NIDD dataset, titled “5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network,” and it is accessible via the official IEEE DataPort repository under HTTPS://DOI.ORG/ 10.21227/xtep-hv36. It was initially created on 2 December 2022, and the version employed in the current study was most recently updated on 1 January 2026. The experiments used the Combined.csv file, and its precise version is specified by the SHA-256 hash given in Appendix A. The reproducibility code, reference outputs, and experimental configuration are publicly available through the GitHub repository and the archived Zenodo release described in the Code Availability Statement below.

Code Availability Statement: The source code, analysis scripts, environment requirements, and configuration files used to reproduce the experiments reported in this study are publicly available in the GitHub repository at https://github.com/Abraheem2026/5G-NIDD-Leakage-Aware-IDS. The archived v1.0.1 release is permanently available through Zenodo at https://doi.org/10.5281/zenodo.21211982. The repository

includes instructions for reproducing the preprocessing, feature-selection, model-evaluation, ablation, and statistical-testing stages.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process: During the preparation of this work, the authors used OpenAI ChatGPT to assist with manuscript organization, English-language refinement, and limited code review and debugging. All AI-assisted outputs were critically reviewed, verified, and edited by the authors. The authors take full responsibility for the accuracy, integrity, and final content of the manuscript.

References

[1] International Telecommunication Union Radiocommunication Sector, "IMT Vision—Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond," Recommendation ITU-R M.2083-0, 2015.

[2] 3rd Generation Partnership Project, "System Architecture for the 5G System (5GS)," 3GPP TS 23.501, Version 18.12.0, Release 18, Dec. 2025.

[3] European Union Agency for Cybersecurity, ENISA Threat Landscape for 5G Networks. ENISA, 2019, https://doi.org/ 10.2824/49299

[4] 3rd Generation Partnership Project, "Security Architecture and Procedures for 5G System." 3GPP TS 33.501, Version 18.10.0, Release 18, Jul. 2025.

[5] M. Bartock, J. Cichonski, M. Souppaya, K. Kent, P. Grayeli, and S. Sharma. "5G Network Security Design Principles: Applying 5G Cybersecurity and Privacy Capabilities." *NIST Cybersecurity White Paper 36E*, Mar. 2026, https://doi.org/ 10.6028/NIST.CSWP.36E.

[6] D. Denning. "An Intrusion-Detection Model." *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987, https://doi.org/ 10.1109/TSE.1987.232894.

[7] M. Talukder, et al. "Machine Learning-Based Network Intrusion Detection for Big and Imbalanced Data Using Oversampling, Stacking Feature Embedding and Feature Extraction." *Journal of Big Data*, vol. 11, art. 33, 2024, https://doi.org/ 10.1186/s40537-024-00886-w.

[8] Y. Siriwardhana, S. Samarakoon, P. Porambage, M. Liyanage, S.-Y. Chang, J. Kim, J. Kim, and M. Ylianttila. "Descriptor: 5G Wireless Network Intrusion Detection Dataset (5G-NIDD)." *IEEE Data Descriptions*, vol. 2, pp. 358–369, 2025, https://doi.org/ 10.1109/IEEEDATA.2025.3592888.

[9] S. Samarakoon, et al. "5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network." *arXiv:2212.01298*, 2022, https://doi.org/ 10.48550/arXiv.2212.01298.

[10] M. Bouke, and A. Abdullah. "An Empirical Assessment of ML Models for 5G Network Intrusion Detection: A Data Leakage-Free Approach." *e-Prime*, vol. 8, art. 100590, 2024, https://doi.org/ 10.1016/j.prime.2024.100590.

[11] P. Patnaik, S. Mishra, B. Panda, and S. K. Kar. "TinyML Driven Intrusion Detection for 5G Network Slices with Leakage-Free Validation." *Next-Generation Computing Systems and Technologies*, vol. 2, no. 1, pp. 10–20, 2026, https://doi.org/ 10.62762/NGCST.2026.664893.

[12] Y.-E. Kim, Y.-S. Kim, and H. Kim. "Effective Feature Selection Methods to Detect IoT DDoS Attack in 5G Core Network." *Sensors*, vol. 22, no. 10, art. 3819, 2022, https://doi.org/ 10.3390/s22103819.

[13] H. Ghani, S. Salekzamankhani, and B. Virdee. "Critical Analysis of 5G Networks' Traffic Intrusion Using PCA, t-SNE, and UMAP Visualization and Classifying Attacks," *in Proceedings of Data Analytics and Management*, *Lecture Notes in Networks and Systems*, vol. 785, pp. 421–437, 2024, https://doi.org/ 10.1007/978-981-99-6544-1_32.

[14] A. Moubayed. "A Complete EDA and DL Pipeline for Softwarized 5G Network Intrusion Detection." *Future Internet*, vol. 16, no. 9, art. 331, 2024, https://doi.org/ 10.3390/fi16090331.

[15] R. Mohammad, et al. "Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach." *Systems*, vol. 12, no. 3, art. 79, 2024, https://doi.org/ 10.3390/systems12030079.

[16] G. Baldini. "Mitigation of Adversarial Attacks in 5G Networks with a Robust Intrusion Detection System Based on Extremely Randomized Trees and Infinite Feature Selection." *Electronics*, vol. 13, no. 12, art. 2405, 2024, https://doi.org/ 10.3390/electronics13122405.

[17] L. Yang, and A. Shami. "Towards Autonomous Cybersecurity: An Intelligent AutoML Framework for Autonomous Intrusion Detection." *in Proceedings of the Workshop on Autonomous Cybersecurity (AutonomousCyber '24)*, Salt Lake City, UT, USA, 2024, https://doi.org/ 10.1145/3689933.3690833.

[18] L. Ilias, G. Doukas, V. Lamprou, C. Ntanos, and D. Askounis. "Convolutional Neural Networks and Mixture of Experts for Intrusion Detection in 5G Networks and Beyond." *Frontiers in Artificial Intelligence*, vol. 8, art. 1708953, 2026, https://doi.org/ 10.3389/frai.2025.1708953.

[19] J. Lau, et al. "BiTAD: An Interpretable Temporal Anomaly Detector for 5G Networks with TwinLens Explainability." *Future Internet*, vol. 17, no. 11, art. 482, 2025, https://doi.org/ 10.3390/fi17110482.

[20] N. Lassoued, I. Filali, and R. Ejbali. "Supervised Machine Learning-Based Intrusion Detection for 5G Networks: Evaluation on the 5G-NIDD Dataset." *Computers*, vol. 15, no. 6, art. 362, 2026, https://doi.org/ 10.3390/computers15060362.

[21] S. Varma, and R. Simon. "Bias in Error Estimation When Using Cross-Validation for Model Selection." *BMC Bioinformatics*, vol. 7, art. 91, 2006, https://doi.org/ 10.1186/1471-2105-7-91.

[22] G. Cawley, and N. Talbot. "On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation." *Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.

[23] M. Osman, F. Elghaffi, L. Ben Dalla, Ö. Karal, and T. Rashid, "A New Approach for Low-Latency, High-Accuracy Anomaly Detection at the Edge: Benchmarking Quantized Autoencoders, LSTMs, and Lightweight Transformers on RT-IoT2022 Time-Series Traffic." *Wadi Alshatti University Journal of Pure and Applied Sciences*, vol. 4, no. 1, pp. 110–121, Jan. 2026, https://doi.org/ 10.63318/waujpasv4i1_12.

[24] S. Kasongo, and Y. Sun. "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset." *Journal of Big Data*, vol. 7, art. 105, 2020, https://doi.org/ 10.1186/s40537-020-00379-6.

[25] Z. Maseer, R. Yusof, N. Bahaman, S. Mostafa, and C. Foozy. "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset." *IEEE Access*, vol. 9, pp. 22351–22370, 2021, https://doi.org/ 10.1109/ACCESS.2021.3056614.

[26] L. Breiman, "Random Forests." *Machine Learning*, vol. 45, pp. 5–32, 2001, https://doi.org/ 10.1023/A:1010933404324.

[27] T. Chen, and C. Guestrin. "XGBoost: A Scalable Tree Boosting System." *in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, San Francisco, CA, USA, 2016, pp. 785–794, https://doi.org/ 10.1145/2939672.2939785.

[28] C. Cortes, and V. Vapnik. "Support-Vector Networks." *Machine Learning*, vol. 20, pp. 273–297, 1995, https://doi.org/ 10.1007/BF00994018.

[29] A. Rahimi, and B. Recht. "Random Features for Large-Scale Kernel Machines." *in Advances in Neural Information Processing Systems 20*, pp. 1177–1184, 2007.

[30] F. Pedregosa, et al. "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[31] T. Fawcett. "An Introduction to ROC Analysis." *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, https://doi.org/ 10.1016/j.patrec.2005.10.010.

[32] D. Powers. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation." *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[33] B. Matthews. "Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme." *Biochimica et Biophysica Acta*, vol. 405, no. 2, pp. 442–451, 1975, https://doi.org/ 10.1016/0005-2795(75)90109-9.

[34] T. Saito, and M. Rehmsmeier. "The Precision–Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets." *PLoS ONE*, vol. 10, no. 3, e0118432, 2015, https://doi.org/ 10.1371/journal.pone.0118432.

[35] R. Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *in Proc. 14th IJCAI*, pp. 1137–1143, 1995.

[36] Q. McNemar. "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages." *Psychometrika*, vol. 12, pp. 153–157, 1947.

[37] S. Holm. "A Simple Sequentially Rejective Multiple Test Procedure." *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[38] QoSient, LLC, "ra(1)—Argus Client Program Manual," Argus Clients 5.0.0 documentation, accessed Jun. 2026.

[39] L. Ben Dalla, Ö. Karal, M. EL-Sseid, and A. Alsharif. "An IoT-Enabled, THD-Based Fault Detection and Predictive Maintenance Framework for Solar PV Systems in Harsh Climates: Integrating DFT and Machine Learning for Enhanced Performance and Resilience." *Wadi Alshatti University Journal of Pure and Applied Sciences*, vol. 4, no. 1, pp. 41–55, Jan. 2026, https://doi.org/ 10.63318/waujpasv4i1_05.

Appendix A. Reproducibility Checklist

Input filename: Combined.csv; SHA-256: fa36f80859585f474504ca69eb951a079b35145537976c86b00aae3aab46ee59

Raw records: 1,215,890; exact duplicates removed: 21; analysis records: 1,215,869.

Split: 70:30 stratified record-level split, random_state = 42.

Five-fold CV: StratifiedKFold, shuffle = True, random_state = 42; fold-specific preprocessing and selection.

Core software: Python 3.13.5, pandas 2.2.3, NumPy 2.3.5, scikit-learn 1.8.0, XGBoost 3.1.3, SciPy 1.17.0, Matplotlib 3.10.8.