

Machine Learning in Network Intrusion Detection: A Cross-Dataset Generalization Study

Sayyad Nida Asif¹ and Prof. D. S. Kshirsagar²

^{1,2}Department of Computer Engineering

Adsul Technical Campus, Chas, India

nidasayyad1906@gmail.com¹, dipask2003@gmail.com²

Abstract: Machine learning-based network intrusion detection systems (ML-NIDS) have gained considerable attention for their ability to identify patterns in normal network traffic and malicious activities. However, achieving high performance on a single benchmark dataset does not guarantee reliable detection in different network environments. Variations in traffic characteristics, feature definitions, attack distributions, labeling practices, and network behaviour can introduce dataset shifts that affect model generalization. This paper proposes a cross-dataset evaluation framework to investigate this challenge using CICIDS2017 and UNSW-NB15, with NSL-KDD included as an optional historical reference. The framework incorporates data cleaning, semantic feature alignment, leakage-aware preprocessing, training on a source dataset, and testing on a target dataset. Random Forest, Support Vector Machine, and Multilayer Perceptron models are considered for comparative evaluation using precision, recall, F1-score, ROC-AUC, and confusion matrices. Particular emphasis is placed on experimental reproducibility and transparent reporting of results. As the study presents an evaluation framework rather than a completed experimental investigation, no unverified performance metrics are reported. This approach provides a systematic basis for distinguishing within-dataset performance from cross-dataset generalization and identifying strategies to improve the robustness and practical applicability of machine learning-based network intrusion detection systems.

Keywords: Network intrusion detection, machine learning, cross-dataset generalization, domain shift, CICIDS2017, UNSW-NB15, Random Forest, cybersecurity.

I. INTRODUCTION

Computer networks play a vital role in communication, online services, cloud-based applications, and connected devices. However, they remain vulnerable to various cyber threats, including denial-of-service (DoS) attacks, reconnaissance, brute-force attempts, exploitation, botnet activities, and other malicious behaviours. Network intrusion detection systems (NIDS) help address these threats by monitoring network traffic and analysing flow records to detect potentially suspicious activities. Although traditional signature-based approaches can effectively identify known attack patterns, their ability to detect new attack variants and evolving threats may be limited.

Machine learning provides an alternative approach by enabling classification models to learn patterns from labelled network traffic. Features such as flow duration, packet and byte counts, protocol information, and traffic-rate statistics can support the identification of normal and malicious activities. However, these models may learn dataset-specific characteristics instead of general patterns associated with cyberattacks. Differences in traffic collection methods, network configurations, and class distributions can therefore reduce their effectiveness when applied to unfamiliar environments.

This limitation raises an important research question: can a machine learning model trained on one publicly available network intrusion dataset maintain its effectiveness when evaluated on a different, independently collected dataset?



Cross-dataset evaluation provides a more rigorous assessment of generalization than conventional random train-test splitting, as it examines whether learned patterns remain applicable across different network environments.

The objective of this paper is to present a reproducible framework for evaluating and comparing commonly used machine learning models under both within-dataset and cross-dataset testing conditions. The proposed approach focuses on evaluation methodology, semantic feature compatibility, prevention of data leakage, and reliable interpretation of experimental findings. Rather than presenting unsupported performance claims, the framework emphasizes transparent reporting and provides a structured foundation for investigating the generalization and robustness of machine learning-based network intrusion detection systems.

II. BACKGROUND AND RELATED WORK

A. Machine-Learning-Based Network Intrusion Detection Systems

Supervised machine learning techniques are widely used in network intrusion detection systems to classify network traffic based on labelled observations. Decision Trees classify records through feature-based decision rules, while Random Forest combines multiple decision trees to improve prediction stability and reduce overfitting. Support Vector Machines (SVMs) identify decision boundaries between classes and can model nonlinear relationships using kernel functions. Multilayer Perceptrons (MLPs) use interconnected neural layers to learn complex patterns in network traffic. Each technique has different computational requirements, interpretability characteristics, and sensitivity to feature scaling and class imbalance.

The effectiveness of these algorithms depends on the characteristics of the dataset and the evaluation environment. Random Forest can handle nonlinear relationships and heterogeneous features with relatively limited preprocessing. SVMs can provide effective classification boundaries but may require substantial computational resources for large datasets. MLPs can capture complex feature interactions, although their performance depends on appropriate network architecture, regularization, and training configuration. Therefore, algorithm selection should consider not only benchmark performance but also generalization across different network environments.

B. Benchmark Datasets

CICIDS2017 is a widely used benchmark dataset for intrusion detection that contains labelled benign traffic and multiple categories of malicious activity. Its flow-based records are generated using CICFlowMeter and include statistical features describing network communication. The dataset documentation covers traffic collection over five weekdays and provides packet captures and machine-learning-ready CSV files [1], [2].

UNSW-NB15 was developed using normal network activities and simulated contemporary attack behaviours. It includes nine attack categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. The dataset provides 49 generated features and predefined training and testing files, making it useful for evaluating supervised intrusion detection methods [3].

NSL-KDD is a traditional intrusion detection benchmark derived from the KDD Cup 1999 dataset. Although it remains useful for historical comparisons, differences in feature definitions, traffic characteristics, and data collection practices create challenges when transferring models between this dataset and more recent benchmarks. Accordingly, NSL-KDD should be treated as an optional independent transfer experiment rather than a directly interchangeable dataset.

C. Cross-Dataset Generalization

Cross-dataset generalization describes a model's ability to maintain effective predictions when the evaluation data differ from the training distribution. In network intrusion detection, such differences may arise from changes in applications, operating systems, traffic volumes, attack techniques, feature extraction methods, and class distributions. Conventional random train-test splitting may not adequately represent these variations because both subsets originate from the same dataset and may share similar collection conditions.

Cross-dataset evaluation addresses this limitation by training a model on one dataset and testing it on another independently collected dataset. A substantial reduction in performance between within-dataset and cross-dataset testing may indicate sensitivity to dataset-specific characteristics. However, this reduction does not necessarily imply



that machine learning is unsuitable for intrusion detection. Instead, it highlights the importance of consistent feature definitions, appropriate validation strategies, probability calibration, and, where justified, adaptation using labelled target-domain samples.

III. PROBLEM STATEMENT AND OBJECTIVES

Machine-learning-based network intrusion detection systems may achieve promising results on benchmark datasets but perform less effectively when deployed in unfamiliar network environments. This discrepancy creates a reliability gap between conventional benchmark evaluation and practical deployment. Differences in traffic distributions, feature representations, attack categories, and labelling procedures can prevent models from transferring learned patterns successfully across datasets.

This study investigates whether supervised machine learning models trained on one public intrusion detection dataset can distinguish benign and malicious traffic in another dataset using features with well-defined semantic correspondence. The primary focus is on cross-dataset evaluation, consistent preprocessing, and transparent performance assessment.

The specific objectives are as follows:

Develop a consistent preprocessing pipeline for the selected intrusion detection datasets.

Identify and align semantically compatible traffic features across CICIDS2017 and UNSW-NB15.

Compare Random Forest, Support Vector Machine, and Multilayer Perceptron classifiers.

Evaluate model performance under both within-dataset and cross-dataset testing conditions.

Examine false-positive and false-negative predictions to identify potential weaknesses in intrusion detection.

Document experimental limitations and reproducibility requirements relevant to real-world deployment.

IV. DATASETS AND FEATURE ALIGNMENT

A. CICIDS2017 Dataset

CICIDS2017 contains labelled network flow records representing benign traffic and various attack types, including DoS/DDoS, brute-force attempts, web attacks, infiltration, and botnet activity. Its statistical flow features provide a foundation for supervised classification and comparative intrusion detection experiments.

Before model development, the dataset must undergo a systematic quality assessment. This process includes identifying duplicate records, handling missing and invalid numeric values, detecting infinite values, and removing identifiers or other attributes that do not represent meaningful traffic behaviour. Features that directly reveal labels or encode information unavailable during real-world prediction must also be excluded. The official dataset documentation and associated publication should be consulted to verify dataset characteristics, download sources, and usage conditions [1].

B. UNSW-NB15 Dataset

UNSW-NB15 combines normal network traffic with multiple categories of simulated malicious activity. Its feature set includes connection attributes, traffic statistics, and other measurements generated using tools such as Argus and Bro, now known as Zeek. The dataset provides predefined training and testing files and documentation describing its structure and intended use [3].

For cross-dataset evaluation, the target dataset must be processed independently while preserving its original labels for final assessment. When binary classification is used, the original labels should be mapped consistently to benign and attack classes. The mapping rules and treatment of ambiguous or unavailable labels must be documented before evaluating the models.

C. Semantic Feature Alignment

Feature compatibility is a fundamental requirement for meaningful cross-dataset evaluation. CICIDS2017 and UNSW-NB15 may differ in column names, measurement units, feature definitions, and extraction procedures. Therefore, matching features by column name or position alone is insufficient.



Potential common features include flow duration, packet counts, byte counts, protocol information, and connection-rate statistics. However, each candidate feature must be checked against its documented definition and measurement unit before inclusion. Similar names do not necessarily indicate equivalent measurements.

A feature mapping dictionary should be created to document the source and target column names, feature descriptions, units, required conversions, missing-value handling, and justification for inclusion. Categorical variables, such as protocol information, must be converted into a consistent representation. Features without a defensible semantic correspondence should be excluded from the primary common-feature experiment.

D. Label Harmonization

Label harmonization establishes a consistent target variable across datasets. For the primary binary classification experiment, records are assigned to either the benign or attack class. This approach is generally more practical than multiclass transfer because the attack taxonomies and labelling conventions of the selected datasets are not identical.

The mapping procedure must be defined explicitly and applied consistently to the training and evaluation data. Original labels should be retained for auditing and detailed error analysis. Multiclass evaluation should be conducted only when attack categories have sufficiently comparable definitions across datasets. Categories with substantially different meanings should not be combined solely to improve model performance.

E. Dataset Shift and Class Imbalance

Differences in class proportions can significantly influence intrusion detection performance. A dataset containing a large proportion of benign records may produce deceptively high accuracy if a classifier predominantly predicts the majority class. Such behaviour can conceal poor detection of malicious traffic.

To address this issue, class distributions should be reported for each dataset and evaluation split. Model performance should be assessed using precision, recall, F1-score, ROC-AUC, and confusion matrices. Precision-recall analysis should also be considered when malicious events are relatively uncommon in the intended deployment environment. These measures provide a more comprehensive understanding of detection capability and classification errors than accuracy alone.

V. PROPOSED METHODOLOGY

The proposed framework follows a sequence of controlled experimental stages designed to support reproducible within-dataset and cross-dataset evaluation.

First, CICIDS2017 and UNSW-NB15 are loaded and independently audited to examine their structure, class distributions, feature types, missing values, duplicate records, and potential data quality issues. Second, identifiers, invalid observations, and features that could introduce target-label leakage are identified and handled using documented rules.

Third, semantically compatible features are selected and mapped to a common representation. The label harmonization procedure is then applied to establish consistent benign and attack classes. Fourth, preprocessing operations, including missing-value imputation, categorical encoding, and feature scaling where required, are fitted exclusively on the training data. The learned transformations are subsequently applied to the corresponding evaluation data without fitting them on the target test set.

Fifth, Random Forest, Support Vector Machine, and Multilayer Perceptron models are trained using the selected source dataset. Each model is evaluated on a held-out test subset from the source dataset and on the separate target dataset using the same compatible feature representation. Finally, performance metrics, confusion matrices, and classification errors are compared to assess the extent to which each model generalizes across datasets.

The framework emphasizes consistent experimental conditions, prevention of data leakage, transparent documentation, and honest reporting. Since the methodology defines the experimental procedure rather than presenting completed benchmark results, numerical performance claims should be reported only after the full pipeline has been executed and the results have been verified.



VI. PROPOSED METHODOLOGY (CONTINUED)

A. Data Cleaning

The data-cleaning process examines column data types, missing values, infinite values, duplicate records, constant features, and inconsistent labels. Each data-removal or correction rule must be supported by a clear justification. Records should not be removed merely because they are difficult to classify, as doing so may introduce bias into the evaluation.

Identifier-related attributes, such as IP addresses, flow identifiers, timestamps, and dataset-specific record IDs, require careful examination. These attributes may contain information associated with the data collection process rather than general characteristics of malicious network activity. Features that introduce dataset-specific shortcuts or reveal target information should be excluded when they are unsuitable for the intended detection scenario.

B. Train-Only Preprocessing

All preprocessing operations, including numerical imputation, feature scaling, feature selection, and dimensionality reduction, must be fitted exclusively on the source training partition. The resulting transformations are then applied consistently to the source test partition and the target dataset.

Using target test labels or fitting preprocessing transformations on the complete target dataset can introduce information leakage and produce overly optimistic evaluation results. To maintain a reliable assessment, the target test set must remain isolated throughout model development and parameter selection.

C. Model Selection

Three supervised machine learning algorithms are selected for comparative evaluation.

Random Forest (RF): An ensemble learning method that combines multiple decision trees to capture nonlinear relationships between traffic features. It generally requires limited feature scaling and provides useful feature-importance estimates.

Support Vector Machine (SVM): A margin-based classification algorithm that can model nonlinear decision boundaries through kernel functions, such as the radial basis function (RBF) kernel. Its numerical inputs generally require feature scaling, and its computational cost may increase considerably with dataset size.

Multilayer Perceptron (MLP): A feed-forward neural network that learns complex relationships through interconnected layers. Appropriate scaling, architecture selection, regularization, and training configuration are important for effective performance.

Hyperparameter optimization must use validation data derived exclusively from the source training data. The same feature definitions and evaluation partitions should be maintained across all three models to ensure a fair comparison.

D. Experimental Scenarios

The experimental framework consists of four scenarios.

Scenario 1: Within-dataset baseline. Each dataset is divided into separate training and testing partitions. Stratified splitting is used where appropriate to preserve class proportions. This experiment establishes a conventional benchmark but does not independently demonstrate generalization to a different network environment.

Scenario 2: Cross-dataset transfer. Models are trained on CICIDS2017 and evaluated on UNSW-NB15, followed by the reverse experiment. Both directions use the same documented feature mapping and binary label definitions. Differences in available features and measurement semantics must be reported.

Scenario 3: Optional historical transfer. NSL-KDD may be incorporated after establishing a documented feature and label mapping. Because its schema and traffic characteristics differ from those of newer datasets, its results should be analysed separately.

Scenario 4: Limited target-domain adaptation. As an extension, a small, explicitly separated labelled sample from the target dataset may be used for calibration or fine-tuning. Its performance should be compared with the zero-shot transfer baseline, in which no target labels are used for model adaptation. The adaptation sample must remain separate from the final target test set.



E. Reproducibility

Experimental reproducibility requires complete documentation of dataset versions, file names, preprocessing procedures, random seeds, software libraries, model configurations, feature mappings, label definitions, and data-splitting strategies. Target test labels must remain isolated until the final evaluation.

When multiple random seeds are used, the mean performance and variability across runs should be reported rather than selecting only the best result. The complete implementation should support execution from the original dataset files through preprocessing, model training, evaluation, and final result generation.

VII. EVALUATION METRICS

The proposed framework uses multiple evaluation metrics to assess the classification performance of machine learning models.

For binary intrusion detection, the confusion matrix contains four outcomes:

True Positive (TP): Malicious traffic correctly classified as an attack.

True Negative (TN): Benign traffic correctly classified as normal.

False Positive (FP): Benign traffic incorrectly classified as malicious.

False Negative (FN): Malicious traffic incorrectly classified as benign.

The principal evaluation metrics are defined as follows:

Accuracy

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall (Detection Rate)

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1-Score

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Specificity

$$\text{Specificity} = \frac{TN}{TN+FP}$$

ROC-AUC is also used to measure how effectively a model ranks attack observations above benign observations across different classification thresholds. It is meaningful when both classes are represented in the evaluation data.

A. Importance of Multiple Evaluation Metrics

Accuracy measures the overall proportion of correct predictions but may conceal poor performance on minority classes. Precision indicates the proportion of predicted attacks that are actually malicious, making it relevant to the management of security alerts. Recall measures the proportion of actual attacks detected and is particularly important when missed intrusions have serious consequences.

The F1-score combines precision and recall into a single measure, while specificity evaluates the model's ability to identify benign traffic correctly. ROC-AUC provides a threshold-independent assessment of ranking performance. However, when malicious traffic is rare, precision-recall curves and metrics measured at a practical operating threshold may offer more useful insights.

The selected classification threshold must be documented. For zero-shot transfer, it should be determined using source-domain validation data. If target-domain calibration is performed, it must be reported separately as an adaptation experiment.

B. Confusion-Matrix Analysis

Confusion matrices should be generated for every source-target evaluation direction to identify the types of classification errors produced by each model. A reduction in recall may indicate that attack patterns learned from the source dataset do not adequately represent malicious activities in the target environment. Reduced precision may indicate that benign target traffic shares characteristics with attack records observed during training.



Error analysis should consider the relevant traffic features, class distributions, and dataset collection characteristics. This approach provides a more informative assessment than relying on a single aggregate performance metric.

C. Fair Model Comparison

All evaluated algorithms must use the same aligned feature set and corresponding data partitions. Hyperparameter selection must be restricted to source-domain training and validation data. Within-dataset and cross-dataset results should be presented side by side to identify differences in generalization performance.

Because the datasets may differ in size and class distribution, accuracy values should not be compared in isolation. Class counts, precision, recall, F1-score, ROC-AUC, and confusion matrices should accompany the reported results.

VIII. EXPECTED ANALYSIS AND DISCUSSION

The proposed framework is intended to investigate whether the performance of machine learning-based intrusion detection models remains consistent across different datasets. One major challenge is distribution shift, which may arise from differences in traffic characteristics, protocol distributions, application behaviour, attack implementations, and class prevalence. Consequently, decision boundaries learned from the source dataset may not remain effective when applied to the target dataset.

Random Forest is expected to provide a useful baseline because it can capture nonlinear feature interactions and generally requires less sensitivity to feature scaling than distance-based methods. Nevertheless, its performance may still depend on source-specific patterns that do not transfer to the target environment. SVM and MLP models may also be sensitive to feature scaling, representation differences, and changes in the target distribution. These observations represent hypotheses for experimental investigation rather than conclusions established by measured results.

A meaningful analysis should distinguish among three major forms of distribution change:

Representation mismatch: Differences in feature definitions, measurement units, or extraction procedures.

Covariate or concept shift: Changes in the distribution of input features or in the relationship between traffic characteristics and their labels.

Label shift: Changes in the prevalence of benign and malicious traffic between datasets.

Preprocessing and feature alignment can reduce certain compatibility problems, but they cannot eliminate every form of distribution shift. The study should therefore document the issues addressed by the experimental design and those that remain unresolved.

Operational considerations are also important. False-positive alerts increase the workload of security analysts, whereas false negatives may allow malicious activities to remain undetected. A decision threshold suitable for one benchmark may not be appropriate for another network. Practical deployment therefore requires continuous monitoring, periodic evaluation using newly labelled traffic, and controlled model recalibration.

No experiments have been executed for this methodology-focused draft, and no measured performance results are claimed. Any final empirical submission should include the results obtained by executing the complete pipeline. This requirement ensures that the reported findings represent actual experimental evidence rather than unsupported assumptions.

IX. THREATS TO VALIDITY AND LIMITATIONS

A. Benchmark-to-Reality Gap

Public intrusion detection datasets are valuable research resources, but they cannot represent every production network. Controlled laboratory environments may not capture the full diversity of operational traffic, encryption patterns, user behaviour, application changes, and emerging attack techniques. Consequently, successful cross-dataset evaluation does not, by itself, establish production readiness.



B. Feature and Label Compatibility

Features with similar names may differ in their calculation methods, units, or flow-direction conventions. Inconsistent or ambiguous labels may also affect model training and evaluation. A documented feature dictionary and label-mapping procedure can reduce these risks, although manual verification may still be necessary.

C. Data Leakage and Duplicate Records

Duplicate or near-duplicate observations and dataset-specific identifiers can allow models to learn collection artefacts rather than meaningful traffic characteristics. If related records occur in both training and testing partitions, within-dataset performance may be overestimated.

Cross-dataset evaluation reduces some forms of dependence on the source dataset but does not eliminate leakage introduced through preprocessing or feature engineering. All transformations must be fitted without access to target test labels, and the data-splitting strategy should minimise overlap between related observations.

D. Class Imbalance and Threshold Selection

Differences in attack prevalence between the source and target datasets can influence precision, accuracy, and other evaluation metrics. Even when ranking performance remains similar, a change in class prevalence may alter the operational value of model predictions.

For zero-shot transfer, the decision threshold should be selected using source-domain validation data. Any target-domain calibration must be identified explicitly as a separate adaptation procedure and evaluated on an independent target test set.

E. Computational Constraints

Large network flow datasets can require substantial memory and processing resources. Sampling may make experiments more manageable, but it can alter class proportions and reduce the representation of rare attacks. If sampling is required, the sampling strategy and resulting class counts must be documented. Results obtained from sampled and full datasets should not be treated as directly equivalent without appropriate qualification.

X. RECOMMENDATIONS FOR ROBUST NETWORK INTRUSION DETECTION

Several practices can improve the reliability of machine learning-based intrusion detection systems.

First, researchers should prioritise semantically consistent traffic features and maintain a version-controlled feature schema. Second, models should be evaluated on multiple independent datasets before making broad generalization claims. Third, evaluation should include precision, recall, F1-score, ROC-AUC, confusion matrices, and class distributions rather than relying exclusively on accuracy.

Fourth, all preprocessing transformations should be fitted on source training data, with target test data isolated until final evaluation. Fifth, domain adaptation should be considered only when target-domain information is legitimately available and can be used under a controlled experimental protocol. Adaptation results should always be compared with a zero-shot baseline.

Sixth, feature-importance methods and explainability techniques may help determine whether predictions rely on meaningful traffic characteristics or dataset-specific artefacts. Finally, machine learning-based intrusion detection should complement other security controls, including network segmentation, security logging, incident-response procedures, and human review.

A practical deployment strategy should incorporate shadow testing, alert-volume monitoring, drift detection, periodic performance assessment, and rollback procedures. Model updates should be version-controlled and validated before production deployment. Training datasets, model files, and supporting artefacts should also be protected against unauthorised modification.

XI. CONCLUSION AND FUTURE WORK

This paper presents a framework for evaluating the cross-dataset generalization of machine learning-based network intrusion detection systems. The primary concern is that a model achieving strong results on a random split of one



benchmark may not maintain comparable performance when exposed to traffic from a different network environment. Dataset shift, feature incompatibility, label differences, class imbalance, and information leakage can all affect the reliability of reported results.

The proposed framework uses CICIDS2017 and UNSW-NB15 as the primary datasets, with NSL-KDD as an optional historical reference. It incorporates semantic feature alignment, source-only preprocessing, consistent label harmonization, and source-to-target evaluation using Random Forest, Support Vector Machine, and Multilayer Perceptron classifiers. Multiple evaluation metrics and confusion-matrix analysis are recommended to assess both missed attacks and false-positive alerts.

The principal contribution is a reproducible experimental design that emphasises methodological consistency, transparent reporting, and reliable interpretation. Since the experimental pipeline has not yet been executed, the paper does not claim that any particular classifier outperforms the others. A completed empirical study should report the implementation environment, feature mappings, model configurations, measured cross-dataset results, variability across repeated runs, and detailed error analysis.

Future research may extend the framework by incorporating newer IoT and enterprise network datasets, evaluating temporal generalization, investigating few-shot domain adaptation, and examining probability calibration strategies. Additional work could explore explainable intrusion detection, robustness against adversarially manipulated traffic, and the computational overhead associated with near-real-time detection.

XII. EXPERIMENTAL REPORTING TEMPLATE

The following table provides a structure for documenting experimental findings after the complete pipeline has been executed. All result fields must be populated using verified measurements.

Table I. Cross-Dataset Evaluation Results

Experiment	Training dataset	Test dataset	Model	Precision	Recall	F1-score	ROC-AUC
Within-dataset	CICIDS2017	Held-out CICIDS2017	RF / SVM / MLP	To measure	To measure	To measure	To measure
Cross-dataset	CICIDS2017	UNSW-NB15	RF / SVM / MLP	To measure	To measure	To measure	To measure
Cross-dataset	UNSW-NB15	CICIDS2017	RF / SVM / MLP	To measure	To measure	To measure	To measure
Optional transfer	NSL-KDD	CICIDS2017 or UNSW-NB15	Selected model	To measure	To measure	To measure	To measure

For each experiment, the report should additionally include the number of evaluated records, benign and attack counts, selected features, preprocessing configuration, decision threshold, and confusion matrix. If a metric is undefined because the evaluation set lacks the required class or predictions, it should be reported as not applicable with an explanation.

REFERENCES

- [1] Canadian Institute for Cybersecurity, "Intrusion Detection Evaluation Dataset (CIC-IDS2017)," University of New Brunswick. Available: <https://www.unb.ca/cic/datasets/ids-2017.html>. Accessed: October 17, 2026.
- [2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018.
- [3] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in Military Communications and Information Systems Conference (MilCIS), 2015. Dataset information: <https://research.unsw.edu.au/projects/unsw-nb15-dataset>.



- [4] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6.
- [5] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.
- [6] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proceedings of the 2010 IEEE Symposium on Security and Privacy, 2010, pp. 305–316.
- [7] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A Survey of Network-Based Intrusion Detection Data Sets," Computers & Security, vol. 86, pp. 147–167, 2019.
- [8] M. Z. Hossain, M. A. R. Khan, M. R. Islam, S. M. S. Islam, and T. Gedeon, "Assessing Generalisation Capability of Machine Learning Models for Intrusion Detection," arXiv:2605.04407, 2026. Verify the bibliographic details and publication status before citing.

