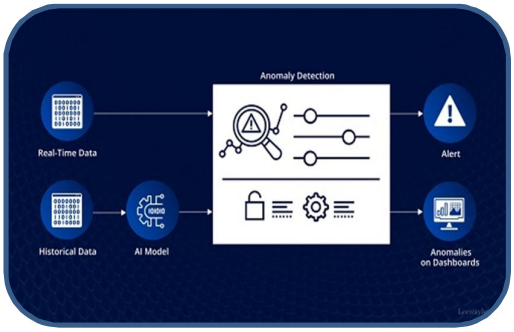
INDIAN STREAMS RESEARCH JOURNAL
ISSN: 2230-7850
IMPACT FACTOR : 5.1651 (UIF)
VOLUME - 10 | ISSUE - 11 | DECEMBER– 2020

AI-BASED ANOMALY DETECTION FOR ENHANCED CYBERSECURITY

Dr. Sayyad Razak Nizam
Department of Computer Science, Ahmednagar College.

ABSTRACT :



The rapid growth of digital technologies, cloud computing, Internet of Things (IoT) devices, and online services has significantly increased the complexity of cybersecurity threats. Traditional security mechanisms, such as signature-based intrusion detection systems, are effective in identifying known attacks but may fail to detect previously unseen threats. Artificial Intelligence (AI) and Machine Learning (ML) offer promising approaches to identifying abnormal patterns in network traffic and detecting potential cyberattacks.

This paper proposes an AI-based anomaly detection framework for enhanced cybersecurity. The proposed system combines supervised machine learning and unsupervised anomaly detection techniques to identify suspicious network activities. Random Forest is considered for classifying known attack patterns, while Isolation Forest is used to identify unusual traffic that may indicate previously unseen attacks. The framework includes network data collection, preprocessing, feature selection, model training, anomaly detection, alert generation, and performance evaluation.

The proposed approach can be evaluated using publicly available cybersecurity datasets, including CICIDS2017 and UNSW-NB15. Its effectiveness will be assessed using accuracy, precision, recall, F1-score, false-positive rate, and detection latency. Rather than relying exclusively on classification accuracy, the framework emphasizes the detection of rare attacks and the reduction of false alarms. The proposed system aims to provide a scalable and adaptable approach to network security that can support security analysts in identifying suspicious activities and responding to potential threats.

KEYWORDS : *Artificial Intelligence, Anomaly Detection, Cybersecurity, Machine Learning, Intrusion Detection System, Random Forest, Isolation Forest, Network Security.*

1. INTRODUCTION

Cybersecurity has become a critical requirement for organizations that depend on digital infrastructure to store, process, and exchange information. Increasing dependence on cloud services, online banking, e-commerce, smart devices, and interconnected networks has created new opportunities for cybercriminals.

Cyberattacks such as distributed denial-of-service (DDoS), brute-force attacks, port scanning, malware communication, unauthorized access, and data exfiltration can compromise the confidentiality, integrity, and availability of information systems.

Traditional cybersecurity mechanisms commonly use predefined signatures, rules, and known attack patterns to identify malicious activities. Although these techniques remain useful, they have limitations when attackers introduce new techniques, modify existing attack patterns, or exploit previously unknown vulnerabilities.

Artificial Intelligence provides an opportunity to improve cybersecurity by learning patterns from network traffic and identifying deviations from expected behavior. Machine learning algorithms can analyze multiple network characteristics, including packet counts, connection duration, traffic volume, protocol types, and communication patterns.

Anomaly detection is a cybersecurity technique that identifies activities that differ significantly from normal system behavior. Such activities may indicate malicious operations, configuration errors, compromised devices, or unusual but legitimate user actions.

This research proposes an AI-based anomaly detection framework that integrates supervised and unsupervised machine learning to improve the identification of suspicious network traffic.

1.1 Problem Statement

Existing intrusion detection systems may struggle to identify unknown attacks, maintain low false-positive rates, and adapt to changing network environments.

Furthermore, highly imbalanced datasets can cause machine learning models to perform well on common traffic while failing to recognize rare but important attack categories.

Therefore, a need exists for a practical anomaly detection framework that combines known-attack classification with the identification of previously unrecognized traffic patterns.

1.2 Research Objectives

The primary objectives of this research are:

1. To investigate the application of AI and machine learning in cybersecurity anomaly detection.
2. To develop a framework for identifying suspicious network traffic.
3. To integrate supervised and unsupervised machine learning algorithms.
4. To evaluate the proposed framework using publicly available cybersecurity datasets.
5. To measure detection performance using multiple evaluation metrics.
6. To investigate methods for reducing false-positive alerts.
7. To assess the feasibility of integrating the proposed framework into real-time network monitoring systems.

1.3 Research Questions

This study addresses the following research questions:

- RQ1: How effectively can machine learning identify malicious network traffic?
- RQ2: Can unsupervised anomaly detection identify unusual traffic that is not recognized by a supervised classifier?
- RQ3: How does combining supervised and unsupervised techniques affect detection performance?
- RQ4: What trade-offs exist between detection accuracy, false-positive rate, and computational overhead?
- RQ5: How can the framework be adapted to changing network behavior?

2. LITERATURE REVIEW

AI-based intrusion detection has attracted considerable research interest because conventional security systems may be insufficient for identifying complex and evolving cyber threats.

2.1 Traditional Intrusion Detection Systems

Traditional intrusion detection systems generally use signature-based or rule-based techniques.

Signature-based systems compare observed traffic with previously identified attack signatures. They can be effective for known threats but may not recognize attacks whose characteristics differ from stored signatures.

Anomaly-based systems establish a baseline of normal activity and identify deviations from that baseline. These systems can potentially identify unknown attacks but may also generate false alarms when legitimate network behavior changes.

2.2 Machine Learning for Intrusion Detection

Machine learning algorithms learn relationships between network characteristics and attack labels.

Commonly investigated algorithms include:

- Decision Tree
- Random Forest
- Support Vector Machine
- K-Nearest Neighbors
- Naive Bayes
- Gradient Boosting
- Artificial Neural Networks

Random Forest is a useful candidate because it combines multiple decision trees and can model nonlinear relationships in network data. Its performance and computational requirements depend on the dataset, feature representation, and model configuration.

Research on machine learning-based intrusion detection has emphasized the importance of systematic benchmarking, representative datasets, and appropriate evaluation metrics.

2.3 Deep Learning-Based Anomaly Detection

Deep learning techniques can identify complex relationships in high-dimensional network data.

Autoencoders learn compressed representations of input data and attempt to reconstruct the original observations. A large reconstruction error may indicate anomalous behavior.

Recurrent neural networks and Long Short-Term Memory networks can analyze sequential patterns, making them useful candidates when the temporal relationships between network events are important.

However, deep learning models may require additional training data, computational resources, and careful parameter tuning.

2.4 Unsupervised Anomaly Detection

Unsupervised algorithms attempt to identify unusual observations without requiring attack labels for every training example.

Isolation Forest isolates observations through randomly selected features and split values. Anomalous observations that are easier to isolate can receive higher anomaly scores.

This method is useful for investigating traffic patterns that differ from the training distribution. However, an anomaly is not necessarily a cyberattack, and the system must be evaluated to determine whether unusual traffic corresponds to genuine threats.

2.5 Research Gap

Existing research identifies several challenges in AI-based cybersecurity, including class imbalance, outdated attack patterns, limited dataset representativeness, adversarial manipulation, and difficulties interpreting model decisions.

A systematic literature study published in 2025 discusses these challenges and compares commonly used intrusion detection datasets and machine learning techniques.

A 2021 study in IEEE Access also benchmarked multiple supervised and unsupervised algorithms using CICIDS2017, highlighting the importance of evaluating detection performance and computational efficiency together.

Research opportunity: This paper proposes a hybrid framework that combines Random Forest classification with Isolation Forest anomaly scoring. Its contribution will be evaluated experimentally rather than assuming that combining the algorithms necessarily produces better results.

3. PROPOSED SYSTEM

The proposed system is an AI-based network anomaly detection framework designed to identify suspicious activities and assist cybersecurity monitoring.

It consists of two complementary detection components:

1. **Supervised detection:** Random Forest classifies network traffic using labeled training data.
2. **Unsupervised detection:** Isolation Forest identifies traffic that deviates from learned patterns of normal behavior.

The outputs of both components are combined using a configurable decision policy to determine whether traffic should be classified as normal, suspicious, or potentially malicious.

3.1 System Architecture

The proposed architecture contains the following stages:

Stage 1: Network Data Collection

Network flow records are collected from authorized monitoring systems or public benchmark datasets. Features may include flow duration, packet counts, byte counts, protocol information, and connection statistics.

Stage 2: Data Preprocessing

The collected data is cleaned to address missing values, duplicate records, invalid entries, and inconsistent feature representations.

Categorical variables are encoded, and numerical features are scaled when required by the selected algorithms.

Stage 3: Feature Selection

Relevant network characteristics are selected to reduce unnecessary computation and remove redundant information.

Feature selection must be performed using training data only to prevent information leakage from the test set.

Stage 4: Model Training

Random Forest is trained using labeled examples of normal and malicious network traffic.

Isolation Forest is trained primarily on normal training traffic so that it can identify observations that differ from expected behavior.

Stage 5: Anomaly Detection

Incoming network records are processed by both models. The supervised classifier predicts a traffic category, while the anomaly detector generates an anomaly score.

Stage 6: Alert Generation

A configurable decision policy combines the model outputs and assigns an alert level. High-risk observations can be forwarded to a security analyst for investigation.

Stage 7: Performance Evaluation

The framework is evaluated using a separate test dataset and metrics that measure detection effectiveness, false alarms, and processing efficiency.

3.2 Proposed Workflow

The overall workflow is:

Network Traffic Collection
↓
Data Cleaning and Preprocessing
↓
Feature Engineering and Selection
↓
Training and Validation
↓

Random Forest Classification + Isolation Forest Scoring
↓
Combined Decision Policy
↓
Anomaly Classification and Alert Generation
↓
Performance Evaluation and Model Improvement

4. RESEARCH METHODOLOGY

4.1 Research Design

The study follows an experimental research methodology. Multiple machine learning configurations will be trained and evaluated under controlled conditions.

The baseline model will use Random Forest. The proposed hybrid model will combine Random Forest predictions with Isolation Forest anomaly scores.

The comparison will determine whether the hybrid approach improves the detection of important attack categories or previously unseen patterns without introducing an unacceptable increase in false alarms.

4.2 Dataset Selection

The following public datasets are suitable candidates for the research.

A. CICIDS2017

CICIDS2017 contains labeled network traffic representing benign activities and multiple attack categories. It is useful for investigating supervised intrusion detection and anomaly detection.

B. UNSW-NB15

UNSW-NB15 contains normal network traffic and several categories of synthetic attack activity. It provides an alternative dataset for assessing the generalizability of the proposed method.
Official dataset information and research documentation should be consulted before selecting specific files and features.

Dataset links:

- CICIDS2017: https://www.unb.ca/cic/datasets/ids-2017.html
- UNSW-NB15: https://research.unsw.edu.au/projects/unsw-nb15-dataset

The experiment should begin with one dataset and, if resources permit, be repeated on the second dataset.

4.3 Data Preprocessing

Data preprocessing is essential for obtaining reliable results.
The following operations will be performed:

1. Remove duplicate records where appropriate.
2. Handle missing and invalid values.
3. Encode categorical features.
4. Identify constant and redundant features.
5. Inspect extreme values without automatically removing potentially meaningful attacks.
6. Separate training, validation, and test data.
7. Address class imbalance using appropriate training-only techniques.
8. Ensure that identifiers and target-derived information do not leak attack labels into model inputs.

Where traffic records from the same capture or time interval are strongly related, a time-based or group-based split should be considered to reduce the risk of overly optimistic evaluation.

4.4 Feature Engineering

Potential input features include:

Feature	Description
Flow Duration	Duration of a network flow
Total Forward Packets	Number of packets in the forward direction
Total Backward Packets	Number of packets in the reverse direction
Total Forward Bytes	Amount of data transmitted forward
Total Backward Bytes	Amount of data transmitted backward
Flow Bytes per Second	Rate of data transfer
Flow Packets per Second	Rate of packet transmission
Protocol	Network protocol associated with the flow
Packet Length Statistics	Statistical characteristics of packet sizes
TCP Flag Counts	Counts of selected TCP control flags
Active Time	Time during which a flow is active
Idle Time	Time during which a flow is inactive

The exact features will depend on the selected dataset and the availability of corresponding fields.

4.5 Algorithm 1: Random Forest

Random Forest is an ensemble learning algorithm that combines predictions from multiple decision trees.

Each tree learns patterns from a sampled training dataset, and the final classification is determined by aggregating the individual tree predictions.

For a classification problem, the predicted class can be represented as:

y^=mode {T1(x),T2(x),…,Tn(x)}\hat{y}=\operatorname{mode}\{T_1(x),T_2(x),\ldots,T_n(x)\}

Where:

- xx represents the input network-flow features.
- TiT_i represents the prediction of the ii-th decision tree.
- nn represents the number of trees.
- y^\hat{y} represents the predicted traffic class.

Random Forest will be used to classify known categories of network traffic.

4.6 Algorithm 2: Isolation Forest

Isolation Forest is an unsupervised anomaly detection algorithm that identifies observations that are relatively easy to isolate from the rest of the data.

Its anomaly score can be expressed conceptually as:

s(x,n)=2−E[h(x)]c(n)s(x,n)=2^{-\frac{E[h(x)]}{c(n)}}

Where:

- xx is an observation.
- E[h(x)]E[h(x)] is the average path length required to isolate the observation.
- c(n)c(n) is a normalization term based on sample size.

A higher anomaly score generally indicates that an observation is easier to isolate.

The exact score direction and threshold interpretation should be verified for the software implementation used.

Isolation Forest will be trained primarily on normal network traffic. Its scores will then be used to identify observations that may require further investigation.

4.7 Hybrid Decision Mechanism

The proposed framework combines the output of Random Forest with the anomaly score generated by Isolation Forest.

Let:

- PA(x)P_A(x) represent the estimated probability that traffic belongs to an attack class.
- SI(x)S_I(x) represent the normalized anomaly score.
- α\alpha represent the relative weight assigned to supervised classification.
- β\beta represent the relative weight assigned to anomaly detection.

A conceptual combined risk score is:

R(x)=αPA(x)+βSI(x)R(x)=\alpha P_A(x)+\beta S_I(x)

Subject to:

α+β=1\alpha+\beta=1

The score can be compared with a threshold τ\tau:

Decision(x)={Suspicious,R(x)≥τNormal,R(x)<τ\text{Decision}(x)= \begin{cases} \text{Suspicious}, & R(x)\geq \tau\\ \text{Normal}, & R(x)<\tau \end{cases}

The threshold and model weights must be calibrated using validation data.

This equation represents a proposed design, not an established optimal decision rule. Random Forest probabilities and anomaly scores must be calibrated or normalized appropriately before they can be meaningfully combined.

In a practical implementation, the system may also maintain separate rules for known attacks, high anomaly scores, and uncertain classifications.

4.8 Model Training and Testing

The experimental procedure will consist of the following steps:

1. Load the selected cybersecurity dataset.
2. Inspect the distribution of normal and malicious records.
3. Clean and preprocess the data.
4. Divide the data into training, validation, and test sets.
5. Train the Random Forest classifier.
6. Train Isolation Forest using normal training observations.
7. Generate predictions and anomaly scores on validation data.
8. Select the hybrid decision threshold using a predefined validation objective.
9. Evaluate all models on the untouched test set.
10. Record the metrics, computational requirements, and error patterns.

To measure generalization, the test set must not be used to select features, tune thresholds, or optimize model parameters.

5. EVALUATION METRICS

Accuracy alone is insufficient for evaluating cybersecurity systems because normal network traffic may greatly outnumber malicious traffic.

The proposed study will use the following metrics.

5.1 Accuracy

Accuracy measures the proportion of correctly classified observations.

Accuracy=TP+TNTP+TN+FP+FNAccuracy=\frac{TP+TN}{TP+TN+FP+FN}

5.2 Precision

Precision measures the proportion of predicted attacks that are genuine attacks.

Precision=TPTP+FPPrecision=\frac{TP}{TP+FP}

5.3 Recall

Recall measures the proportion of actual attacks correctly detected.

Recall=TPTP+FNRecall=\frac{TP}{TP+FN}

5.4 F1-Score

The F1-score is the harmonic mean of precision and recall.

F1=2×Precision×RecallPrecision+RecallF1=2\times\frac{Precision\times Recall}{Precision+Recall}

5.5 False-Positive Rate

The false-positive rate measures the proportion of normal observations incorrectly classified as attacks.

FPR=FPFP+TNFPR=\frac{FP}{FP+TN}

5.6 Detection Latency

Detection latency measures the time between the availability of an observation and the production of its detection result.

Latency should be measured under a clearly specified hardware configuration, traffic volume, and processing setup.

5.7 Computational Overhead

Computational overhead can be assessed using:

- Model training time.
- Prediction time per observation.
- Throughput in records per second.
- Memory consumption.
- CPU and, where applicable, GPU utilization.

The study should report precision, recall, F1-score, and false-positive rate alongside accuracy. For imbalanced data, the precision-recall curve and area under that curve can also be informative.

6. EXPERIMENTAL SETUP

The proposed implementation can be developed using Python and commonly available machine learning libraries.

Component	Proposed Tool
Programming Language	Python
Data Processing	Pandas, NumPy
Machine Learning	Scikit-learn
Data Visualization	Matplotlib, Seaborn
Development Environment	Jupyter Notebook or Google Colab
Dataset	CICIDS2017
Secondary Dataset	UNSW-NB15
Supervised Model	Random Forest
Unsupervised Model	Isolation Forest

6.1 Baseline Models

To evaluate the contribution of the proposed hybrid model, the following configurations should be compared:

- Model A: Random Forest only.
- Model B: Isolation Forest only.
- Model C: Random Forest combined with Isolation Forest.
- Model D: A suitable additional baseline, such as a Decision Tree or Support Vector Machine.

The comparison should use identical test observations and consistent evaluation procedures wherever applicable.

6.2 Experimental Controls

For a fair comparison:

1. Use the same training and test partitions for comparable models.
2. Document preprocessing operations and hyperparameters.

3. Prevent test-data leakage.
4. Report results for individual attack categories.
5. Evaluate multiple random seeds where appropriate.
6. Record hardware specifications and software versions.
7. Preserve the original class distribution in the test set unless a different evaluation scenario is explicitly justified.

7. RESULTS AND DISCUSSION

Note: The following section defines the planned analysis. Actual performance values must be obtained by running the experiments.

7.1 Proposed Results Table

After implementation, the results can be presented using the following format.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	Detection Time
Random Forest	To measure	To measure	To measure	To measure	To measure	To measure
Isolation Forest	To measure	To measure	To measure	To measure	To measure	To measure
Hybrid Model	To measure	To measure	To measure	To measure	To measure	To measure

The table must be populated with measured results before the paper is submitted as an experimental research study.

7.2 Expected Observations

The experiments will investigate the following possibilities:

Improved detection coverage: The hybrid model may identify suspicious observations missed by the supervised classifier because it also considers deviations from normal traffic patterns.

Reduction in missed attacks: Combining the two detection signals may improve recall for certain attack categories. However, this must be verified experimentally.

False-positive challenges: Unusual but legitimate traffic may receive high anomaly scores. Threshold selection and analyst review will therefore be important.

Computational requirements: The combined model may require additional computation compared with a single model. Its performance should be measured under realistic traffic volumes.

Generalization: Performance on a second dataset may differ from performance on the training dataset. Cross-dataset evaluation can help identify weaknesses caused by dataset-specific patterns.

7.3 Interpretation of Results

A hybrid model should not be considered superior solely because it achieves a higher accuracy value.

Its practical usefulness depends on whether it improves the detection of relevant threats while maintaining an acceptable false-positive rate, processing speed, and resource requirement.

For example, a model with high recall but an extremely high false-positive rate may overwhelm security analysts. Conversely, a model with few false alarms but low recall may fail to identify important attacks.

The final model should therefore be selected according to the intended cybersecurity environment and its operational requirements.

8. APPLICATIONS

The proposed AI-based anomaly detection framework may support several cybersecurity applications.

8.1 Enterprise Network Security

Organizations can use anomaly detection to identify unusual internal traffic, suspicious connections, and potentially compromised devices.

8.2 Cloud Security
Cloud environments generate large volumes of network and system activity. AI-based monitoring may help identify unusual access patterns and suspicious communication.

8.3 Internet of Things
IoT devices may have limited computing resources and inconsistent security controls. Anomaly detection can help identify unusual device behavior and unexpected communication patterns.

8.4 Financial Services
Financial organizations can use anomaly detection to investigate suspicious network activity and unusual access to sensitive systems. Network anomaly detection should complement, rather than replace, transaction fraud detection and other dedicated security controls.

8.5 Educational Institutions
Colleges and universities can investigate unusual network traffic, repeated failed login attempts, unauthorized access attempts, and unexpected communication with external systems.

9. CHALLENGES AND LIMITATIONS
Despite its potential benefits, the proposed framework faces several challenges.

9.1 Imbalanced Datasets
Cybersecurity datasets often contain substantially more normal observations than attack observations. This imbalance may reduce the model's ability to identify rare attack categories.

9.2 False-Positive Alerts
Legitimate changes in user activity, applications, or network traffic can trigger anomaly alerts. Excessive false positives can reduce the usefulness of the system.

9.3 Concept Drift
Normal network behavior changes over time. Models trained on historical traffic may gradually become less effective as applications, devices, and user behavior evolve.

9.4 Adversarial Manipulation
Attackers may attempt to modify their behavior to avoid detection. Models should therefore be evaluated for robustness and should not be treated as the sole security control.

9.5 Dataset Limitations
Public benchmark datasets cannot represent every real-world network environment. High performance on a benchmark does not guarantee equally high performance in a live organization.

9.6 Privacy and Data Protection
Network monitoring may expose sensitive information about users and organizational activity. Data collection should follow applicable privacy requirements, access restrictions, and data-retention policies.

9.7 Explainability
Security analysts need understandable explanations for high-risk alerts. Feature importance, interpretable models, and event-level evidence can help analysts investigate suspicious behavior.

Risk-based alert prioritization: The framework could incorporate asset importance, threat intelligence, and contextual information to prioritize alerts for human investigation.

11. CONCLUSION

This paper proposes an AI-based anomaly detection framework for enhanced cybersecurity through the integration of supervised and unsupervised machine learning.

Random Forest is used to classify known network traffic patterns, while Isolation Forest identifies unusual observations that may warrant further investigation. A configurable decision mechanism combines these signals to produce suspicious-traffic alerts.

The proposed methodology includes data preprocessing, feature engineering, model training, validation, testing, and evaluation using accuracy, precision, recall, F1-score, false-positive rate, and detection latency.

The primary research contribution is a testable hybrid framework that can be compared against individual machine learning models under controlled experimental conditions. Its effectiveness will depend on dataset quality, feature selection, threshold calibration, computational requirements, and the ability to generalize to unseen network conditions.

Future experimental work should evaluate the proposed framework on CICIDS2017 and UNSW-NB15, report actual measurements, and investigate its suitability for real-world network monitoring.
AI-based anomaly detection should be regarded as a component of a broader cybersecurity strategy that includes access control, patch management, endpoint protection, logging, incident response, and human oversight.

REFERENCES

1. A Systematic Literature Study of Machine Learning Techniques Based Intrusion Detection: Datasets, Models, Challenges, and Future Directions. *Journal of Big Data*, 2025. Available at: https://link.springer.com/article/10.1186/s40537-025-01323-2
2. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., and Hotho, A. (2019). A Survey of Network-Based Intrusion Detection Data Sets. *Computers & Security*, 86, 147–167. https://doi.org/10.1016/j.cose.2019.06.005
3. Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). 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)*.
4. Benchmarking of Machine Learning for Anomaly-Based Intrusion Detection Systems in the CICIDS2017 Dataset. *IEEE Access*, 9, 22351–22370, 2021. https://doi.org/10.1109/ACCESS.2021.3056614
5. Deep Learning Approaches for Anomaly-Based Intrusion Detection Systems: A Survey, Taxonomy, and Open Issues. *Knowledge-Based Systems*, 2019. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0950705119304897
6. Choi, S., Kim, S., and Ryu, J. (2026). A Survey of Deep Learning-Based Network Anomaly Detection: Benchmarking on NSL-KDD, UNSW-NB15, and CICIDS2017. *IEIE Transactions on Smart Processing & Computing*, 15(3), 410–425. https://doi.org/10.5573/IEIESPC.2026.15.3.410
7. University of New Brunswick. CICIDS2017 Dataset. Available at: https://www.unb.ca/cic/datasets/ids-2017.html
8. Australian Centre for Cyber Security, UNSW Canberra. UNSW-NB15 Dataset. Available at: https://research.unsw.edu.au/projects/unsw-nb15-dataset