African Journal of Education Science and Technology (AJEST) Vol. 8 No. 4 (May, 2026)
University of Eldoret, Kenya, Mount Kenya University, Kenya, Chukwuemeka Odumegwu Ojukwu University, Nigeria, Kyambogo University, Uganda and University of Makeni, Sierra Leone

A Multi-Layer Hybrid Machine Learning Framework for Intrusion Detection Using the UNSW-NB15 Dataset

***Levine Nakhone[1], Anthony Kibira Wanjoya[2] & Mwirigi Kiula[3]**
[1]*Department of Computer Science and Information Technology, Co-operative University of Kenya*
[2]*Department of Mathematical Sciences, Co-operative University of Kenya*
[3]*Department of Computer Science, St. Paul's University, Kenya*

Corresponding E-mail: mugenilevine@gmail.com

Abstract

Network intrusion detection increasingly combines rule-based, supervised, and anomaly-based methods on the premise that complementary detectors can increase robustness. In this study, that assumption is tested using the UNSW-NB15 benchmark and a structured three-layer framework comprising interpretable high-confidence rules, supervised machine-learning classifiers, and a normal-only autoencoder. A review of the data revealed considerable repetition among the predictor profiles, so leakage-aware grouped validation was used during model selection to ensure that identical predictor profiles did not appear in both the development and validation subsets. Logistic Regression, Random Forest, and Histogram-based Gradient Boosting were trained under the original class distribution, with balanced class weighting and SMOTENC. The autoencoder achieved a validation ROC-AUC of 0.9190, an attack recall of 0.7396, and a false-positive rate of 0.0757, while class-weighted Gradient Boosting provided the best balance among the supervised methods. Although the initially proposed ungated hierarchy improved sensitivity to attacks, it also led to an accumulation of false positives; therefore, a confidence-gated rule was introduced that allowed the autoencoder to override a supervised normal prediction only if the supervised attack probability was at least 0.45. On the untouched official test set, class-weighted Gradient Boosting achieved an accuracy of 0.9068, a balanced accuracy of 0.9031, an attack precision of 0.8964, an attack recall of 0.9393, and a false-positive rate of 0.1330. The gated hybrid raised recall to 0.9450 but lowered balanced accuracy to 0.8871 and increased the false-positive rate to 0.1708. McNemar's exact test showed significantly different paired error performance ($p < .001$): the hybrid detected 259 additional attacks but introduced 1,399 extra false positives. On a separate test subset, the gap in balanced accuracy decreased to 0.0019. These results indicate that hybridization is not inherently better; rather, its usefulness depends on leakage-aware evaluation, controlled fusion, category-specific complementarity, and the operational cost of false alarms.

Keywords: *intrusion detection systems; hybrid machine learning; autoencoder; gradient boosting; UNSW-NB15*

Introduction

Intrusion detection systems (IDSs) remain an important defensive measure for organizations whose digital services rely on continuously connected networks. While signature- and rule-based systems are interpretable and effective for known threats, they may fail to detect traffic that does not fit predefined patterns. Machine-learning methods, by contrast, address this shortcoming by learning decision boundaries from labelled flow data or by identifying deviations from learned normal behaviour (Moustafa & Slay, 2015, 2016; Ahmad et al., 2021; Song et al., 2021). Recent reviews have also identified machine learning as a major approach for protecting networked and critical infrastructure (Pinto et al., 2023).

The UNSW-NB15 benchmark includes normal traffic and nine types of attacks: Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, and Worm, and it is commonly used to evaluate supervised, unsupervised, and hybrid IDS systems (Moustafa & Slay, 2015, 2016). Yet class imbalance, class overlap, duplicate observations, and repeated patterns can influence reported performance, meaning that the evaluation protocol is just as important as the choice of model (Al-Daweri et al., 2020; Zoghi & Serpen, 2024). Moreover, leakage during partitioning or preprocessing can lead to an overestimation of model performance if information is allowed to cross from the training to the validation phase (Bouke & Abdullah, 2023; Saputra et al., 2026).

The motivation for research into hybrid IDSs comes from the complementarity of their detection methods. Supervised classifiers can identify known attack patterns accurately, while anomaly detectors can identify uncertain traffic. Recent studies have combined Random Forest with autoencoders or integrated machine-learning and neural-network components (Wang et al., 2023; Sun et al., 2023; Kumar et al., 2025). However, because each component can independently declare an attack, sensitivity may increase at the expense of a higher number of false alarms. This is operationally important because

too many alerts can overwhelm analysts. Balanced accuracy, precision, recall, F1-score, ROC-AUC, specificity, and the false-positive rate should therefore be considered alongside accuracy (Bai & Fossaceca, 2025; Shanmugam et al., 2025).

Study Positioning and Research Objectives

This study examines a structured intrusion detection system combining three elements: one, an interpretable high-confidence rule layer; two, supervised classifiers for distinguishing between normal and attack cases, and three, a normal-only autoencoder that uses reconstruction error as an anomaly score. It compares class weighting with SMOTENC, employs grouped validation to avoid identical predictor profiles appearing in both the development and validation splits, examines both ungated and confidence-gated fusion, and tests the frozen final architecture on the official UNSW-NB15 test subset. A sensitivity analysis also removes test predictor profiles that appear in the training data. The aims were to design the three component layers, compare the performance of the individual components with that of the hybrid system, assess the impact of addressing class imbalance, and determine whether the benefits of the hybrid approach remain under independent testing. The contribution is therefore not an assumption that greater model complexity is better, but rather an empirical investigation into whether adding detection layers improves generalization after accounting for the false-positive cost of fusion.

Materials and Methods

Research Design, Dataset and Leakage Controls

The experiment used a positivist quantitative experimental approach in a Python/Jupyter Notebooks, with Pandas and NumPy for data handling, scikit-learn for supervised modeling and preprocessing, imbalanced-learn for SMOTENC, and PyTorch for the autoencoder. Where possible, a random seed value of 42 was fixed. The study used the official training and testing files from the carefully curated UNSW-NB15 dataset. Each analysed record included 34 predictors together with attack_cat and a binary label; a label of 0 indicated normal traffic, while a label of 1 indicated attack traffic. The attack_cat field was excluded from the predictor set and used only to interpret results at the category level. The categorical predictors were proto, service, and state. All other predictors were numerical.

The data audit revealed no missing values, infinities, constant predictors, or datatype inconsistencies, although it detected a high level of repetition. Specifically, the training subset included 78,519 exact duplicate rows and 83,243 duplicate predictor profiles, whereas the test subset had 32,361 exact duplicate rows and 34,101 duplicate predictor profiles. In addition, 20,851 of the official-test predictor profiles were also found in the training dataset. To prevent identical predictor profiles from appearing in both the development and validation partitions, a group identifier was generated by hashing all 34 predictor values and then applying StratifiedGroupKFold with five folds, shuffling, and a random_state of 42. This resulted in 140,273 development records and 35,068 validation records with no overlap in predictor profiles or exact rows. A final profile-novelty sensitivity analysis removed all official-test predictor profiles that were present in the training set, leaving 61,481 records.

Table 11: Dataset Splits Used in Model Development and Evaluation

Table 1. Dataset Splits Used in Model Development and Evaluation

Dataset split	Total records	Normal records	Attack records	Normal (%)	Attack (%)
Official training	175,341	56,000	119,341	31.94	68.06
Leakage-safe validation	35,068	11,200	23,868	31.94	68.06
Official test	82,332	37,000	45,332	44.94	55.06
Non-overlapping test subset	61,481	32,601	28,880	53.03	46.97

Preprocessing and Model Development

The preprocessing parameters were determined using the development training data. For Logistic Regression and Random Forest, one-hot encoding was applied to the categorical variables, and standardization was used for the numerical variables. For Histogram-based Gradient Boosting, ordinal encoding was used for the categorical variables and the numerical variables were left unstandardized. Candidate interpretable features were selected using mutual information, shallow-tree importance, and numerical correlation. A constrained entropy-based decision tree (with a maximum depth of 4 and a minimum of 500 records per leaf) was used to discover rules. Attack leaves were retained only if they had at least 500 development-training records, at least 100 validation records, a development attack precision of at least 0.90, and a validation precision of at least 0.85. Six leaves satisfied these conditions; the rule layer was therefore treated as a high-confidence attack detector rather than a complete classifier.

Three supervised classifiers were assessed: Logistic Regression, Random Forest, and Histogram-based Gradient Boosting. Logistic Regression employed SAGA with L2 regularization. Random Forest used 150 trees with a maximum depth of 25. Gradient Boosting used a learning rate of 0.08, a maximum of 200 iterations, and early stopping. The models were tested using the original distribution and balanced class weighting, while SMOTENC was applied only to the development training data. By contrast, the validation and test data were never resampled. Since attacks constitute the binary majority in the curated training split, balancing increased the weight of normal traffic rather than directly balancing the rare attack families.

Autoencoder and Hybrid Decision Logic

The autoencoder was trained using only normal development traffic. The normal predictor profiles were divided into 80% training and 20% calibration sets using GroupShuffleSplit to ensure that the same normal profiles did not appear in both groups. The numerical variables were standardized and the categorical variables were one-hot encoded. The PyTorch network had a symmetric fully connected architecture consisting of input-128-64-32-64-128-output, with ReLU activation functions in the hidden layers and a linear output function. Training used mean-squared reconstruction error, the Adam optimization method (with a learning rate of 0.001 and a weight decay of 1e-5), a batch size of 512, a maximum of 100 epochs, and early stopping with a patience level of 10. Candidate thresholds were obtained from the 90th to 99.5th percentiles of the normal calibration reconstruction error, while the 92.5th percentile (threshold value 0.0013928439) was selected because it maximised the geometric mean among candidates meeting a validation false-positive-rate constraint of 10% or less.

The hierarchy initially gave attack priority to any retained rule, then to any anomaly detected by the autoencoder that exceeded the threshold, and otherwise to the supervised classifier. Because this logical combination led to an accumulation of false positives, a confidence-gated alternative was selected before examining the official test set. At the final stage, a high-confidence rule was given unconditional priority; otherwise, class-weighted Gradient Boosting predicted an attack when the supervised probability was at least 0.50. If the supervised model judged the case as normal, the autoencoder could override that decision only if the record was anomalous and the supervised attack probability was still at least 0.45. The gate and anomaly threshold were fixed after validation-based selection.

Evaluation and Statistical Analysis

Performance was assessed using accuracy, balanced accuracy, attack precision, attack recall, F1-score, normal recall/specificity, false-positive rate (FPR), Matthews Correlation Coefficient (MCC), confusion matrices, and ROC-AUC when continuous scores were available. Ablation studies compared the standalone components, two-layer combinations, the original hierarchy, and the gated hybrid. McNemar's exact two-sided test was used to compare paired predictions on the official test set for the best standalone supervised model and the final gated hybrid at a significance level of alpha = 0.05. Category-level detection rates were computed using attack_cat only after the model had made its prediction. The non-overlapping test subset was treated as a profile-novelty sensitivity analysis, not as proof of controlled zero-day detection.

Results And Discussion

Standalone Performance and Class-Imbalance Handling

The six-rule detector achieved a validation attack precision of 0.9847 and a false-positive rate of 0.0208, confirming that the retained rules functioned as a highly specific first-stage detector, although attack recall was deliberately lower at 0.6278. The normal-only autoencoder achieved a validation ROC-AUC of 0.9190, with an attack precision of 0.9542, attack recall of 0.7396, and a false-positive rate of 0.0757. The median reconstruction error was 0.000153 for normal traffic and 0.020150 for attacks, but performance varied substantially by attack family, showing that reconstruction error was useful but provided an uneven anomaly signal.

For the supervised models, class weighting offered the most effective practical balance for the tree-based approaches. Weighted Random Forest reached a validation attack recall of 0.9518, while weighted Gradient Boosting had the highest balanced accuracy (0.9407), the highest attack precision (0.9751), and the lowest false-positive rate (0.0506) among the weighted supervised models. SMOTENC, on the other hand, did not provide a consistent performance benefit and came with a high computational cost: Gradient Boosting took approximately 1,252 seconds with SMOTENC compared with about 16 seconds using class weighting, while Random Forest took approximately 1,347 seconds rather than 85 seconds. These findings therefore align with the general observation that resampling can improve minority-class representation without ensuring better overall generalization (Chawla et al., 2002; Shanmugam et al., 2025).

Table 12: Validation and Official-Test Performance of the Main Configurations

Table 2. Validation and official-test performance of the main configurations

Configuration	Accuracy	Balanced Accuracy	Attack Precision	Attack Recall	F1-Score	Specificity	FPR
Validation - Weighted GB	0.9376	0.9407	0.9751	0.9321	0.9531	0.9494	0.0506
Validation - Gated hybrid	0.9416	0.9401	0.9692	0.9441	0.9565	0.9362	0.0638
Validation - Ungated hybrid	0.9274	0.9146	0.9437	0.9501	0.9468	0.8791	0.1209
Official test - Weighted GB	0.9068	0.9031	0.8964	0.9393	0.9173	0.8670	0.1330
Official test - Gated hybrid	0.8929	0.8871	0.8714	0.9450	0.9067	0.8292	0.1708
Official test - Ungated hybrid	0.8648	0.8546	0.8266	0.9546	0.8860	0.7546	0.2454

Hybrid Fusion and Independent-Test Generalization

Initially, validation indicated that the gated hybrid could provide a useful compromise. Compared with weighted Gradient Boosting alone, it raised accuracy from 0.9376 to 0.9416, increased recall from 0.9321 to 0.9441, boosted the F1-score from 0.9531 to 0.9565, and improved the MCC from 0.8619 to 0.8682, although balanced accuracy remained almost unchanged. The original ungated hierarchy was clearly less advantageous because, although recall rose to 0.9501, the FPR more than doubled to 0.1209 and balanced accuracy dropped to 0.9146. This shows that fusion logic, rather than simply the presence of multiple detectors, is also a major modelling choice.

On the official test partition, class-weighted Gradient Boosting showed the most balanced generalization, achieving an accuracy of 90.68%, a balanced accuracy of 90.31%, and a false-positive rate (FPR) of 0.1330. Although the hybrid architectures improved attack recall, they did so at the cost of overall precision. Specifically, the gated hybrid increased recall

to 0.9450 but reduced balanced accuracy to 0.8871 and raised the FPR to 0.1708, while the ungated full hybrid boosted recall to a maximum of 0.9546 but experienced a sharp rise in FPR to 0.2454. Overall, the greater sensitivity of the hybrid models did not result in better performance when tested independently, so class-weighted Gradient Boosting remained the better-balanced option.

The confusion-matrix comparison makes the trade-off clear. Weighted Gradient Boosting correctly classified 42,579 attacks and 32,079 normal entries, with 2,753 false negatives and 4,921 false positives. The gated hybrid correctly classified 42,838 attacks and 30,680 normal entries, with 2,494 false negatives and 6,320 false positives. Compared with the standalone model, the hybrid detected 259 more attacks but also added 1,399 more false positives, equivalent to about 5.4 additional false alarms for every extra attack recovered. McNemar's exact test showed a significant difference in the paired errors (there were 1,399 cases in which weighted Gradient Boosting was correct but the hybrid was not, compared with 259 cases in the other direction; $p < .001$), thus favouring the standalone model overall. An operational environment in which the cost of missing an attack is much greater than that of a false alarm could still choose the hybrid, but that would be a decision based on cost considerations rather than a general improvement in performance.

Attack-Specific Behavior and Novel-Profile Sensitivity

The benefits of the hybrid approach were specific to each category. On the official test set, Fuzzer detection rose from 0.6341 with weighted Gradient Boosting to 0.6729 with the gated hybrid, representing an improvement of 3.88 percentage points. There was only a slight increase for DoS, Exploits, and Reconnaissance, while Analysis, Backdoor, Generic, Shellcode, and Worms showed no practical improvement. Normal specificity decreased from 0.8670 to 0.8292. The autoencoder performed well in several categories but was weak for Shellcode, Reconnaissance, and Worms. These results justify per-family evaluation because overall binary metrics can hide the extent to which anomaly evidence is complementary (Song et al., 2021; Park et al., 2025).

On the 61,481-record non-overlapping test subset, weighted Gradient Boosting was still the best-balanced model, achieving a balanced accuracy of 0.8960, a recall of 0.9128, and an FPR of 0.1209; the gated hybrid achieved a balanced accuracy of 0.8941, a recall of 0.9185, and an FPR of 0.1303. The gap in balanced accuracy therefore decreased from 0.0160 on the full official test set to 0.0019 on the non-overlapping subset, although the hybrid retained a recall advantage of 0.0057. This indicates that layered detection becomes more competitive when predictor profiles are novel. But the result should not be taken to mean that it achieves zero-day detection because the same attack families were present in both the development and test data.

Table 13: Sensitivity Analysis of Predictor Profiles not Present in the Training Data

Table 3. Sensitivity analysis of predictor profiles not present in the training data

Model	Records	Accuracy	Balanced Accuracy	Attack Precision	Attack Recall	F1-score	FPR
Weighted Gradient Boosting	61,481	0.8949	0.8960	0.8700	0.9128	0.8909	0.1209
Final gated hybrid	61,481	0.8926	0.8941	0.8620	0.9185	0.8893	0.1303
Original ungated hybrid	61,481	0.8552	0.8597	0.7943	0.9337	0.8583	0.2143
Autoencoder only	61,481	0.7928	0.7872	0.8360	0.6953	0.7592	0.1208
Rule only	61,481	0.7197	0.7027	0.9608	0.4205	0.5850	0.0152

Methodological and Practical Implications

The drop in validation performance relative to test results supports concerns regarding benchmark leakage and the presence of repeated patterns. The balanced accuracy of weighted Gradient Boosting decreased from 0.9407 to 0.9031, while its false-positive rate increased (worsened) from 0.0506 to 0.1330. For the gated hybrid, balanced accuracy dropped from 0.9401 to 0.8871 and its false-positive rate increased from 0.0638 to 0.1708. Attack recall remained similar or improved slightly, indicating that the main cause of the generalization loss was normal traffic being misclassified as attack traffic. Grouping the validation samples prevented one type of leakage that might have occurred if identical predictor profiles crossed a row-level split, thus confirming the concerns expressed by Al-Daweri et al. (2020), Bouke and Abdullah (2023), and Saputra et al. (2026). Therefore, benchmark scores must be interpreted with regard to split design, feature representation, duplicate handling, target definition, and balancing strategy, rather than compared solely on raw accuracy values.

The component detectors have several practical applications. Because it has a very low false-positive rate, the rule layer is appropriate for high-confidence triage, whereas the autoencoder should be regarded as a secondary risk indicator rather than a component that can unconditionally override decisions. The impact of anomaly detection varies according to the point at which it enters the decision process (Wang et al., 2023). Class weighting was better than SMOTENC in this binary case because it provided a better balance between performance and computational cost, although binary class balance should not be confused with rare attack-family balance (Rao & Babu, 2023; Sayegh et al., 2024; Musthafa et al., 2024).

There are several limitations to the findings. Rather than using actual organizational traffic, the study relied on benchmark traffic. The curated dataset included 34 predictors; the rules were derived from and specific to that dataset; while the autoencoder was not subjected to a category-holdout design. Once validated, the final supervised model was kept fixed to maintain experimental separation. Real-time throughput, memory usage, adversarial robustness, and concept drift were not examined. Therefore, before deployment in a production environment, the system would need to be recalibrated and prospectively validated using local traffic.

Conclusion and Recommendations

A three-layer intrusion-detection framework was developed and assessed using the UNSW-NB15 dataset, combining high-confidence rules, supervised machine learning, and normal-only autoencoder anomaly detection. Grouped validation was employed because repeated predictor profiles could otherwise cause identical patterns to appear in both the development and validation sections. Following a comparison between class weighting and SMOTENC, a confidence gate was added after the original ungated hybrid was found to generate excessive false positives. Class-weighted Gradient Boosting was the best-performing model on the untouched official test set, reaching an accuracy of 90.68%, a balanced accuracy of 90.31%, an attack recall of 93.93%, an F1-score of 91.73%, a specificity of 86.70%, and a false-positive rate of 13.30%. The gated hybrid improved attack recall to 94.50%, but balanced accuracy dropped to 88.71% and the false-positive rate increased (worsened) from 13.30% to 17.08%. The paired-error test showed that the overall difference significantly favored the standalone model.

The main point is that hybridization should be justified by operational objectives rather than model complexity alone. Where balanced normal-versus-attack discrimination is the priority, class-weighted Gradient Boosting is preferred in this experiment. High-confidence rules can be retained for interpretable triage, while autoencoder output is better used as supporting evidence unless a cost model demonstrates that the additional false-positive burden is acceptable. Future work should use category-holdout and temporally ordered evaluations, calibrate cost-sensitive fusion thresholds, test retrained deployment specifications, validate on live traffic, and assess explainability, latency, throughput, memory use, adversarial robustness, and concept-drift adaptation. Such evaluation would help determine whether the narrower advantage observed under novel predictor profiles can be converted into a practically useful hybrid IDS.

REFERENCES

Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S. A. and Khan, M. S. (2021) Intrusion detection in the Internet of Things using supervised machine learning based on application and transport layer features using the UNSW-NB15 data set. EURASIP Journal on Wireless Communications and Networking, 2021, article 10. https://doi.org/10.1186/s13638-021-01893-8

Al-Daweri, M. S., Ariffin, K. A. Z., Abdullah, S., & Senan, M. F. E. M. (2020). An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system. Symmetry, 12(10), 1666. https://doi.org/10.3390/sym12101666

Bai, K. Z., & Fossaceca, J. M. (2025). EM-AUC: A novel algorithm for evaluating anomaly-based network intrusion detection systems. Sensors, 25(1), 78. https://doi.org/10.3390/s25010078

Bouke, M. A., & Abdullah, A. (2023). An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability. Expert Systems with Applications, 230, 120715. https://doi.org/10.1016/j.eswa.2023.120715

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

Kumar, A., Radhakrishnan, R., Sumithra, M., Kaliyaperumal, P., Balusamy, B., & Benedetto, F. (2025). A scalable hybrid autoencoder-extreme learning machine framework for adaptive intrusion detection in high-dimensional networks. Future Internet, 17(5), 221. https://doi.org/10.3390/fi17050221

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1-3), 18-31. https://doi.org/10.1080/19393555.2015.1125974

Musthafa, M. B., Huda, S., Kodera, Y., & Ali, M. A. (2024). Optimizing IoT intrusion detection using balanced class distribution, feature selection, and ensemble machine learning techniques. Sensors, 24(13), 4293. https://doi.org/10.3390/s24134293

Park, H., Shin, D., Park, C., Jang, J., & Shin, D. (2025). Unsupervised machine learning methods for anomaly detection in network packets. Electronics, 14(14), 2779. https://doi.org/10.3390/electronics14142779

Pinto, A., Herrera, L.-C., Donoso, Y., & Gutierrez, J. A. (2023). Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure. Sensors, 23(5), 2415. https://doi.org/10.3390/s23052415

Rao, Y. N., & Babu, K. S. (2023). An imbalanced generative adversarial network-based approach for network intrusion detection in an imbalanced dataset. Sensors, 23(1), 550. https://doi.org/10.3390/s23010550

Saputra, A., Ramli, K., Nugroho, A. S., Nugraha, I. G. D., & Pranggono, B. (2026). A novel hybrid IGL1 feature selection method for high-performance intrusion detection on the UNSW-NB15 dataset using multiple machine learning models. Big Data and Cognitive Computing, 10(6), Article 182. https://doi.org/10.3390/bdcc10060182

Sayegh, H. R., Dong, W., & Al-madani, A. M. (2024). Enhanced intrusion detection with LSTM-based model, feature selection, and SMOTE for imbalanced data. Applied Sciences, 14(2), 479. https://doi.org/10.3390/app14020479

Shanmugam, V., Razavi-Far, R., & Hallaji, E. (2025). Addressing class imbalance in intrusion detection: A comprehensive evaluation of machine learning approaches. Electronics, 14(1), 69. https://doi.org/10.3390/electronics14010069

Song, Y., Hyun, S., & Cheong, Y.-G. (2021). Analysis of autoencoders for network intrusion detection. Sensors, 21(13), 4294. https://doi.org/10.3390/s21134294

Sun, D., Zhang, L., Jin, K., Ling, J., & Zheng, X. (2023). An intrusion detection method based on hybrid machine learning and neural network in the industrial control field. Applied Sciences, 13(18), 10455. https://doi.org/10.3390/app131810455

Wang, C., Sun, Y., Wang, W., Liu, H., & Wang, B. (2023). Hybrid intrusion detection system based on combination of random forest and autoencoder. Symmetry, 15(3), 568. https://doi.org/10.3390/sym15030568

Zoghi Z. and Serpen G. (2024) 'Building an intrusion detection system on UNSW-NB15: Reducing the margin of error to deal with data overlap and imbalance', Concurrency and Computation: Practice and Experience, 36(25), e8242. https://doi.org/10.1002/cpe.8242