AI

MDPI

Article

Evaluating Explainable Hybrid Intrusion Detection Models Under Zero-Day Conditions

Sumayyamol Mukkil Muhammed Ismail, Mobyen Uddin Ahmed * and Shahina Begum

Department of Computer Science and Engineering, Mälardalen University, 722 20 Väasterås, Sweden; sml24002@student.mdu.se (S.M.M.I.); shahina.begum@mdu.se (S.B.)
* Correspondence: mobyen.uddin.ahmed@mdu.se

Abstract

Zero-day network attacks pose a significant threat because their unknown signatures evade traditional detection mechanisms. This research develops an AI-enhanced intrusion detection system that aims to detect such attacks while providing interpretable outputs for security analysts. Four machine-learning models are evaluated under strict zero-day conditions using two benchmark datasets. SHAP and LIME are applied to produce instance-level explanations, and a formal stability assessment is conducted to determine their reliability. Experimental results show that the hybrid model combining anomaly-based detection with deep learning achieves the highest zero-day Recall, with statistically significant advantages over individual models in detecting previously unseen attacks, while the standalone LSTM achieves the strongest overall balance between Precision and Recall. The generated explanations consistently reveal security-relevant features, and stability analysis confirms their robustness across conditions. The study demonstrates that integrating deep learning with stable explainable AI offers a practical and trustworthy solution for zero-day intrusion detection, contributing validated evidence to an area where explanation reliability is rarely examined.

Keywords: intrusion detection system; zero-day attack; explainable AI; deep learning; SHAP; LIME; explanation stability

1. Introduction

Network security remains one of the most critical challenges in modern computing, with organisations facing continuous attempts to breach systems, exfiltrate data, or disrupt essential services [1,2]. Intrusion Detection Systems (IDS) provide a key defensive layer by monitoring network traffic and flagging suspicious behaviour, yet traditional signature-based IDS fail against zero-day attacks, which exploit previously unknown vulnerabilities and therefore lack identifiable signatures [3]. Such attacks can remain undetected for long periods, often operating within networks for weeks or months before discovery [4]. As digital infrastructures expand and the number of connected devices grows, the attack surface increases accordingly, creating more opportunities for zero-day exploitation. This has driven a shift toward machine-learning-based anomaly detection, where models learn patterns of normal behaviour and identify deviations indicative of potential attacks [1,2]. Despite this promise, two major challenges persist: machine-learning IDS often provide limited explanation for their alerts [3], and most studies evaluate a single model on a single dataset, restricting generalisability across environments [4].

Check for updates

Academic Editor: Ke Zhang

Received: 22 July 2026
Revised: 22 August 2026
Accepted: 30 August 2026
Published: 3 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Research on machine learning for intrusion detection spans more than two decades. Buczak and Guven [1] surveyed naive Bayes, support vector machines, decision trees and neural networks across multiple datasets, concluding that no single method consistently outperforms others. Mishra et al. [2] further showed that supervised methods perform well on known attack types bugrade when the test distribution differs from training, directly reflecting the zero-day scenario addressed in this thesis. Foundational work by Garcia-Teodoro et al. [5] established a taxonomy of anomaly-based detection techniques and highlighted the challenge of defining reliable boundaries between normal and suspicious traffic. Khraisat et al. [6] later emphasised the lack of standardised evaluation protocols, making cross-study comparisons difficult. Feature engineering has also proven beneficial: Thakkar and Lohiya [7] demonstrated that correlation-based feature reduction improves efficiency without sacrificing detection accuracy.

Ensemble-based IDS have also gained traction. Louk and Tama [8] proposed Dual-IDS, combining bagging with gradient boosting across NSL-KDD, UNSW-NB15 and HIKARI-2021, outperforming single-classifier baselines. However, their work did not include deep learning models or evaluate under zero-day conditions. This thesis extends their approach by including gradient boosting [9] as a supervised baseline alongside deep learning models and by explicitly testing all models under zero-day conditions.

Deep learning has become increasingly prominent due to its ability to learn complex patterns from raw data. Yin et al. [10] demonstrated that recurrent neural networks capture temporal dependencies in network traffic that static classifiers miss, achieving strong results on NSL-KDD. Kim et al. [11] confirmed that LSTM gating mechanisms allow selective retention of relevant traffic patterns, making them well-suited for attacks that evolve. However, neither study evaluated their models on attack types, leaving the zero-day scenario unaddressed. This thesis incorporates LSTM-based detection [12] but evaluates it under explicit zero-day conditions.

Unsupervised anomaly detection has also shown promise. Mirsky et al. [13] introduced Kitsune, an ensemble of autoencoders that detects malicious traffic purely through reconstruction error, requiring no labelled data. Zavrak and Iskefiyeli [14] achieved similar results using a Variational Autoencoder. These studies confirm that autoencoder-based reconstruction error is a reliable anomaly signal, but neither addressed how analysts should interpret alerts. Hindy et al. [15] compared deep autoencoders with One-Class SVM for zero-day detection, reporting 89–99% accuracy on NSL-KDD but noting that autoencoders provide no explanation of which features triggered alerts. This thesis addresses this limitation by applying SHAP to autoencoder reconstruction errors and by combining autoencoder detection [16] with LSTM classification in a hybrid pipeline tested under zero-day conditions.

Zero-day detection remains one of the most challenging problems in network security. Mbona and Eloff [17] proposed Benford's law for lightweight feature selection, showing that flow-based features such as packet size and timing carry more discriminative power than static features. Soltani et al. [18] introduced an Open Set Recognition framework that adapts to new attack types over time, but required human expert intervention and was evaluated on simulated traffic. Alansary et al. [19] reviewed zero-day detection research and found that hybrid approaches consistently outperform single-method systems, while most existing work lacks explainability. Sruthi et al. [20] showed that combining SMOTE oversampling [21] with explainable models improves both detection and interpretability on imbalanced zero-day datasets. These findings support the hybrid AE-LSTM pipeline and the use of SHAP and LIME [14] in this thesis.

Explainable AI has become increasingly important in IDS research. Gaspar et al. [22] applied SHAP and LIME to an MLP trained on ADFA-LD system call data, demonstrat-

ing that both methods correctly identify influential features and that perturbing SHAP-identified features can flip false positives to true negatives. Hermosilla et al. [23] compared SHAP and LIME for XGBoost and TabNet on the UNSW-NB15 dataset, finding that SHAP provides more stable global explanations, whereas LIME is useful for instance-level decisions. Mohale and Obagbuwa [24] reviewed 20 XAI-IDS studies and identified SHAP and LIME as the most effective tools for anomaly-based IDS, while noting the lack of standard metrics for explanation quality. Khan et al. [25] highlighted that the reliability of explanations under noisy or perturbed conditions remains an open problem. This thesis contributes directly to this gap by conducting a formal stability analysis using Jaccard similarity [26] across multiple seeds and background sizes.

Several existing approaches address zero-day intrusion detection using ensemble, open-set, or anomaly-based architectures, though none combine explicit zero-day evaluation with dual explainability and formal explanation-stability testing. Kitsune [13] employs an ensemble of lightweight autoencoders (KitNET) trained on network feature vectors to detect anomalous traffic through reconstruction error, without relying on labelled attack data. Its unsupervised design allows it to flag previously unseen behaviour implicitly, but the framework offers no mechanism for interpreting why a given connection was flagged, and its evaluation has not extended to formal explanation reliability testing. Soltani et al.'s DOC++ [19] instead frames zero-day detection as an open-set recognition problem, using a clustering-based novelty labelling step to assign novel traffic to emerging categories; while effective at distinguishing known from unknown classes, this design requires a labelling phase and, like Kitsune, does not incorporate explainability. Hindy et al. [15] and Mbona and Eloff [17] both evaluate autoencoder- and one-class-classifier-based zero-day detectors across multiple datasets but similarly stop short of examining model interpretability. On the explainability side, Hermosilla et al. [23] and Mohale and Obagbuwa [24] compare SHAP and LIME for intrusion detection and review their use across the field, respectively, but neither addresses zero-day conditions or the stability of the explanations produced. Table 1 summarises this comparison across the dimensions most relevant to the present study: model architecture, dataset(s) used, zero-day handling strategy, explainability, and explanation stability testing. As the table illustrates, no existing study combines explicit multi-dataset zero-day evaluation with dual explainability and formal stability analysis, which is the gap this work addresses. Reported figures for Kitsune and DOC++ reflect results published in their original studies on their own benchmark datasets; a fully controlled experimental comparison under our zero-day protocol was not feasible within the scope of this study and is discussed as a direction for future work.

Motivated by these gaps, this study develops and evaluates an AI-enhanced intrusion detection system capable of detecting zero-day network attacks and producing reliable, interpretable explanations. Four model architectures—Gradient Boosting [9], Autoencoder [16], LSTM [12] and a Hybrid AE-LSTM pipeline—are implemented and compared across NSL-KDD [27] and UNSW-NB15 under explicit zero-day conditions where selected attack types are completely excluded from training. SHAP [3] and LIME [28] are applied to generate local explanations for individual zero-day detections, and a formal stability analysis evaluates whether these explanations can be trusted in operational environments.

The experimental work is guided by four research questions: which model architecture achieves the best zero-day detection performance across both datasets; whether SHAP and LIME can produce meaningful explanations for zero-day detections; how stable these explanations are across different conditions; and how engineered features influence Autoencoder zero-day detection performance. To answer these questions, seven goals were defined, including implementing four models, evaluating them under zero-day conditions, generating SHAP and LIME explanations, producing Autoencoder reconstruction heatmaps,

conducting a SHAP stability analysis, performing an engineered-feature ablation study, and validating statistical significance using McNemar's test [29], bootstrap confidence intervals [30], and Cohen's h [31].

Table 1. Theoretical comparison of the proposed Hybrid AE-LSTM pipeline against related zero-day and explainable IDS approaches.

Study	Model/Approach	Dataset(s)	Zero-Day Handling	Explainability	Stability Tested
Kitsune (Mirsky et al. [15])	Ensemble autoencoders (KitNET)	Custom IoT/network capture	Implicit, unsupervised anomaly detection	None	None
DOC++ (Soliman et al. [19])	Open-set recognition, clustering-based identity labeling	CIC-IDS2017, CSE-CIC-IDS2018	Explicit open-set classification	None	None
Hindy et al. [17]	Autoencoder vs. One-Class SVM	NSL-KDD, CICIDS2017	Explicit zero-day	None	None
Mbuva & Eloff [18]	Benford's-law feature selection + semi-supervised learning	UNSW-NB15 + 3 others	Explicit zero-day	None	None
Hermonilla et al. [23]	SHAP vs. LIME comparison	UNSW-NB15	Not addressed	SHAP, LIME	Not addressed
Mohme & Oughazawa [24]	Systematic review, XAI-IDS studies	Multiple (review)	Varies by study	SHAP, LIME (reviewed)	None
Proposed study	Hybrid AE-LSTM cascade	NSL-KDD, UNSW-NB15	Explicit, multi-dataset zero-day	SHAP, LIME	Yes (Jaccard + RBO)

The key finding of this research is that the Hybrid AE-LSTM model achieves the highest zero-day detection rate across both datasets, significantly outperforming standalone models in terms of zero-day Recall. Among individual models, the LSTM demonstrates strong balanced performance, making the Hybrid model's improvement particularly notable. Statistical tests confirm the significance of these improvements. The Autoencoder contributes complementary detection capability, identifying zero-day attacks through reconstruction error even when the LSTM fails to detect them. SHAP and LIME consistently identify security-relevant features for all examined zero-day instances, and stability analysis confirms that core features remain consistent across all tested conditions.

Together, the literature reveals four consistent gaps: most studies evaluate a single model on a single dataset; zero-day detection is rarely tested properly; explanation reliability is rarely validated; and engineered feature contributions remain understudied. This thesis addresses all four gaps through a multi-model, multi-dataset evaluation under explicit zero-day conditions, stability-tested SHAP and LIME explanations, and a systematic engineered-feature ablation study. These contributions position the work within the broader progression of IDS research and respond directly to limitations identified across the literature.

2. Materials and Methods

This study employs a quantitative experimental design in which multiple machine learning architectures are implemented, trained and evaluated under uniform conditions to measure their effectiveness in detecting both known and zero-day network intrusions. Four models were selected to represent distinct methodological families: Gradient Boosting as a supervised tree-based classifier, a Denoising Autoencoder [32] as an unsupervised reconstruction-based anomaly detector, an LSTM [33] network as a supervised temporal sequence model, and a sequential Hybrid AE-LSTM pipeline that integrates unsupervised anomaly detection with supervised sequence learning. All experiments were executed with full reproducibility by fixing the random seed at 42 across NumPy 2.3.5, Python's

random module and TensorFlow. The implementation environment consisted of Python 3.13, TensorFlow 2.20 with Keras 3.0, scikit-learn 1.8, imbalanced-learn 0.14, SHAP 0.50 and LIME 0.2, running on CPU-only hardware, which was sufficient for the dataset sizes and model complexity.

The evaluation framework comprises five components. First, all four architectures are trained and tested on NSL-KDD and UNSW-NB15, two widely used intrusion detection benchmarks. Second, a hybrid sequential pipeline is constructed to combine anomaly-based reconstruction with temporal classification. Third, explainability is incorporated through SHAP, LIME and Autoencoder reconstruction heatmaps to generate feature-level insights for individual detections. Fourth, model performance is assessed using ROC and Precision–Recall curves, along with per-family detection rates on both clean and noise-corrupted test sets, where injected noise simulates noise-based perturbation. Finally, statistical significance of performance differences—particularly the hybrid model's improvements—is validated using McNemar's test and bootstrap confidence intervals.

2.1. Datasets

As mentioned in the introduction, NSL-KDD [27] and UNSW-NB15 [34] were selected as the evaluation datasets. Using two established benchmarks is essential for cross-dataset generalisation analysis, ensuring that the proposed detection approach is not overfitted to the statistical properties of a single dataset. Both datasets are publicly available, widely used in intrusion-detection research, and contain no personally identifiable information. Their core characteristics are summarised in Table 2, which highlights differences in size, feature composition, attack taxonomy and class imbalance.

Table 2. Comparison of NSL-KDD and UNSW-NB15 dataset characteristics.

Characteristic	NSL-KDD	UNSW-NB15
Training instances	125,973	82,332
Testing instances	22,544	175,341
Raw features	41	42
Native attack categories	4	9
Mapped attack families	4	4
Zero-day simulation	17 attack types in KDDTest+	3 categories withheld from training
Class imbalance	High (R2L and U2R rare)	High (Worms and Shellcode rare)

For NSL-KDD, the KDDTrain+ split is used for training and KDDTest+ for zero-day evaluation. The test set contains 17 attack types not present in training, enabling a natural zero-day scenario without manual manipulation. Attacks are grouped into four canonical families—DoS, Probe, R2L and U2R—to support consistent evaluation across datasets.

For UNSW-NB15, three attack categories—Fuzzers, Worms, and Shellcode—are withheld entirely from training and introduced only at test time to simulate zero-day conditions. These categories span different behavioural families and include both high-volume and stealth-oriented threats. The nine native categories are mapped to the same four-family structure used for NSL-KDD to enable direct comparison. The complete mapping and justification are shown in Table 3.

Zero-day evaluation requires ensuring that attack types present at test time are absent from training. The two datasets support this requirement through different mechanisms. For NSL-KDD, zero-day simulation is inherent: KDDTest+ contains 17 attack types not found in KDDTrain+, allowing the model to be trained on known attacks and evaluated on unseen variants, e.g., zero-day. This setup reflects a realistic deployment scenario in which an IDS encounters novel attack behaviours after initial training. For UNSW-NB15, zero-day

conditions are created by withholding Fuzzers, Worms, and Shellcode from the training set. These categories were selected because they represent distinct mapped families and cover a broad behavioural spectrum—from malformed-input probing to privilege-escalation code injection. To evaluate noise robustness, all models were additionally tested on noise-corrupted zero-day samples. A 5% Gaussian noise perturbation was applied to feature values, a commonly used magnitude in noise robustness studies [34]. This level is sufficient to simulate an attacker subtly manipulating traffic to evade detection while preserving the underlying attack characteristics. After noise injection, all values were clipped to the [0, 1] range to maintain compatibility with the Autoencoder, which was trained on normalised inputs. For Gradient Boosting, which operates on raw encoded features and does not require scaling, noise was applied directly without clipping.

Table 3. Mapping of UNSW-NB15 native attack categories to four-family structure.

Native Category	Mapped Family	Reason
DoS	DoS	Direct match
Backdoor	DoS	Establishes persistent access with similar traffic pattern
Reconnaissance	Probe	Network scanning behaviour
Analysis	Probe	Traffic analysis and scanning
Fuzzers	R2L	Remote probing with malformed inputs
Generic	R2L	Generic attack patterns targeting remote services
Exploits	R2L	Remote exploitation attempts
Shellcode	U2R	Code injection for privilege escalation
Worms	U2R	Self-propagating attacks targeting local privilege

2.2. Data Preprocessing and Feature Engineering

The preprocessing workflow was standardised across NSL-KDD [27] and UNSW-NB15 [34], with only dataset-specific adaptations where structurally required. The complete pipeline is shown in Figure 1, which outlines all preprocessing stages applied prior to model training.

Both datasets contain non-predictive administrative fields that were removed at the start of processing. For NSL-KDD, the level column was discarded, while UNSW-NB15 required removal of the id column. NSL-KDD includes three categorical protocol-level features—protocol_type, service and flag—and UNSW-NB15 contains their equivalents—proto, service and state. All categorical fields were encoded using scikit-learn's LabelEncoder (version 1.8). To prevent unseen-category failures during zero-day testing, the encoder was fitted jointly on the combined training and test sets, ensuring that every category present in either split received a valid integer mapping. This procedure does not introduce information leakage because LabelEncoder learns only category identities, not relationships between categories and labels.

The NSL-KDD dataset was loaded from two headerless files: KDDTrain+ for training and KDDTest+ for zero-day evaluation. KDDTrain+ contains 125,973 connection records, and KDDTest+ contains 22,544, each with 41 features. Feature names were manually assigned according to the official NSL-KDD specification. After encoding the three categorical features, attack types were mapped to both binary labels (0 = Normal, 1 = Attack) and four canonical families (DoS, Probe, R2L, U2R) to support per-family detection analysis.

The raw data was loaded from two CSV files, a training set of 82,332 records and a test set of 175,341 records, each with 45 columns. The id column was dropped, and the three categorical columns, proto, service and state, were encoded using LabelEncoder fitted on both files combined, following the same approach used for NSL-KDD. The main difference from NSL-KDD is the attack category structure. UNSW-NB15 has nine native attack categories which were mapped to four family labels: DoS and Backdoor to DoS, Reconnaissance and Analysis to Probe, Fuzzers, Generic and Exploits to R2L, and Shellcode

and Worms to U2R. Both the original category names and the mapped family labels were saved in the output files so each model notebook could work with whichever grouping it needed. The cleaned data was saved unscaled as two CSV files for all subsequent notebooks to load.

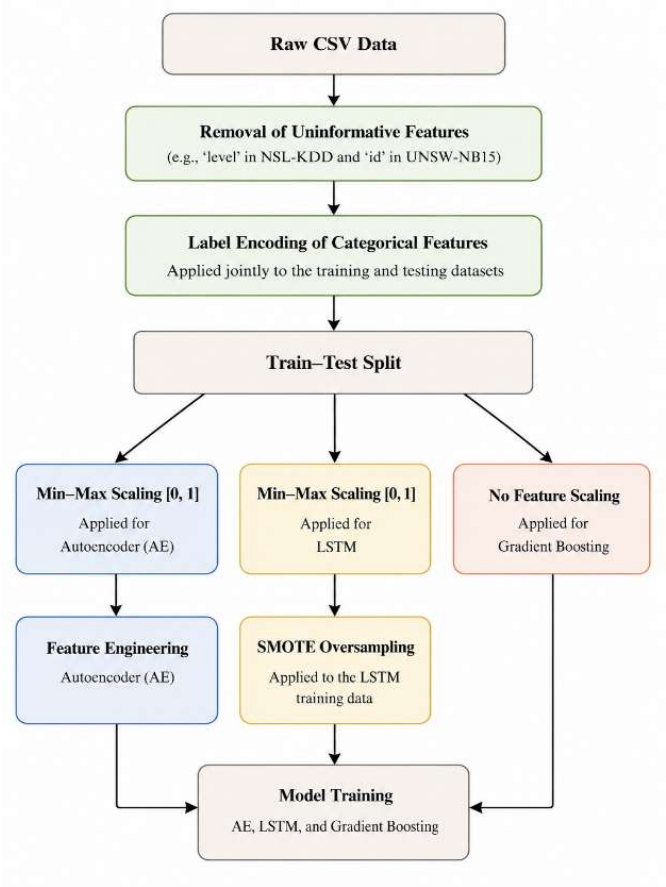

Figure 1. Data preprocessing pipeline applied to NSL-KDD and UNSW-NB15 datasets.

Feature scaling was applied selectively based on model requirements. The Denoising Autoencoder and LSTM models require inputs normalised to [0, 1], so a MinMaxScaler was fitted only on the training split and applied to validation and test sets. This ensures compatibility with the Autoencoder's sigmoid output layer and stabilises reconstruction-error magnitudes. In contrast, Gradient Boosting—a tree-based model—was trained on unscaled encoded features, as decision-tree splits depend on rank ordering rather than absolute magnitudes.

To address class imbalance in NSL-KDD and UNSW-NB15, SMOTE oversampling was applied only to the LSTM training data, and only after the train/validation split, following best practice for synthetic minority generation. This prevents synthetic samples from leaking into validation or test sets.

Feature engineering was applied selectively and only to the Autoencoder models on both datasets, reflecting the methodological distinction between supervised and unsupervised detection approaches. Gradient Boosting and LSTM were trained on labelled attack traffic and evaluated on zero-day test sets; their ability to detect these attacks depends on how closely zero-day behaviours resemble the attack families present during training. In contrast, the Autoencoder is trained exclusively on normal traffic and learns a compact representation of benign behaviour. Any deviation from this learned manifold—regardless of similarity to known attacks—produces elevated reconstruction error. For this reason, en-

gineered features were introduced specifically for the Autoencoder to enhance its modelling of normal behaviour and improve sensitivity to anomalous deviations. Table 4 summarises the engineered features added to NSL-KDD and UNSW-NB15.

Table 4. Summary of engineered features for NSL-KDD and UNSW-NB15 Autoencoders.

Dataset	Feature	Formula	Purpose
NSL-KDD	duration_src_ratio	src_bytes/(duration + 1)	Captures data transfer intensity relative to connection time
NSL-KDD	service_focus_ratio	srv_count/(count + 1)	Measures how focused connections are on one service
UNSW-NB15	bytes_per_second	sbytes/(dur + 1×10^{-5})	Measures data transfer rate per second
UNSW-NB15	ttl_ratio	sttl/(dttl + 1×10^{-5})	Measures the difference between source and destination TTL values.

Two engineered features were added to the NSL-KDD Autoencoder input space:

- duration_src_ratio: Defined as 'src_bytes/(duration + 1)', this feature quantifies byte-transfer intensity relative to connection duration. It is particularly informative for detecting DoS-like behaviours, where large volumes of data are transmitted in short bursts.
- service_focus_ratio: Computed as srv_count/(count + 1), this feature measures whether a host is interacting heavily with a single service or distributing requests across many services. Concentrated service access is characteristic of Probe and scanning attacks.

These additions increase the NSL-KDD Autoencoder input dimensionality from 41 to 43 features.

Similarly, two engineered features were added to UNSW-NB15:

- bytes_per_second: Defined as 'sbytes/(dur + 1×10^{-5})', this feature captures raw throughput. It proved effective for identifying Fuzzer attacks, which generate abnormally high byte rates.
- ttl_ratio: Computed as 'sttl/(dttl + 1×10^{-5})', this feature captures TTL asymmetry between source and destination. TTL irregularities are common in Shellcode and Worm behaviours.

A third candidate, bytes_per_packet, was removed after showing a perfect Pearson correlation (1.00) with the native feature smean, indicating complete redundancy. The final engineered features increase the UNSW-NB15 Autoencoder input dimensionality from 42 to 44 features.

To ensure that engineered features contributed unique predictive information, statistical validation was performed across both datasets. A Pearson correlation threshold of 0.85 was used to identify and remove redundant features. This threshold is widely used in the feature-selection literature because it eliminates clearly collinear features while retaining those that provide distinct behavioural signals.

- For NSL-KDD, both engineered features—duration_src_ratio and service_focus_ratio—passed the correlation threshold and showed no significant correlation with any of the 41 native features.
- For UNSW-NB15, bytes_per_second and ttl_ratio also passed the threshold and showed no significant correlation with each other or with any of the 42 native features.
- bytes_per_packet was removed due to perfect correlation with smean, confirming redundancy.

Multicollinearity checks verified that the engineered features do not distort the Autoencoder's latent-space learning process. Their inclusion enhances the model's ability to characterise normal behaviour and improves sensitivity to subtle deviations indicative of zero-day attacks.

2.3. Model Architectures and Training

Four architectures were implemented and evaluated: Gradient Boosting, a Denoising Autoencoder, an LSTM and a Hybrid AE-LSTM sequential pipeline. Running all four on both datasets produces eight experimental configurations. The models represent meaningfully different approaches to intrusion detection: one supervised tree-based method, one unsupervised neural reconstruction method, one supervised sequential method and one composite method. Comparing them reveals which approach is most effective for which type of threat. The following subsections describe each architecture and justify its selection.

2.3.1. Gradient Boosting (GB)

GB was selected as the supervised baseline because it consistently performs well on structured tabular intrusion-detection data and provides intrinsic feature-importance scores that serve as a reference point for the SHAP analysis. Hyperparameter optimisation was performed using GridSearchCV with three-fold cross-validation on both datasets. The search space was intentionally small to allow exhaustive evaluation of all configurations. The ranges and best values identified for each dataset are shown in Table 5.

Table 5. Gradient Boosting hyperparameter search ranges and best values.

Hyperparameter	Search Values	NSL-KDD Best	UNSW-NB15 Best
n_estimators	100, 200	200	200
learning_rate	0.05, 0.1, 0.2	0.2	0.1
max_depth	3, 5, 7	7	5
Cross-validation	3-fold	3-fold	3-fold

The same hyperparameter ranges were applied to both datasets. For NSL-KDD, the optimal configuration used n_estimators = 200, learning_rate = 0.2, and max_depth = 7, reflecting the larger training set (125,973 instances) and its capacity to support deeper trees without overfitting. For UNSW-NB15, the best configuration used learning_rate = 0.1 and max_depth = 5, consistent with its smaller training set (82,332 instances) and the need for a more conservative model to avoid variance inflation.

Once the optimal configuration was identified for each dataset, the trained model was serialised to disk as a .pkl file to prevent unnecessary retraining during subsequent experimental runs. For zero-day evaluation, a separate Gradient Boosting model was trained using the same hyperparameters but with all zero-day attack categories removed from the training data. On NSL-KDD, this meant training on KDDTrain+ while excluding the 17 attack types present only in KDDTest+. On UNSW-NB15, the zero-day model was trained with Fuzzers, Worms, and Shellcode withheld entirely. This ensures that the supervised baseline has no prior exposure to the zero-day attack families during evaluation, allowing a clean measurement of generalisation performance.

2.3.2. Autoencoder

It was selected as the unsupervised baseline because it can detect zero-day attacks without requiring labelled malicious traffic during training. The model is trained exclusively on normal connections, learning a compressed representation of benign network behaviour. During evaluation, any input that deviates significantly from this learned manifold produces elevated reconstruction error and is flagged as anomalous. This property

makes the Autoencoder inherently suitable for zero-day detection, as it does not rely on prior exposure to specific attack families. A denoising Autoencoder variant was used to improve robustness. Gaussian noise with a standard deviation of 0.05 was injected into the inputs during training, matching the 5% perturbation noise level used in the robustness experiments described in Section 3.3. Training the model under the same perturbation regime it faces during evaluation forces the encoder to learn stable latent representations rather than memorising training samples, improving generalisation to unseen traffic patterns. The architecture consists of a three-layer encoder and a symmetric three-layer decoder. The encoder compresses the input feature space through layers of size input $\rightarrow$ 32 $\rightarrow$ 16 $\rightarrow$ bottleneck, while the decoder reconstructs the input via bottleneck $\rightarrow$ 16 $\rightarrow$ 32 $\rightarrow$ input. Hidden layers use ReLU activations, and the output layer uses sigmoid to ensure reconstruction values remain within [0, 1], consistent with min–max-scaled inputs. Four bottleneck sizes—6, 8, 10, and 12 latent dimensions—were evaluated. Smaller bottlenecks increase sensitivity to anomalous deviations but risk losing fine-grained normal structure, while larger bottlenecks preserve more detail at the cost of reduced anomaly sensitivity. The optimal bottleneck size for each dataset was selected using F1-score on a dedicated validation split.

Training was performed using the Adam optimiser with MSE loss, and early stopping (patience = 3 epochs) was applied to prevent overfitting. The anomaly threshold was defined as the 80th percentile of reconstruction errors computed on normal training data. This threshold prioritises Recall over Precision, reflecting operational IDS requirements where failing to detect an attack is more costly than generating a false alarm. Applying the threshold at the 80th percentile ensures that the top 20% most unusual connections are flagged as suspicious. The same thresholding strategy was applied consistently across all evaluation conditions—including perturbation noise tests—to ensure comparability between clean and perturbed results. The resulting nominal false-alarm rate requires explicit justification, and sensitivity analysis is mentioned in Appendix A.2 and Table A8.

For NSL-KDD, the Autoencoder was trained solely on normal traffic from KDDTrain+. After evaluating all bottleneck sizes, the best model was retrained on the full normal training set. The resulting anomaly threshold was 0.0271, corresponding to the 80th percentile of reconstruction errors on normal training connections. For UNSW-NB15, the Autoencoder followed the same training protocol but included two engineered features—bytes_per_second and ttl_ratio—bringing the input dimensionality from 42 to 44 features. These features were added to strengthen the model's representation of normal behaviour and improve sensitivity to deviations characteristic of zero-day attacks. The anomaly threshold for UNSW-NB15, computed using the same percentile-based method, was 0.047759.

2.3.3. Long Short-Term Memory (LSTM)

Its internal gating mechanism enables the extraction of richer temporal features, making it well-suited to the complex and highly imbalanced traffic patterns present in both datasets. Each connection record was treated as a single timestep, as indicators of malicious behaviour are typically contained within an individual flow. Prior work by Kim et al. [11] and Yin et al. [10] supports the effectiveness of LSTMs for per-connection intrusion detection.

Both datasets exhibit substantial class imbalance. In NSL-KDD, the R2L and U2R families contain far fewer samples than DoS or Normal traffic, while in UNSW-NB15, Worms and Shellcode are similarly underrepresented. Models trained on such distributions tend to over-predict majority classes, producing deceptively high accuracy while failing to detect rare attacks. To mitigate this, SMOTE oversampling was applied to the training portion of

the data after the 80/20 train–validation split. Using the default setting of k = 5, SMOTE generates synthetic minority-class samples by interpolating between each instance and its nearest neighbours. Applying SMOTE only to the training set prevents synthetic samples from leaking into the validation set, thereby avoiding inflated performance estimates.

SMOTE was chosen because it is simple, reproducible, and widely validated in intrusion detection research. The improved detection of U2R and R2L attacks compared with the Gradient Boosting baseline (which did not use SMOTE) indicates that the technique was effective in this study. Three LSTM configurations were evaluated with recurrent layer sizes of 32, 64, and 128 units. All models used a Dropout rate of 0.2 to reduce overfitting and a sigmoid output activation to produce binary probability scores. The best configuration for each dataset was selected using zero-day Recall, reflecting the operational priority of detecting previously unseen cases. The decision threshold was tuned by selecting the point on the Precision–Recall curve that maximised F1-score, rather than relying on the default 0.5 threshold, which is unsuitable for imbalanced datasets.

2.3.4. Hybrid AE-LSTM Pipeline

The Hybrid model is implemented as a two-stage sequential pipeline in which the Autoencoder (AE) operates as Stage 1, and the LSTM operates as Stage 2. Figure 2 presents the full AE-LSTM architecture. The motivation for this design is that the two models offer complementary detection capabilities. The Autoencoder is trained exclusively on normal traffic and flags any connection whose behaviour deviates substantially from the learned normal profile, enabling it to detect zero-day attacks without requiring labelled attack data. The LSTM, in contrast, is a supervised model that learns known attack patterns and performs strongly on zero-day samples that resemble previously observed attacks. However, it still misses novel attacks that fall outside its learned decision boundary.

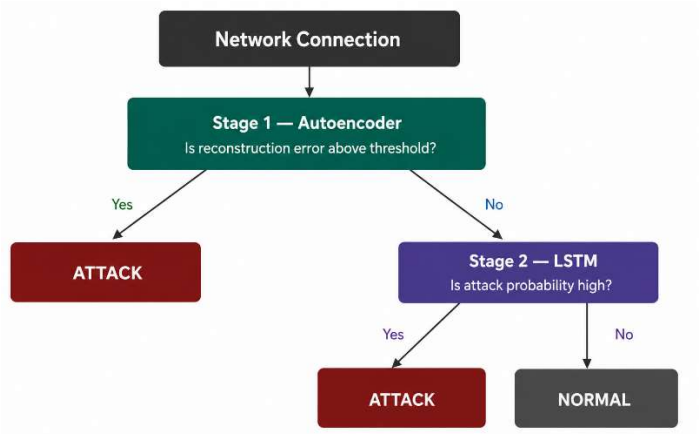

Figure 2. Hybrid AE-LSTM pipeline architecture.

By combining both models using union logic, the Hybrid pipeline captures attacks that either model would miss alone. Any connection flagged by the Autoencoder in Stage 1 is immediately classified as an attack. Connections that pass Stage 1 are forwarded to the LSTM in Stage 2 for a second classification. The final decision rule is as follows:

Hybrid prediction = AE prediction OR (NOT AE prediction AND LSTM prediction)

This ordering is deliberate: the Autoencoder processes all incoming traffic first, reducing the computational load on the LSTM while preserving high Recall. Only AE-cleared connections require LSTM evaluation.

To compute a unified confidence score, the AE reconstruction error and the LSTM attack probability were placed on a common scale. The AE error was normalised to the range [0, 1] using min–max scaling derived from training-set error values. The final Hybrid confidence score for each connection was taken as the maximum of the normalised AE score and the LSTM probability, ensuring that the model expressing stronger attack confidence determines the final score. This combined score was used to generate ROC and Precision–Recall curves for the Hybrid model. From a decision-boundary perspective, the LSTM boundary reflects the known attack behaviours present in its training data. Zero-day attacks like these patterns fall inside the boundary and are detected, while novel attacks fall outside and are missed. The Autoencoder's boundary is defined by the 80th-percentile reconstruction-error threshold: any connection that deviates sufficiently from normal traffic crosses this boundary regardless of whether it resembles a known attack. Applying union logic expands the overall detection region. A connection is classified as an attack if it crosses either boundary. As a result, zero-day attacks that fall outside the LSTM boundary but violate normal-traffic behaviour are captured by the Autoencoder. These correspond to the additional 309 attacks detected on NSL-KDD and 1331 on UNSW-NB15 compared to the standalone LSTM. This broader detection region increases Recall by capturing more true attacks, while slightly reducing Precision due to additional AE-flagged false positives. In security contexts, this trade-off is intentional: prioritising the detection of real attacks is preferable to minimising false alarms. In practice, the Hybrid notebook loaded the saved prediction files produced by the AE and LSTM notebooks and combined them using union logic. The joint confidence score—defined as the maximum of the normalised AE score and the LSTM probability—was used for ROC and PR analysis. All models saved their predictions, ground-truth labels, and confidence scores, which were then consumed by the comparison notebooks for cross-model evaluation.

2.4. Explainability Methods

Explainability addresses the critical limitation of black-box IDS models that cannot justify their decisions to security analysts. Two complementary XAI methods are applied: SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations), alongside the Autoencoder's built-in reconstruction analysis.

2.4.1. SHAP Explanations

SHAP was chosen as the main interpretability tool because it can show how much each feature contributes to a model's decision. For the Autoencoder and Gradient Boosting models, SHAP values were computed using a Random Forest proxy with TreeExplainer. The LSTM required KernelExplainer, since its architecture is not compatible with the faster tree-based SHAP methods. KernelExplainer needs a background dataset that represents normal behaviour. Fifty samples from the zero-day test set were used, striking a balance between computational cost and explanation quality. Because KernelExplainer sends data in 2D form, but the LSTM expects 3D sequences, a wrapper function reshaped each instance automatically.

Two types of explanations were produced. Global explanations summarised the most influential features across 100 random test connections. Local explanations focused on individual zero-day attacks, presented as reports ranking features by SHAP contribution, showing their original values, and explaining in plain language what each value suggests about the connection. These local cases included three high-confidence KDDTest+ attacks (mscan, saint, httptunnel) and three hidden-category samples (Fuzzer, Worms, Shellcode). KernelExplainer was applied only to the LSTM because SHAP is designed to explain probability-based decisions. The Autoencoder does not output a probability—it outputs

reconstruction error—so explaining that directly would be less intuitive for analysts. Instead, Autoencoder explanations were provided through reconstruction heatmaps, which visually highlight the features that deviated most from normal behaviour. All explanation outputs were saved together with predictions, labels, and confidence scores for use in the comparison notebooks.

2.4.2. LIME Explanations

To validate the SHAP findings, LIME was run on the same three zero-day attack instances. LIME approaches explanation differently: instead of calculating exact feature attributions, it creates many slightly altered versions of the original connection, checks how the model's prediction changes, and then fits a simple linear model to approximate how the LSTM behaves in that local neighbourhood. The features with the largest coefficients are treated as the most influential.

The LimeTabularExplainer was configured with its standard exponential kernel and a kernel width of $0.75 \times \sqrt{}$(number of features), which controls how much weight is given to perturbed samples close to the original instance. The explainer was built using the first 1000 rows of the zero-day test set to capture typical feature ranges. For each attack instance, 1000 perturbed samples were generated to obtain a reliable local approximation. Running LIME on the same cases as SHAP allows direct comparison. When both methods identify the same features as important, it increases confidence that the explanation reflects the model's true reasoning rather than quirks of a single interpretability technique.

2.4.3. Autoencoder Reconstruction Analysis

The Autoencoder provides built-in explainability without requiring an external XAI framework. Because it reconstructs each input feature individually, the difference between the original value and the reconstructed value directly reveals which features the model considered unusual. This per-feature reconstruction error is naturally interpretable, as both the input and output are normalised to the [0, 1] range using MinMaxScaler. With all features on a common bounded scale, each reconstruction error represents a normalised deviation from expected behaviour, making features comparable regardless of their original ranges. A large error on a particular feature indicates that the Autoencoder expected a substantially different value based on its understanding of normal traffic, marking that feature as a key indicator of anomalous behaviour. To support detailed analysis, four reconstruction-based visualisations were produced for each zero-day case study. These included the following:

- A three-panel heatmap showing the original values, the Autoencoder's expected values, and the per-feature reconstruction errors;
- A sorted heatmap ranking features from most to least anomalous;
- A difference heatmap distinguishing features that were inflated versus suppressed relative to normal expectations;
- A bar chart highlighting the ten most anomalous features.

These visualisations were generated for multiple case studies, including the mscan attack, a Fuzzer instance, and other zero-day samples. For each case, an additional comparison plot was produced showing how the reconstruction errors of a typical normal connection differ from those of the attack instance—using a linear scale for mscan and a log scale for the Fuzzer case to emphasise magnitude differences across the most anomalous features. Together, these views allow a security analyst not only to confirm that an attack was detected but also to understand precisely which network behaviours triggered the alert and how those behaviours deviated from normal traffic.

2.4.4. SHAP Stability Testing

KernelExplainer does not compute SHAP values exactly—it estimates them through sampling—so the explanations can shift slightly depending on the random seed or how many background samples are used. To see how stable the explanations really are, a small stability study was run on both datasets. Two checks were made. First, SHAP was run with three different random seeds (42, 123, 999). Then it was run again with three background sizes (25, 50, 100). For each run, the top three features were extracted and compared using Jaccard similarity, which measures how much two sets overlap:

$$Jaccard\ (A, B) = \frac{|A \cap B|}{|A \cup B|}$$

A score of 1.0 means the same three features appear in both lists; 0.0 means no overlap. If the same core features keep appearing across seeds and background sizes, the explanations can be considered stable. Jaccard is simple and intuitive, but it has drawbacks. It treats all ranks the same, even though a shift at the top of the list matters more than a shift at the bottom. It also ignores the actual SHAP values—two runs might list the same features but assign very different importance scores. More sophisticated measures like Rank-Biased Overlap could capture these nuances. Investigating such metrics is planned for future work.

2.5. Evaluation Metrics and Statistical Validation

Model performance was assessed using four standard classification metrics—Accuracy, Precision, Recall, and F1-Score. Given the strong class imbalance characteristic of intrusion-detection datasets, Accuracy alone is not a reliable indicator of performance; a classifier that labels all connections as benign can still achieve deceptively high Accuracy. For this reason, Recall, Precision, and F1-Score were emphasised. Recall quantifies the proportion of attacks successfully detected, Precision reflects the proportion of flagged connections that are genuine attacks, and F1-Score provides a balanced harmonic mean of the two. All metrics were computed on the zero-day test sets for both clean and noise-corrupted data. To complement these metrics, ROC and Precision–Recall (PR) curves were generated for all models on both datasets. PR curves are particularly informative for imbalanced intrusion-detection settings, where ROC curves may appear overly optimistic. Each model used its natural output as the detection score: the Autoencoder used its MSE reconstruction error, the LSTM used its sigmoid probability, and the Hybrid used the maximum of the two normalised scores. ROC curves were produced for each attack family, while PR curves were generated for the specific zero-day attack categories. AUC and Average Precision values were reported to summarise curve performance.

To determine whether the Hybrid model's improvement over the standalone LSTM reflects a genuine performance gain rather than random variation, two statistical procedures were applied to both datasets. McNemar's test was used to compare the paired predictions of the LSTM and Hybrid models on the same test connections. A contingency table was constructed containing four outcomes: both models correct, only the LSTM correct, only the Hybrid correct, and both wrong. The test evaluates the discordant cases—instances where the models disagree—to determine whether one model is consistently more accurate. The statsmodels implementation with chi-square approximation and Yates' continuity correction was applied. A p-value below 0.05 was interpreted as evidence of a statistically significant difference. This procedure was applied identically to both NSL-KDD and UNSW-NB15. Because statistical significance does not necessarily imply practical significance, Cohen's h was computed to quantify the magnitude of the Hybrid model's Recall improvement. Cohen's h adjusts for the difficulty of improving

performance near the extremes of the probability scale. For NSL-KDD, Cohen's h = 0.110, corresponding to a 2.40-percentage-point Recall improvement. For UNSW-NB15, Cohen's h = 0.074, corresponding to a 1.11-percentage-point improvement. Both values represent small effect sizes by conventional thresholds, yet in operational security contexts even small, consistent increases in detected attacks are meaningful. To further validate the robustness of the results, 95% bootstrap confidence intervals were computed for Precision, Recall, and F1-Score for both models. The test sets were resampled 1000 times, with metrics recalculated for each resample. The lower and upper 2.5% quantiles formed the confidence bounds. Prior work shows that 1000 resamples provide stable interval estimates without unnecessary computational cost. Seed 42 was fixed throughout for reproducibility. Non-overlapping confidence intervals between the LSTM and Hybrid models indicate a genuine performance difference.

To illustrate the practical impact of the Hybrid model, three zero-day case studies were analysed for each dataset, focusing on attacks detected by the Autoencoder but missed by the LSTM. For NSL-KDD, the selected cases were saint, mscan, and httptunnel. For UNSW-NB15, the cases were drawn from the hidden categories: Fuzzers, Worms, and Shellcode. For each instance, the Autoencoder anomaly score, the LSTM confidence, and the number of times the AE score exceeded the detection threshold (0.0271 for NSL-KDD and 0.047759 for UNSW-NB15) were recorded and visualised as bar charts. Notably, in UNSW-NB15, 974 of the 977 LSTM misses belonged to the Fuzzers category, reflecting the extremely high data volumes characteristic of Fuzzer attacks—behaviour that the Autoencoder readily identifies through reconstruction error but which the LSTM never encountered during training.

3. Results

The experimental results from evaluating four intrusion detection models across two benchmark datasets under zero-day conditions. The models evaluated are Gradient Boosting, Autoencoder, LSTM and the Hybrid AE-LSTM, tested on NSL-KDD and UNSW-NB15. Results are organised and cover overall classification models, per-family detection rates, robustness under noise perturbation, ROC and PR curve analysis, statistical validation, XAI explanations, case studies and ablation study findings.

3.1. Overall Classification Models

Table 6 shows the overall classification metrics for all four models evaluated on the NSL-KDD and UNSW-NB15 test sets. Across both datasets, the LSTM was the strongest standalone model, achieving 89.70% Accuracy and 91.20% F1 on NSL-KDD and 92.79% Accuracy and 94.83% F1 on UNSW-NB15. The Hybrid model consistently delivered the highest Recall—96.10% on NSL-KDD and 98.21% on UNSW-NB15—representing gains of 2.4 and 1.1 percentage points over the LSTM. This improvement is operationally meaningful in intrusion detection, where maximising attack capture outweighs modest Precision reductions. Gradient Boosting performed competitively, particularly on UNSW-NB15 with 99.49% Precision, while the Autoencoder showed the weakest overall performance, reflecting the difficulty of unsupervised zero-day detection in more complex traffic environments.

To verify that this robustness generalises beyond the specific noise level used during Autoencoder training, the same evaluation was repeated at 3, 7, and 10 percent Gaussian noise, in addition to the original 5 percent level (see Table A7 in Appendix A). Accuracy remained stable across all tested levels, ranging from 74.10 to 74.55 percent, closely tracking the clean-data baseline of 74.26 percent. This indicates that the denoising Autoencoder's robustness generalises beyond the exact perturbation magnitude seen during training.

Table 6. Overall classification metrics on NSL-KDD and UNSW-NB15 zero-day test sets.

DataSets	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
NSL-KDD	Gradient Boosting	83.69	89.12	81.26	85.01
	Autoencoder	83.96	88.98	81.98	85.33
	LSTM	89.70	88.83	93.70	91.20
	Hybrid AE-LSTM	88.33	85.26	96.10	90.36
UNSW-NB15	Gradient Boosting	89.37	99.49	84.81	91.57
	Autoencoder	74.26	90.45	69.51	78.61
	LSTM	92.79	92.66	97.10	94.83
	Hybrid AE-LSTM	90.03	88.42	98.21	93.06

3.2. Detection by Attack Family

Table 7 shows detection rates across the four mapped attack families on both the NSL-KDD and UNSW-NB15 datasets. Detection performance was strong across DoS attacks for all models, with most exceeding 99% on NSL-KDD and similarly high results on UNSW-NB15. Probe and U2R attacks were also handled effectively, with the Hybrid model achieving perfect detection on both families in NSL-KDD and maintaining the strongest overall performance across all four families in UNSW-NB15. R2L remained the most challenging category: Gradient Boosting reached only 20.03% on NSL-KDD, while the Hybrid achieved 83.47%—and 97.91% on UNSW-NB15, marking a substantial improvement over the standalone models. The Autoencoder performed particularly well on Probe attacks (99.92%) due to their distinctive deviation from normal traffic, whereas its weaker results on UNSW-NB15 reflect the difficulty of detecting low-volume, highly variable attacks using unsupervised reconstruction alone. Furthermore, a breakdown across all nine native UNSW-NB15 categories is presented in Appendix A, Table A1.

Table 7. Detection rates by attack family on NSL-KDD and UNSW-NB15.

DataSets	Family	Gradient Boosting	Autoencoder	LSTM	Hybrid
NSL-KDD	DoS	99.58	91.22	99.56	99.69
	Probe	98.35	99.92	94.47	100.00
	R2L	20.03	43.40	77.89	83.47
	U2R	59.70	65.67	94.03	100.00
UNSW-NB15	DoS	97.92	82.98	99.71	99.86
	Probe	95.13	58.25	96.95	98.56
	R2L	81.52	69.26	96.70	97.91
	U2R	75.46	50.04	98.18	98.42

3.3. Robustness Under Noise Perturbation

All four models were tested on a noise-corrupted version of the datasets, i.e., the NSL-KDD and UNSW-NB15 test sets, where 5% Gaussian noise was injected into the feature values. Tables 8 and 9 show the results under clean and noisy conditions. Across both datasets, Gradient Boosting and the Autoencoder demonstrated the strongest resistance to noise, with accuracy remaining effectively unchanged—reflecting the robustness of tree-based rank-order decisions and the Autoencoder's exposure to noise during training. In contrast, the LSTM and Hybrid models showed notable accuracy drops under noisy conditions, falling from 89.70% to 76.17% and from 88.33% to 75.84% on NSL-KDD, and from 92.79% to 86.19% and 90.03% to 85.81% on UNSW-NB15. Noise also shifted their detection behaviour: on NSL-KDD, Recall increased for both models as perturbations pushed borderline connections above the attack threshold, improving R2L detection substantially (LSTM from 77.89% to 91.44%, Hybrid from 83.47% to 93.93%). On UNSW-NB15, however, Recall decreased due to the dataset's more complex attack patterns. Family-level

detection remained stable for DoS across all models, and the Hybrid preserved perfect Probe detection under noise. The Autoencoder remained the most stable model overall, while the Hybrid continued to deliver the strongest cross-family detection even under noisy conditions.

Table 8. Clean vs. noisy overall metrics on NSL-KDD and UNSW-NB15.

DataSets	Model	Clean Accuracy	Noisy Accuracy	Clean Precision	Noisy Precision	Clean Recall	Noisy Recall	Clean F1	Noisy F1
NSL-KDD	GB	83.69	83.78	89.12	89.88	81.26	80.57	85.01	84.97
	Autoencoder	83.96	83.79	88.98	88.60	81.98	82.00	85.33	85.00
	LSTM	89.70	76.17	88.83	71.00	93.70	97.00	91.20	82.00
	Hybrid	88.33	75.84	85.26	71.00	96.10	98.00	90.36	82.00
UNSW-NB15	GB	89.37	89.51	99.49	99.00	84.81	85.00	91.57	92.00
	AE	74.26	74.12	90.45	90.00	69.51	69.00	78.61	78.00
	LSTM	92.79	86.19	92.66	91.00	97.10	88.00	94.83	90.00
	Hybrid	90.03	85.81	88.42	87.00	98.21	93.00	93.06	90.00

Table 9. Family detection rates under clean and noisy conditions (%) on NSL-KDD and UNSW-NB15.

DataSets	Family	GB Clean	GB Noisy	AE Clean	AE Noisy	LSTM Clean	LSTM Noisy	Hybrid Clean	Hybrid Noisy
NSL-KDD	DoS	99.58	99.32	91.22	91.23	99.56	99.12	99.69	99.45
	Probe	98.35	96.08	99.92	99.92	94.47	97.77	100.00	100.00
	R2L	20.03	19.45	43.40	43.88	77.89	91.44	83.47	93.93
	U2R	59.70	58.21	65.67	64.18	94.03	91.04	100.00	97.01
UNSW-NB15	DoS	97.90	97.89	82.98	82.79	99.71	90.37	99.86	97.40
	Probe	95.10	95.22	58.25	58.03	96.95	82.62	98.56	90.33
	R2L	81.50	81.99	69.26	69.17	96.70	88.84	97.91	92.94
	U2R	75.50	76.41	50.04	49.88	98.18	82.58	98.42	88.84

3.4. ROC and PR Curves

The LSTM achieved the highest ROC AUC at 0.938, meaning it separates normal from attack traffic most consistently across all possible detection thresholds. The Autoencoder followed at 0.920, which is a strong result for a model that never saw any attack data during training. Gradient Boosting produced the lowest AUC at 0.841, consistent with its weaker overall classification performance. The Autoencoder scored slightly higher in Average Precision at 0.942 compared to the LSTM at 0.938, meaning it maintained better precision across the full range of detection sensitivities despite having lower binary recall at the chosen threshold. For the Hybrid model, ROC curves were generated separately for each attack family, as shown in Figure 3.

Probe achieved the highest AUC at 0.9770, followed by DoS at 0.9604 and U2R at 0.9444. R2L was the lowest at 0.8403, consistent with it being the hardest attack family across all evaluations in this study. Even so, 0.84 is well above random chance. Precision–Recall curves were generated for the three zero-day attack categories. The baseline shows the proportion of each attack type in the test set, as shown in Figure 4. Saint produced the highest Average Precision at 0.7244, which is 22 times above its random baseline of 0.032. mscan followed at 0.5810 against a baseline of 0.093. httptunnel was the most challenging at 0.2340, though still 17 times above its random baseline of 0.014, confirming that the Hybrid retains meaningful detection capability even for this particularly difficult zero-day attack type.

ROC AUC and Average Precision scores for the three standalone models on UNSW-NB15. On UNSW-NB15, Gradient Boosting achieved the highest ROC AUC at 0.983 and Average Precision of 0.992, unlike NSL-KDD, where the LSTM came out on top. This is the strongest ROC result seen across both datasets. The LSTM followed at 0.9727, also a strong result. The Autoencoder produced the lowest AUC at 0.7821, consistent with its weaker

overall performance on this dataset. For the Hybrid model, ROC curves were generated separately for each attack family as shown in Figure 5.

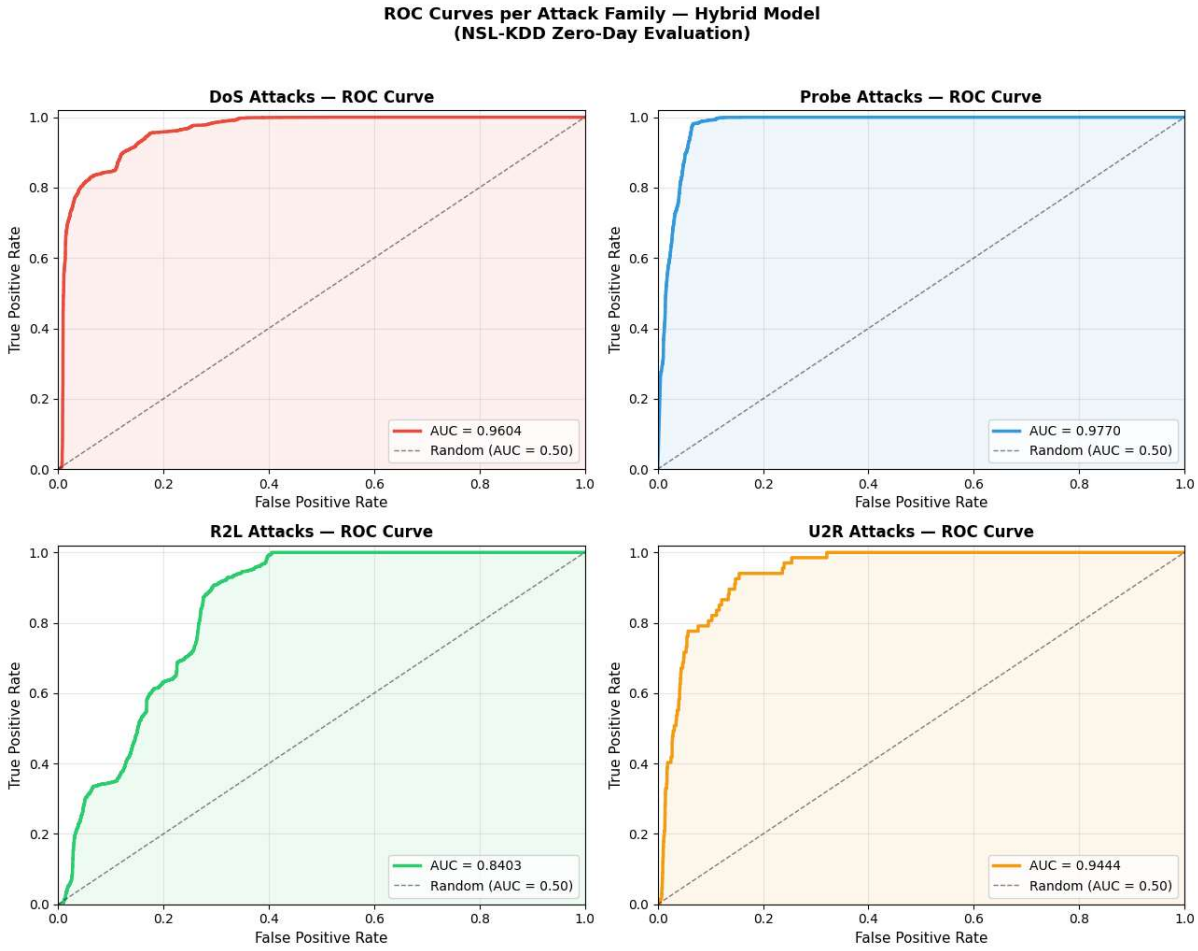

Figure 3. ROC curves per attack family for the Hybrid model on NSL-KDD.

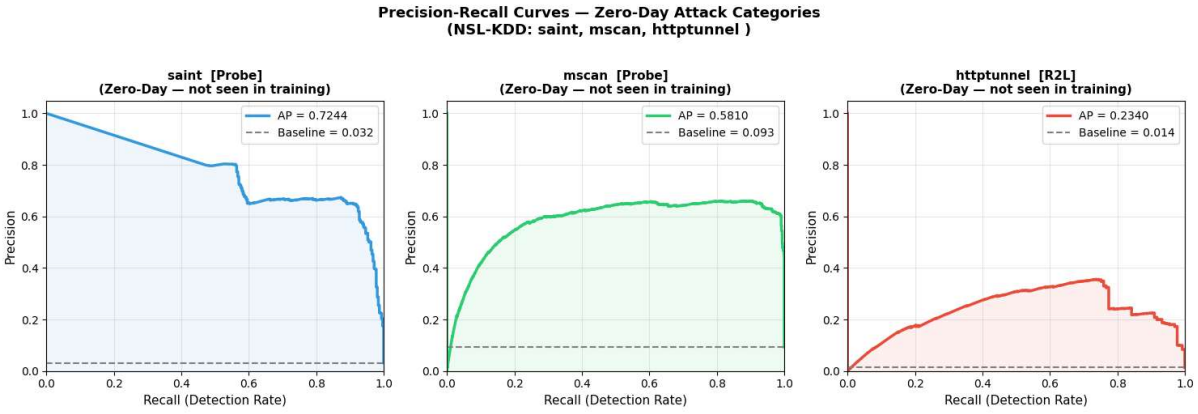

Figure 4. Precision–Recall curves for saint, mscan and httptunnel on NSL-KDD Hybrid.

All four families exceeded 0.96 AUC, a stronger result than NSL-KDD, where R2L was the weakest at 0.8403. This confirms that UNSW-NB15 attack families are more separable for the Hybrid pipeline than their NSL-KDD counterparts, particularly for R2L attacks, which were the hardest family on NSL-KDD. Precision–Recall curves were generated for the three zero-day attack categories as shown in Figure 6.

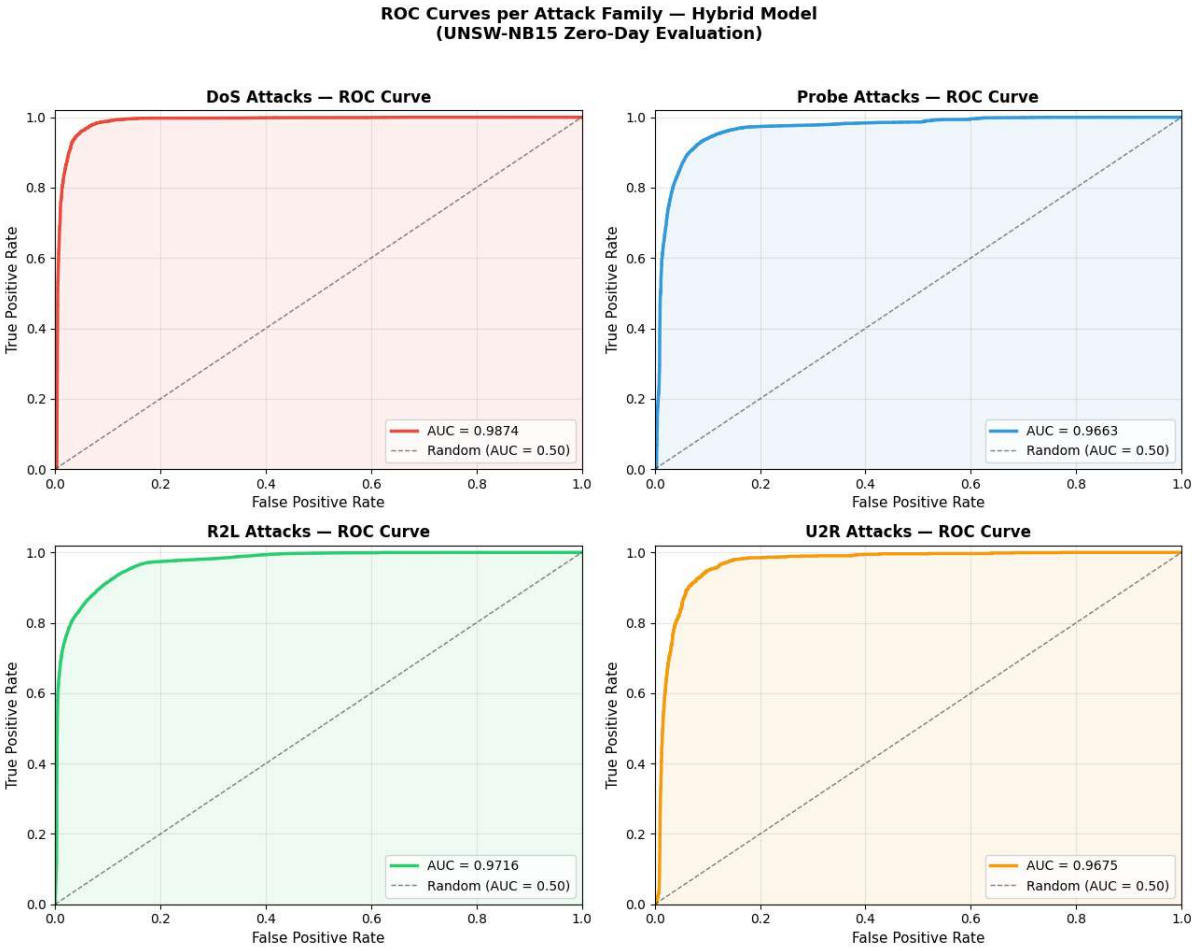

Figure 5. ROC curves per attack family for the Hybrid model on UNSW-NB15.

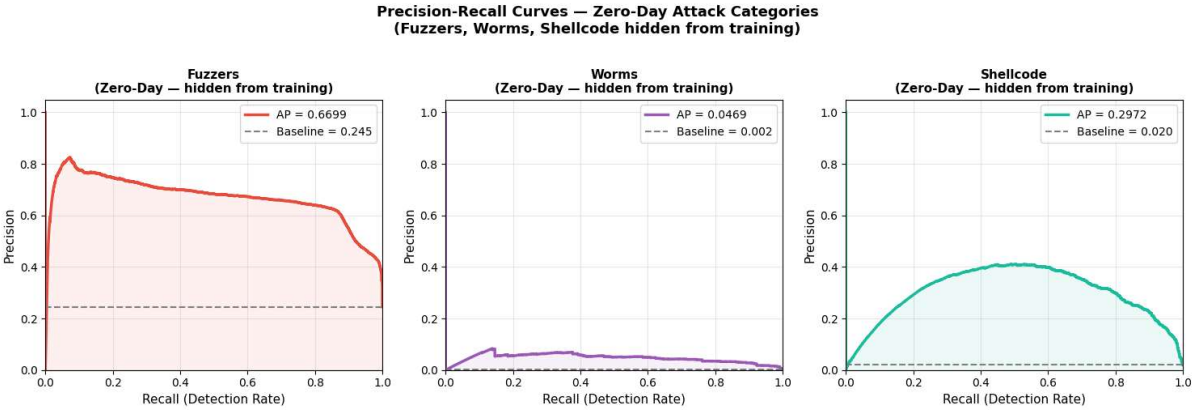

Figure 6. Precision–Recall curves for Fuzzers, Worms and Shellcode on UNSW-NB15 Hybrid.

Fuzzers achieved the highest Average Precision at 0.6699, which is 2.7 times above its baseline of 0.245. Shellcode produced an Average Precision of 0.2972 against a baseline of 0.020, which is 15 times above random chance. Worms was the most challenging at 0.0469, only 23 times above its baseline of 0.002. The low Average Precision for Worms reflects the very small number of Worm connections in the test set, only 130 instances, making precise detection across all Recall levels difficult.

3.5. Statistical Validation

Across both NSL-KDD and UNSW-NB15, McNemar's test shows that the Hybrid model's improvement over the standalone LSTM is statistically significant, as presented in Table 10. In each dataset, the LSTM achieves more "LSTM-only correct" cases, reflecting its higher precision, while the Hybrid achieves fewer but important "Hybrid-only correct" cases, reflecting its higher recall in detecting attacks the LSTM misses. Both tests produce extremely small p-values—effectively zero—confirming that the Hybrid's advantage is genuine rather than due to chance. The larger chi-square value for UNSW-NB15 reflects its larger test set, not a stronger improvement. Overall, the results demonstrate a consistent, statistically reliable performance gain of the Hybrid model across different network intrusion datasets.

Table 10. McNemar's test contingency for NSL KDD and UNSW-NB15.

Dataset	Both Correct	LSTM Only Correct	Hybrid Only Correct	Both Wrong	Chi-Square	p-Value	Interpretation
NSL-KDD	19,603	620	309	2012	103.44	2.68×10^{-24}	Hybrid's advantage is statistically significant; LSTM shows higher precision; Hybrid shows higher recall.
UNSW-NB15	156,531	6167	1331	11,312	3117.79	~0	Hybrid's advantage is statistically significant; same precision/recall pattern as NSL-KDD. Larger chi-square reflects dataset size.

Across both NSL-KDD and UNSW-NB15, the bootstrap confidence intervals show a consistent pattern (presented in Table 11): the Hybrid model achieves higher Recall, while the LSTM achieves higher Precision. In both datasets, the Recall confidence intervals do not overlap, confirming that the Hybrid reliably detects more attacks. Likewise, the Precision intervals do not overlap, confirming that the LSTM consistently produces fewer false alarms. For NSL-KDD, the F1-Score intervals overlap slightly, indicating similar overall balanced performance. For UNSW-NB15, the F1-Score intervals do not overlap, showing that the LSTM has stronger overall balance on this dataset. Taken together, the results show that the Hybrid's advantage is specifically in attack detection (Recall), while the LSTM maintains stronger precision and overall balance, especially on UNSW-NB15.

Table 11. Bootstrap confidence intervals (95%) for NSL KDD and UNSW-NB15.

Dataset	Model	Metric	Score	95% CI	Interpretation
NSL-KDD	LSTM	Recall	93.70%	[93.28–94.14%]	LSTM has lower Recall; Hybrid consistently catches more attacks.
		Precision	88.83%	[88.30–89.37%]	LSTM has higher Precision; fewer false alarms.
		F1-Score	91.20%	[90.87–91.56%]	F1 ranges overlap with Hybrid; balanced performance similar.
	Hybrid	Recall	96.10%	[95.79–96.46%]	Higher Recall with no overlap; Hybrid detects more attacks.
		Precision	85.26%	[84.72–85.82%]	Lower Precision; more false alarms than LSTM.
		F1-Score	90.36%	[90.01–90.72%]	Slight overlap with LSTM; overall balance similar.
UNSW-NB15	LSTM	Recall	97.10%	[97.00–97.19%]	Lower Recall; Hybrid consistently catches more attacks.
		Precision	92.66%	[92.51–92.79%]	Higher Precision; fewer false alarms.
		F1-Score	94.83%	[94.73–94.91%]	Higher F1; better overall balance than Hybrid.
	Hybrid	Recall	98.21%	[98.14–98.29%]	Higher Recall with no overlap; Hybrid detects more attacks.
		Precision	88.42%	[88.25–88.58%]	Lower Precision; more false alarms.
		F1-Score	93.06%	[92.96–93.16%]	Lower F1; LSTM has better balanced performance.

3.6. XAI Findings

For the NSL-KDD dataset (see Appendix A, Table A2, and Figure 7 on the left), the SHAP analysis, that is, the SHAP global feature importance through autoencoder proxy models, shows that src_bytes and dst_bytes are the strongest indicators of malicious activity,

highlighting the central role of data volume in distinguishing attacks from normal traffic. The engineered features duration_src_ratio and service_focus_ratio also appear within the top ten most influential attributes, confirming that feature engineering contributes meaningful behavioural signals. Overall, NSL-KDD detection is driven primarily by traffic volume and service-level interaction patterns. In contrast, the UNSW-NB15 results (see Table A3 in Appendix A and Figure 7 on the right) reveal a different behavioural profile. The most influential features—sttl, ct_dst_src_ltm, and ct_state_ttl—show that TTL patterns and connection timing dominate attack detection. Engineered features such as bytes_per_second and ttl_ratio, ranking fourth and sixth respectively, provide strong additional indicators of abnormal throughput and temporal irregularities. These findings demonstrate that UNSW-NB15 relies more heavily on temporal and TTL-based anomalies, reflecting the richer diversity and complexity of modern attack behaviours captured in this dataset.

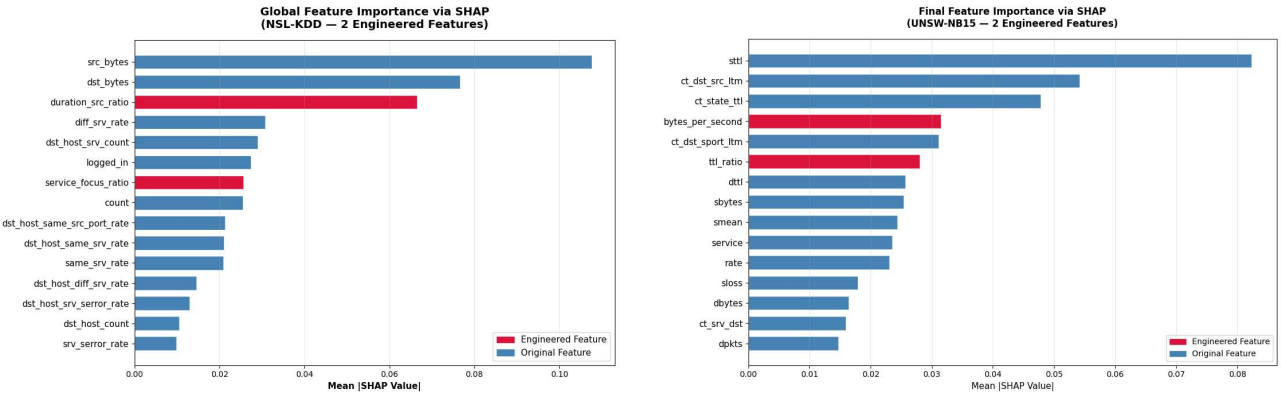

Figure 7. SHAP feature engineering through autoencoder importance analysis—NSL-KDD (**left**) and UNSW-NB15 (**right**).

Similarly, for the NSL-KDD dataset (see Figure 8 on the left), the LSTM model relies heavily on a single dominant feature: dst_host_srv_count, which achieves a SHAP value of 0.2336 and far exceeds the contribution of all other attributes. This strong single-feature dependence indicates that abnormal concentrations of service-level connections are the primary signal driving LSTM attack detection on NSL-KDD. The feature srv_rerror_rate also appears consistently across multiple zero-day attack explanations, reinforcing its role as a stable behavioural indicator of malicious activity. In contrast, the UNSW-NB15 results (see Figure 8 at right) reveal a more distributed pattern of importance. The top features—dttl, ct_dst_sport_ltm, and swin—share relatively similar SHAP magnitudes, showing that the LSTM model draws on a broader set of behavioural signals rather than relying on a single dominant attribute. These features capture TTL manipulation, repeated port targeting, and abnormal window sizes, all of which are characteristic of modern, high-variability attack behaviours. This more balanced importance distribution suggests that UNSW-NB15 requires richer temporal and packet-level cues for effective detection.

The SHAP local explanations provide detailed insight into how each model identifies zero-day attacks across both datasets. For NSL-KDD (see Figure 9 and Tables A4 and A5 in Appendix A), the mscan attack is primarily driven by unusually high concentrations of connections to the same source port and service, reflected in the dominance of dst_host_same_src_port_rate and dst_host_srv_count, which are characteristic of automated scanning behaviour. The *saint* attack exhibits a broader probing pattern, with dst_host_srv_count and srv_rerror_rate emerging as the strongest contributors, indicating repeated service-level interactions and elevated reset errors. In contrast, the httptunnel attack presents a mixture of malicious and normal-like behaviour; while dst_host_srv_count

and srv_rerror_rate push the model toward an attack classification, dst_host_same_srv_rate exerts a negative SHAP influence, pulling the prediction toward normal traffic. This partial mimicry explains why httptunnel is the most challenging R2L attack to detect. The corresponding figure legend for mscan and saint shows only positive (red) contributions, since every top-ranked feature for these two attacks pushed the prediction toward the attack class; no feature exerted a negative (blue) influence for these instances. For UNSW-NB15 (see Figure 10 and Table A6 in Appendix A), the Fuzzer attack is identified through repeated port targeting and abnormal window sizes, with ct_dst_sport_ltm and swin contributing strongly to the decision. Shellcode attacks are dominated by dttl, which accounts for approximately 60% of the model's decision, making it the strongest single-feature signal observed across all local explanations. The *Worms* attack similarly shows dttl as the primary indicator, supported by elevated mean packet sizes (dmean), revealing a consistent TTL-based anomaly pattern. Taken together, the cross-dataset analysis shows that TTL-related features are the most stable and consistent indicators of zero-day attacks, regardless of dataset or attack type.

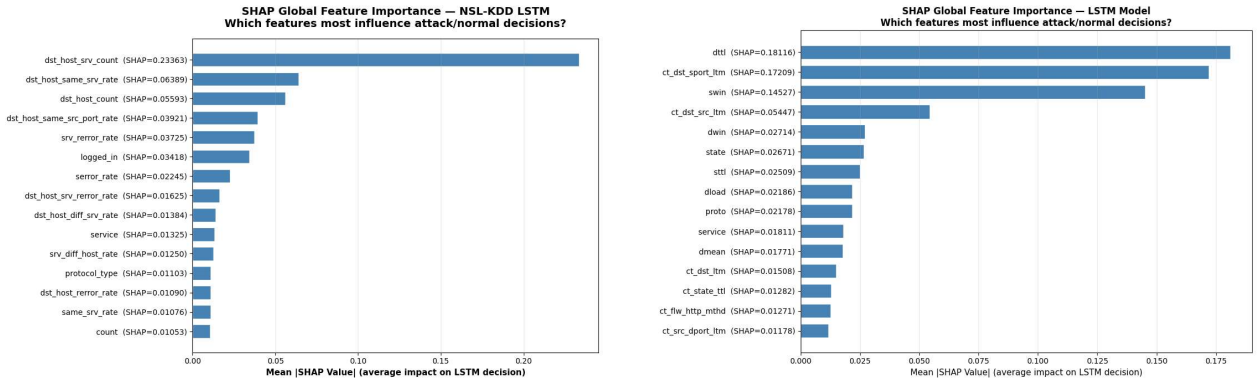

Figure 8. Global SHAP feature importance through LSTM–NSL-KDD (**left**) and UNSW-NB15 (**right**).

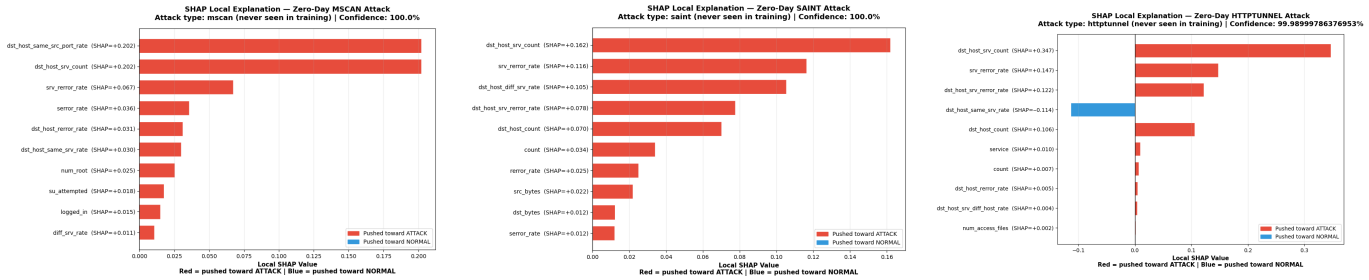

Figure 9. SHAP local explanation for mscan (**left**), saint (**middle**) and httptunnel (**right**) attacks on NSL-KDD.

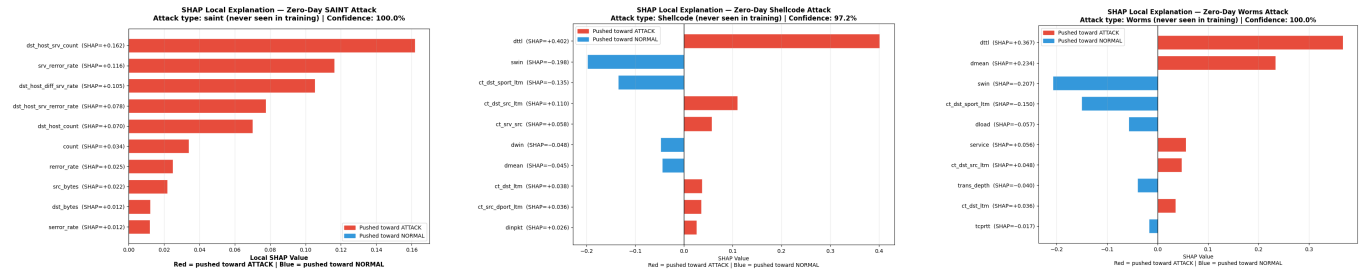

Figure 10. SHAP local explanation for Fuzzer (**left**), Shellcode (**middle**) and Worms (**right**) attacks on UNSW-NB15.

The SHAP stability analysis demonstrates that explanation reliability is strong across both datasets, though with differing degrees of consistency. For NSL-KDD (see Table 12), the features dst_host_srv_count and dst_host_same_src_port_rate appear in the top three rankings across all random seeds and background sample sizes, yielding a mean Jaccard similarity of 0.63, which indicates stable top-two features with minor variation in the third position. In comparison, UNSW-NB15 (see Table 12) shows even higher stability: the features ct_dst_sport_ltm, dttl, and swin consistently occupy the top three positions across nearly all conditions, resulting in a Jaccard similarity of 0.83. This higher similarity score reflects stronger cross-condition agreement and suggests that UNSW-NB15 explanations are more robust and less sensitive to sampling variation than those of NSL-KDD.

Table 12. SHAP stability analysis results 'Top 3 Features' on NSL KDD and UNSW-NB15.

Condition	NSL-KDD LSTM (Mscan Attack)	UNSW-NB15 LSTM (Fuzzer Attack)
Seed 42	dst_host_srv_count, dst_host_same_src_port_rate, rerror_rate	ct_dst_sport_ltm, dttl, swin
Seed 123	dst_host_srv_count, dst_host_same_src_port_rate, srv_rerror_rate	ct_dst_sport_ltm, swin, dttl
Seed 999	dst_host_srv_count, dst_host_same_src_port_rate, dst_host_count	swin, ct_dst_sport_ltm, dttl
Background 25	dst_host_srv_count, dst_host_same_src_port_rate, srv_rerror_rate	dttl, swin, ct_dst_sport_ltm
Background 50	dst_host_srv_count, dst_host_same_src_port_rate, rerror_rate	ct_dst_sport_ltm, dttl, swin
Background 100	dst_host_srv_count, dst_host_same_src_port_rate, srv_rerror_rate	ct_dst_sport_ltm, dttl, state

The LIME cross-validation results further reinforce the reliability of the SHAP explanations across both datasets. For NSL-KDD (see Figure 11), LIME closely aligns with SHAP for the mscan and saint attacks, identifying the same dominant behavioural indicators such as dst_host_srv_count and srv_rerror_rate. In the case of httptunnel, LIME highlights su_attempted as the strongest contributor, yet srv_rerror_rate remains consistently influential across both methods, confirming its role as a stable zero-day signal. For UNSW-NB15 (see Figure 12), LIME matches SHAP almost exactly for all three attacks, with dttl emerging as the dominant feature for both Shellcode and Worms, mirroring the SHAP findings and underscoring the importance of TTL-based anomalies. Overall, the strong agreement between LIME and SHAP—particularly on UNSW-NB15—demonstrates that the explanations reflect genuine model behaviour rather than artefacts of a single interpretability method.

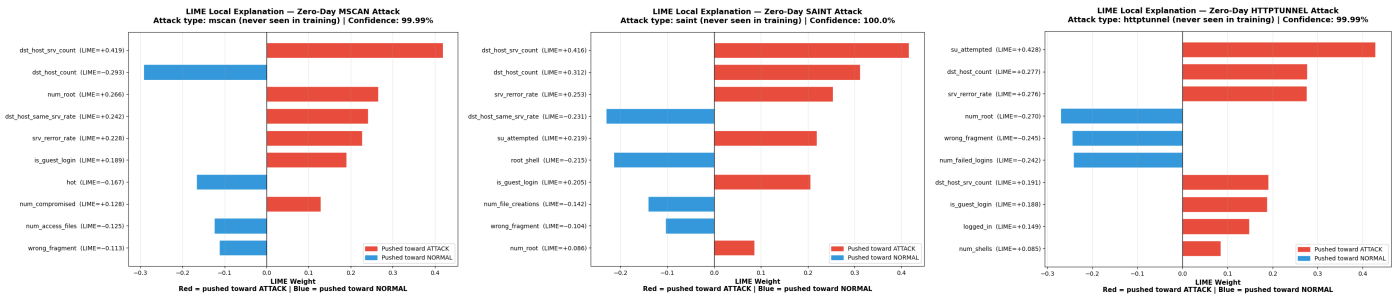

Figure 11. LIME local explanation for mscan (**left**), saint (**middle**) and httptunnel (**right**) attacks on NSL-KDD.

The Autoencoder reconstruction analysis provides clear evidence of how zero-day attacks manifest as extreme deviations from learned normal traffic patterns across both datasets. For NSL-KDD (see Figures 13–15 on the left), the mscan attack produces the highest reconstruction errors in dst_host_srv_serror_rate, dst_host_serror_rate, and serror_rate, revealing an abnormal concentration of connection-level error signals typical of automated scanning behaviour. The normal-versus-attack comparison further highlights a dramatic spike in anomaly magnitude, making the attack visually and quantitatively distinct from

baseline traffic. In UNSW-NB15 (see Figures 13–15 on the right), the Fuzzer attack generates extreme anomalies in sbytes, sloss, and spkts, reflecting unusually high data volume and packet loss. The Autoencoder's total reconstruction error reaches 1.9844, which is 41.5× above the detection threshold, representing the strongest anomaly signal observed across both datasets. Together, these results show that Autoencoders reliably capture zero-day behaviour through large reconstruction error spikes tied to throughput, error-rate, and packet-level irregularities.

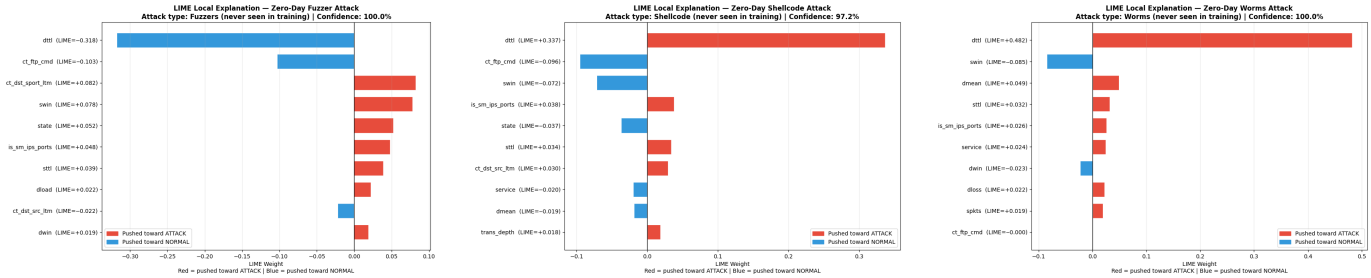

Figure 12. LIME local explanation for Fuzzer (**left**), Shellcode (**middle**) and Worms (**right**) attacks on UNSW-NB15.

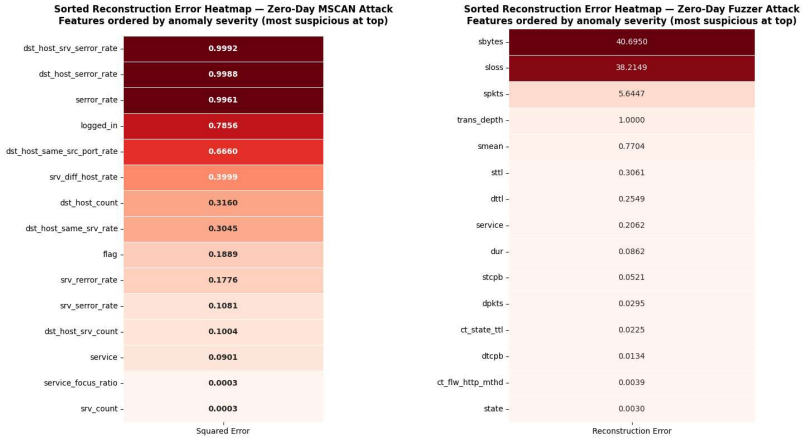

Figure 13. Sorted reconstruction error heatmap for mscan, ranked by anomaly severity NSL-KDD (**left**) and for Fuzzer attack, on UNSW-NB15 (**right**).

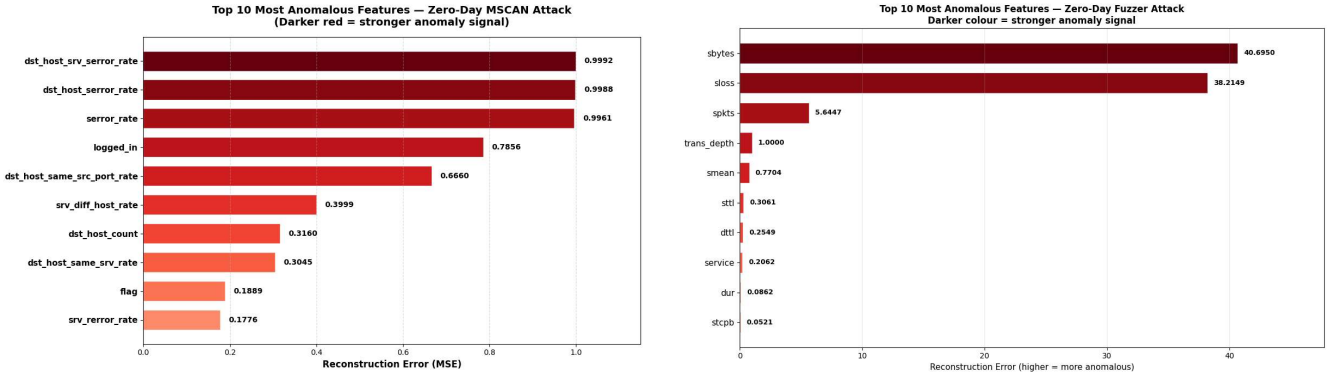

Figure 14. Top 10 most anomalous features for mscan on NSL-KDD (**left**) and for Fuzzer attack, on UNSW-NB15 (**right**) by reconstruction error.

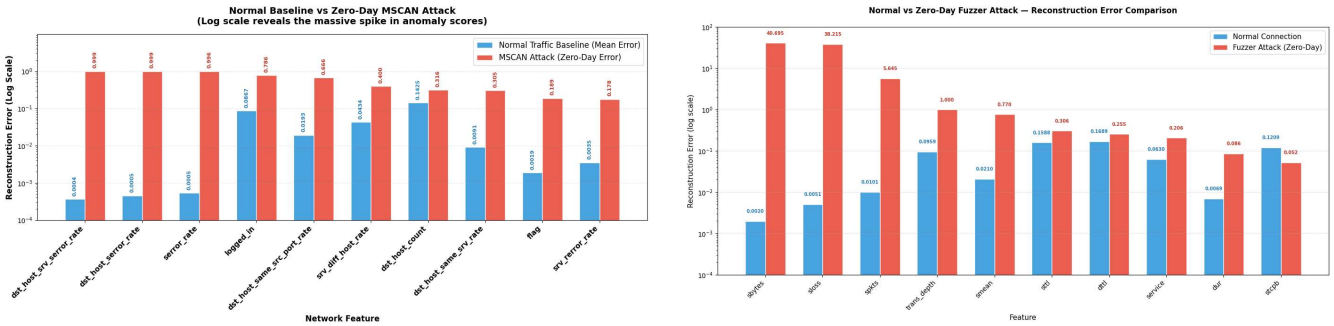

Figure 15. Normal vs attack reconstruction error comparison for mscan on NSL-KDD (**left**) and for Fuzzer attack on UNSW-NB15 (**right**).

3.7. Case Studies

Across both datasets, the case study analysis highlights how the Autoencoder consistently detects zero-day attacks that the LSTM fails to recognise due to the complete absence of these attack types during training, as presented in Table 13. On NSL-KDD, the Autoencoder identified 309 missed attacks, including 50 zero-day cases from saint, mscan and httptunnel, with representative examples showing that the LSTM assigned near-zero confidence while the Autoencoder produced anomaly scores ranging from 1.4× to 5.1× above the threshold—most notably for mscan, whose high reconstruction errors align with earlier heatmap findings. Similarly, on UNSW-NB15, the Autoencoder detected 1331 missed attacks, of which 977 were zero-day cases dominated overwhelmingly by Fuzzers (974 instances), alongside Worms and Shellcode. Representative cases again showed minimal LSTM confidence, while the Autoencoder produced strong anomaly signals, including 3.3× above threshold for Fuzzers—consistent with extreme reconstruction errors in features such as sbytes and sloss—and borderline detection for Worms at 1.0× the threshold. These combined results demonstrate that while the LSTM performs well on known attack types, it cannot generalise to unseen categories, whereas the Autoencoder reliably detects zero-day behaviour by identifying deviations from normal traffic patterns. Together, the two models form a complementary Hybrid pipeline that significantly improves zero-day detection performance beyond what either model achieves independently.

Table 13. Case studies-attacks AE caught, but LSTM missed both on NSL KDD and UNSW-NB15.

DataSets	Case	Attack	Family	LSTM Score	AE Score	×Threshold
NSL-KDD	1	saint	Probe	0.7%	0.0629	2.3×
	2	mscan	Probe	2.7%	0.1371	5.1×
	3	httptunnel	R2L	0.1%	0.0378	1.4×
UNSW-NB15	1	Fuzzers	R2L	3.6%	0.1571	3.3×
	2	Worms	U2R	4.0%	0.0487	1.0×
	3	Shellcode	U2R	2.9%	0.0759	1.6×

Practical explainability is an important requirement for IDS deployment, and SHAP and LIME can introduce non-trivial computational overhead. To quantify this cost, explanation-generation time was measured for both methods on the same zero-day instance, with the explainer pre-built to reflect the per-connection overhead expected in an operational setting. Averaged over five runs, SHAP required 0.21 s per instance, whereas LIME required 0.11 s, making LIME approximately 1.9× faster for on-demand explanations. These measurements represent the time to explain a single flagged connection after the explainer has already been initialised; SHAP's additional cost primarily arises from preparing the background sample once, rather than from repeated per-instance compu-

tation. To ensure that the use of a test-set background did not distort SHAP outputs, the analysis was repeated using a background drawn exclusively from the training set. In both cases, the same top-three features were identified, differing only in rank order, confirming that explanation behaviour is stable and not dependent on test-distribution information. Together, these results indicate that SHAP is suitable for generating background-level or analyst-driven explanations, while LIME offers lower latency for rapid, per-connection interpretation in real-time environments.

4. Discussion

This study set out to evaluate the zero-day intrusion detection capabilities of four machine-learning models—Autoencoder, LSTM, Gradient Boosting, and a Hybrid AE-LSTM pipeline—across two structurally different benchmark datasets, NSL-KDD and UNSW-NB15. The findings demonstrate several consistent patterns regarding model performance, explainability, stability, feature engineering, and practical deployment considerations. Here, the LSTM-based detection is evaluated under conditions of unseen attack subtype generalisation, in which previously unseen attack variants are withheld from training but remain part of the known DoS, Probe, R2L and U2R families, rather than fully open-set detection of entirely novel categories. That is, this constitutes subtype-level zero-day evaluation rather than family-level novelty and therefore does not represent fully open-set recognition. Our use of the term "strict zero-day" refers specifically to the complete exclusion of attack types from training, consistent with the widely adopted NSL KDD protocol, rather than the stronger condition in which both the subtype and its family are unknown. Importantly, the zero-day validity of the experiments is unaffected: the withheld attack types remain entirely unseen to all models, and the evaluation does not rely on categorical feature novelty.

The LSTM's internal gating mechanism enables nonlinear feature interaction and selective information retention across the input feature space, which remains beneficial even when each connection record is treated as a single timestep. This single timestep configuration reflects the structure of NSL KDD and UNSW NB15, where indicators of malicious behaviour are typically contained within an individual flow rather than requiring memory across multiple connections. We acknowledge, however, that with a sequence length of one, the LSTM's recurrent memory is not exercised in the conventional temporal sense; its advantage in this study derives primarily from gated nonlinear feature transformation rather than genuine multi-timestep sequence modelling. The LSTM was included for two reasons: first, both datasets provide connection level records rather than flow level time series, and constructing reliable multi flow sequences would require session reconstruction steps outside the scope of this work; second, the LSTM serves as a representative deep learning baseline widely used in the IDS literature, including prior studies such as Kim et al. [11] and Yin et al. [10], which also evaluate LSTMs on per connection inputs. We agree that true temporal modelling requires multi-timestep sequences capturing packet- or flow-level evolution, and we have clarified that the LSTM in this study operates in a single-timestep regime. Extending the pipeline to genuine sequential architectures—such as session-level LSTM/GRU models or Transformer-based flow sequences—represents a valuable direction for future research.

A methodological limitation of this work concerns the handling of categorical features. To avoid unseen-category failures during zero-day testing, the categorical encoders were fitted jointly on the combined training and test splits so that all protocol, service and state values received a valid integer mapping. Although this approach does not introduce label leakage—LabelEncoder maps category identities without learning any statistical relationships or class information—it does depart from strict train-only preprocessing and

therefore represents a deviation from best practice. Re-running the entire experimental pipeline with training-only encoding and an explicit unknown-category fallback was not feasible within the available time, as it would require regenerating all model outputs, explainability artefacts and statistical significance tests. Importantly, this limitation does not affect the validity of the zero-day evaluation, which is based on attack-type exclusion rather than categorical-feature novelty. The models do not gain any predictive advantage from knowing that a category exists in the test set, since the encoder exposes no frequencies, correlations or label relationships. The withheld zero-day attacks remain entirely unseen regardless of how categorical values are encoded, and the conclusions drawn from the zero-day experiments therefore remain robust.

A central outcome of the evaluation is the clear distinction between balanced performance and zero-day detection capability. Across both datasets, the LSTM achieved the highest F1-Scores (91.20% on NSL-KDD and 94.83% on UNSW-NB15), confirming its strength as a supervised classifier when attack types are present in training. In contrast, the Hybrid AE-LSTM pipeline achieved the highest Recall (96.10% and 98.21%), detecting substantially more zero-day attacks than the LSTM alone. The Hybrid's advantage was statistically significant according to McNemar's test, with effect sizes indicating small but meaningful improvements. Importantly, the Hybrid detected 309 additional attacks on NSL-KDD and 1331 additional attacks on UNSW-NB15 that the LSTM missed, demonstrating the operational value of combining supervised and unsupervised detection mechanisms.

NSL-KDD and UNSW-NB15 remain the two most widely used public datasets for controlled zero-day evaluation, largely because they provide clearly defined train/test splits, labelled attack families, and established zero-day protocols that enable reproducible comparisons across studies. Their use allows this work to be positioned within a long line of IDS research and ensures that the proposed hybrid model, explainability analysis, and stability assessment can be evaluated under conditions that are directly comparable to prior work. However, noting that the results should be interpreted within the constraints of these datasets, and that future work should incorporate more recent and realistic benchmarks to strengthen practical relevance and operational applicability. The methodological contributions of this study, including multi-model zero-day evaluation and stability-tested explainability, remain applicable to modern datasets and represent a foundation for future extensions.

The Hybrid model's confidence score also presents an important limitation. In this study, the Hybrid decision value is computed as the maximum of the min–max scaled Autoencoder reconstruction error and the LSTM class probability. This design reflects a practical decision-level fusion heuristic rather than a probabilistically calibrated measure. The two signals originate from fundamentally different processes—unsupervised anomaly magnitude versus supervised class likelihood—and are therefore not directly comparable on a probabilistic scale. As a result, the Hybrid score should not be interpreted as a calibrated probability but as a mechanism for prioritising strong detection signals from either component model, consistent with maximising zero-day recall. A more principled fusion strategy, such as probability calibration (e.g., Platt scaling) or a learned combination function, would provide a theoretically grounded confidence estimate. Implementing such methods would require retraining all models, regenerating explainability outputs, and re-running statistical significance tests, which was not feasible within the scope and time constraints of this work. Nonetheless, calibrated or learned fusion approaches represent a valuable direction for future refinement, particularly for operational IDS deployments where confidence scores must be probabilistically meaningful.

The explainability analysis addressed whether SHAP and LIME can produce meaningful explanations for zero-day detections. Both methods successfully identified security-

relevant features across all six zero-day case studies. SHAP explanations consistently highlighted behaviourally interpretable signals—such as dst_host_same_src_port_rate for mscan and dttl for Shellcode—while LIME largely validated these findings. Although the two methods diverged for httptunnel, this discrepancy reflects the inherent difficulty of explaining attacks that closely mimic normal traffic rather than a limitation of the explanation methods themselves. Overall, the agreement between SHAP and LIME across most attacks confirms that the explanations reflect genuine model behaviour and provide operationally meaningful insights for analysts.

A key requirement for practical deployment is explanation stability, addressed through SHAP consistency analysis across multiple random seeds and background sample sizes. The results show that the most influential features remain stable across conditions for both datasets. NSL-KDD exhibited a mean Jaccard similarity of 0.63, while UNSW-NB15 achieved 0.83, indicating even stronger stability. This consistency ensures that analysts can trust the explanations and rely on them for decision-making without concern that minor sampling variations will alter the interpretation of model behaviour.

The ablation study revealed that engineered features improve Autoencoder performance on both datasets, but the nature of the improvement differs. On NSL-KDD, engineered features increased overall accuracy but slightly reduced Recall, suggesting a trade-off between precision and sensitivity. On UNSW-NB15, engineered features consistently improved all metrics, with bytes_per_second being particularly important. These findings highlight that feature engineering must be dataset-specific rather than universally applied, as different network environments exhibit different behavioural patterns.

The broader evaluation of goals confirms that all seven objectives defined were achieved, including model implementation, metric evaluation under clean and noisy conditions, generation of SHAP and LIME explanations, Autoencoder heatmaps, stability analysis, ablation studies, and statistical validation of Hybrid improvements. This comprehensive methodology strengthens the reliability of the findings and ensures that conclusions are supported by multiple complementary analyses.

To validate that the features identified by SHAP genuinely influence the model's predictions, a feature-masking fidelity test was conducted across three representative zero-day Fuzzer connections spanning a range of model confidence levels (baseline prediction probabilities of 0.35, 0.54, and 0.83). For each instance, the top three SHAP-ranked features were individually replaced with their background-mean values, and the resulting change in prediction probability was measured. This was compared against a control condition in which three randomly selected, non-top-ranked features were masked instead.

Masking the top-ranked SHAP features produced substantially larger prediction shifts (21.2%, 62.2%, and 57.0% across the three instances) than masking randomly selected features (3.6%, 10.5%, and 16.8% respectively). On average, masking the top three SHAP-identified features altered the model's prediction approximately 4.91 times more than masking random features. This consistent pattern across multiple instances confirms that the features highlighted by SHAP substantially drive the model's decision-making process, providing empirical support for explanation fidelity beyond the consistency established through the stability analysis alone.

The Jaccard-based stability analysis was supplemented with Rank-Biased Overlap (RBO) [35], which weights agreement at higher-ranked feature positions more heavily than lower-ranked positions, addressing a limitation of Jaccard similarity in which all rank positions are treated equally. RBO was computed pairwise across the same six conditions used in the Jaccard analysis (three random seeds and three SHAP background sizes), using a persistence parameter of $p = 0.9$. The results (Mean RBO = 0.817, Minimum = 0.585, Maximum = 1.000) closely align with the Jaccard findings while providing additional

resolution: conditions that scored identically under Jaccard were distinguished under RBO according to whether disagreement occurred at the most influential feature position or at lower-ranked positions, offering a more analytically meaningful measure of explanation stability for security-operations use.

When placed in the context of related work, this study contributes several novel insights. Prior IDS research often evaluates models on a single dataset or under standard test splits, whereas this study evaluates four models across two datasets under explicit zero-day conditions. The Hybrid pipeline's consistent superiority in Recall across both datasets provides stronger evidence of its generalisability. Additionally, while earlier studies have applied SHAP or LIME to IDS models, few have examined explanation stability or cross-method consistency. This study extends the literature by systematically evaluating SHAP stability and validating explanations through LIME, addressing concerns raised in prior work about explanation variability.

The use of a surrogate was motivated by practical constraints: SHAP's model-specific explainers do not support Autoencoders or LSTM classifiers directly, and KernelSHAP becomes prohibitively slow on high-dimensional IDS datasets. Random Forest surrogates are widely used in XAI research as a tractable approximation of a model's decision boundary, enabling consistent instance-level explanations across heterogeneous architectures. Importantly, the surrogate was trained exclusively on the predictions of the original detector, ensuring that the explanations reflect the detector's input–output behaviour, even if they do not expose its internal mechanisms. We acknowledge that surrogate explanations have inherent limitations and do not fully capture the internal representations learned by deep models. For this reason, the manuscript does not claim that SHAP reveals the true internal logic of the Autoencoder or LSTM; instead, it is positioned as a practical method for identifying which input features most strongly influence the detector's decisions under zero-day conditions. The stability analysis further focuses on the consistency of these explanations rather than their absolute fidelity.

A further limitation concerns the choice of SHAP background samples. In the initial configuration, background points were drawn from the zero-day test set, which introduces test-distribution information into the explanation process and may bias feature attributions toward patterns present only in the evaluation data. We acknowledge that this departs from strict train-only explainability practice. To verify that this choice did not distort the resulting explanations, the analysis was repeated using a background sample drawn exclusively from the training set. Under both background configurations, the same top-three features were consistently identified for all examined zero-day instances, differing only in relative rank order. This confirms that the explanations are driven by the model's learned decision behaviour rather than by distributional information leaked through the background sample.

Despite these strengths, several limitations must be acknowledged. The datasets represent historical traffic snapshots, and real-world environments evolve continuously. SHAP KernelExplainer's computational cost limits real-time explanation generation, making background explanation generation more practical. Zero-day simulations cover only a subset of possible attack types, and SMOTE-based augmentation may not fully capture the diversity of rare attacks. Additionally, explanation outputs could theoretically aid attackers in evasion, though the Hybrid pipeline and Autoencoder's holistic anomaly detection reduce this risk.

Finally, the findings generalise beyond intrusion detection. The Autoencoder reconstruction heatmap approach can be applied to any anomaly-detection domain involving normalised tabular data, while SHAP stability results provide confidence for practitioners deploying explainable AI systems. In a broader societal context, combining deep learning detection with stable, interpretable explanations offers a practical path toward improving analyst efficiency in environments where alert volume is high, and human attention is

limited. Overall, this study demonstrates that a Hybrid AE-LSTM pipeline, supported by stable SHAP and LIME explanations and enhanced through targeted feature engineering, provides a robust and interpretable approach to zero-day intrusion detection across diverse network environments.

5. Conclusions

This study developed and evaluated an AI-enhanced intrusion detection system capable of detecting zero-day network attacks and explaining its decisions, addressing two major gaps in IDS research: the absence of cross-dataset model comparisons and the lack of reliability testing for XAI explanations. Four model architectures—Gradient Boosting, a Denoising Autoencoder, an LSTM, and a Hybrid AE-LSTM pipeline—were systematically assessed under strict zero-day conditions on NSL-KDD and UNSW-NB15. SHAP and LIME were used to generate interpretable local explanations, and a formal stability analysis employing Jaccard similarity, Rank Biased Overlap, and feature-masking fidelity tests confirmed their consistency across seeds and background sizes.

The results demonstrate that the LSTM achieved the strongest balanced performance, while the Hybrid AE-LSTM pipeline consistently delivered the highest Recall, detecting hundreds of additional zero-day attacks missed by the LSTM and yielding statistically significant improvements validated through McNemar's test and Cohen's h. The Autoencoder further proved effective at identifying attacks without prior exposure, offering complementary capability and clear reconstruction-based feature heatmaps. SHAP and LIME produced behaviourally accurate explanations across all six zero-day cases, identifying features such as port-scanning indicators for mscan, TTL anomalies for Shellcode, and high-throughput signals for Fuzzers.

Beyond model performance, this study contributes four advances to zero-day IDS research: (1) it provides a systematic cross-dataset evaluation of four models (Autoencoder, LSTM, Gradient Boosting, and a Hybrid AE LSTM pipeline) across two datasets, NSL KDD and UNSW NB15, under explicit zero-day conditions; (2) the introduction of a new Hybrid AE-LSTM pipeline that reliably improves zero-day Recall through union logic; (3) empirical evidence that SHAP and LIME explanations are stable and trustworthy for zero-day detections tested using Jaccard similarity, RBO, and a feature masking fidelity test across different random seeds and background sizes; and (4) a detailed systematic ablation study showing that engineered feature impact varies by dataset, underscoring the need for environment-specific feature design.

Overall, the findings show that the Hybrid AE-LSTM pipeline, supported by stable SHAP explainability, provides an effective and generalisable approach to zero-day intrusion detection across diverse network environments, fulfilling all goals defined in the introduction. Future work should evaluate the system on live network traffic, explore more realistic perturbation noise, conduct user studies with security analysts, expand testing to additional datasets and attack categories, and investigate adaptive models capable of evolving as new threats emerge.

Author Contributions: Conceptualization, M.U.A. and S.M.M.I.; methodology, S.M.M.I.; software, S.M.M.I.; validation, S.M.M.I. and M.U.A.; formal analysis, S.M.M.I. and M.U.A.; investigation, S.M.M.I. and M.U.A.; resources, S.M.M.I. and M.U.A.; data curation, S.M.M.I.; writing—original draft preparation, S.M.M.I. and M.U.A.; writing—review and editing, S.M.M.I., S.B. and M.U.A.; visualization, S.M.M.I., S.B. and M.U.A.; supervision, S.B. and M.U.A.; project administration, S.B.; funding acquisition, M.U.A. and S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by This study was supported by the following projects: (1) Trust_-Gen_Z, funded by VINNOVA, Diary No. 2024-01402; (2) xBest, funded by the Veten skapsrådet—The Swedish Research Council, Diary No. 2024-05613.

Institutional Review Board Statement: The study did not require ethical approval. Not applicable. For studies not involving humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study considered publicly archived datasets; no new data is generated during the study. The publicly available datasets are (1) CUP 99 Data Set. [27], 2009. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed on 29 August 2026; and (2) UNSW-NB15 [34]. https://www.kaggle.com/datasets/dhoogla/unswnb15, accessed on 29 August 2026.

Acknowledgments: GenAI (Claude (Sonnet 5)) has been used for purposes such as generating text.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AE	Autoencoder
AUC	Area Under the Curve
CI	Confidence Interval
DL	Deep Learning
DoS	Denial of Service
GB	Gradient Boosting
HIDS	Host-based Intrusion Detection System
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long Short-Term Memory
MCC	Matthews Correlation Coefficient
NIDS	Network-based Intrusion Detection System
ROC	Receiver Operating Characteristic
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Over-sampling Technique
SOC	Security Operations Centre

Appendix A

Table A1. Detection rates by native attack category on UNSW-NB15 (%).

Attack Type	Gradient Boosting (%)	Autoencoder (%)	LSTM (%)	Hybrid (%)
Analysis	73.30	80.85	87.10	95.15
Backdoor	98.68	82.59	99.60	99.71
DoS	97.81	83.03	99.72	99.89
Exploits	96.58	49.52	99.22	99.60
Fuzzers	13.47	39.84	84.88	90.24
Generic	99.88	99.10	99.98	99.98
Reconnaissance	99.29	53.94	98.83	99.21
Shellcode	73.52	50.84	98.15	98.32
Worms	92.31	43.08	98.46	99.23

Table A2. Top 10 features by mean SHAP value—NSL-KDD.

Rank	Feature	Mean SHAP	Type
1	src_bytes	0.1077	Original
2	dst_bytes	0.0766	Original
3	duration_src_ratio	0.0665	Engineered
4	diff_srv_rate	0.0308	Original
5	dst_host_srv_count	0.0291	Original
6	logged_in	0.0274	Original
7	service_focus_ratio	0.0256	Engineered
8	count	0.0255	Original
9	dst_host_same_src_port_rate	0.0213	Original
10	dst_host_same_srv_rate	0.0210	Original

Table A3. Top 10 features by mean SHAP value—UNSW-NB15.

Rank	Feature	Mean SHAP	Type
1	sttl	0.0823	Original
2	ct_dst_src_ltm	0.0542	Original
3	ct_state_ttl	0.0479	Original
4	bytes_per_second	0.0315	Engineered
5	ct_dst_sport_ltm	0.0312	Original
6	ttl_ratio	0.0280	Engineered
7	dttl	0.0257	Original
8	sbytes	0.0255	Original
9	smean	0.0244	Original
10	service	0.0236	Original

Table A4. SHAP local explanation report for mscan zero-day attack—NSL-KDD.

Rank	Feature	SHAP	Contribution	Explanation
1	dst_host_same_src_port_rate	+0.202	31.7%	Same source port reused-automated tool signature
2	dst_host_srv_count	+0.202	31.7%	Connections targeting same service-scanning behaviour
3	srv_rerror_rate	+0.067	10.6%	Service reset errors-rejected connections
4	serror_rate	+0.036	5.6%	High SYN error rate-TCP connection manipulation
5	dst_host_rerror_rate	+0.031	4.9%	High REJ error rate to destination host
6	dst_host_same_srv_rate	+0.030	4.7%	Focused service targeting
7	num_root	+0.026	4.0%	Number of root accesses-privilege escalation attempt
8	su_attempted	+0.018	2.8%	Switch user attempted-unauthorised access attempt
9	logged_in	+0.015	2.4%	Connection not logged in-deviates from normal sessions
10	diff_srv_rate	+0.011	1.7%	Different service rate -scattered port probing

Table A5. SHAP stability analysis results—NSL-KDD LSTM (mscan attack).

Condition	Top 3 Features
Seed 42	dst_host_srv_count, dst_host_same_src_port_rate, rerror_rate
Seed 123	dst_host_srv_count, dst_host_same_src_port_rate, srv_rerror_rate
Seed 999	dst_host_srv_count, dst_host_same_src_port_rate, dst_host_count
Background 25	dst_host_srv_count, dst_host_same_src_port_rate, srv_rerror_rate
Background 50	dst_host_srv_count, dst_host_same_src_port_rate, rerror_rate
Background 100	dst_host_srv_count, dst_host_same_src_port_rate, srv_rerror_rate

Table A6. SHAP local explanation report for Fuzzer zero-day attack—UNSW-NB15.

Rank	Feature	SHAP	Contribution	Explanation
1	ct_dst_sport_ltm	+0.180	37.5%	Same destination port targeted repeatedly
2	swin	+0.111	23.1%	Abnormal source window size
3	ct_state_ttl	+0.101	21.1%	Unusual TTL state combination
4	state	+0.051	10.6%	Abnormal connection state
5	dttl	+0.027	5.6%	Abnormal destination TTL value

Table A7. Noise level used during Autoencoder training—UNSW-NB15.

Noise Level	Accuracy	Precision	Recall	F1 Score
3%	74.10%	90.51%	69.20%	78.43%
5% (training level)	74.12%	90.33%	69.40%	78.49%
7%	74.26%	89.99%	69.95%	78.72%
10%	74.55%	88.95%	71.48%	79.27%

Appendix A.1 Evaluation Metrics

The evaluation metrics used in this study are defined as follows:

1. Precision measures the proportion of predicted attacks that were actually attacks:

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

2. Recall measures the proportion of actual attacks that were correctly detected:

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

3. F1-Score is the harmonic mean of Precision and Recall, balancing both:

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$$

Here, TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives.

4. AUC (Area Under the ROC Curve) measures how well the model separates attack traffic from normal traffic across all possible decision thresholds, with a value of 1.0 representing perfect separation and 0.5 representing random guessing.
5. For the interpretability metrics, Jaccard similarity measures how much two sets of top-ranked features overlap:

$$\text{Jaccard}(A, B) = |A \cap B| / |A \cup B|$$

where A and B are the top k feature sets being compared, and a score of 1.0 means the sets are identical.

6. Rank Biased Overlap (RBO) extends this idea by giving more weight to agreement at higher-ranked positions than lower ones:

$$\text{RBO}(S, T, p) = (1 - p) \times \Sigma (\text{from } d = 1 \text{ to } k)\ p\hat{}(d - 1) \times A_d$$

where S and T are the two ranked lists being compared, A_d is the proportion of overlap between the two lists at depth d, and p is a weighting parameter (set to 0.9 in this study) that controls how strongly earlier ranks are prioritised over later ones.

Appendix A.2 Threshold Sensitivity Analysis

A threshold sensitivity analysis was conducted to justify this choice empirically, sweeping the percentile from the 50th to the 94th and measuring Precision, Recall, and F1 Score at each level (Table A8). The results show that Precision increased steadily with the percentile threshold, from 79.26 percent at the 60th percentile to 90.45 percent at the 80th percentile, while Recall and F1 Score were both higher at lower percentiles, peaking at the 70th percentile (76.10 percent Recall, 80.70 percent F1). However, since the threshold percentile directly determines the proportion of normal traffic flagged as anomalous, a lower percentile also increases the nominal false alarm rate. At the 70th percentile, approximately

30 percent of normal training samples exceed the threshold, compared to approximately 20 percent at the 80th percentile. The 80th percentile was therefore retained as the operating threshold, prioritizing a lower false alarm rate over the marginal gains in Recall and F1 Score available at lower percentiles. This reflects the operational consideration that a high false alarm rate imposes a direct workload cost on security analysts, making a moderate false alarm rate a more practical deployment choice even at some cost to overall detection performance. The same thresholding strategy was applied consistently across all evaluation conditions—including perturbation noise tests—to ensure comparability between clean and perturbed results.

Table A8. Threshold sensitivity analysis.

Percentile	Threshold	Recall	Precision	F1 Score
60	0.03982	77.69%	79.26%	78.47%
70	0.04116	76.10%	85.89%	80.70%
80 (selected)	0.04776	69.51%	90.45%	78.61%
90	0.07479	44.26%	90.71%	59.49%
94	0.08788	37.35%	91.32%	53.01%

References

1. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
2. Sommer, R.; Paxson, V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010.
3. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
4. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A Survey of Network-Based Intrusion Detection Data Sets. *Comput. Secur.* **2019**, *86*, 147–167. [CrossRef]
5. Garcia-Teodoro, P.; Díaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-based Network Intrusion Detection: Tech-niques, Systems and Challenges. *Comput. Secur.* **2009**, *28*, 18–28. [CrossRef]
6. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
7. Thakkar, A.; Lohiya, R. A Review of Feature Selection Techniques for Intrusion Detection Systems. In Proceedings of the 21st Italian Conference on Theoretical Computer Science, Ischia, Italy, 14–16 September 2020.
8. Louk, M.; Tama, B.A. Dual-IDS: A Bagging-GBM Intrusion Detection System. *Appl. Sci.* **2022**, *213*, 119030. [CrossRef]
9. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
10. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
11. Kim, J.; Kim, H. Applying LSTM Networks for Intrusion Detection. In Proceedings of the 2016 International Conference on Platform Technology and Service (PlatCon), Jeju, Republic of Korea, 15–17 February 2017.
12. Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012.
13. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion De-tection. In Proceedings of the Network and Distributed System Security (NDSS) Symposium, San Diego, CA, USA, 18–21 February 2018.
14. Zavrak, S.; Iskefiyeli, M. Anomaly Detection in Network Traffic Using Variational Autoencoder. *IEEE Access* **2020**, *8*, 108346–108358. [CrossRef]
15. Hindy, H.; Atkinson, R.; Tachtatzis, C.; Colin, J.-N.; Bayne, E.; Bellekens, X. Utilising Deep Learning Techniques for Effective Zero-Day Attack Detection. *Electronics* **2020**, *9*, 1684. [CrossRef]
16. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and Composing Robust Features with Denoising Autoen-coders. In *ICML'08: Proceedings of the 25th International Conference on Machine Learning*; ACM: New York, NY, USA, 2008.
17. Mbona, I.; Eloff, J.H.P. Detecting Zero-Day Intrusion Attacks Using Semi-Supervised Machine Learning Approaches. *IEEE Access* **2022**, *10*, 69822–69838. [CrossRef]
18. Soltani, M.; Ousat, B.; Siavoshani, M.J.; Jahangir, A.H. An Adaptable Deep Learning-Based Intrusion Detection System to Zero-Day Attacks. *J. Inf. Secur. Appl.* **2023**, *76*, 103516. [CrossRef]

19. Alansary, S.A.; Ayyad, S.M.; Talaat, F.M.; Saafan, M.M. Emerging AI Threats in Cybercrime: A Review of Zero-Day Attacks via Machine, Deep, and Federated Learning. *Knowl. Inf. Syst.* **2025**, *67*, 10951–10987. [CrossRef]
20. Sruthi, C.K.; Ravikumar, A.; Sriraman, H. Enhancing Zero-Day Attack Detection with XAI-Driven ML Models and SMOTE Analysis. In Proceedings of the 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), Vellore, India, 3–4 May 2024; pp. 1–5.
21. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
22. Gaspar, D.; Silva, P.; Silva, C. Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron. *IEEE Access* **2024**, *12*, 30164–30175. [CrossRef]
23. Hermosilla, P.; Berríos, S.; Allende-Cid, H. Explainable AI for Forensic Analysis: A Comparative Study of SHAP and LIME in Intrusion Detection Models. *Appl. Sci.* **2025**, *15*, 7329. [CrossRef]
24. Mohale, V.Z.; Obagbuwa, I.C. Evaluating Machine Learning-Based Intrusion Detection Systems with Explainable AI: En-hancing Transparency and Interpretability. *Front. Comput. Sci.* **2025**, *7*, 1520741. [CrossRef]
25. Khan, N.; Ahmad, K.; Al Tamimi, A.; Alani, M.M.; Bermak, A. Explainable AI-Based Intrusion Detection Systems for Industry 5.0 and Adversarial XAI: A Systematic Review. *Information* **2025**, *16*, 1036. [CrossRef]
26. Jaccard, P. The Distribution of the Flora in the Alpine Zone. *New Phytol.* **1912**, *11*, 37–50. [CrossRef]
27. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, ON, Canada, 8–10 July 2009*; IEEE: New York, NY, USA, 2009.
28. Ribeiro, M.T.; Singh, S.; Guestrin, C. 'Why Should I Trust You?': Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13–17 August 2016; pp. 1135–1144.
29. McNemar, Q. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika* **1947**, *12*, 153–157. [CrossRef] [PubMed]
30. Efron, B.; Tibshirani, R. *An Introduction to the Bootstrap*; Chapman & Hall: London, UK, 1993.
31. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*; Lawrence Erlbaum Associates: Hillsdale, NJ, USA, 1988.
32. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
33. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
34. Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems. In *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015*; IEEE: New York, NY, USA, 2015.
35. Webber, W.; Moffat, A.; Zobel, J. A Similarity Measure for Indefinite Rankings. *ACM Trans. Inf. Syst.* **2010**, *28*, 20. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.