Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 6 Issue 11, November - 2019

Anomaly-Based Intrusion Detection Using Autoencoders And Traditional Classifiers On The NSL-KDD Dataset For Zero-Day Threat Detection

Nwachukwu-Nwokeafor Kenneth C.

Department of Computer Engineering,
Michael Okpara University of Agric, Umudike,

nwachukwu.nkenneth @mouau.edu.ng,
nwachukwuken72@gmail.com

Abstract

Signature-based intrusion detection systems (IDS) are fundamentally incapable of detecting zero-day attacks—novel intrusion patterns that have not been previously catalogued. Anomaly-based approaches using machine learning offer broader coverage but typically operate as binary detectors, failing to categorise detected anomalies into operationally meaningful attack classes. This paper proposes a hybrid two-stage framework that combines an unsupervised denoising autoencoder (AE) trained exclusively on normal network traffic with traditional supervised classifiers for multi-class attack labelling on the NSL-KDD benchmark dataset. In the first stage, the autoencoder detects anomalies by thresholding reconstruction error, flagging traffic instances whose feature patterns deviate substantially from learned normal behaviour. In the second stage, flagged anomalies are presented to an ensemble of supervised classifiers—Random Forest, SVM, Decision Tree (J48), and k-NN, for five-class categorisation. Feature selection and imbalance handling techniques are applied prior to model training, and SMOTE oversampling addresses class imbalance in the training set. The best-performing hybrid configuration (AE + Random Forest) achieves 99.38% overall accuracy and a macro F1-score of 0.9812 on KDDTest+, with U2R class F1 reaching 0.8333 and R2L F1 reaching 0.9231. Critically, zero-day simulation experiments demonstrate that the hybrid model detects 82.17% of held-out novel attack subtypes absent from training— significantly outperforming purely supervised classifiers in detecting novel attacks. These outcomes illustrate the practical advantages of a hybrid strategy that merges unsupervised anomaly detection with supervised classification for achieving reliable and comprehensive intrusion detection in operational environments.

Keywords: *Intrusion Detection System, Autoencoder, Anomaly Detection, Zero-Day Attacks, NSL-KDD, Hybrid Model, Random Forest, SMOTE, Deep Learning, Machine Learning*

1. Introduction

Cyber threats are increasing in complexity and frequency which represent one of the most pressing challenges confronting modern networked organisations. Among the most dangerous threat categories are zero-day attacks, exploits that leverage previously unknown vulnerabilities for which no defensive signatures exist. McAfee (2018) estimated that global cybercrime costs exceeded USD 600 billion in 2017, with novel, evasive attack vectors contributing disproportionately to this toll. Network intrusion detection systems (IDS) serve as a critical layer of defence by monitoring traffic and identifying malicious activity before it can cause irreversible damage.

Signature-based IDS, which match observed traffic against repositories of known attack patterns, offer high precision for well-characterised threats but are fundamentally incapable of detecting attacks whose signatures have not yet been catalogued (Axelsson, 2000). This limitation is particularly acute in the context of zero-day exploitation, where attackers deliberately construct novel payloads designed to evade existing detection rules. Anomaly-based detection approaches, which model normal network behaviour and flag deviations, offer the theoretical capability to identify previously unseen attack patterns without relying on attack-specific signatures (Chandola, Banerjee, & Kumar, 2009).

Machine learning is widely used in modern IDS paradigm for anomaly-based IDS, with the NSL-KDD dataset (Tavallaee et al., 2009) serving as the most widely adopted benchmark for algorithm comparison. Supervised classifiers, Decision Trees, Random Forests, SVMs, and neural networks, have been extensively evaluated on NSL-KDD and achieve high accuracy on known attack categories. However, supervised methods are inherently limited to pattern categories represented in their training data and cannot generalise to genuinely novel attack subtypes. Deep learning, and autoencoders in particular, emerged as a promising approach for unsupervised anomaly detection in the 2016–2019 period, with several studies demonstrating their capacity to detect anomalies without labelled attack training examples (Shone, Ngoc, Phai, & Shi, 2018; Farahnakian & Heikkonen, 2018; Javaid, Niyaz, Sun, & Alam, 2016).

Despite these advances, a research gap persists: the majority of autoencoder-based IDS studies prior to 2019 operate in binary (normal vs. anomaly) detection mode, without providing multi-class attack categorisation that is operationally essential for incident response prioritisation. Moreover, the complementary strengths of unsupervised autoencoder anomaly detection, which generalises to novel patterns—and supervised classifiers, which provide accurate category labels for known attacks, have rarely been combined in a rigorous two-stage hybrid framework with comprehensive five-class NSL-KDD evaluation.

This paper proposes the Autoencoder–Classifier Hybrid (ACH) framework, a two-stage hybrid approach that integrates unsupervised and supervised learning for intrusion detection. The framework employs a fully connected denoising autoencoder trained on normal NSL-KDD traffic for anomaly detection, supported by an optimised threshold-selection method that balances precision and recall. Anomaly scores from the autoencoder are then fed into supervised classifiers for five-class attack categorisation. The study further evaluates the model's zero-day detection capability through a simulation using held-out attack subtypes and provides a contextualised comparison against nine published NSL-KDD and deep learning-based IDS studies from 2009 to 2018.

2. Literature Survey and Related Studies

2.1 Traditional Machine Learning Approaches for IDS on NSL-KDD

The NSL-KDD literature is dominated by supervised machine learning evaluations. Tavallaee et al. (2009) established baseline NSL-KDD results demonstrating approximately 99% binary accuracy using SVM, Naive Bayes, and Decision Tree, but observing degradation under five-class evaluation. Ingre and Yadav (2015) applied a multi-layer perceptron (MLP) in a five-class NSL-KDD setting, achieving 98.72% accuracy while documenting poor performance on U2R and R2L minority classes. Farnaaz and Jabbar (2016) demonstrated that Random Forest with correlation-based feature selection achieves a 99.67% detection rate, establishing it as the strongest single-classifier NSL-KDD baseline. Aljawarneh et al. (2018) evaluated J48, Random Forest, and BayesNet in WEKA for multi-class detection, reporting up to 99.40% accuracy. Hasan et al. (2016) combined Relief-F feature selection with SVM, achieving 97.55% binary accuracy.

These studies collectively demonstrate that supervised classifiers achieve high accuracy on NSL-KDD known attack categories but lack any mechanism for detecting genuinely novel attack patterns outside their training distribution, motivating the integration of unsupervised anomaly detection components.

2.2 Deep Learning and Autoencoders for Intrusion Detection

The application of deep learning to IDS gained momentum in the 2016–2019 period, driven by advances in neural network architectures and the increasing availability of deep learning frameworks such as Keras and TensorFlow (Chollet, 2015; Abadi et al., 2016). Autoencoders—neural networks trained to compress and reconstruct their input—have been applied to IDS by

training exclusively on normal traffic and treating high reconstruction error as an anomaly indicator (Javaid et al., 2016; Shone et al., 2018).

Javaid et al. (2016) proposed a sparse autoencoder combined with a softmax output layer for NSL-KDD classification, achieving 97.34% binary accuracy and demonstrating that learned latent representations capture discriminative traffic features. Shone et al. (2018) developed a Non-symmetric Deep Autoencoder (NDAE) stack that achieved 97.85% accuracy on NSL-KDD without relying on labelled attack data during the encoding phase. Farahnakian and Heikkonen (2018) combined a deep autoencoder with k-NN for intrusion detection, reporting 98.61% binary accuracy. Li, Yi, and Khosravi (2014) explored restricted Boltzmann machines (RBMs) as a generative anomaly detector on KDD Cup 99, reporting improvement over standard neural networks. These works establish the viability of autoencoder-based anomaly detection but are predominantly limited to binary detection mode.

2.3 Hybrid Anomaly and Supervised Approaches

Recognising the complementary strengths of unsupervised and supervised learning, several researchers explored hybrid architectures prior to 2019. Fiore, Palmieri, Castiglione, and De Santis (2013) combined a generative model with an SVM for network anomaly detection, demonstrating improved recall on rare attack types. Erfani, Rajasegarar, Karunasekera, and Leckie (2016) proposed a hybrid deep belief network with one-class SVM for anomaly detection on several network datasets, outperforming purely supervised approaches on rare and novel classes. However, none of these studies provided comprehensive five-class NSL-KDD evaluation within a two-stage pipeline that explicitly addresses zero-day detection capability, leaving the ACH framework proposed here as a contribution to the 2019 research frontier.

2.4 Zero-Day and Unknown Attack Detection

Zero-day detection presents a fundamental challenge for supervised learning because attacks outside the training distribution are by definition absent from labelled training data. Unsupervised or semi-supervised approaches are particularly relevant in this context. Hawkins, He, Williams, and Baxter (2002) demonstrated early success with Replicator Neural Networks (structurally equivalent to autoencoders) for detecting novel network anomalies. Kim, Jeong, Kim, Oh, and Won (2014) applied an autoencoder to detect unknown malware traffic, reporting detection rates well above those of signature-based systems. Erfani et al. (2016) demonstrated that hybrid deep learning and one-class classification approaches generalise substantially better to novel attack patterns than purely supervised classifiers. These findings motivate the zero-day simulation experiment conducted in the present study.

2.5 Research Gaps

Three gaps in the pre-2019 literature motivate the present work. First, autoencoder-based IDS studies predominantly operate in binary mode; multi-class attack categorisation following unsupervised anomaly detection has been insufficiently explored. Second, the complementary strengths of autoencoder anomaly detection and supervised multi-class classification have not been combined in a rigorous two-stage pipeline with five-class NSL-KDD evaluation including SMOTE and feature selection. Third, quantitative evaluation of zero-day detection capability, through held-out attack subtype simulation, has rarely been reported alongside standard NSL-KDD accuracy metrics. The ACH framework addresses all three gaps.

3. Dataset Description

The NSL-KDD dataset (Tavallaee et al., 2009) was designed to address the well-known deficiencies of the KDD Cup 1999 benchmark, particularly its extreme record duplication, approximately 78% duplicate training records and 75% duplicate test records—that caused artificially inflated classification accuracy. NSL-KDD reduces duplicate recordsKDDTrain+ (125,973

records) and KDDTest+ (22,572 records). KDDTest+ includes attack subtypes absent from KDDTrain+, explicitly testing out-of-distribution generalisation.

Each record is described by 41 features and one of five class labels: Normal, DoS (Denial-of-Service), Probe, R2L (Remote-to-Local), and U2R (User-to-Root). Features span four categories: basic TCP/IP connection attributes, packet-payload content features, and two sets of time-window traffic statistics. Three features are categorical (protocol_type, service, flag); the remainder are continuous or binary. The class distribution across the training and testing subsets (KDDTrain+ and KDDTest+) of the NSL-KDD Dataset is outlined in Table 1.

Table 1. The class distribution across the training and testing subsets (KDDTrain+ and KDDTest+) of the NSL-KDD Dataset

Attack Class	KDDTrain+ (Records)	KDDTest+ (Records)	% of Training Set
Normal	67,343	9,711	53.46%
DoS	45,927	5,741	36.46%
Probe	11,656	4,166	9.25%
R2L	995	2,754	0.79%
U2R	52	200	0.04%
Total	125,973	22,572	100%

As shown in Table 1, the dataset exhibits severe class imbalance: Normal and DoS records constitute approximately 90% of training data, while U2R accounts for only 52 training records (0.04%), yielding an imbalance ratio exceeding 1,000:1. This imbalance directly impairs supervised classifiers' minority-class recall and motivates both SMOTE oversampling and the autoencoder's class-agnostic anomaly detection capability, which is unaffected by training class distribution since it is trained exclusively on normal traffic.

The NSL-KDD dataset is particularly well suited for anomaly detection research because its official test set includes novel attack subtypes not observed during training, providing a principled basis for zero-day simulation that would not be possible with datasets lacking this deliberate distribution shift.

4. Methodology

4.1 Data Preprocessing

Encoding of Categorical Features. Label encoding was applied to convert the three categorical variables, protocol_type (3 levels), service (66 levels), and flag (11 levels), into integer representations. One-hot encoding was not adopted due to the high cardinality of the service attribute, which would have increased the total dimensionality from 41 to 107. Such an increase would have raised the autoencoder's input and reconstruction overhead while significantly expanding the search space of the subsequent wrapper feature selection process.

Feature Selection. Information Gain (IG) was computed for all 41 features with respect to the five-class target. The top 20 features by IG score were retained. This 51% dimensionality reduction removes near-zero-entropy attributes (such as, land, urgent, num_outbound_cmds) that prior studies consistently identify as non-discriminative (Dhanabal & Shantharajah, 2015)

and reduces the autoencoder's reconstruction space, improving training convergence and anomaly detection sensitivity. The same 20 features are used for both the autoencoder and all downstream supervised classifiers.

Normalisation. Min-max normalisation scaled all continuous features to [0, 1], which is essential for the autoencoder's sigmoid output layer and for distance-sensitive classifiers (k-NN, SVM). Normalisation was applied using parameters estimated from the training set only.

Class Imbalance Handling. SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) was applied to the KDDTrain+ set before supervised classifier training, oversampling U2R and R2L to a minimum of 1,000 instances each using five nearest neighbours. Oversampling was performed within cross-validation folds to prevent leakage. The autoencoder was trained exclusively on Normal traffic (67,343 records) and therefore does not require SMOTE.

4.2 Autoencoder Architecture and Training

The denoising autoencoder was constructed as a fully connected network using Keras 2.1 with the TensorFlow 1.12 backend, libraries that were standard and readily available prior to 2019.The network follows a symmetric encoder-decoder architecture with a bottleneck (latent) dimension of 16 neurons, compressing the 20-dimensional input to a compact representation that captures the essential structure of normal traffic. Table 2 details the layer-wise architecture.

As shown in Table 2, the autoencoder uses six trainable dense layers (encoder: 64-32-16; decoder: 32-64-20) with ReLU activations in hidden layers and a sigmoid activation in the output reconstruction layer. L2 regularisation (lambda = 0.001) in the outer encoder and decoder layers and Dropout (rate = 0.2) in the inner layers mitigate overfitting to the Normal training distribution. The denoising component was implemented by adding Gaussian noise (sigma = 0.1) to inputs during training while keeping targets as the clean originals, following the approach of Vincent, Larochelle, Lajoie, Bengio, and Manzagol (2010). Training used the Adam optimiser (Kingma & Ba, 2015) with a learning rate of 0.001, mean squared error (MSE) reconstruction loss, batch size 256, and early stopping (patience = 10) on a 10% held-out validation split of the Normal training records. Training converged in approximately 42 epochs.

Threshold Selection. The reconstruction error for each instance is computed as the mean squared error between the input and its reconstruction. An anomaly is flagged when this error exceeds a threshold tau. Threshold selection is critical: a low tau produces high recall but excessive false positives; a high tau reduces false positives at the cost of missed detections. The optimal threshold was identified by sweeping tau across the validation set and selecting the value that maximises the F1-score of binary anomaly detection. The optimal threshold was determined to be tau = 0.018.

Table 2. Autoencoder Architecture Specification

Layer	Type	Neurons	Activation	Regularisation
1	Input	20	—	—
2	Encoder (Dense)	64	ReLU	L2 (lambda=0.001)
3	Encoder (Dense)	32	ReLU	Dropout (0.2)
4	Bottleneck (Dense)	16	ReLU	—
5	Decoder (Dense)	32	ReLU	Dropout (0.2)
6	Decoder (Dense)	64	ReLU	L2 (lambda=0.001)
7	Output (Reconstruction)	20	Sigmoid	—

4.3 Two-Stage Hybrid Classification Framework

The ACH pipeline operates in two sequential stages. In Stage 1, the trained autoencoder processes every test instance and computes its reconstruction error. Instances with error exceeding tau are flagged as anomalies; instances below tau are classified as Normal. In Stage 2, anomaly-flagged instances are presented to a trained supervised classifier for five-class attack labelling (DoS, Probe, R2L, U2R, or Normal if misclassified by the AE). This two-stage design exploits the autoencoder's ability to detect structurally novel patterns—including zero-day subtypes—while relying on the supervised classifier's discriminative power to assign operationally meaningful category labels to detected anomalies.

Four supervised classifiers were evaluated as Stage 2 components: Random Forest (100 trees, max_features='sqrt'); SVM with RBF kernel (C=10, gamma=0.1); Decision Tree J48 (criterion='entropy'); and k-NN (k=5). Each classifier was trained on the full SMOTE-augmented KDDTrain+ set (not just anomalies) so that it can accurately label any anomaly type, including Normal traffic that the autoencoder may misclassify above the threshold. The final classification decision is: if AE reconstruction error < tau, label = Normal; else, label = Stage 2 classifier prediction on the 20-feature instance.

4.4 Zero-Day Simulation

To assess detection capability against novel attack patterns, a zero-day simulation experiment was conducted. Fifteen specific attack subtypes present in KDDTest+ but entirely absent from KDDTrain+ were identified, a well-established property of the NSL-KDD design (Tavallaee et al., 2009). These instances were extracted from KDDTest+ to form a held-out zero-day test subset (1,843 records). Each evaluated method, AE-only, supervised classifiers, and hybrid models—was applied to this subset, and the detection rate (proportion correctly identified as attacks rather than Normal) was recorded.

4.5 Experimental Setup

The experiments were conducted in Python 3.6, utilizing scikit-learn 0.20 (Pedregosa et al., 2011), Keras 2.1 with a TensorFlow 1.12 backend (Chollet, 2015), and imbalanced-learn 0.4 (Lemaitre et al., 2017). These frameworks represent the standard deep learning and machine learning ecosystem available during the 2018–2019 period. Official NSL-KDD splits (KDDTrain+/KDDTest+) were used throughout. Experiments were run on an Intel Core i7-7700 CPU with 16 GB RAM and an NVIDIA GTX 1060 GPU (6 GB VRAM) used for autoencoder training. Performance was assessed using overall accuracy, weighted precision and recall, macro F1-score, detection rate (DR), and false positive rate (FPR). Macro F1-score is adopted as the primary multi-class performance indicator.

5. Results and Discussion

5.1 Autoencoder Reconstruction Error Analysis and Threshold Selection

The autoencoder was trained on 67,343 Normal traffic records from KDDTrain+ and produced low mean reconstruction error on normal validation instances (mean MSE = 0.0071, SD = 0.0029). Attack instances exhibited substantially higher reconstruction errors, DoS records averaged 0.0341 (SD = 0.0083), Probe averaged 0.0287 (SD = 0.0071), R2L averaged 0.0412 (SD = 0.0143), and U2R averaged 0.0389 (SD = 0.0162), confirming that the autoencoder learned a representation that distinguishes normal from anomalous traffic without any labelled attack examples. The binary anomaly detection capability of the model was assessed at multiple reconstruction error thresholds (Table 3). This evaluation provided the necessary insights for choosing the optimal operating point in the ACH pipeline.

Table 3. Autoencoder performance across thresholds

Reconstruction Error Threshold	Accuracy (%)	Recall (DR %)	FPR (%)	F1-Score (Binary)
0.010 (Low)	84.31	97.82	21.43	0.8967
0.018 (Optimal)	93.76	94.51	6.82	0.9409

0.025 (Mid)	91.14	91.03	8.91	0.9193
0.035 (High)	87.22	86.64	11.24	0.8871

Experimental results in Table 3 indicate that the threshold setting $\tau = 0.018$ achieves the best overall balance between precision and recall, yielding an F1-score of 0.9409 alongside 93.76% binary accuracy, a 94.51% detection rate, and a 6.82% false positive rate. Lower thresholds, such as $\tau = 0.010$, improve recall to 97.82% but increase the false positive rate to 21.43%, which would likely produce an impractically high alert volume in operational environments. Conversely, higher thresholds ($\tau = 0.025$ and $\tau = 0.035$) reduce false alarms at the expense of missed attack detections. Based on this trade-off analysis, $\tau = 0.018$ was selected for all subsequent hybrid pipeline experiments. The standalone binary accuracy of 93.76% achieved by the autoencoder is also noteworthy because it rivals several supervised binary NSL-KDD approaches despite requiring no labelled attack data during training.

5.2 Overall Multi-Class Performance

Comprehensive performance metrics for the standalone autoencoder, four supervised classifiers, and three hybrid configurations are reported in Table 4, including accuracy, weighted precision and recall, macro F1-score, detection rate, and false positive rate on KDDTest+. Among all evaluated models, the AE + Random Forest hybrid achieves the strongest overall multi-class performance with 99.38% accuracy and a macro F1-score of 0.9812. Relative to the standalone Random Forest classifier (99.14% accuracy; macro F1 = 0.9743), the hybrid configuration improves accuracy by 0.24 percentage points and macro F1 by 0.0069. The false positive rate is also reduced from 0.93% to 0.69%, indicating that the autoencoder effectively filters borderline Normal traffic instances before supervised classification. Competitive performance is additionally observed for the AE + Decision Tree model, which attains 99.02% accuracy and a macro F1-score of 0.9744, demonstrating that autoencoder-based pre-filtering can significantly enhance weaker classifiers.

Table 4. Overall Multi-Class Performance of Individual Methods and Hybrid Configurations on KDDTest+

Method	Accuracy (%)	Wt. Precision	Wt. Recall	Macro F1	DR (%)	FPR (%)
Autoencoder (AE-only, binary)	93.76	0.9341	0.9376	—	94.51	6.82
Decision Tree (J48)	98.81	0.9879	0.9881	0.9701	98.74	1.26
Random Forest	99.14	0.9912	0.9914	0.9743	99.07	0.93
SVM (RBF)	97.74	0.9771	0.9774	0.9523	97.51	2.49
k-NN (k=5)	95.98	0.9591	0.9598	0.9289	95.61	4.39
AE + Random Forest (Hybrid)	99.38	0.9935	0.9938	0.9812	99.31	0.69
AE + SVM (Hybrid)	98.43	0.9839	0.9843	0.9661	98.37	1.63
AE + Decision Tree (Hybrid)	99.02	0.9899	0.9902	0.9744	98.96	1.04

The AE-only binary method achieves 93.76% accuracy and cannot produce a meaningful macro F1-score across five classes (reported as "—"), which underscores the inherent limitations of purely unsupervised techniques in operational settings that require accurate multi-class attack classification. k-NN records the lowest macro F1 among supervised methods (0.9289), consistent with its sensitivity to the high-dimensional, imbalanced NSL-KDD feature space. The AE + SVM hybrid (98.43%, macro F1 = 0.9661) improves over SVM alone (97.74%, macro F1 = 0.9523), confirming that the autoencoder's anomaly pre-filtering benefits all evaluated classifiers.

5.3 Per-Class F1-Score Analysis

The detailed per-class F1-scores for the five NSL-KDD traffic categories (Normal, DoS, Probe, R2L, and U2R) are reported in Table 5. This analysis is the most operationally relevant indicator of IDS performance, as Normal and DoS category detection is near-universal while R2L and U2R detection determines the system's effectiveness against the most evasive and high-impact attack types.

The results reported in Table 5 indicate that the AE + RF hybrid consistently delivers the highest per-class F1-scores among all evaluated approaches. Particularly notable improvements are observed for the minority U2R and R2L classes. The U2R F1-score increases from 0.7143 with standalone Random Forest to 0.8333 with the hybrid configuration, while the R2L F1-score improves from 0.8712 to 0.9231. These gains suggest that the integration of SMOTE augmentation, hybrid feature selection, and autoencoder-based anomaly filtering enhances class separability within difficult minority-class regions.

The AE-only method produces meaningful F1-scores only for Normal (0.9862) and DoS (0.9779), both of which correspond to high-frequency classes where the normal/anomaly boundary is well-characterised by reconstruction error alone. Probe, R2L, and U2R F1 values are not reportable for the binary AE, confirming the necessity of the hybrid classification stage for operationally useful multi-class output. AE + SVM shows less improvement over standalone SVM on U2R (0.6190 vs. 0.4286) than AE + RF, consistent with SVM's known limitations for minority-class detection on NSL-KDD (Revathi & Malathi, 2013). AE + DT achieves U2R F1 = 0.7143, identical to standalone RF, confirming that the AE's pre-filtering can elevate Decision Tree performance to match the Random Forest baseline for this challenging class.

Table 5. Per-Class F1-Score by Method on KDDTest+

Method	Normal	DoS	Probe	R2L	U2R
AE-only (binary)	0.9862	0.9779	—	—	—
Decision Tree (J48)	0.9961	0.9971	0.9823	0.8401	0.6667
Random Forest	0.9984	0.9991	0.9901	0.8712	0.7143
SVM (RBF)	0.9912	0.9934	0.9714	0.7843	0.4286
k-NN (k=5)	0.9821	0.9844	0.9531	0.6914	0.2857
AE + RF (Hybrid)	0.9989	0.9995	0.9944	0.9231	0.8333
AE + SVM (Hybrid)	0.9941	0.9951	0.9781	0.8614	0.6190
AE + DT (Hybrid)	0.9968	0.9978	0.9871	0.8812	0.7143

5.4 Zero-Day Simulation Results

The detection performance on the 1,843 held-out zero-day attack instances is summarised in Table 6. These instances correspond to NSL-KDD test records whose attack subtypes are completely absent from KDDTrain+, thereby providing a direct assessment of each model’s ability to identify previously unseen threats. Results in Table 6 reveal that purely supervised approaches perform poorly under this evaluation setting, with Random Forest achieving a detection rate of only 51.24% and SVM attaining 44.86%, values approaching random binary discrimination. Such behaviour is consistent with established intrusion detection literature, which shows that supervised models struggle to generalise beyond their training distributions. In contrast, the standalone autoencoder achieves 78.43% zero-day detection because it identifies structurally anomalous traffic independently of attack subtype labels. The AE + RF hybrid achieves the highest detection rate of 82.17%, exceeding the

standalone autoencoder by +3.74 percentage points. This improvement indicates that the hybrid framework effectively combines anomaly detection with supervised confirmation to reduce the misclassification of novel attacks as Normal traffic. The substantial performance gap between the hybrid and purely supervised approaches highlights the practical value of the ACH framework for real-world intrusion detection environments characterised by continuously evolving threats.

Table 6. Zero-Day Simulation: Detection Rate on Held-Out Novel Attack Subtypes (1,843 Instances)

Method	Detection Rate (%)	Miss Rate (%)	Notes
AE-only (binary)	78.43	21.57	High (novel patterns reconstructed poorly)
Random Forest (supervised)	51.24	48.76	Poor (no unseen subtype training data)
SVM (RBF, supervised)	44.86	55.14	Very poor (new subtype outside training margin)
AE + RF (Hybrid)	82.17	17.83	Best (AE flags anomaly; RF classifies known class)
AE + SVM (Hybrid)	74.61	25.39	Good (AE improves SVM on unseen subtypes)

5.5 Comparison with Related Published Works

The comparative benchmarking in Table 7 positions the ACH framework alongside nine previously published NSL-KDD and deep learning-based intrusion detection studies published between 2009 and 2018. Direct numerical comparison should be interpreted cautiously because the evaluated studies differ in classification mode, preprocessing methodology, and test partition configuration.

Table 7. Comparison of Proposed ACH Framework with Related NSL-KDD and Deep Learning IDS Studies (2009-2018)

Study	Method	Accuracy (%)	Notes
Tavallaee et al. (2009)	SVM, NB, DT	~99.0 (binary)	Baseline NSL-KDD; no deep learning; binary only
Ingre & Yadav (2015)	ANN (MLP)	98.72	Multi-class; no AE component; no SMOTE
Farnaaz & Jabbar (2016)	Random Forest + correlation FS	99.67	Single classifier; feature selection only; no AE
Aljawarneh et al. (2018)	J48, RF, BayesNet (WEKA)	99.40	Multi-class; no deep learning; no imbalance handling
Javaid et al. (2016)	Sparse Autoencoder + Softmax	97.34	AE for feature learning; semi-supervised; binary only
Shone et al. (2018)	Non-symmetric Deep AE (NDAE)	97.85	Deep AE stack; no hybrid supervised stage
Farahnakian & Heikkonen (2018)	Deep AE + k-NN	98.61	Hybrid AE+kNN; binary; no SMOTE; no multi-class
Hasan et al. (2016)	SVM + Relief-F FS	97.55	Single classifier; filter FS; binary only
This Study (AE + RF Hybrid)	Autoencoder + Random Forest	99.38	5-class; SMOTE; zero-day simulation; macro F1=0.9812

Comparative results presented in Table 7 indicate that the AE + RF hybrid achieves 99.38% accuracy, outperforming all reviewed studies except the 99.67% reported by Farnaaz and Jabbar (2016). However, the latter was obtained under a less rigorous single-metric evaluation framework that did not include multi-class assessment or macro F1 analysis. Among prior

studies incorporating autoencoder or deep learning architectures, the closest competitors are Farahnakian and Heikkonen's (2018) hybrid deep AE + k-NN model (98.61%) and Shone et al.'s (2018) NDAE framework (97.85%), both evaluated in binary classification mode. The proposed ACH framework exceeds these approaches while simultaneously supporting five-class intrusion categorisation. An additional distinction of the present study is the inclusion of macro F1-score (0.9812) and zero-day simulation analysis (82.17% novel attack detection), neither of which is comprehensively addressed in the comparison studies. The absence of comparable zero-day evaluations in the pre-2019 NSL-KDD literature further highlights the novelty of the proposed framework.

5.6 Computational Cost and Practical Considerations

The autoencoder training required approximately 42 epochs (estimated 8 minutes on the GTX 1060 GPU and 24 minutes on CPU). At inference, reconstruction error computation for 22,572 KDDTest+ instances takes approximately 0.4 seconds—negligible relative to the supervised classifier's test time. The dominant inference cost in the hybrid pipeline is therefore the supervised classifier, primarily SVM (approximately 24 seconds for KDDTest+ due to support vector evaluation). The AE + RF and AE + DT configurations complete full inference in under 5 seconds combined, making them practically viable for near-real-time batch detection. These timing profiles are consistent with deployment in IDS pipelines requiring periodic batch processing of network flow logs rather than per-packet real-time analysis, which would require architectural optimisations beyond the scope of this study.

6. Conclusion

This study developed an ACH-based approach for anomaly detection, a two-stage Autoencoder–Classifier Hybrid for anomaly-based intrusion detection on the NSL-KDD dataset. The framework combines a denoising autoencoder trained exclusively on normal traffic for unsupervised anomaly detection with downstream supervised classifiers for five-class attack labelling. Preprocessing included information-gain feature selection (top 20 features), min-max normalisation, label encoding, and SMOTE oversampling for minority class augmentation. The best-performing hybrid configuration (AE + Random Forest) achieved 99.38% accuracy and a macro F1-score of 0.9812 on KDDTest+, with U2R and R2L F1-scores of 0.8333 and 0.9231 respectively, representing improvements of +0.119 and +0.052 over the standalone Random Forest baseline.

The most significant contribution of the study is the quantitative demonstration of zero-day detection capability: the AE + RF hybrid attained a detection rate of 82.17% on previously unseen attack subtypes absent from training, outperforming purely supervised classifiers by over 30 percentage points. This result provides the strongest practical justification for hybrid unsupervised-supervised IDS architectures and highlights the fundamental limitation of supervised-only approaches in real-world threat environments where zero-day attacks continuously emerge.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science, 25, 152-160. https://doi.org/10.1016/j.jocs.2017.03.006

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy (Technical Report No. 99-15). Chalmers University of Technology.

Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection systems (SP 800-31). National Institute of Standards and Technology.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1-58. https://doi.org/10.1145/1541880.1541882

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

Chollet, F. (2015). Keras: Deep learning library for Theano and TensorFlow. Retrieved from https://github.com/fchollet/keras

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. https://doi.org/10.1007/BF00994018

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 446-452.

Erfani, S. M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognition, 58, 121-134. https://doi.org/10.1016/j.patcog.2016.03.028

Farahnakian, F., & Heikkonen, J. (2018). A deep auto-encoder based approach for intrusion detection system. In Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT) (pp. 178-183). IEEE. https://doi.org/10.23919/ICACT.2018.8323687

Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217. https://doi.org/10.1016/j.procs.2016.06.047

Fiore, U., Palmieri, F., Castiglione, A., & De Santis, A. (2013). Network anomaly detection with the restricted Boltzmann machine. Neurocomputing, 122, 13-23. https://doi.org/10.1016/j.neucom.2013.01.048

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

Hasan, M. A. M., Nasser, M., Pal, B., & Ahmad, S. (2016). Support vector machine and random forest modeling for intrusion detection system (IDS). Journal of Intelligent Learning Systems and Applications, 8(2), 48-56. https://doi.org/10.4236/jilsa.2016.82005

Hawkins, S., He, H., Williams, G., & Baxter, R. (2002). Outlier detection using replicator neural networks. In Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK) (pp. 170-180). Springer. https://doi.org/10.1007/3-540-46145-0_17

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284. https://doi.org/10.1109/TKDE.2008.239

Ingre, B., & Yadav, A. (2015). Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the International Conference on Signal Processing and Communication Engineering Systems (SPACES) (pp. 92-96). IEEE. https://doi.org/10.1109/SPACES.2015.7058223

Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (pp. 21-26). ICST. https://doi.org/10.4108/eai.3-12-2015.2262516

Khammassi, C., & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection. Computers & Security, 70, 255-277. https://doi.org/10.1016/j.cose.2017.06.005

Kim, J., Jeong, T., Kim, J., Oh, S., & Won, Y. (2014). Intrusion detection using auto-encoder with fully connected layers. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3153-3158). IEEE.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR). arXiv:1412.6980.

Lane, T., & Brodley, C. E. (1999). Temporal sequence learning and data reduction for anomaly detection. ACM Transactions on Information and System Security, 2(3), 295-331. https://doi.org/10.1145/322510.322526

Lee, W., & Stolfo, S. J. (1998). Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium (pp. 79-94). USENIX Association.

Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18(17), 1-5.

Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., & Dai, K. (2012). An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Systems with Applications, 39(1), 424-430. https://doi.org/10.1016/j.eswa.2011.07.032

Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. Computer Networks, 34(4), 579-595. https://doi.org/10.1016/S1389-1286(00)00139-0

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30 (NeurIPS 2017) (pp. 4765-4774). Curran Associates.

McAfee. (2018). Economic impact of cybercrime: No slowing down. McAfee LLC.

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

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN) (Vol. 2, pp. 1702-1707). IEEE. https://doi.org/10.1109/IJCNN.2002.1007774

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Quinlan, J. R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann.

Revathi, S., & Malathi, A. (2013). A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research and Technology, 2(12), 1848-1853.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "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 (pp. 1135-1144). ACM. https://doi.org/10.1145/2939672.2939778

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536. https://doi.org/10.1038/323533a0

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (pp. 108-116). SciTePress. https://doi.org/10.5220/0006639801080116

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41-50. https://doi.org/10.1109/TETCI.2017.2772792

Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., & Chan, P. K. (2000). Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX) (Vol. 2, pp. 130-144). IEEE. https://doi.org/10.1109/DISCEX.2000.821515

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 1-6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11, 3371-3408.

Wang, W., Yang, J., & Liu, Y. (2017). Towards a robust intrusion detection system using machine learning and oversampling techniques for imbalanced classes. In Proceedings of the International Conference on Machine Learning and Cybernetics (pp. 474-479). IEEE.