SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X VOLUME 4, ISSUE 3, 2026

EXPLAINABLE DEEP LEARNING FOR NETWORK INTRUSION DETECTION USING AUTOENCODERS AND SHAP

Maryam Javaid[1], Hamza Gulshad[2], Imran Ullah[*3], Maha Asghar[4], Ali Jawad[5]

[1,4,5]*Department of Computer Science, University of Lahore, Sargodha Campus, Pakistan*
[2]*Department of Information Technology, University of Haripur, Pakistan*
[*3]*Department of Computer Science, Hazara University Dhodial, Mansehra, Pakistan*

[*3]imranullah.ajm@gmail.com

DOI: https://doi.org/10.5281/zenodo.23559960

Keywords
Network intrusion detection, autoencoder, anomaly detection, explainable AI, SHAP, deep learning, NSL-KDD, cybersecurity.

Article History
Received: 01 January 2026
Accepted: 16 March 2026
Published: 31 March 2026

Copyright @Author
Corresponding Author: *
Imran Ullah
Email:imranullah.ajm@gmail.com

Abstract
Network intrusion detection is a critical defence against the growing volume and sophistication of cyber-attacks, yet signature-based systems cannot recognise novel threats and many machine learning detectors operate as opaque black boxes that security analysts cannot trust. This paper presents an explainable deep learning framework for network intrusion detection that combines an unsupervised deep autoencoder with SHapley Additive exPlanations (SHAP). The autoencoder is trained only on normal traffic from the NSL-KDD benchmark so that it learns to reconstruct benign behaviour, and intrusions are identified as records whose reconstruction error exceeds a threshold set from the normal error distribution, which enables the detection of attacks unseen during training. The method is compared against two established anomaly detectors, Isolation Forest and One-Class SVM, across accuracy, precision, recall, F1-score, and the areas under the ROC and precision-recall curves. On the NSL-KDD test set of 22,544 records, the autoencoder achieves the highest-ranking quality with a ROC-AUC of 0.945 and a precision-recall AUC of 0.942, together with 92.3 percent precision, while the One-Class SVM attains the best fixed-threshold F1-score of 0.903. SHAP is then applied to the reconstruction-error function to attribute each detection to individual traffic features, identifying file-creation activity, host indicators, service types, and connection flags as the dominant intrusion indicators, in agreement with known attack behaviour. The framework delivers accurate anomaly detection together with transparent, feature-level explanations that support analyst trust and practical deployment.

1. INTRODUCTION

Modern computer networks are under constant and evolving attack, and intrusion detection systems (IDS) are a central line of defence, monitoring traffic to identify malicious activity [1]. Traditional signature-based detectors match traffic against a database of known attack patterns and are accurate for previously catalogued threats, but they are fundamentally unable to recognise novel or zero-day attacks whose signatures have not yet been recorded [2]. As attackers continually devise new techniques, this limitation has driven a shift

toward anomaly-based detection, which models normal behaviour and flags deviations from it, and is therefore capable of detecting previously unseen attacks [3].

Machine learning has become the dominant approach to anomaly-based intrusion detection, learning the statistical structure of network traffic from data rather than from hand-written rules [4]. Supervised classifiers can achieve high accuracy when abundant labelled attack data is available, but labelled attacks are scarce, quickly become outdated, and cannot cover attack types that have never been observed [5]. Unsupervised and semi-supervised methods address this by learning only from normal traffic, which is plentiful, and treating anything that deviates as a potential intrusion [6]. Among these, the autoencoder, a neural network trained to reconstruct its input through a compressed bottleneck, has become a popular and effective anomaly detector: trained on normal traffic, it reconstructs benign records accurately but fails to reconstruct attacks, so a high reconstruction error signals an intrusion [7], [8].

Despite their accuracy, deep learning detectors are widely criticised for their opacity [9]. A security analyst who receives an alert needs to understand why a connection was flagged, both to decide how to respond and to distinguish true threats from false alarms, which are costly and erode trust in the system [10]. A detector that outputs only a score, with no indication of which traffic characteristics drove it, is difficult to trust, to audit, and to act upon, which has slowed the adoption of deep learning in operational security settings [11]. This has motivated the application of explainable artificial intelligence (XAI) to intrusion detection, so that detections can be accompanied by human-understandable reasons [12].

Among XAI methods, SHapley Additive exPlanations (SHAP) has become the standard for attributing a model output to its input features, providing a theoretically grounded decomposition based on cooperative game theory that gives both a global view of feature importance and a local explanation for any single prediction [13], [14]. Applying SHAP to an autoencoder is less straightforward than to a classifier, because the autoencoder produces a reconstruction error rather than a class probability; the explanation must therefore attribute the reconstruction error itself to the input features. Doing so reveals which traffic characteristics make a record anomalous, turning an opaque anomaly score into an actionable, feature-level explanation.

This paper presents an explainable deep learning framework for network intrusion detection that unites an unsupervised autoencoder detector with SHAP-based explanation. The contributions are as follows. First, a deep autoencoder is trained on normal NSL-KDD traffic and used to detect intrusions by reconstruction-error thresholding, and its detection performance is rigorously compared against Isolation Forest and One-Class SVM baselines across six metrics including the areas under the ROC and precision-recall curves. Second, SHAP is applied to the reconstruction-error function of the autoencoder to produce global and local explanations, identifying the traffic features that most drive intrusion detections and validating them against known attack behaviour. Third, the analysis surfaces an honest and practically important trade-off between ranking quality, at which the autoencoder excels, and fixed-threshold classification, providing guidance for deployment. The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 presents the methodology. Section 4 reports the results and discussion. Section 5 concludes.

2. Literature Review

This section reviews machine learning for intrusion detection, autoencoder-based anomaly detection, and explainability in security systems.

2.1 Machine Learning for Intrusion Detection

Machine learning for intrusion detection has been studied extensively, with the KDD Cup 99 and its refined successor NSL-KDD serving as the dominant benchmarks [15]. Tavallaee et al. introduced NSL-KDD to address the redundancy and imbalance of the original KDD dataset, and it has since become the standard testbed for intrusion-detection research [15]. Early work applied classical classifiers such as decision trees, support vector machines, and naive Bayes to the

labelled data, achieving strong results on known attack types [4], [16]. Buczak and Guven surveyed the field and noted that while supervised methods achieve high accuracy on catalogued attacks, their dependence on labelled data limits their ability to detect novel intrusions [4].

Deep learning subsequently improved detection accuracy by learning hierarchical representations of traffic. Convolutional and recurrent networks have been applied to intrusion detection with strong supervised performance [17], [18]. However, these supervised deep models inherit the fundamental limitation of requiring labelled attacks, and they typically perform poorly on attack types absent from training. This has sustained interest in unsupervised and semi-supervised approaches that model normal traffic and detect deviations, which are better suited to the open-world nature of real intrusion detection [6], [19].

2.2 Autoencoder-Based Anomaly Detection

Autoencoders have become a widely used tool for unsupervised anomaly detection across domains [20]. An autoencoder compresses its input to a low-dimensional bottleneck and reconstructs it, and when trained only on normal data it learns a representation that reconstructs normal inputs well but reconstructs anomalies poorly, so the reconstruction error serves as an anomaly score [7]. For intrusion detection specifically, several studies have applied autoencoders to NSL-KDD and related datasets and reported competitive detection of attacks, including those unseen during training [8], [21]. Variants such as denoising, variational, and deep sparse autoencoders have been explored to improve the separation between normal and anomalous reconstruction errors [22].

Autoencoder detectors are commonly compared against classical unsupervised baselines. Isolation Forest isolates anomalies by random partitioning, under the principle that anomalies are easier to separate from the bulk of the data and therefore require fewer splits [23]. One-Class SVM learns a boundary that encloses the normal data in a high-dimensional feature space and flags points outside it as anomalous [24]. These methods are efficient and well understood, and they form the standard comparison set for neural anomaly detectors, which the present work adopts. While autoencoders often match or exceed these baselines in ranking quality, their advantage is not uniform across all operating points, a nuance that this paper examines directly.

2.3 Explainability in Intrusion Detection

As machine learning detectors move toward operational deployment, the need to explain their decisions has become pressing [9], [10]. Security analysts must triage large volumes of alerts, and an unexplained alert is difficult to prioritise or act upon, while regulatory and accountability requirements increasingly demand that automated security decisions be justifiable [11]. Among explanation methods, SHAP [13] has become the standard because it unifies earlier approaches under the theory of Shapley values and provides consistent global and local attributions, and its efficient variants make it practical for complex models [14].

Recent work has begun to apply SHAP and related methods to intrusion detection to interpret which features drive detections [12], [25]. Most of these studies explain supervised classifiers, where SHAP attributes a class probability to the input features. Explaining an unsupervised autoencoder is less common and more involved, because the quantity to be explained is the reconstruction error rather than a probability. Attributing the reconstruction error to the input features, as done in this work, directly answers the operationally important question of which traffic characteristics make a connection anomalous. Relatively few studies combine an accurate unsupervised autoencoder detector, a rigorous comparison against standard baselines, and SHAP-based explanation of the reconstruction error within a single framework, which is the gap this paper addresses.

3. Proposed Methodology

The proposed framework detects network intrusions by training a deep autoencoder on normal traffic and flagging records with high reconstruction error, then explains the detections with SHAP. Fig. 1 presents the overall framework,

Fig. 2 the autoencoder architecture, and Fig. 3 the detection and explanation workflow.

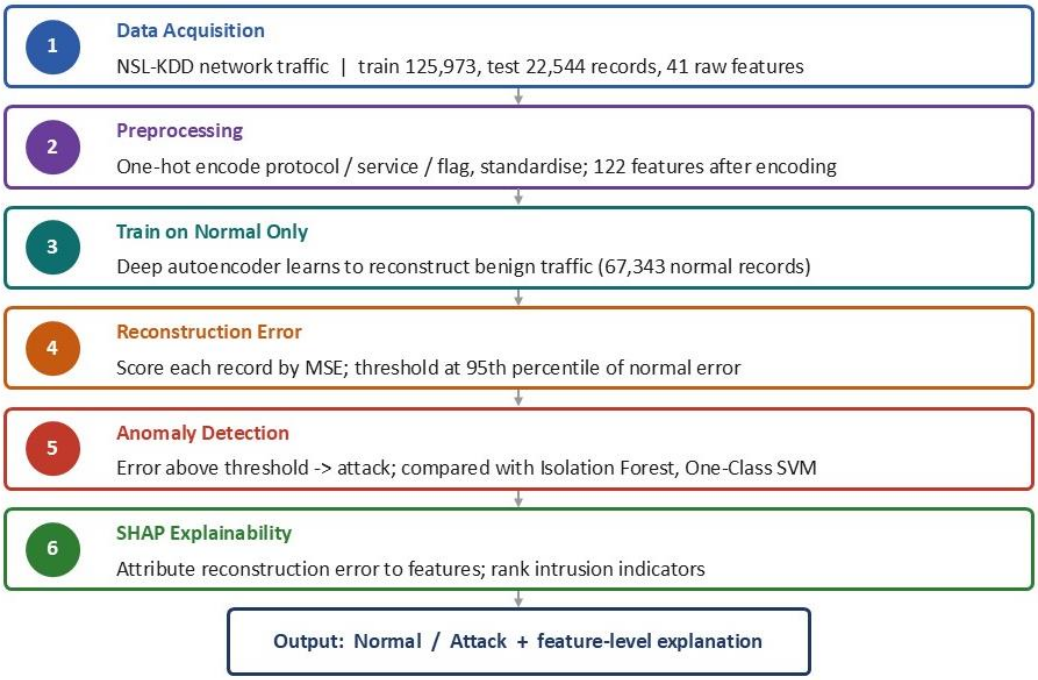


Fig. 1. Proposed explainable autoencoder framework for network intrusion detection, from data acquisition and preprocessing through training on normal traffic, reconstruction-error anomaly detection, baseline comparison, and SHAP explainability to the normal or attack decision.

3.1 Dataset and Preprocessing

The study uses the NSL-KDD benchmark [15], comprising 125,973 training and 22,544 test records, each described by 41 features that capture connection properties such as duration, protocol, service, flags, byte counts, and host- and service-level traffic statistics, together with a label. The multi-class attack labels are collapsed to a binary target, normal or attack. The categorical features protocol type, service, and flag are one-hot encoded, producing 122 features after encoding, and all features are standardised to zero mean and unit variance according to

$$z = (x - \mu) / \sigma \qquad (1)$$

where μ and σ are estimated from the training set. Crucially, the autoencoder is trained only on the 67,343 normal records, so that it learns the structure of benign traffic. Fig. 4 shows the class distribution and the most frequent attack types.

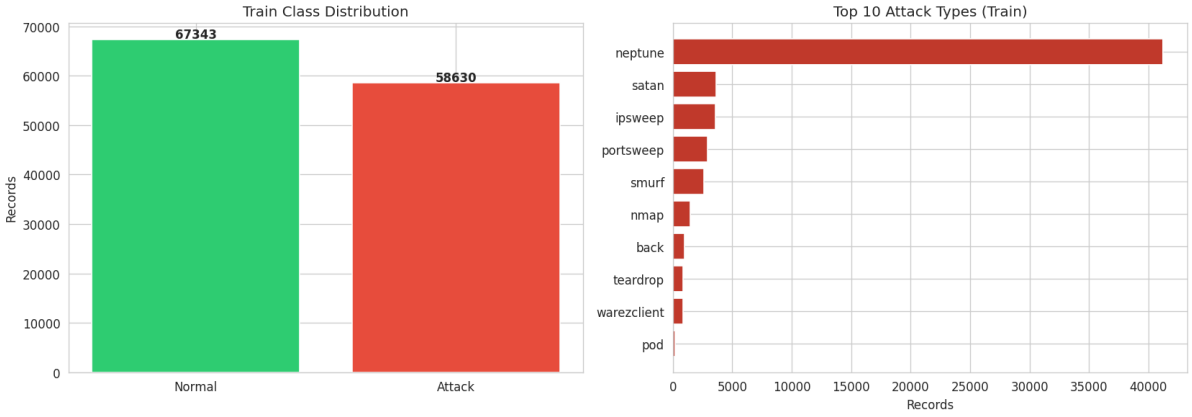


Fig. 4. NSL-KDD overview. Left: training class distribution of normal and attack records. Right: the ten most frequent attack types in the training set.

3.2 Deep Autoencoder

The detector is a deep autoencoder, shown in Fig. 2, consisting of a symmetric encoder and decoder. The encoder maps the 122-dimensional input through hidden layers of 64 and 32 units to a 16-dimensional bottleneck, and the decoder mirrors this structure to reconstruct the input. Each hidden layer uses a rectified linear activation and dropout for regularisation, and the output layer is linear. Formally, the encoder computes a latent representation

$$z = f_{enc}(x) = \sigma(W_2 \sigma(W_1 x + b_1) + b_2) \quad (2)$$

and the decoder reconstructs the input as

$$\hat{x} = f_{dec}(z) \quad (3)$$

The network is trained to minimise the mean squared reconstruction error over the normal training records

$$L = (1/N) \Sigma \, || \, x_i - \hat{x_i} \, ||^2 \quad (4)$$

IEER
Institute for Excellence in Education & Research

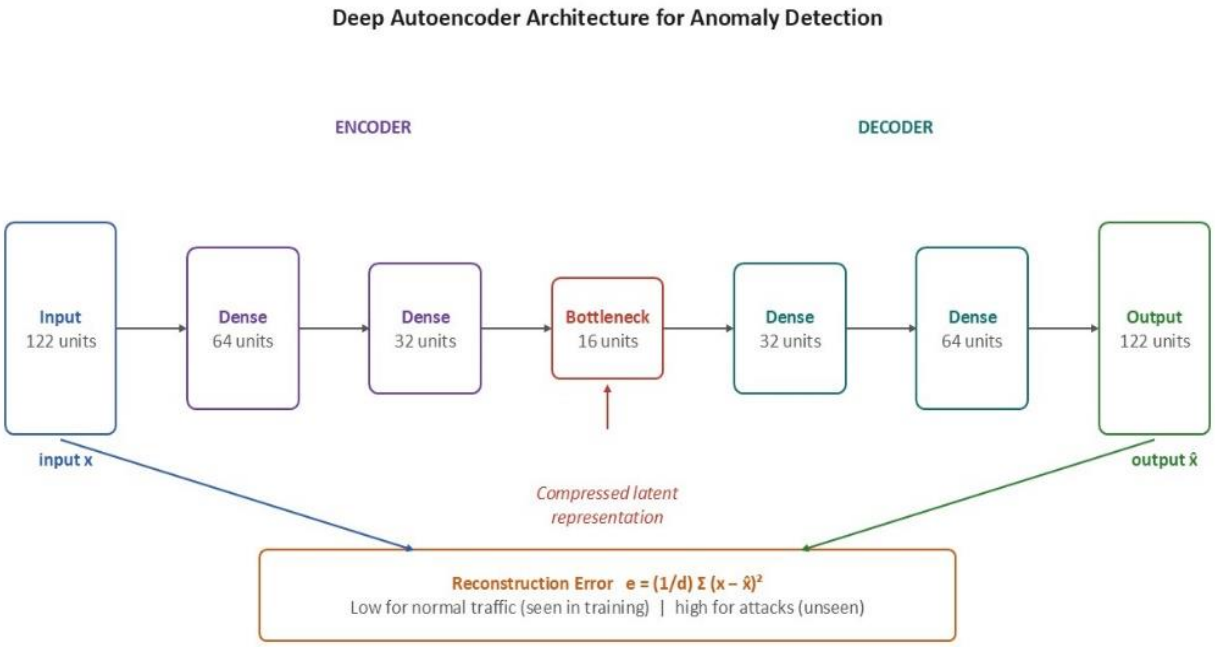


Fig. 2. Deep autoencoder architecture. The encoder compresses the 122-dimensional input to a 16-unit bottleneck and the decoder reconstructs it; the reconstruction error is low for normal traffic seen during training and high for unseen attacks.

3.3 Anomaly Detection

After training, each record is scored by its reconstruction error, the mean squared difference between the input and its reconstruction

$$e(x) = (1/d)\,\Sigma\,(x_j - \hat{x_j})^2 \qquad (5)$$

where d is the number of features. A detection threshold τ is set at the 95th percentile of the reconstruction error on normal training traffic, and a record is flagged as an attack when its error exceeds it

$$y(x) = 1 \;\; if \;\; e(x) > \tau, \;\; else \;\; 0 \qquad (6)$$

The autoencoder is compared against two standard unsupervised baselines fitted on the same normal traffic: Isolation Forest [23], which isolates anomalies by random partitioning, and One-Class SVM [24], which encloses the normal data within a learned boundary. Algorithm 1 summarises the full procedure.

Training uses the Adam optimiser with early stopping on a validation split. Table I lists the configuration.

TABLE I. Autoencoder and experimental configuration.

Parameter	Value
Input features	122 (after one-hot encoding)
Encoder / decoder units	64, 32 / 32, 64
Bottleneck dimension	16
Activation	ReLU (linear output)
Dropout	0.1

Optimiser	Adam
Epochs / batch size	50 (early stopping) / 256
Detection threshold	**95th percentile of normal error**

Algorithm 1: Explainable Autoencoder Intrusion Detection

```
Input: train set X (with labels), test set X_test
Output: detections and SHAP feature attributions
encode categoricals, standardise features
X_normal <- training records labelled normal
train autoencoder AE on X_normal (minimise MSE)
e_normal <- reconstruction error of AE on X_normal
tau <- 95th percentile of e_normal
for each record x in X_test do
    e(x) <- mean squared error of AE on x
    y_hat(x) <- 1 if e(x) > tau else 0   // attack / normal
fit Isolation Forest and One-Class SVM on X_normal (baselines)
evaluate all detectors (Acc, Prec, Rec, F1, ROC-AUC, PR-AUC)
g(x) <- reconstruction-error function of AE
phi <- KernelSHAP(g, background, X_explain)
rank features by mean |phi|; produce local explanations
return detections, phi
```

3.4 SHAP Explanation

To explain the detections, SHAP [13] is applied to the reconstruction-error function of the autoencoder, treating the mapping from input features to reconstruction error as the model to be explained. The SHAP value of feature i is its average marginal contribution to the reconstruction error across all feature coalitions

$$\varphi_i = \Sigma \left[|S|!(M-|S|-1)! / M! \right] \left[g(S \cup \{i\}) - g(S) \right] \quad (7)$$

where g is the reconstruction-error function, M is the number of features, and S is a coalition not containing i. The global importance of feature i is the mean absolute SHAP value over the explained records, and local explanations decompose the error of an individual flagged record into per-feature contributions. Fig. 3 illustrates how detection and explanation combine in operation.

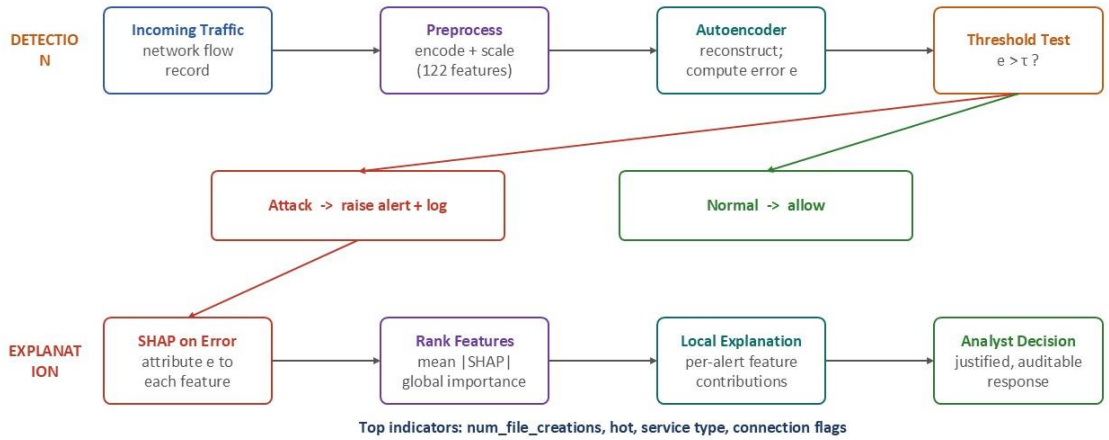


Fig. 3. Intrusion detection and explanation workflow. Incoming traffic is preprocessed, scored by the autoencoder, and thresholded; flagged attacks are explained with SHAP on the reconstruction error, giving analysts the feature-level reasons behind each alert.

Detection quality is measured by accuracy, precision, recall, and F1-score, with the F1-score the harmonic mean of precision and recall

$$F1 = 2 \cdot (Precision \cdot Recall) / (Precision + Recall) \quad (8)$$

and by the areas under the receiver operating characteristic curve (ROC-AUC) and the precision-recall curve (PR-AUC), which assess ranking quality independently of the chosen threshold and are well suited to the class balance of the test set.

4. Results and Discussion

The framework is evaluated on the NSL-KDD test set of 22,544 records, of which 57 percent are attacks. The autoencoder was trained on 67,343 normal records for 50 epochs. Results cover detection performance, the reconstruction-error separation, and SHAP explainability.

4.1 Detection Performance

Table II reports the performance of the autoencoder and the two baselines. The autoencoder achieves the highest ranking quality, with a ROC-AUC of 0.945 and a precision-recall AUC of 0.942, and the highest precision at 0.923, indicating that the records it ranks as most anomalous are reliably attacks. At the fixed 95th-percentile threshold its F1-score is 0.856, close to Isolation Forest at 0.855, while the One-Class SVM attains the highest fixed-threshold F1-score of 0.903 and accuracy of 89.00 percent. This presents an honest and operationally meaningful picture: the autoencoder is the strongest at ranking records by how anomalous they are, as measured by the threshold-independent ROC-AUC and PR-AUC, whereas the One-Class SVM gives the best performance at a single fixed operating point. The autoencoder's F1-score is limited by the choice of threshold rather than by its ranking ability, and its superior ROC-AUC indicates that an appropriately tuned threshold would raise its F1-score. Figs. 5, 6, and 7 present the metric comparison, confusion matrices, and ROC and precision-recall curves.

TABLE II. Detection performance on the NSL-KDD test set. Best value in each column in bold.

Model	Acc (%)	Prec	Rec	F1	AUC	PR-AUC
Autoencoder (proposed)	**84.67**	**0.923**	**0.797**	**0.856**	**0.945**	**0.942**
Isolation Forest	84.13	0.890	0.823	0.855	0.939	0.952

One-Class SVM	89.00	0.903	0.903	0.903	0.929	0.929

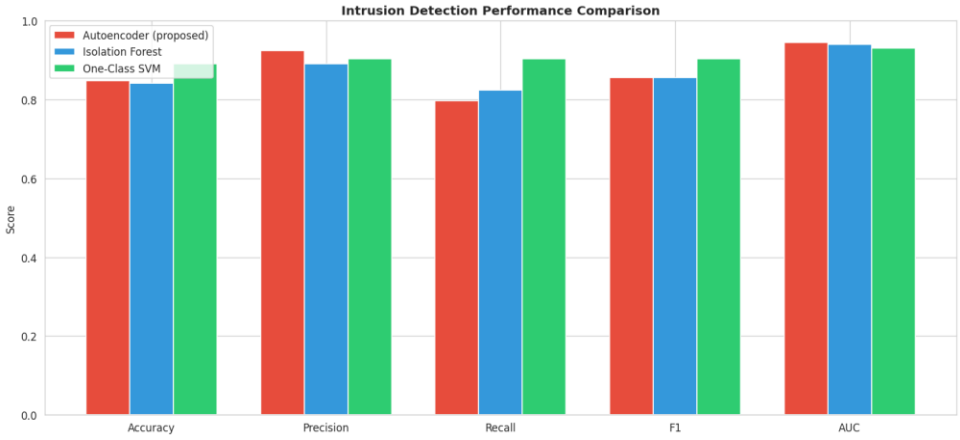


Fig. 5. Detection performance comparison of the autoencoder, Isolation Forest, and One-Class SVM across accuracy, precision, recall, F1-score, and AUC on the NSL-KDD test set.

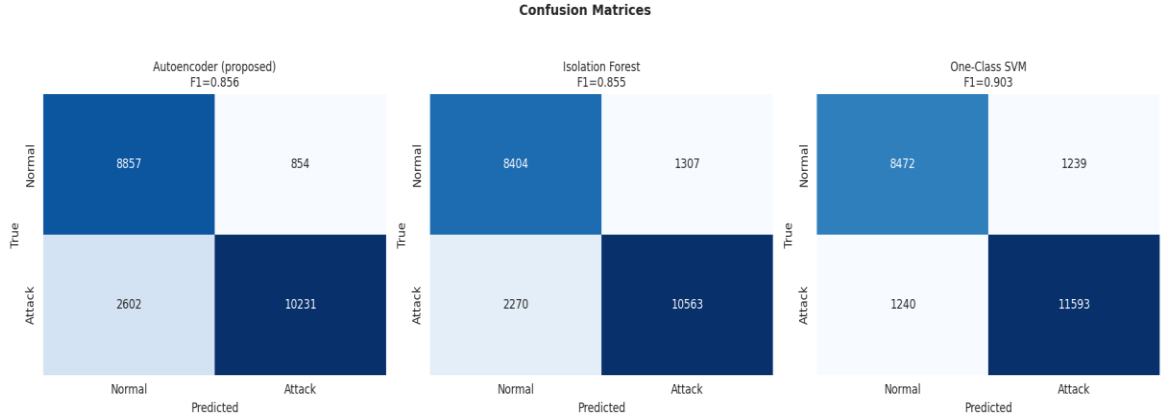


Fig. 6. Confusion matrices on the NSL-KDD test set of 22,544 records. The autoencoder achieves high precision with relatively few false alarms, while the One-Class SVM balances precision and recall.

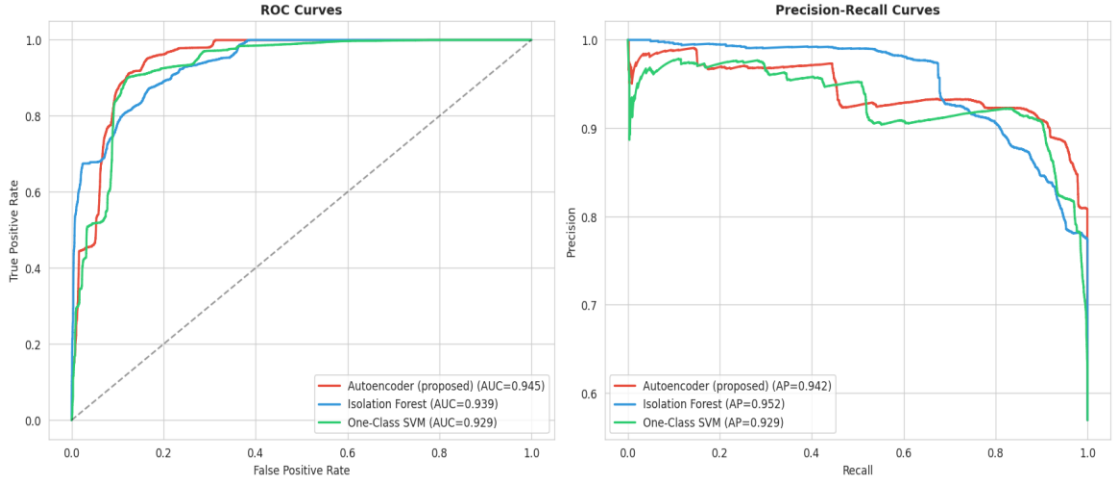


Fig. 7. ROC curves (left) and precision-recall curves (right). The autoencoder attains the highest ROC-AUC of 0.945, confirming the best ranking quality among the three detectors.

4.2 Reconstruction-Error Separation

Fig. 8 shows the autoencoder training curve and Fig. 9 the distribution of reconstruction error on the test set for normal and attack records. The training and validation losses decrease smoothly and converge without overfitting, confirming that the autoencoder learns a stable model of normal traffic. The error distribution shows a clear separation: normal records concentrate at low reconstruction error, while attack records exhibit a long tail of high error, which is exactly the behaviour that makes reconstruction error an effective anomaly score. The detection threshold, set at the 95th percentile of normal error, sits between the two populations, and the modest overlap between them accounts for the residual detection errors.

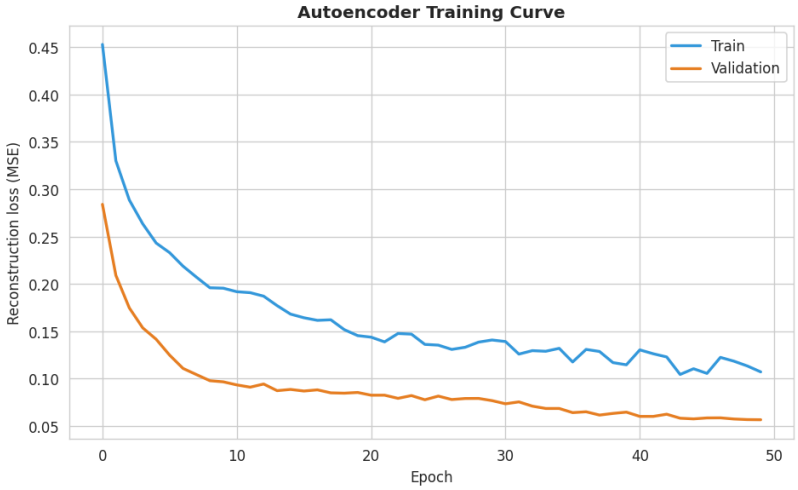


Fig. 8. Autoencoder training and validation reconstruction loss over 50 epochs, converging smoothly without overfitting.

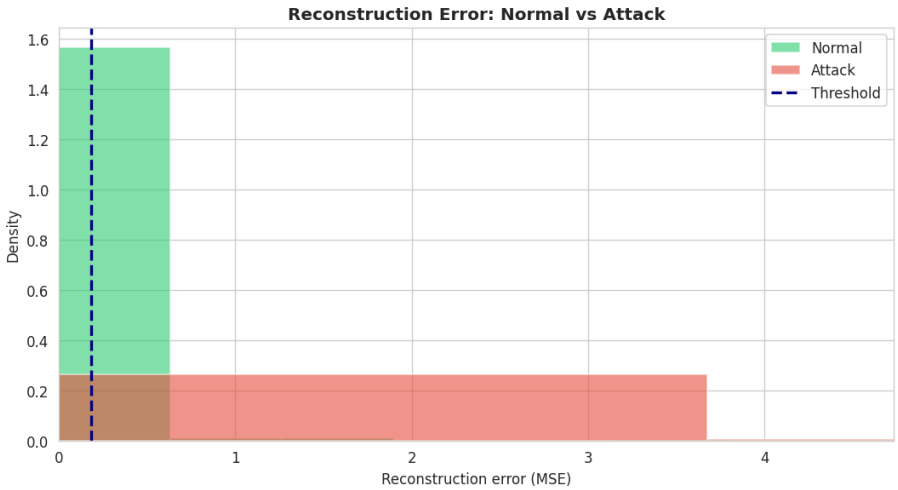


Fig. 9. Distribution of reconstruction error on the test set for normal and attack records, with the detection threshold marked. Normal traffic concentrates at low error while attacks form a high-error tail.

4.3 SHAP Explainability

SHAP was applied to the reconstruction-error function of the autoencoder to explain its detections. Fig. 10 ranks the features by mean absolute SHAP value and Fig. 11 shows the SHAP summary. The features that most drive records toward being flagged as intrusions are the number of file creations, the count of hot indicators, specific services such as the time, pop, and imap services, and connection flags such as RSTO and

RSTR, together with indicators such as guest login and root shell. Table III lists the ten most important features. These indicators are consistent with known attack behaviour: elevated file-creation and root-shell activity are hallmarks of privilege-escalation and compromise, hot indicators capture suspicious access to sensitive files, and abnormal connection flags such as RSTO and RSTR reflect the aborted or reset connections characteristic of scanning and denial-of-service activity. The agreement between the SHAP attributions and established intrusion signatures provides evidence that the autoencoder detects genuinely malicious behaviour rather than spurious artefacts, and the per-feature attributions give analysts the concrete, auditable reasons behind each alert.

TABLE III. Ten most intrusion-indicative features ranked by mean absolute SHAP value.

Feature	Mean |SHAP|
num_file_creations	**0.0565**
hot	0.0443
service_tim_i	0.0362
service_pop_2	0.0355
flag_RSTO	0.0305
service_imap4	0.0208
is_guest_login	0.0154
flag_RSTR	0.0152
service_sunrpc	0.0148
diff_srv_rate	0.0138

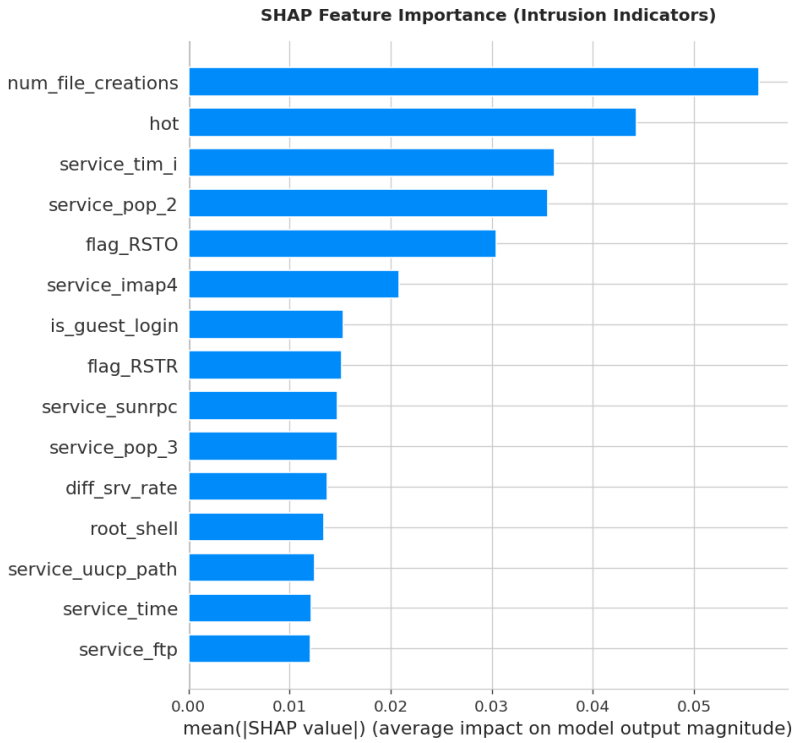


Fig. 10. Global feature importance ranked by mean absolute SHAP value of the reconstruction error, identifying the dominant intrusion indicators.

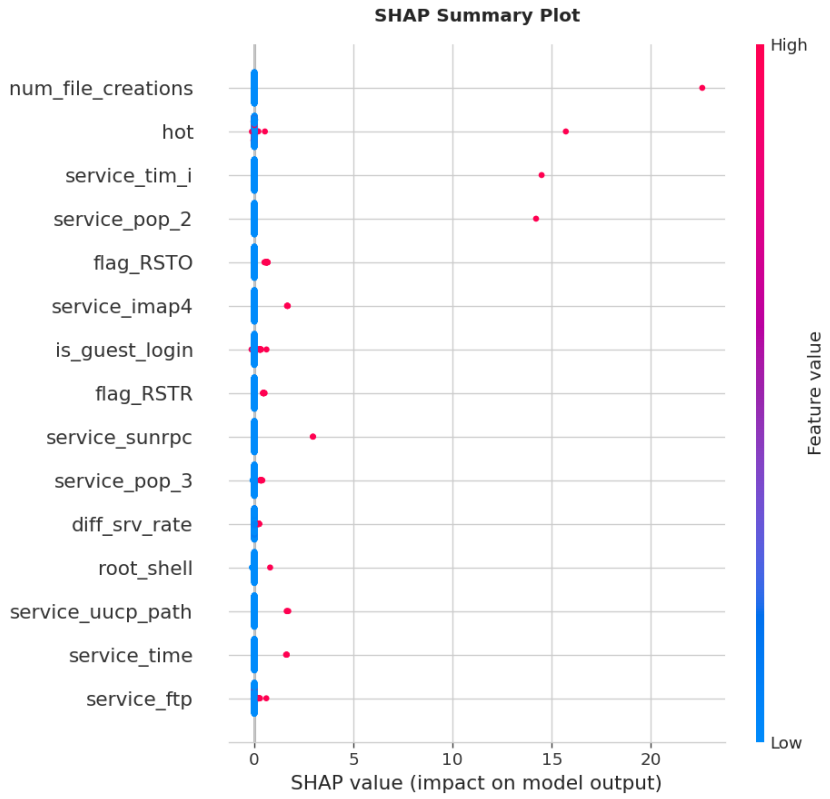


Fig. 11. SHAP summary plot for the reconstruction-error function. Each point is a record; horizontal position gives the feature contribution to the anomaly score and colour gives the feature value from low (blue) to high (red).

4.4 Discussion

The results show that the autoencoder is an accurate and, through SHAP, interpretable intrusion detector. Its superior ROC-AUC and PR-AUC indicate that it ranks records by anomaly more effectively than the baselines, and its high precision means its strongest detections are trustworthy, which is valuable in operational settings where false alarms are costly. The honest trade-off surfaced here, that the One-Class SVM achieves a higher F1-score at the fixed threshold while the autoencoder achieves better threshold-independent ranking, is practically important: it indicates that the autoencoder's operating point can be tuned to improve its F1-score, and that the choice of detector should reflect whether ranking quality or a fixed operating point matters more for the deployment. A limitation of this study is that detection is evaluated on the NSL-KDD benchmark, which, while standard, does not capture the full diversity of modern network traffic, so validation on contemporary traffic such as CIC-IDS datasets is needed. A further consideration is that the detection threshold was set by a single percentile rule; adaptive or cost-sensitive thresholding is likely to improve the fixed-threshold metrics. These considerations motivate the directions outlined in the conclusion.

5. Conclusion and Future Work

This paper presented an explainable deep learning framework for network intrusion detection that combines an unsupervised deep autoencoder with SHAP-based explanation. The autoencoder, trained only on normal NSL-KDD traffic, detects intrusions by reconstruction-error thresholding and thereby identifies attacks unseen during training. Evaluated against Isolation Forest and One-Class SVM baselines, the autoencoder

achieved the best ranking quality with a ROC-AUC of 0.945 and a precision-recall AUC of 0.942, together with the highest precision of 0.923, while the One-Class SVM attained the best fixed-threshold F1-score of 0.903. SHAP applied to the reconstruction-error function identified file-creation activity, host indicators, service types, and connection flags as the dominant intrusion indicators, in agreement with known attack behaviour, and provided per-alert explanations that make the detector transparent and auditable. The framework thus delivers accurate anomaly detection together with the interpretability needed for analyst trust and practical deployment.

Several directions remain for future work. The most immediate is to validate the framework on contemporary intrusion datasets such as the CIC-IDS collections to confirm that the results generalise beyond NSL-KDD. A second direction is to replace the fixed-percentile detection threshold with an adaptive or cost-sensitive criterion that optimises the operating point for the deployment, which is expected to raise the fixed-threshold F1-score of the autoencoder. A third direction is to explore more expressive autoencoder variants, such as variational and denoising autoencoders, to improve the separation between normal and anomalous reconstruction errors. Finally, integrating the SHAP explanations into an analyst-facing security operations interface, presenting the feature-level reasons alongside each alert, would support the practical adoption of explainable deep learning in real intrusion-detection workflows.

REFERENCES

[1] H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013.

[2] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," Computers and Security, vol. 28, no. 1–2, pp. 18–28, 2009.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, 2009.

[4] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Communications Surveys and Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.

[5] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE Symp. Security and Privacy, 2010, pp. 305–316.

[6] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19–31, 2016.

[7] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in Proc. MLSDA Workshop, 2014, pp. 4–11.

[8] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in Proc. ACM SIGKDD, 2017, pp. 665–674.

[9] Z. C. Lipton, "The mythos of model interpretability," Communications of the ACM, vol. 61, no. 10, pp. 36–43, 2018.

[10] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable AI in intrusion detection systems," in Proc. IEEE IECON, 2018, pp. 3237–3243.

[11] A. Nadeem, D. Vos, C. Cao, L. Pajola, S. Dieck, R. Baumgartner, and S. Verwer, "SoK: Explainable machine learning for computer security applications," in Proc. IEEE EuroS&P, 2023, pp. 221–240.

[12] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," IEEE Access, vol. 9, pp. 152379–152396, 2021.

[13] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in Proc. Advances in Neural Information Processing Systems (NIPS), 2017, pp. 4765–4774.
[14] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," Nature Machine Intelligence, vol. 2, no. 1, pp. 56–67, 2020.
[15] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6.
[16] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system," Expert Systems with Applications, vol. 67, pp. 296–303, 2017.
[17] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.
[18] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," IEEE Access, vol. 6, pp. 50850–50859, 2018.
[19] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, 2018.
[20] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," arXiv preprint arXiv:1901.03407, 2019.
[21] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in Proc. Network and Distributed System Security Symposium (NDSS), 2018.
[22] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in Proc. IEEE ICACT, 2018, pp. 178–183.
[23] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2008, pp. 413–422.
[24] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Computation, vol. 13, no. 7, pp. 1443–1471, 2001.
[25] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," IEEE Access, vol. 8, pp. 73127–73141, 2020.

IEER
Institute of Excellence in Education & Research