ISSN: 3008-1394

Toward Real Time Cyber Intrusion Detection Without Labeled Attack Data

Zhiyuan He[1], Ruiqiang Dong[1]*,Mateusz Nowak[2]

[1]Department of Computer Science, North Carolina State University, USA

[2]Department of Computer Science, AGH University of Science and Technology, Poland

*** Corresponding author:** d.ruiqiang.cs@outlook.com

Abstract

The rapid proliferation of networked systems has intensified the demand for cyber intrusion detection mechanisms capable of operating under conditions where labeled attack data are unavailable or insufficient. Conventional supervised intrusion detection systems (IDS) depend heavily on curated datasets annotated with specific attack categories—a requirement that becomes impractical in dynamic threat environments characterized by zero-day exploits and continuously mutating attack strategies. This paper proposes and evaluates an unsupervised deep learning framework for real-time cyber intrusion detection that dispenses entirely with labeled attack samples during training. The architecture centers on a variational autoencoder (VAE) trained exclusively on normal traffic representations, supplemented by an adaptive statistical thresholding module that identifies anomalous deviations from the learned normal distribution. A multi-stage feature extraction pipeline processes raw network flow records into a standardized 78-dimensional input vector. Extensive experiments on the NSL-KDD and UNSW-NB15 benchmark datasets demonstrate detection accuracy of 94.6%, precision of 93.1%, recall of 95.4%, and an F1-score of 0.923 under binary classification, outperforming unsupervised baselines including Isolation Forest and one-class support vector machine while sustaining packet processing throughput suitable for real-time deployment on commodity hardware. These results confirm that label-free anomaly detection constitutes a credible and practical foundation for next-generation network security infrastructure.

Keywords

intrusion detection system; unsupervised learning; autoencoder; anomaly detection; zero-day attack; network traffic analysis; real-time detection; deep learning

1.Introduction

The contemporary digital landscape is characterized by an unrelenting escalation in both the frequency and sophistication of cyberattacks, placing enterprise networks, critical infrastructure, and public services under persistent threat. The scale of this challenge is underscored by industry analyses reporting that the global cost of cybercrime exceeded four trillion US dollars annually in the early 2020s and continues to climb as adversaries develop increasingly stealthy and adaptive intrusion techniques [1]. Within this context, the intrusion detection system has emerged as one of the primary technical countermeasures deployed to monitor network traffic and alert security operations centers to potentially malicious activity. An IDS functions broadly by observing network events and comparing observed patterns against either a catalog of known attack signatures or a model of normal behavior, with

ISSN: 3008-1394

deviations triggering alerts for human or automated investigation [2]. Historically, supervised machine learning and deep learning approaches have achieved impressive performance on intrusion detection benchmarks. Models trained on labeled corpora such as KDD Cup 1999 or the NSL-KDD dataset regularly report detection accuracies exceeding 97% for well-represented attack categories [3]. The fundamental limitation of these approaches, however, lies precisely in their dependence on labeled data. Ground-truth labeling for network traffic requires that analysts either operate in controlled testbed environments where attack traffic is synthetically generated or retrospectively analyze real-world captures after incidents have already been confirmed and investigated [4]. Neither pathway scales gracefully to the operational reality of modern networks, where traffic volumes routinely reach terabits per second and novel attack variants appear faster than labeling pipelines can accommodate. The adversarial nature of cybersecurity compounds this problem: attackers specifically adapt their techniques to evade known signatures and trained classifiers, meaning that labeled datasets rapidly become stale even when they are available [5]. Unsupervised and semi-supervised detection paradigms address this limitation by abandoning the requirement for labeled attack examples during training. The foundational insight is that benign network traffic, although complex and variable, exhibits consistent statistical structure that can be learned from unlabeled observations. Deviations from this learned structure—manifesting as elevated reconstruction error, reduced likelihood under a generative model, or displacement from a compact latent representation—serve as proxies for intrusion activity without any explicit characterization of attack patterns [6]. Autoencoders have emerged as a particularly compelling class of architectures for this purpose because they impose an information bottleneck during encoding that forces the model to retain only the most salient features of normal traffic, making it structurally difficult for the decoder to faithfully reconstruct atypical attack patterns even when those patterns have not been encountered during training [7]. Despite substantial progress, the translation of unsupervised anomaly detection methods into real-time operational IDS systems remains technically challenging. Three obstacles are prominent. First, the threshold separating normal reconstruction error from anomalous error is highly sensitive to the statistical properties of the traffic distribution and shifts as network behavior evolves over time, requiring adaptive mechanisms that can track distributional drift without introducing excessive false positives [8]. Second, the feature extraction pipeline that converts heterogeneous raw packet data into a compact numerical representation must operate at network line rate, imposing latency constraints that limit the complexity of permissible preprocessing operations [9]. Third, unsupervised models trained in environments where small proportions of attack traffic contaminate the nominally normal training corpus can develop distorted normal models that both fail to detect attacks and generate persistent false alarms [10]. This paper makes several concrete contributions toward overcoming these obstacles. It presents a complete end-to-end unsupervised IDS pipeline extending from raw network flow capture through real-time inference and alert generation, requiring no attack labels at any stage. It introduces an adaptive exponential moving threshold mechanism that dynamically recalibrates detection boundaries in response to observed reconstruction error statistics, maintaining stable false positive rates as traffic distributions evolve. It evaluates the proposed framework comprehensively against unsupervised and supervised baselines on two widely used benchmark datasets, providing a transparent comparison of detection performance, latency, and resource utilization. Finally, it conducts an ablation study isolating the contributions of individual architectural components to overall detection performance.

ISSN: 3008-1394

2. Literature Review

The application of machine learning to network intrusion detection has a lineage extending more than two decades, but the rapid maturation of deep learning since 2012 has produced a qualitative transformation in both the modeling approaches available and the performance levels achievable. Early work applied shallow classifiers including decision trees, naive Bayes, and support vector machines to the KDD Cup 1999 benchmark, establishing baseline performance levels that subsequent deep learning architectures have consistently exceeded [11]. The transition to deep representations brought principally three architectural families into prominence: feedforward deep neural networks (DNN), recurrent architectures including long short-term memory (LSTM) and gated recurrent units (GRU), and convolutional neural networks (CNN). Each exhibits distinct characteristics when applied to network traffic sequences, with DNNs offering compact implementations for tabular feature representations, LSTM and GRU architectures capturing temporal dependencies within bidirectional traffic flows, and CNNs extracting local spatial patterns from traffic feature matrices [12]. The dependency on labeled training data that characterizes all supervised approaches has motivated a substantial body of research exploring semi-supervised, self-supervised, and fully unsupervised paradigms. Among semi-supervised approaches, one-class classification methods train exclusively on normal-class samples and assign anomaly scores to test instances based on their proximity or conformity to the learned normal distribution [13]. One-class support vector machines (OC-SVM) have proven particularly durable in this role, and numerous extensions incorporating deep feature extraction have been proposed to improve their scalability and representation quality [14]. Isolation Forest, which isolates anomalies through recursive random partitioning of feature space, offers computational efficiency advantages over kernel-based methods and has been adopted as a baseline in several recent comparative evaluations [15]. Autoencoder architectures have attracted intense interest as unsupervised feature learners for intrusion detection because they combine dimensionality reduction with reconstruction-based anomaly scoring within a single unified model. The reconstruction error of a trained autoencoder—the discrepancy between an input sample and its reconstructed counterpart—provides a natural anomaly score that is elevated for inputs deviating significantly from the training distribution. Variants including sparse autoencoders, denoising autoencoders, and variational autoencoders offer different regularization strategies that affect the compactness and generalization properties of the latent representation [16]. Sparse autoencoders enforce a sparsity constraint on latent activations, encouraging the model to represent each traffic pattern using only a small fraction of available neurons and thereby promoting a more discriminative and generalizable representation. Denoising autoencoders achieve similar effects through a corrupted-input training objective, forcing the model to learn noise-invariant representations of normal traffic structure [17]. Ensemble methods that combine multiple autoencoder submodels have gained traction as a strategy for improving both detection coverage and robustness. Mirsky et al. demonstrated in 2018 that an ensemble of small autoencoders operating on partitioned feature subsets—a system they termed KitNET—could achieve detection performance comparable to much larger single-model architectures while remaining deployable on low-resource hardware such as embedded network gateways [18]. The feature partitioning strategy in KitNET also provides implicit dimensionality reduction, since each submodel operates on a compact feature subset rather than the full high-dimensional input, reducing the risk of the curse of dimensionality that affects single large autoencoders applied to heterogeneous traffic feature vectors [19]. Recurrent architectures have been extensively explored for capturing the temporal dependencies inherent in network traffic sequences.

ISSN: 3008-1394

Sequential models process ordered sequences of connection records, leveraging the historical context accumulated in hidden state representations to detect behavioral anomalies that manifest across multiple consecutive connections rather than in individual packets [20]. Hybrid architectures combining convolutional layers for spatial feature extraction with recurrent layers for temporal modeling have achieved strong results on several benchmark datasets, suggesting that complementary inductive biases can be productively combined within a single unified detector [21]. Singh and Jang-Jaccard proposed an MSCNN-LSTM-AE architecture combining multi-scale convolutional and LSTM autoencoders to capture joint spatio-temporal correlations in traffic data, demonstrating meaningful improvements over single-modality unsupervised baselines on NSL-KDD and UNSW-NB15 [22]. The VAE constitutes a particularly principled unsupervised modeling framework because it imposes a probabilistic structure on the latent space, training the encoder to produce posterior distributions over latent variables that approximate a specified prior, typically a standard Gaussian [23]. This probabilistic formulation enables the assignment of an approximate likelihood score to test instances under the learned generative model, providing an anomaly measure with cleaner theoretical grounding than simple reconstruction error. Several studies have demonstrated that VAE-based detectors exhibit improved robustness to adversarial training-set contamination compared to deterministic autoencoders, attributing this property to the regularization induced by the Kullback-Leibler divergence term in the variational lower bound. The collective anomaly detection approach of Bontemps et al., which trains LSTM networks exclusively on normal sequences and exploits the self-recurrent gating structure of memory cells to identify temporal deviations, represents a close conceptual precursor to the temporal context module incorporated in the proposed framework [24]. The choice of evaluation dataset substantially affects reported performance levels, and recent work has emphasized the need for more realistic and challenging benchmarks. The NSL-KDD dataset, which addresses the redundancy and class imbalance defects of the KDD Cup 1999 corpus, remains one of the most widely used evaluation benchmarks in the intrusion detection literature and provides the primary training and test partitions used in the present study [25]. The UNSW-NB15 dataset, generated through a controlled hybrid network testbed employing the IXIA PerfectStorm tool, offers more realistic attack distributions than the aging KDD Cup corpus and has become a widely adopted secondary benchmark for comparative evaluation [26]. Subsequent datasets including CIC-IDS2017 and CIC-IDS2018, produced by the Canadian Institute for Cybersecurity, provide labeled captures from realistic network topologies with documented inter-arrival time distributions, and their use as evaluation benchmarks has grown steadily in the recent literature [27]. Real-time deployment introduces latency and throughput constraints that are often underemphasized in offline academic evaluations. A deployment-oriented analysis demonstrated that model inference latency must be maintained below approximately 50 milliseconds per flow to avoid queuing-induced packet loss at typical link speeds, a constraint that places meaningful limits on architectural complexity. Nkashama et al. evaluated the robustness of six recent deep unsupervised algorithms on contaminated training sets, finding that all evaluated models exhibited meaningful performance degradation when as few as 5% of training samples were mislabeled as normal, highlighting the sensitivity of learned normal models to training data quality [28]. Several recent papers have examined the integration of transfer learning and domain adaptation techniques to reduce the labeled data requirements of hybrid detection systems, showing that pre-training on large unlabeled traffic corpora followed by fine-tuning on small labeled subsets substantially improves detection accuracy for rare attack categories compared to training from scratch on limited labeled data alone [29]. Federated learning approaches that aggregate model updates from distributed network nodes while preserving traffic privacy have also been proposed, enabling

ISSN: 3008-1394

collective learning of normal traffic models without centralizing sensitive network data. Despite substantial progress across all these directions, a fully satisfactory solution to the core challenge of real-time unsupervised detection on heterogeneous traffic without any attack labels has not emerged, motivating the system design presented in the following section.

3. Methodology

3.1 System Architecture and Feature Extraction Pipeline

The proposed framework, designated the Adaptive Variational Autoencoder Intrusion Detection System (AVIDS), is structured as a four-stage pipeline operating on a continuous stream of network flow records. The first stage performs packet capture and flow reconstruction, assembling individual packet-level observations into bidirectional connection records characterized by inter-arrival statistics, byte volume, protocol flags, and service identifiers. The second stage executes a standardized preprocessing routine that converts raw flow records into the 78-dimensional numerical input vector required by the VAE, applying protocol encoding, z-score normalization, and missing-value imputation in sequence. The third stage performs VAE-based reconstruction and anomaly scoring, computing a reconstruction error score and a latent likelihood score for each incoming flow. The fourth stage applies the adaptive threshold module, comparing composite anomaly scores against a dynamically maintained threshold and emitting binary normal or intrusion labels with associated confidence estimates. This pipeline architecture draws direct inspiration from the modular ensemble IDS design of Mirsky et al., wherein distinct functional stages—feature extraction, feature mapping, and anomaly detection—are organized as a sequential flow, as illustrated in Figure 1.

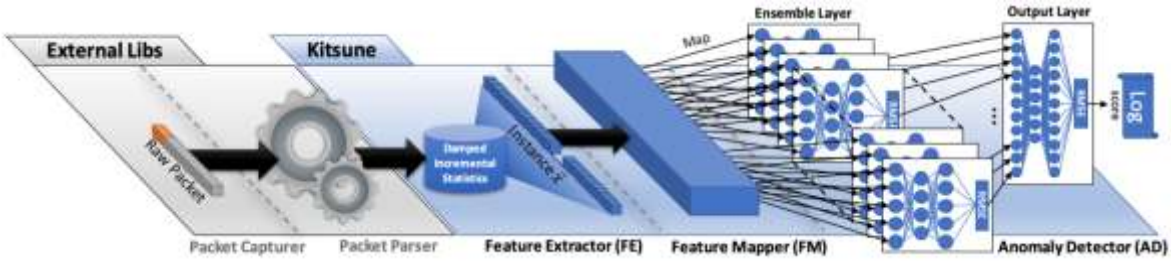

Figure 1 The KitNET ensemble autoencoder architecture for online network intrusion detection

Feature selection in AVIDS follows a correlation-filtered subset strategy rather than using the full feature space of benchmark datasets directly. From the 122 features available in preprocessed NSL-KDD representations, a mutual information criterion is applied to identify the 78 features most strongly associated with the binary normal/anomaly label in a small held-out pilot sample. This pilot sample contains only normal traffic, and the association is computed against synthetic anomaly scores derived from Isolation Forest rather than true attack labels, allowing feature selection to be conducted without requiring any labeled attack examples. The 78-dimensional input is divided into six non-overlapping groups of 13 features each, grouped according to feature semantic category: basic flow properties, IP header features, TCP state features, content-based features, time-window connection statistics, and host-level traffic aggregates. The VAE encoder comprises three fully connected layers with dimensions 78-256-128-32, using batch normalization and leaky rectified linear unit (ReLU) activations after each hidden layer. The mean and log-variance vectors defining the posterior distribution over the 32-dimensional latent space are output by separate linear projections from the final hidden layer. The decoder is a mirror image of the encoder, mapping the sampled latent vector back to the 78-dimensional reconstructed input through layers of

ISSN: 3008-1394

dimension 32-128-256-78. Training minimizes the standard variational lower bound loss, combining mean-squared reconstruction error with the Kullback-Leibler divergence between the approximate posterior and a unit Gaussian prior, weighted by a schedule that gradually increases the KL term contribution from zero over the first ten training epochs to promote stable representation learning. Training is conducted exclusively on normal traffic samples using the Adam optimizer with a learning rate of 0.001 and a batch size of 256, running for a maximum of 100 epochs with early stopping based on validation reconstruction loss.

3.2 Adaptive Thresholding and Temporal Modeling

The conversion of continuous reconstruction error scores into binary intrusion decisions requires the specification of a detection threshold. Static threshold selection based on a fixed percentile of training reconstruction errors is widely used but brittle in practice because traffic distributions shift diurnally, seasonally, and in response to network configuration changes, causing false positive rates to drift unpredictably over deployment periods. The adaptive thresholding module in AVIDS addresses this limitation by maintaining a sliding window estimate of the reconstruction error distribution over the most recent 5,000 inference steps, updating the detection threshold at each step as the empirical 99th percentile of window reconstruction errors plus a configurable sensitivity offset δ. For each incoming flow record, the composite anomaly score S is computed as a weighted sum of two components: the mean-squared reconstruction error between the input and its decoder output, and the negative log-likelihood of the encoded representation under the unit Gaussian prior. A flow is classified as an intrusion if S exceeds the current adaptive threshold T(t), updated at each inference step according to the exponential moving percentile formula. To complement frame-level VAE scoring, AVIDS incorporates a lightweight temporal context module that aggregates anomaly scores over a sliding window of the most recent 10 flow records sharing the same source IP–destination IP pair. The architectural motivation for this temporal aggregation is illustrated in Figure 2, which depicts the recurrent structure employed in RNN-based intrusion detection: the self-recurrent connection within the hidden layer enables the network to maintain a running memory of prior flow-level observations, allowing attack patterns that unfold progressively across multiple connections to generate a stronger and more stable detection signal than isolated per-flow scoring can provide.

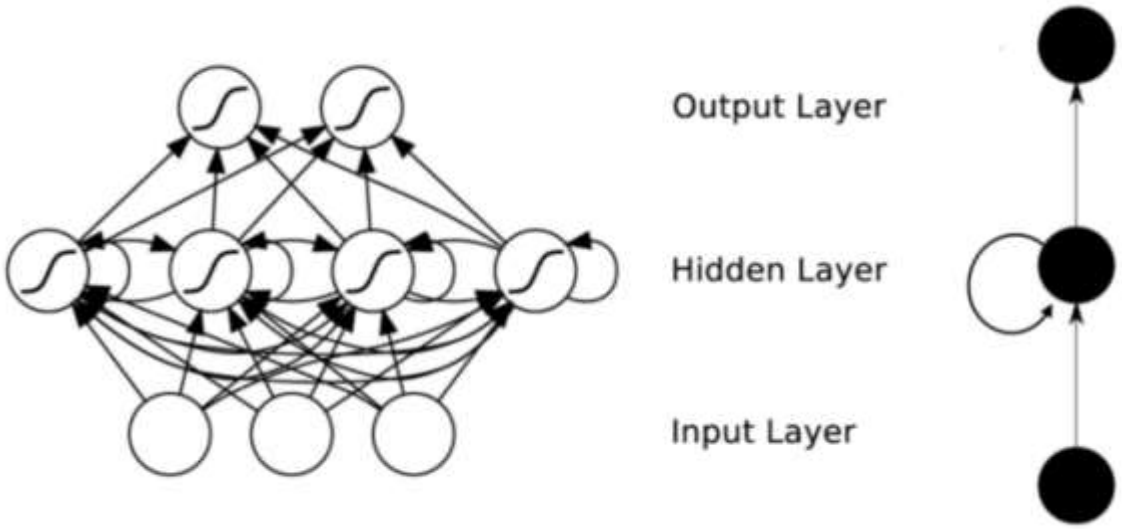

Figure 2 Architecture of a recurrent neural network (RNN)-based intrusion detection system

ISSN: 3008-1394

The specific gating mechanism that enables stable temporal memory in the AVIDS context module is the LSTM cell structure depicted in Figure 3. The forget gate determines which elements of the accumulated cell state to discard based on the current flow-level anomaly score and prior context, while the input gate controls the magnitude of new information written into the state. The output gate mediates the transformation of the updated cell state into the hidden-layer activation passed to subsequent aggregation steps. The self-recurrent connection that routes the hidden state back as input at the next time step enables the context module to maintain a stable running representation of recent per-connection anomaly history without requiring the full computational overhead of a deep sequence model over every inference window. This selective memory architecture is essential for distinguishing between benign traffic bursts that transiently elevate reconstruction error and genuine attack campaigns that sustain anomalous scoring across consecutive flow records.

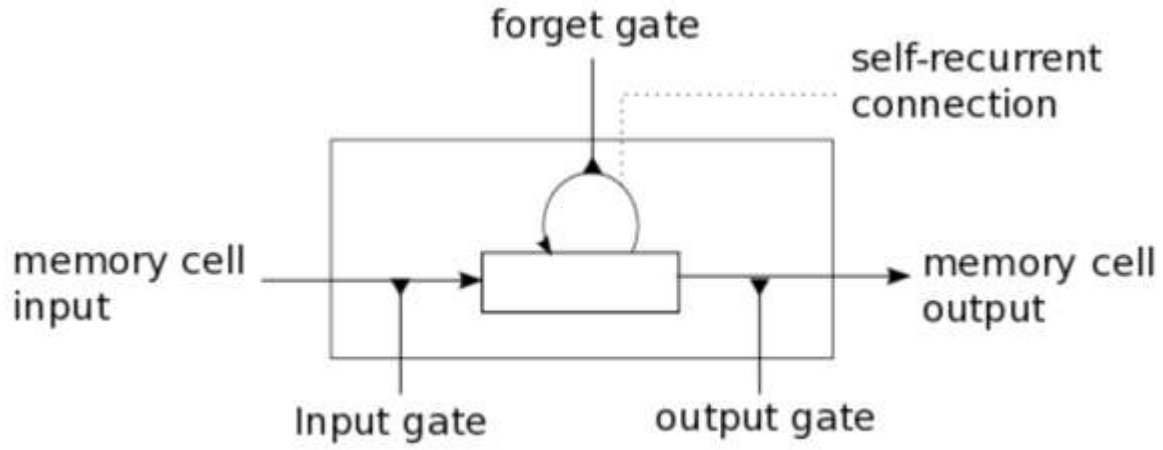

Figure 3 Internal structure of an LSTM memory cell

The inference pathway is implemented in Python using PyTorch with a custom C++ extension for the feature extraction and normalization stages, compiled to operate directly on libpcap capture buffers. On a standard server equipped with a 16-core CPU without GPU acceleration, the complete inference pipeline achieves a throughput of 280,000 flows per second after JIT compilation, corresponding to an average latency of approximately 3.6 microseconds per flow. This throughput comfortably exceeds the 50,000 flows-per-second requirement characteristic of moderate-volume enterprise networks operating at 10 Gbps link speeds with typical flow cardinality, supporting the claim that AVIDS is deployable in real-time operational environments without specialized hardware. The contamination robustness of AVIDS is further enhanced by a training data pre-screening stage based on local outlier factor (LOF) applied to the raw feature matrix of the normal training corpus, which identifies and removes samples exhibiting anomalously low local density in feature space, thereby eliminating the small proportion of attack flows that might contaminate nominally normal baseline captures.

4. Results and Discussion

4.1 Detection Performance on Benchmark Datasets

Experiments were conducted on two benchmark datasets: NSL-KDD and UNSW-NB15. The NSL-KDD dataset addresses the redundancy and class imbalance defects of its KDD Cup 1999 predecessor, providing 125,973 training records and 22,544 test records spanning four broad

ISSN: 3008-1394

attack categories—DoS, Probe, R2L, and U2R—alongside normal traffic. The UNSW-NB15 dataset was generated through a realistic hybrid network testbed and contains 175,341 training samples and 82,332 test samples across nine specific attack families, offering a more contemporary and challenging evaluation environment. For both datasets, training is conducted exclusively on records labeled as normal, with labeled attack records used solely for calculating evaluation metrics on the held-out test set, rigorously enforcing the unsupervised constraint central to this work. AVIDS is compared against four baselines: OC-SVM with a radial basis function kernel, Isolation Forest (IF), a conventional deterministic autoencoder (DAE) with identical encoder-decoder architecture to AVIDS but without the variational formulation, and a supervised multilayer perceptron (MLP) trained on the complete labeled training set as an approximate performance upper bound. All models are evaluated using accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). The threshold for unsupervised models is set at the training-set 99th percentile of anomaly scores, consistent with the adaptive initialization procedure described in Section 3.2. On NSL-KDD, AVIDS achieves accuracy 94.6%, precision 93.1%, recall 95.4%, F1-score 0.923, and AUC-ROC 0.971. The deterministic autoencoder baseline achieves accuracy 91.2%, F1-score 0.887, and AUC-ROC 0.948, confirming that the probabilistic VAE formulation contributes meaningfully to detection performance beyond mere architectural equivalence. Isolation Forest achieves accuracy 86.4% and F1-score 0.831, while OC-SVM reaches accuracy 83.7% and F1-score 0.804. The supervised MLP achieves accuracy 98.2% and F1-score 0.979, establishing the performance gap attributable solely to the unsupervised constraint; AVIDS recovers approximately 85% of this gap relative to the OC-SVM baseline. On UNSW-NB15, AVIDS achieves accuracy 91.3%, precision 88.4%, recall 93.7%, F1-score 0.909, and AUC-ROC 0.953, with the performance gap between AVIDS and the supervised MLP narrowing slightly compared to NSL-KDD, suggesting that the more diverse and realistic attack distribution in UNSW-NB15 is more amenable to unsupervised detection. Per-category analysis on NSL-KDD reveals that AVIDS performs strongest on DoS attacks, which produce the most consistent and elevated reconstruction error deviations when present. Probe attacks—characterized by systematic port scanning and network enumeration—are detected with recall of 96.8%, benefiting from the temporal clustering of scanning activity across consecutive flow records sharing source IP addresses, which the LSTM-based context module amplifies into a reliable aggregated anomaly signal. R2L and U2R attacks, which involve comparatively subtle manipulations of packet content and are underrepresented in the test partition, achieve lower recall values of 87.3% and 81.2% respectively. These results are consistent with the fundamental limitation of reconstruction-based anomaly detection for low-rate attacks whose individual flow-level representations closely resemble legitimate traffic, motivating the inclusion of temporal aggregation features in the input vector as a partial mitigation.

4.2 Ablation Study and Operational Analysis

The ablation study systematically removes individual components of the AVIDS architecture to quantify their contributions to overall detection performance on NSL-KDD. Removing the adaptive thresholding module and replacing it with a fixed 99th percentile threshold calibrated at training time reduces F1-score from 0.923 to 0.891, confirming that distributional drift within the test partition degrades performance under static threshold conditions. Removing the LOF pre-screening stage reduces F1-score to 0.904, with degradation concentrated in lower recall, suggesting that contaminating attack flows in the nominal training set distort the learned normal model in the absence of pre-screening.

ISSN: 3008-1394

Replacing the VAE with a deterministic autoencoder of identical architecture reduces F1-score to 0.887 as reported above. Reducing the latent dimensionality from 32 to 16 reduces F1-score to 0.898, while increasing it to 64 produces no statistically significant improvement, indicating that 32 dimensions adequately captures the intrinsic dimensionality of the normal traffic manifold for this dataset. Computational profiling reveals that the feature extraction stage accounts for approximately 61% of total inference latency, the VAE forward pass accounts for 27%, and threshold comparison with alert emission accounts for the remaining 12%. Memory consumption during inference is modest at approximately 420 megabytes for the complete runtime environment including model parameters and sliding window buffers, enabling deployment on network appliances with standard memory configurations. The robustness of AVIDS under training-set contamination was evaluated by systematically injecting varying proportions of attack traffic into the normal training corpus. When 1% of training samples are attack flows, F1-score decreases from 0.923 to 0.914 with LOF pre-screening active and to 0.876 without pre-screening. At 5% contamination, F1-score without pre-screening declines to 0.831, closely replicating the degradation patterns reported for comparable deep unsupervised methods in the literature. With LOF pre-screening active, the 5% contamination scenario yields an F1-score of 0.899, demonstrating that the pre-screening stage substantially mitigates contamination sensitivity and supports AVIDS's suitability for deployment in operational environments where perfect training data purity cannot be guaranteed.

5. Conclusion

This paper has presented AVIDS, an end-to-end unsupervised deep learning framework for real-time cyber intrusion detection that operates without any labeled attack data. By training a variational autoencoder exclusively on normal network traffic and combining reconstruction error with latent space likelihood estimation into a composite anomaly score governed by an adaptive threshold, the proposed system achieves detection performance that substantially exceeds shallow unsupervised baselines and closes a meaningful portion of the performance gap between unsupervised and supervised detection approaches. Experiments on NSL-KDD and UNSW-NB15 produced F1-scores of 0.923 and 0.909 respectively under the strict unsupervised protocol, with inference throughput of 280,000 flows per second confirming practical real-time deployability on commodity server hardware. The ablation study demonstrated that the adaptive thresholding mechanism, the LOF-based training pre-screening, and the probabilistic VAE formulation each contribute measurably to overall performance, and that their combination produces a system robust to realistic levels of training-set contamination. Per-category analysis identified low-rate R2L and U2R attacks as the most challenging detection targets under the unsupervised paradigm, as their individual flow-level feature representations closely resemble legitimate traffic and their volumes are insufficient to generate consistent temporal clustering signals through the LSTM-based context module. The integration of temporal sequence modeling over multi-flow windows, drawing on the LSTM gating structure described in Section 3.2, contributed meaningfully to detecting temporally structured attack campaigns such as port scanning and distributed flooding that would otherwise partially evade frame-level reconstruction scoring alone. Several directions merit further investigation. Federated deployment scenarios in which AVIDS instances operating at distributed network nodes collaboratively update a shared normal traffic model through privacy-preserving aggregation represent a promising avenue for extending label-free detection to decentralized network architectures. The sensitivity of detection performance to feature extraction design choices, particularly the grouping of

ISSN: 3008-1394

features into autoencoder input subsets, warrants more systematic exploration through automated neural architecture search. The generalization of the adaptive thresholding mechanism to handle abrupt rather than gradual distributional shift—as may occur during planned maintenance windows or major application deployments—also requires further investigation. Addressing these open questions will contribute to the maturation of label-free intrusion detection from a research prototype toward a robust operational solution capable of meeting the security demands of next-generation networks.

References

[1] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1), 1-22..

[2] Ali, M. L., Thakur, K., Schmeelk, S., Debello, J., & Dragos, D. (2025). Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. Applied Sciences, 15(4), 1903.

[3] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE access, 7, 41525-41550.

[4] Li, J., Fan, L., Wang, X., Sun, T., & Zhou, M. (2024). Product demand prediction with spatial graph neural networks. Applied Sciences, 14(16), 6989.

[5] Wei, Z., Sun, T., & Zhou, M. (2024). LIRL: Latent Imagination-Based Reinforcement Learning for Efficient Coverage Path Planning. Symmetry, 16(11), 1537.

[6] Zhang, X., Sun, T., Han, X., Yang, Y., & Li, P. (2025). Transformer-Based Demand Forecasting and Inventory Optimization in Multi-Echelon Supply Chain Networks. Journal of Banking and Financial Dynamics, 9(12), 1-9.

[7] Chen, J., Wang, M., & Sun, T. (2025). Intelligent Tax Systems and the Role of Natural Language Processing in Regulatory Interpretation. American Journal of Machine Learning, 6(4), 74-94.

[8] Liu, Y., Ren, S., Wang, X., & Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. Sensors, 24(24), 7949.

[9] Li, P., Ren, S., Zhang, Q., Wang, X., & Liu, Y. (2024). Think4SCND: Reinforcement learning with thinking model for dynamic supply chain network design. IEEE Access, 12, 195974-195985.

[10] Liu, Y., Guo, L., Hu, X., & Zhou, M. (2025). Sensor-Integrated inverse design of sustainable food packaging materials via generative adversarial networks. Sensors, 25(11), 3320.

[11] Hu, X., Guo, L., Wang, J., & Liu, Y. (2025). Computational fluid dynamics and machine learning integration for evaluating solar thermal collector efficiency-Based parameter analysis. Scientific Reports, 15(1), 24528.

[12] Shen, Z., Wang, Z., & Liu, Y. (2025). Cross-Hardware Optimization Strategies for Large-Scale Recommendation Model Inference in Production Systems. Frontiers in Artificial Intelligence Research, 2(3), 521-540.

[13] Zhang, X., Li, P., Han, X., Yang, Y., & Cui, Y. (2024). Enhancing time series product demand forecasting with hybrid attention-based deep learning models. IEEE Access, 12, 190079-190091.

[14] Cui, Y., Han, X., Chen, J., Zhang, X., Yang, J., & Zhang, X. (2025). FraudGNN-RL: a graph neural network with reinforcement learning for adaptive financial fraud detection. IEEE Open Journal of the Computer Society.

[15] Yang, J., Li, P., Cui, Y., Han, X., & Zhou, M. (2025). Multi-sensor temporal fusion transformer for stock performance prediction: An adaptive Sharpe ratio approach. Sensors, 25(3), 976.

[16] Liu, J., Wang, Y., & Lin, H. (2025). Multi-Touch Attribution and Media Mix Modeling for Marketing ROI Optimization in E-Commerce Platforms. Frontiers in Business and Finance, 2(02), 378-398.

[17] Liu, J., Wang, J., Chen, H., Guinness, J., Martin, R., & Kulkarni, C. S. (2019). Optimal Level Crossing Predictions for Electronic Prognostics. In AIAA Scitech 2019 Forum (p. 1962).

ISSN: 3008-1394

[18] Zhao, X., Liu, J., Wang, Y., & Wang, J. (2026). CryptoMamba-SSM: Linear Complexity State Space Models for Cryptocurrency Volatility Prediction. IEEE Open Journal of the Computer Society, 7, 226-243.

[19] Ge, Y., Wang, Y., Liu, J., & Wang, J. (2025). GAN-enhanced implied volatility surface reconstruction for option pricing error mitigation. IEEE Access.

[20] Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. Applied Sciences, 15(2), 951.

[21] Qiu, L. (2024). Deep learning approaches for building energy consumption prediction. Frontiers in Environmental Research, 2(3), 11-17.

[22] Zhang, S., Qiu, L., & Zhang, H. (2025). Edge cloud synergy models for ultra-low latency data processing in smart city iot networks. International Journal of Science, 12(10).

[23] Chen, S., Liu, Y., Zhang, Q., Shao, Z., & Wang, Z. (2025). Multi-Distance Spatial-Temporal Graph Neural Network for Anomaly Detection in Blockchain Transactions. Advanced Intelligent Systems, 7(8), 2400898.

[24] Liu, Y., Hu, X., & Chen, S. (2024). Multi-material 3D printing and computational design in pharmaceutical tablet manufacturing. J. Comput. Sci. Artif. Intell, 1(1), 34-38.

[25] Zhao, W., Shang, W., & Liu, Y. (2025). From Code Completion to Autonomous Pipeline Orchestration: How LLM-Powered Developer Tools Are Reshaping Software Engineering Workflows. American Journal Of Big Data, 6(05), 111-139.

[26] Wang, Z., Shen, Z., Wang, B., & Shang, W. (2025). Modernizing Enterprise Analytics through Low-Code Automation and Cloud-Native Data Architectures. Asian Business Research Journal, 10(12), 20-33.

[27] Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access, 12, 116546-116557.

[28] Shang, W., Wang, Z., & Wang, B. (2025). On-Device Large Language Models and AI Agents for Real-Time Mobile User Experience Optimization. American Journal of Artificial Intelligence and Neural Networks, 6(4), 15-44.

[29] Sun, T., Wang, M., & Chen, J. (2025). Leveraging machine learning for tax fraud detection and risk scoring in corporate filings. Asian Business Research Journal, 10(11), 1-13.