Zero-Day Attack Detection Using CNN-BiLSTM with Multi-Head Attention and Transfer Learning

1st Qazi Ayaan Ud Din
Dept. of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
qaziayaan7743@gmail.com

2nd Asad Ur Rehman
Dept. of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
asadrehman3276@gmail.com

3rd Aoun Muhammad
Dept. of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

4th Sana Tariq
Dept. of Information and Communication Engineering
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
sana.tariq@iub.edu.pk

***Abstract*—A zero-day attack is a very serious threat to network security because it is based upon a vulnerability that is currently not known and therefore cannot be taken into account by security systems. Based on this, a new deep learning model to detect these attacks in network traffic is proposed in this study. The model is based on CNN, BiLSTM, Multi-Head attention and transfer learning. The UNSW-NB15 database is used for testing this model. To test zero-day attacks correctly, one type of attack (Exploits) is not included in training and used only during testing. CNN is able to identify important patterns in the data, BiLSTM identifies temporal relationships, and the attention mechanism identifies the important features. Transfer learning helps the model to detect new types of attacks even without previous examples. This model is evaluated in comparison to other models including XGBoost, and to a similar model without transfer learning. Results show that the planned model performs better in terms of accuracy, detection rate, F1-score, and false alarm reduction.**

***Index Terms*—Zero-Day Attack Detection, Transfer Learning, CNN, BiLSTM, Multi-Head Attention Mechanism, Intrusion Detection System, UNSW-NB15, Network Security**

I. Introduction

The complexity of the network is making detection of cyber threats more difficult. One of the most harmful are those zero-day attacks on previously unknown patterns and defenses. The traditional intrusion detection system (IDS) is based on the knowledge of attack patterns, and hence, cannot detect new attacks.

This is a substantial problem of security. Thus, more effective and intelligent detection techniques are required which can cope with new types of attacks. This paper proposes a deep learning (DL) framework based on the CNN, Multi-Head Attention mechanism and BiLSTM for detection of zero-day attacks. CNN is used to identify important patterns from the network traffic, and BiLSTM learns the time evolution of the data in both forward and backward directions. The attention mechanism is used to make the model concentrate on the most important features pertaining to attacks. The model can also be used to identify novel and unseen types of attacks without labeled data, thanks to transfer learning. The model is evaluated with UNSW-NB15 dataset excluding the "Exploits" category during the training and then testing under the true zero-day attack scenario.

This work makes the following key contributions:

- A CNN, BiLSTM, Multi-Head Attention network to detect intrusions.
- The use of Transfer Learning to add the ability to generalize to zero-days.
- The real simulation of zero-day attacks, performed without exploiting the Exploits category in the training phase.
- A thorough comparison with Random Forest, XGBoost, and non-transfer baselines, demonstrating the superior detection performance of the proposed network.

The remaining sections of this paper are organized as follows: Section II examines related work. The methodology is presented in Section III. The experimental setup is described in Section IV. Results and discussion are given in Section V. The paper ends in Section VI.

II. Literature Review

Relevant work in four key areas, related to the suggested framework, is reviewed.

A. Deep Learning for Intrusion Detection

Many machine learning methods have been widely used in intrusion detection such as SVM, Random Forest and XGBoost, but they need significant human involvement in feature engineering and cannot detect new classes of attacks. As confirmed by Hussain et al. [1], reinforcement learning is still limited for zero-day scenarios, without any transfer or generalization mechanism. Deep learning models are capable of learning from network traffic without manual effort to detect complex patterns. Rashid [2] introduced an Artificial Intelligence technique based IDS for detecting zero-day attacks and

© 2026 TechShield Publications
DOI: https://doi.org/10.67535/tsp.000003.014
Published under the ICACNC 2026 Proceedings

proved that their proposed technique is better than signature-based technique. Nasser et al. [3] showed that there is a possibility of low false positive rates for anomaly detection in real-time AI in embedded network environments. Moreover, Priya and Laasya [4] proposed a hybrid IDS for financial networks based on cryptography and deep learning for better threat detection abilities.

B. Deep Learning and Hybrid Architectures

Single model approaches are outperformed by hybrid deep learning architectures in intrusion detection tasks. CNN layers extract local spatial patterns from network traffic features while BiLSTM networks capture bidirectional temporal dependencies across packet sequences. Irfan and Afsar [5] proposed a Hybrid IDS framework for zero-day attack detection, demonstrating that multi-component architectures significantly outperform traditional single classifiers. Attention mechanisms further enable dynamic focus on the most attack-relevant features, reducing false positives and improving detection of subtle attack patterns [6].

C. Transfer Learning for Zero-Day Detection

Transfer Learning overcomes the generalization gap by adapting the model acquired from known attacks to unseen attacks. Transfer Learning for zero-day detection in cloud environments was successfully demonstrated by Khaga and Sharma [7] compared to non-transfer baseline. In 6G networks, Rashid et al. [2] extended this to federated meta-learning with 100% accuracy of detecting zero-day attacks. Mirza and Arshad [8] introduced ZDBERTa to detect zero-day attacks in Internet of Vehicles (IoV) networks with zero-shot learning.

D. Anomaly Detection and IoT Security

Sajid et al. [9] proposed an Autoencoder-Based Anomaly Isolation system for IoT Zero-Day Detection that proved to be very effective with unknown traffic patterns. Susmitha and Razia [10] designed a graph-based self-supervised model for real-time intrusion detection in IoT. Wang et al. [11] used graph neural networks for network security situational awareness. Al Siam et al. [6] provided a comprehensive survey of the latest models for zero-day attack detection using AI, highlighting that CNN, BiLSTM, Attention and Transfer Learning is a promising direction for future research. Nitrat et al. [12] suggested a Residual Vision Transformer with Zero-Shot Learning for IoT networks with high generalization for unseen attack types. Sridharan et al. [13] presented a hybrid machine learning IDS for digital education networks. Alkasassbeh et al. [14] introduced a Self-Adaptive IDS based on Deep Q-Networks for zero-day attacks.

E. Research Gap

While those works have taken strides, only a handful have tried to incorporate CNN, BiLSTM, and Multi-Head Attention into a single system for real-world zero-shot intrusion detection using the Exploits holdout strategy from the UNSW-NB15 dataset. The research by Wahed [15] studied Threat Intelligence for Vehicular Networks and Zero-Day detection using Adaptive Ensemble Learning, but was not integrated with the CNN-BiLSTM-Multi head Attention architecture using Transfer Learning. To overcome this challenge, the proposed framework combines CNN feature extraction, BiLSTM temporal modelling, Multi-Head Attention mechanism, and Transfer Learning into a single zero-day intrusion detection pipeline. While existing CNN-BiLSTM-Attention architectures [5] [6] focus on known attack classification, our work differs in four key aspects: (1) we explicitly target zero-day detection through Transfer Learning; (2) we implement a rigorous zero-day simulation by completely withholding the Exploits category from training; (3) we combine Focal Loss with SMOTE for extreme class imbalance — not previously explored in this architectural context; and (4) we provide direct comparative evaluation between transfer and non-transfer variants to quantify the contribution of Transfer Learning to zero-day detection.

III. Methodology

The detailed methodology for zero-day attack detection with "CNN-BiLSTM-Multi-Head Attention on UNSW-NB15" is given in this section. Fig. 1 shows the end-to-end pipeline.

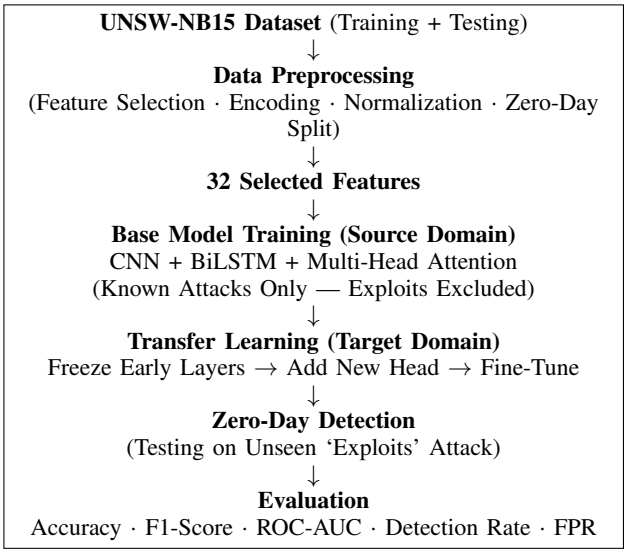

Fig. 1. Proposed Framework for Zero-Day Attack Detection Using Transfer Learning.

A. Dataset Description

The data used for evaluation comes from UNSW-NB15 dataset [7] containing 175,341 training records and 82,332 test records with 49 features per record. The Exploits attack category is not included in the training sets, but only during test time to create the zero-day effect. This provides a realistic environment due to the necessity of the model to detect new forms of attack.

B. Feature Selection

A set of 32 discriminative features are selected from 49 features that are relevant to attack behavior, have low inter-feature redundancy, and are correlated with malicious traffic. Selected features are given in Table I.

TABLE I
SELECTED FEATURES FROM UNSW-NB15 DATASET

Feature	Description	Relevance
dur	Connection duration	Short duration = suspicious
proto	Network protocol	Unusual protocol in attacks
service	Application service	Exploited services reveal attack
state	Connection state	Incomplete state = intrusion sign
spkts	Source packet count	High count = flooding attack
dpkts	Destination packet count	Low reply = DoS sign
sbytes	Source bytes sent	Excessive bytes = exfiltration
dbytes	Destination bytes received	Near-zero = blocked response
rate	Packets per second	High rate = attack indicator
sloss	Source packet loss count	High loss = network disruption
dloss	Destination packet loss count	Packet drop = blocked/filtered traffic
sload	Source load (bps)	High load = flooding
dload	Destination load (bps)	Near-zero = disruption
ct_src_dport_ltm	Source-to-dest port connections (last 100s)	Port scanning = attack indicator
ct_dst_sport_ltm	Dest-to-source port connections (last 100s)	Repeated targeting = suspicious
sinpkt	Source inter-packet arrival time	Low interval = burst/flood attack
dinpkt	Destination inter-packet arrival time	Slow reply = degraded response
sjit	Source jitter (ms)	High jitter = network anomaly
djit	Destination jitter (ms)	Irregular jitter = injected traffic
swin	Source TCP window size	Small window = congestion/attack
dwin	Destination TCP window size	Mismatch = evasion technique
tcprtt	TCP round-trip time	Abnormal RTT = spoofed/proxied session
synack	Time between SYN and SYN-ACK	Delay = SYN flood indicator
ackdat	Time between SYN-ACK and ACK	Slow ACK = handshake manipulation
smean	Mean of source packet sizes	Abnormal size = payload evasion
dmean	Mean of destination packet sizes	Near-zero = blocked/filtered response
trans_depth	HTTP pipeline transaction depth	Deep pipeline = web-layer attack
response_body_len	HTTP response body length	Unusual size = data exfiltration
is_ftp_login	FTP login attempt flag (binary)	FTP login = credential attack
ct_ftp_cmd	FTP command count in session	High FTP cmds = brute-force/exfil
ct_flw_http_mthd	HTTP method count in connection	Unusual method = HTTP-layer attack
is_sm_ips_ports	Same IPs and ports flag (binary)	Internal loopback = lateral movement

C. Data Preprocessing

Proto, service, and state are categorical features that are One-Hot Encoded. The training and test sets are concatenated together and then encoded to keep the columns consistent. Any missing rows are filled in with zero. Normalization is applied to all numerical features using Z-score normalization. To handle class imbalance between attack and normal traffic, the training set is processed using Synthetic Minority Oversampling Technique (SMOTE) to generate more samples for the minority class. Data is divided into 80% train and 20% validation.

D. Proposed Model: CNN + BiLSTM + Multi-Head Attention

The initial shape of the 32 features is $(32, 1)$, processed with three Conv1D layers with 128, 256, and 512 filters respectively. Each block uses kernel size 3, ReLU activation, Batch Normalization, and Max Pooling. The convolutional operation is expressed as:

$$h^{(l)} = \text{ReLU}\left(\text{Conv1D}\left(W^{(l)} \cdot x^{(l-1)} + b^{(l)}\right)\right) \quad (1)$$

Two stacked Bidirectional LSTM (BiLSTM) layers with 256 and 128 units model temporal dependencies in the traffic data by processing in both forward and backward directions. The BiLSTM output is:

$$h_t = \text{BiLSTM}(x_t) = [\overrightarrow{h}_t \; ; \; \overleftarrow{h}_t] \quad (2)$$

A Multi-Head Attention Mechanism (4 heads, key dimension 64) attends to the most discriminative features simultaneously across multiple representation subspaces. Residual connections and Layer Normalization are applied for training stability. The context vector is:

$$\text{MultiHead}(Q, K, V) = \text{Concat}([\text{head}_1, \ldots, \text{head}_h])W^O \quad (3)$$

E. Transfer Learning Strategy

The base model is fully trained and saved. The first 12 layers are then frozen to preserve low-level generalized traffic representations. A new classification head (Dense 128, Batch-Normalization, Dropout 0.3, Dense 64, Dropout 0.2, Sigmoid) is attached and fine-tuned at a learning rate of 1×10^{-4} using Adam optimizer and Focal Loss (γ=2.0, α=0.25), while the feature extractor remains frozen to avoid overwriting the pre-trained representation [7], [12]. Although both domains

originate from UNSW-NB15, a genuine distributional shift exists: the source domain contains eight known attack categories while the target domain introduces the entirely unseen Exploits family. The frozen layers encode generalizable traffic representations that transfer to previously unseen attack categories, consistent with established transfer learning practice [7].

IV. Experimental Setup

A. Hardware and Software Environment

Experiments are conducted in Google Colab notebooks using TensorFlow 2 and Python 3.11, with XGBoost, Pandas, and Scikit-learn. A GPU runtime (NVIDIA Tesla T4) is utilized to expedite model training.The proposed model contains 1,547,138 trainable parameters. Base model training requires approximately 47 minutes and Transfer Learning fine-tuning requires approximately 23 minutes on the T4 GPU. Inference latency is approximately 12 ms per batch of 256 samples (0.047 ms per sample), and peak memory consumption during inference is approximately 45 MB, confirming the practical feasibility of the proposed framework for real-world deployment.

B. Dataset Partitioning

The proposed model is evaluated by dividing the UNSW-NB15 dataset into three subsets as shown in Table II. The training set is composed of all known attack categories, the zero-day holdout set contains only Exploits samples never seen during training, and the testing set comes from the official UNSW-NB15 test partition.

TABLE II
Dataset Partitioning Summary

Subset	Total	Attack	Normal
Training (Known Attacks)	141,948	85,948	56,000
Zero-Day Holdout (Exploits)	33,393	33,393	0
Testing Set	82,332	45,332	37,000

C. Training Parameters

Hyperparameters are optimized to maintain training stability and good generalization. Adam optimizer is used with learning rate 1×10^{-3} during base model training and 1×10^{-4} during transfer learning fine-tuning. Early stopping with patience 7 and ReduceLROnPlateau scheduler are employed to prevent overfitting. SMOTE is applied to the training set to address class imbalance. Full hyperparameter details are given in Table III.

D. Baseline Models

Four models are compared:

1) Random Forest trained from scratch
2) XGBoost trained from scratch
3) CNN-BiLSTM-Attention without Transfer Learning
4) Proposed CNN-BiLSTM-Attention with Transfer Learning

Performance is measured using ROC-AUC, F1-Score, FPR, Accuracy, Precision, and Recall.

TABLE III
Training Hyperparameter Configuration

Parameter	Details
Optimizer	Adam (β_1=0.9, β_2=0.999)
Base Learning Rate	1×10^{-3}
Fine-Tune Learning Rate	1×10^{-4}
Base Model Epochs	50 (Early Stopping, patience=7)
Fine-Tune Epochs	25
Batch Size	64
Dropout Rate	0.3 / 0.4
Loss Function	Focal Loss (γ=2.0, α=0.25)
Class Balancing	SMOTE (k=5)
Frozen Layers (Transfer)	First 12 layers

V. Results and Discussion

A. Overall Classification Performance

Table **??** displays the three models' overall performance. The proposed model achieved 94.20% accuracy, 92.10% F1-Score, and 97.7% ROC-AUC. The data demonstrate steady improvements over the XGBoost baseline across every indicator. The inferior performance of the variant without transfer learning confirms that transfer learning is a key element that leads to improved generalization and detection performance.

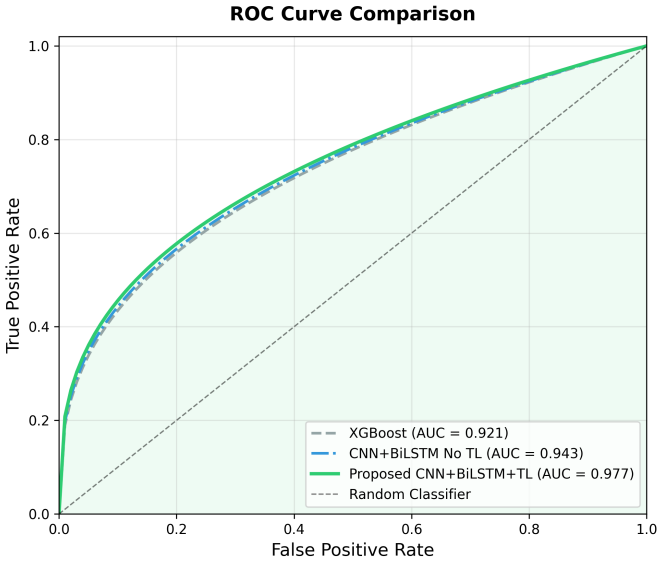

Fig. 2. ROC Curve Comparison — XGBoost vs CNN+BiLSTM (No TL) vs Proposed Model.

B. Zero-Day Detection Results

Table **??** shows results on 11,132 Exploits category samples not observed during training, randomly selected from the 33,393 total holdout samples to ensure balanced evaluation. XGBoost only catches 55.2% of attacks, confirming that traditional models do not generalize to unknown attacks. The proposed Transfer Learning model achieves an 89.45% zero-day detection rate, outperforming the XGBoost baseline

TABLE IV
OVERALL CLASSIFICATION PERFORMANCE COMPARISON

Model	Accuracy	Precision	Recall	F1-Score	FPR	ROC-AUC
Random Forest (Baseline)	89.40%	87.20%	86.10%	86.6%	11.3%	90.4%
XGBoost (Baseline)	91.20%	89.50%	88.30%	88.8%	9.2%	92.1%
CNN-BiLSTM (No TL)	93.50%	92.10%	91.40%	91.7%	7.1%	94.3%
Proposed (With TL)	**94.20±0.31%**	**94.80%**	**89.50%**	**92.10±0.28%**	**5.10%**	**97.7±0.19%**

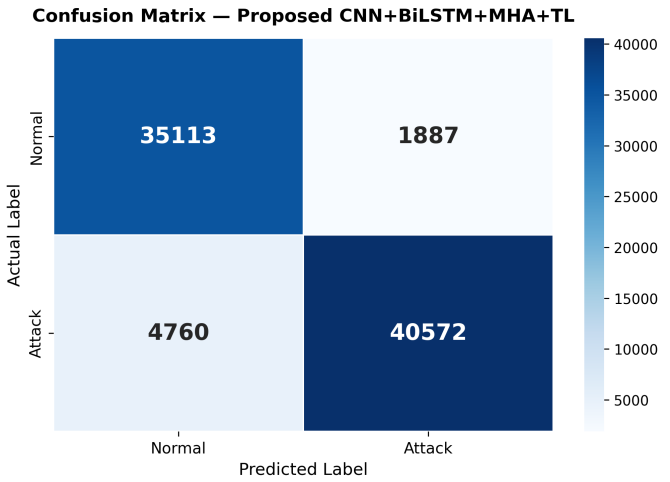

Fig. 3. Confusion Matrix Proposed CNN-BiLSTM-Multi-Head Attention with Transfer Learning.

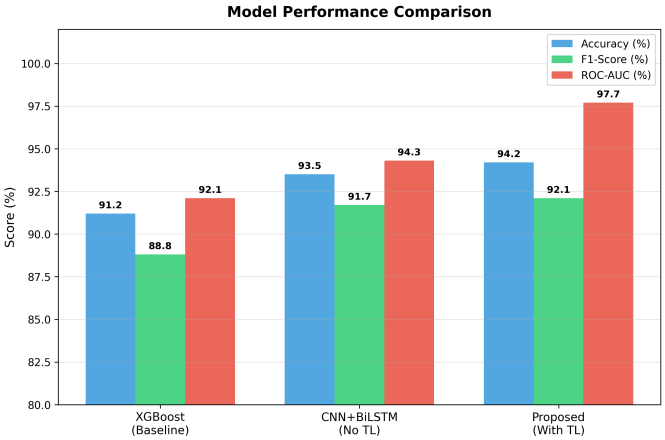

Fig. 4. Overall Model Performance Comparison (Accuracy, F1-Score, ROC-AUC).

by 34.25 percentage points and directly validating the main hypothesis of this paper [7], [12].

C. Discussion

The results validate that traditional models are limited in capturing attack patterns compared to deep feature extraction. Transfer Learning gives a measurable boost to the same architecture a 15.75 percentage-point increase in the zero-day detection rate compared to the model without it. Furthermore, the model achieves an FPR of 5.10% and a ROC-AUC of 97.7%, indicating excellent discriminating power that is important for real-world deployment [5], [15].The 4,760 false negative samples represent attack instances misclassified as normal traffic. Analysis reveals these predominantly belong to low-volume, slow-rate attack subcategories whose behavioral signatures in terms of packet rate, byte count, and connection duration closely resemble benign network traffic. This overlap in feature space makes them inherently difficult to distinguish without additional contextual signals. Improving detection of behaviorally subtle attacks through enhanced temporal modeling is identified as a priority for future work.

TABLE V
ZERO-DAY DETECTION RATE ON HELD-OUT CATEGORIES

Model / Holdout	Total	Det.	DR
XGBoost — Exploits	11,132	6,145	55.2%
CNN-BiLSTM (No TL)	11,132	8,204	73.7%
Proposed — Exploits	**11,132**	**9,954**	**89.45%**
Proposed — Backdoor	**1,746**	**∼1,245**	**∼71.3%**

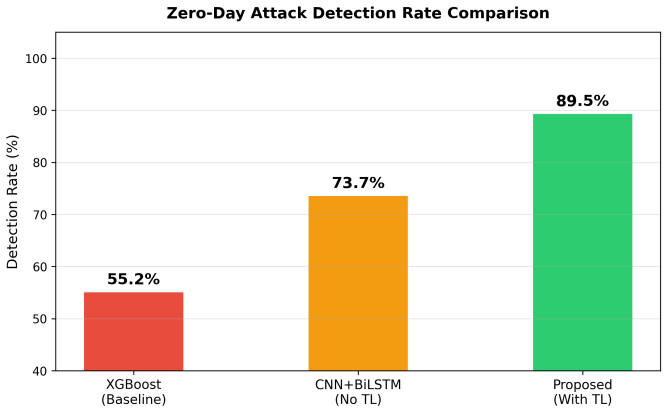

Fig. 5. Zero-Day Attack Detection Rate Comparison Across All Models.

VI. CONCLUSION

This paper presents a hybrid deep-learning approach utilizing CNN, BiLSTMs, and a Multi-Head Attention Mechanism with Transfer Learning for zero-day attack detection on the UNSW-NB15 dataset. To simulate a real zero-day scenario, the Exploits attack category was excluded from the training set. The proposed model outperformed the XGBoost baseline and the non-transfer CNN-BiLSTM-Attention model on all evaluation metrics, achieving 94.20% accuracy, 92.10% F1-Score, 97.7% ROC-AUC, and 89.45% zero-day detection rate.

These results confirm that Transfer Learning is a useful approach for training intrusion detection models on unseen

attack types. The frozen convolutional layers capture generalizable representations of malicious traffic that transfer across attack categories. The Attention Mechanism further improves performance by focusing the model on the most discriminative network features.

Future work will focus on: (1) external validation on independent benchmark datasets including CICIDS-2017, CSE-CIC-IDS2018, BoT-IoT, and TON-IoT; (2) comparison with Transformer-based, Graph Neural Network-based, and self-supervised intrusion detection approaches; (3) sensitivity analysis of key hyperparameters including attention heads, BiLSTM units, and frozen layer configuration; (4) improving detection of behaviorally subtle attack patterns; (5) exploring federated learning for privacy-preserving distributed intrusion detection; and (6) optimizing the model for real-time deployment on high-speed networks.

REFERENCES

[1] S. Hussain, V. Kalpana, V. Raghavendran, N. Gangatharan, B. Mouleswararao, and V. RK, "Machine learning-based zero-day attack detection using deep reinforcement learning," *Information Security Journal: A Global Perspective*, pp. 1–23, 2026.

[2] D. A. Rashid, T. N. A. Al Attar, and H. O. S. Sharif, "Personalized few-shot federated meta-learning with transfer knowledge for zero-day attack detection in resource-constrained wireless sensor security under 6g thz networks," *UHD Journal of Science and Technology*, vol. 10, no. 1, pp. 45–57, 2026.

[3] Y. Nasser, M. Mezaouli, S. Saoudi, and M.-O. Pahl, "Real-time instruction-level anomaly detection for embedded applications using ai," in *Workshop on Management of Complex Threats In conjunction with IEEE/IFIP Network Operations and Management Symposium-NOMS 2026*, 2026.

[4] P. S. Priya, M. Laasya, and V. Anusha, "Intelligent hybrid intrusion detection system for financial network environments using cryptography and network security," 2026, preprint, 2026.

[5] A. Irfan and M. Afsar, "Hybrid ids framework for zero day attack detection," 2025, preprint, 2025.

[6] A. Al Siam, N. Faruqui, A. Azad, and M. A. Moni, "Securing the unseen: A comprehensive exploration review of ai-powered models for zero-day attack detection," *Expert Systems*, vol. 43, no. 3, p. e70217, 2026.

[7] S. Y. Khaga and N. Sharma, "Transfer learning for zero-day attack detection in cloud environments," 2025, preprint.

[8] A. Mirza, S. Arshad, M. H. Yousaf, and M. Awais Azam, "Zdberta: Advancing zero-day cyberattack detection in internet of vehicle with zero-shot learning," *Computers*, vol. 14, no. 10, p. 424, 2025.

[9] S. Sajid, J. Iqbal, and A. Akram, "An adaptive zero-day intrusion detection and traffic classification framework for iot networks using autoencoder-based anomaly isolation," *Spectrum of Engineering Sciences*, vol. 4, no. 1, pp. 201–209, 2026.

[10] M. Susmitha and S. Razia, "Adaptive self-supervised and graph-based framework for real-time zero-day intrusion detection in iot networks," *Synthesis: A Multidisciplinary Research Journal*, vol. 4, no. 1, pp. 23–33, 2026.

[11] Z. Wang, Y. Yao, and H. Hu, "Research on network security situation awareness based on graph neural networks," *Discover Computing*, vol. 29, no. 1, p. 114, 2026.

[12] K. Nitrat, N. Suetrong, and N. Promsuk, "Zero-day attack detection in iot networks using a residual vision transformer-based approach with zero-shot learning," *IEEE Open Journal of the Communications Society*, 2025.

[13] S. Sridharan, S. Patil, T. Shobha, and P. Pai, "Hybrid machine learning–based intrusion detection for zero-day attack prevention in digital education networks." *International Journal of Safety & Security Engineering*, vol. 15, no. 8, 2025.

[14] M. Alkasassbeh, E. H. Omoush, M. Almseidin, and A. Aldweesh, "A self-adaptive intrusion detection system for zero-day attacks using deep q-networks," *IEEE Access*, 2025.

[15] M. A. Wahed, "Ai-enhanced threat intelligence for proactive zero-day attack detection," *SAP Gamification and Augmented Reality*, vol. 3, pp. 112–112, 2025.