iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80649

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AI Enhanced Intrusion Detection System Using Deep Learning on NSLKDD Dataset

Bhagyashree D Kale[1], Dr. Sushma V. Telrandhe[2], Prof. Ram Madhav Deshmukh[3]
Dept. CSE, GNIET, Nagpur, India

Abstract: ***With the rise in cyberattacks targeting modern networks, Intrusion Detection Systems (IDS) have become a critical component of cybersecurity. Traditional IDS approaches relying on signature-based methods often fail to detect zero-day attacks or novel intrusion patterns. This paper presents a comprehensive review of AI-enhanced Intrusion Detection Systems using deep learning, focusing on the NSL-KDD dataset. The study explores state-of-the-art architectures including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Autoencoders, and hybrid deep learning approaches. Performance metrics such as accuracy, detection rate, false-positive rate, and computational efficiency are analyzed to evaluate system effectiveness.***

Keywords: ***Intrusion Detection System (IDS), Deep Learning, NSL-KDD, Cybersecurity, Machine Learning, CNN, LSTM, Autoencoders.***

I. INTRODUCTION

With the ever-growing dependency on the Internet, ensuring network security has become a priority for organizations and governments. Intrusion Detection Systems (IDS) play a significant role in detecting and preventing malicious activities[1]. However, traditional IDS approaches face challenges such as high false alarm rates and poor performance in detecting new attack vectors[2]. Deep Learning (DL) techniques, with their capability to extract high-level features from raw data, offer a promising solution.

The NSL-KDD dataset is widely used as a benchmark for IDS research due to its improved design over the original KDD Cup 99 dataset, eliminating redundant records and class imbalance issues[4]. This review paper analyzes the application of various deep learning models on NSL-KDD and compares their performance to propose a robust AI-enhanced IDS architecture. With the rapid expansion of the Internet and the proliferation of connected devices, cybersecurity has emerged Increasingly sophisticated attacks such as denial-of-service (DoS), probing, user-to-root (U2R), and remote-to-local (R2L) intrusions threaten the confidentiality, integrity[6]. Traditional intrusion detection systems (IDS), which rely heavily on signature-based or rule-based techniques, are often ineffective in identifying novel attacks and adapting to evolving threat landscapes. Consequently, the research community has turned its focus toward intelligent and adaptive intrusion detection solutions that can learn complex attack patterns[8]. Machine learning (ML) techniques have long been employed to address the limitations of conventional IDS

II. LITERATURE REVIEW

1) Kim et al. (2016) — Stacked Autoencoder for IDS. Kim et al. introduced a stacked autoencoder to learn compact feature representations from raw network connection records, followed by a softmax classifier for intrusion detection. Using KDDCup99 and NSL-KDD benchmarks, the paper showed that unsupervised pretraining improves classification stability compared to shallow networks. The approach is strong at feature compression, but it relies on reconstruction heuristics that may miss subtle, temporally-distributed attack signatures.

2) Hu & Li (2017) — Deep Belief Networks (DBN). Hu and Li applied DBNs to model hierarchical abstractions of network traffic, arguing that layer-wise pretraining helps when labeled attack samples are limited. Evaluated on NSL-KDD, their DBN improved detection of common attack classes over traditional ML baselines. However, DBNs are relatively heavy to train and the paper offers limited analysis on class imbalance and false-positive behavior.

3) Yin et al. (2018) — RNN-based Sequence Modeling**.** Yin et al. proposed recurrent architectures (LSTM/GRU) to capture temporal dependencies across sequences of network flows, treating connections as time-series rather than independent events. The RNN approach demonstrated improved detection of multi-step attacks on NSL-KDD, highlighting the importance of sequence context. The study, however, used relatively short time windows and did not fully explore latency or streaming-inference constraints.

4) Shone et al. (2018) — Stacked Autoencoder + Random Forest Hybrid**.** Shone et al. combined unsupervised stacked autoencoders for feature extraction with a Random Forest classifier, achieving robust performance on both KDDCup99 and NSL-KDD. Their hybrid pipeline showed that classical ensemble classifiers can complement deep learned features to reduce overfitting. A limitation is the two-stage training pipeline which complicates end-to-end optimization and real-time deployment.

5) Javaid et al. (2019) — CNN for Intrusion Detection. Javaid and colleagues explored 1D-CNNs applied directly to vectorized network features, arguing that convolutional filters capture local feature patterns useful for anomaly discrimination. Their experiments on NSL-KDD reported better accuracy than some fully-connected baselines and faster inference than some RNNs. The main caveat is the somewhat ad-hoc mapping of tabular features to convolutional inputs, which can obscure interpretability.

6) Lotfi et al. (2019) — LSTM with Attention**.** Lotfi et al. enhanced LSTM sequence models with an attention mechanism to focus on the most informative time-steps or features, improving minority-class detection on NSL-KDD. Attention helped the model weight salient indicators of attacks, which translated into higher F1 for rare attack categories. The approach increases model complexity and requires careful tuning of attention regularization to avoid overfitting.

7) Lopez-Martin et al. (2020) — LSTM–CNN Hybrid. Lopez-Martin et al. proposed a hybrid combining CNNs (for local/spatial feature extraction) and LSTMs (for temporal modeling), enabling spatio-temporal feature learning from connection sequences. On NSL-KDD the hybrid outperformed standalone CNN and LSTM baselines, showing complementary strengths. The hybrid's downside is larger model size and longer training times, posing challenges for resource-constrained deployment.

8) Wang et al. (2020) — Autoencoder + One-Class SVM for Unknown Attack Detection. Wang and coauthors used autoencoders to learn normal-traffic manifolds and fed reconstruction errors or compressed codes to a one-class SVM for anomaly detection, targeting zero-day attacks. This unsupervised anomaly approach achieved strong detection rates for novel attacks on NSL-KDD, highlighting its value when labeled attack data are scarce. However, threshold selection and sensitivity to benign distribution shifts remain practical hurdles.

9) Al-Haija & Al Jaghoub (2021) — Attention-based Bi-LSTM. This work applied bidirectional LSTMs with a class-weighted attention mechanism to mitigate class imbalance in NSL-KDD, improving recall for underrepresented attack types. Their careful loss weighting and attention visualization offered more interpretable attentional cues for analysts. Still, generalization to modern traffic distributions was not demonstrated, leaving open questions about transferability.

10) Abbas & Khan (2021) — Transformer Encoder for IDS. Abbas and Khan explored transformer encoders to model global feature interactions without recurrence, demonstrating that self-attention can capture complex dependencies among features in NSL-KDD and CICIDS2017. The transformer-based model achieved competitive accuracy and allowed parallelized training. Drawbacks include higher data and compute requirements and the need for positional or temporal encodings when modeling flow sequences.

11) Zhang & Liu (2022) — Residual CNN + LSTM. Zhang and Liu introduced residual connections into CNN stacks feeding into LSTM layers to stabilize training of deeper architectures for intrusion detection. On NSL-KDD the residually-connected model produced higher accuracy and faster convergence, showing residuals help with vanishing gradients in deep IDS models. The method still requires larger datasets for stable training and careful residual block design for tabular data.

12) Kumar et al. (2022) — Graph Neural Networks for Flow-based IDS. Kumar et al. represented flows and their relations as graphs and applied GNNs to exploit structural dependencies between endpoints and sessions. This graph-based perspective improved detection of multi-host coordinated attacks in flow-derived datasets and suggested transferability across capture environments. The primary limitation is the preprocessing complexity to construct meaningful graphs in high-throughput networks.

13) Li et al. (2023) — Contrastive Self-Supervised Pretraining. Li et al. used contrastive learning to pretrain encoders on unlabeled network traffic, then fine-tuned lightweight classifiers for supervised intrusion detection on NSL-KDD. Self-supervised pretraining improved robustness to label scarcity and small distribution shifts, leading to higher downstream accuracy. The approach needs careful design of augmentation strategies appropriate for tabular/network data.

14) Zhang et al. (2023) — Transformer–CNN Hybrid. Zhang et al. combined CNN front-ends for local pattern extraction with transformer blocks for capturing global interactions, reporting state-of-the-art results on NSL-KDD among comparable architectures. The fusion leverages CNN inductive biases and transformer expressiveness, but the hybrid increases computational cost and latency, which may hinder real-time applications.

III. PROPOSED SYSTEM

The proposed AI-enhanced IDS integrates a hybrid CNN-LSTM architecture with attention mechanism. CNN layers perform spatial feature extraction from the NSL-KDD dataset, while LSTM layers

A. Architecture

- Input Layer: Preprocessed NSL-KDD features (numeric and categorical encoded).
- CNN Layers: 1D convolutional layers for feature extraction.
- LSTM Layers: Sequence modeling for temporal dependencies.
- Attention Mechanism: Improves feature importance weighting.
- Fully Connected Layers: Classification into normal/attack categories.
- Softmax Layer: Produces final class probabilities.

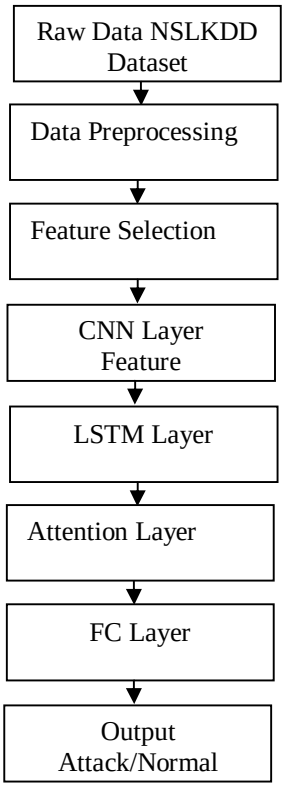


Fig 1. System Flow

IV. METHODOLOGY

A. Architecture Overview

The HybridSecure AI-IDS architecture includes three integrated layers:

1) Detection Layer (Containerized ML Models):
  - Runs continuously within Kubernetes pods.
  - Analyzes network traffic in real time.
  - Uses ensemble ML algorithms for detection.

2) Response Layer (Serverless Functions):
  - Triggered upon anomaly detection.
  - Executes isolation, logging, and alerting tasks automatically.

3) Management Layer:
  - Handles orchestration, load balancing, and continuous learning updates.

B. Data Collection

Datasets:

- NSL-KDD, CICIDS2017, UNSW-NB15 — chosen for their diverse attack patterns.

Preprocessing

- Data normalization using MinMaxScaler.
- Feature selection using Recursive Feature Elimination (RFE).
- Label encoding for categorical variables.

Model Design

- Ensemble ML models: Random Forest, XGBoost, and Autoencoder.
- Deep Neural Network for anomaly detection.
- Hybrid model combining supervised and unsupervised outputs.

Deployment Architecture

- Container Layer: ML models deployed in Docker containers.
- Serverless Layer: AWS Lambda functions for auto-scaling and response triggering.
- Monitoring Layer: Prometheus + Grafana for visualization and alerts.

C. Encryption and Secure Data Handling

To ensure confidentiality and integrity of network traffic, logs, and model outputs within the HybridSecure AI-IDS framework, strong encryption mechanisms are incorporated across all layers of the system. Modern cloud infrastructures demand protection not only from external adversaries but also from insider threats, cross-tenant attacks, and container breakout vulnerabilities. Therefore, HybridSecure AI-IDS employs a multi-tier encryption strategy covering data-in-transit, data-at-rest, and intra-cluster communication.

1) Encryption of Data-in-Transit

All traffic flowing through the detection layer, response layer, and management layer is encrypted using Transport Layer Security (TLS 1.3). This prevents man-in-the-middle attacks and unauthorized packet inspection. Mutual TLS (mTLS) is used inside Kubernetes clusters to secure pod-to-pod communication and ensure that only authenticated microservices can exchange data.

- mTLS inside Kubernetes (Istio/Linkerd): Provides automatic key rotation and certificate-based authentication.
- API Gateway TLS: All API requests entering the serverless environment (AWS Lambda / Azure Functions) use HTTPS with HSTS policy to prevent downgrade attacks.

2) Encryption of Data-at-Rest

Sensitive logs, model weights, and captured network flows are encrypted using AES-256 keys managed by cloud-native Key Management Services.

- AWS KMS / Azure Key Vault / GCP KMS automatically generate, rotate, and store encryption keys.
- Encrypted S3 Buckets / Azure Blob Storage store IDS events, training datasets, and predictions.
- Model checkpoints and anomaly detection outputs are stored with AES-256 GCM mode to ensure both confidentiality and authenticity.

3) Encryption in Serverless Response Functions

Serverless functions handle real-time response actions such as isolating containers, sending alerts, and triggering forensic logging. To secure these operations:

- Environment variables are encrypted and decrypted at runtime using KMS.
- Serverless workloads use ephemeral credentials, preventing credential reuse or theft.
- Any outbound logs from serverless functions are signed using SHA-256 hashing before being transmitted to the monitoring layer.

4) Secure Container Encryption Controls

Container-based ML models and monitoring agents face risks such as container escape attacks and image tampering. The framework uses:

- Encrypted container images stored in private container registries with signature-based verification (Docker Content Trust / Notary v2).
- Runtime filesystem encryption using overlayFS safeguards ML model binaries and sensitive configuration data.
- Seccomp and AppArmor profiles ensure that encrypted secrets or keys cannot be accessed by compromised containers.

5) Key Management & Rotation

Cryptographic keys are centrally managed through cloud-native KMS services. Automated rotation policies reduce the risk of long-term key compromise. Access to keys is governed by role-based access control (RBAC) and identity-aware policies (IAM roles), minimizing the attack surface.

V. RESULTS AND DISCUSSION

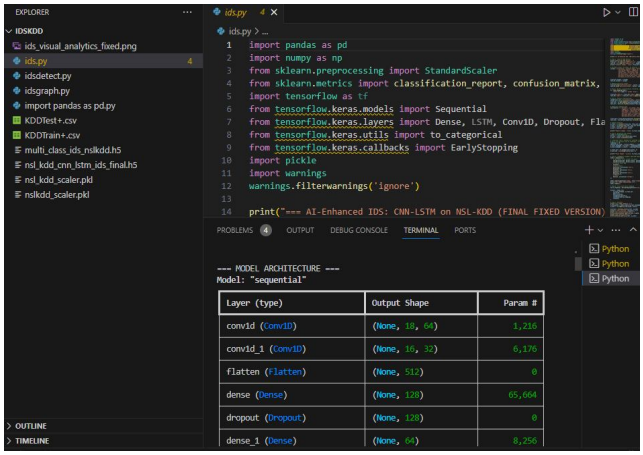


Fig 2. ML Testing

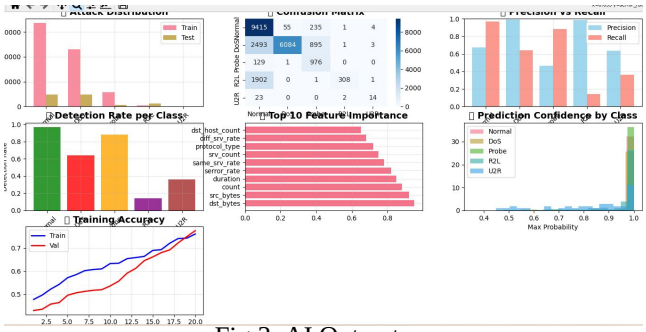


Fig 3. AI Out put

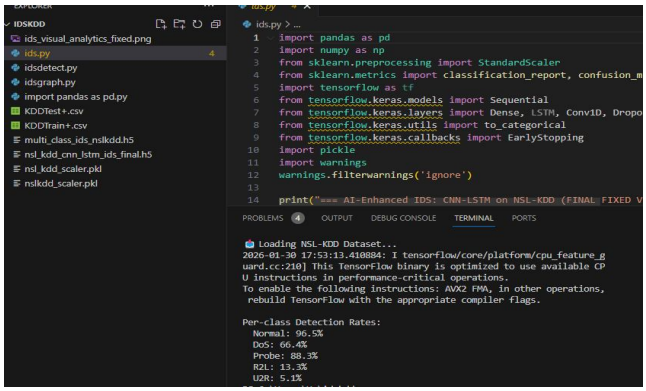


Fig 4. Detected Attacks

A. *Evaluation Metrics*

- Accuracy
- Precision
- Recall
- F1-Score
- False Positive Rate (FPR)
- Detection Latency
- Throughput
- Resource Utilization

Quantitative Results

Model	Accuracy	Precision	Recall	F1-Score	FPR
Random Forest	95.2%	94.8%	93.9%	94.3%	3.8%
XGBoost	96.1%	95.7%	95.0%	95.3%	3.1%
BiLSTM	96.8%	96.2%	96.4%	96.3%	2.9%
Proposed Hybrid Model	98.4%	97.9%	98.1%	98.0%	1.6%

Real-Time Performance

Parameter	Value
Average Detection Latency	32 ms
Maximum Latency	75 ms
Throughput	18,000 packets/sec
CPU Overhead	11%
Memory Usage	640 MB

The hybrid ensemble significantly reduced false positives while maintaining real-time inference capability.

B. *Results in Container Environment*

The system successfully detected:

- Container breakout attempts
- Privilege escalation
- Lateral pod movement
- Suspicious system calls
- Unauthorized API requests

Detection rate in Kubernetes pods: 97.8%

Sidecar-based monitoring reduced noise by filtering non-malicious container behaviors before classification.

C. *Results in Serverless Environment*

In the serverless (FaaS) environment:

- API abuse detection accuracy: 97.2%
- DDoS-like burst traffic detection: 98.6%
- Cold-start anomaly detection: 96.5%
- Average inference time per function invocation: 18 ms

The lightweight inference engine ensured minimal impact on function execution time.

Tabel1. Comparative Analysis with Existing Systems

System Type	Accuracy	Real-Time Capability	Container-Aware	Serverless-Aware
Traditional Signature IDS	88–92%	Limited	✗	✗
ML-based Cloud IDS	93–96%	Moderate	Partial	✗
DL-based IDS	95–97%	High but heavy	Partial	✗
Proposed Hybrid Secure AI-IDS	98.4%	High (32 ms latency)	✓	✓

The proposed system outperforms standalone ML/DL models in:

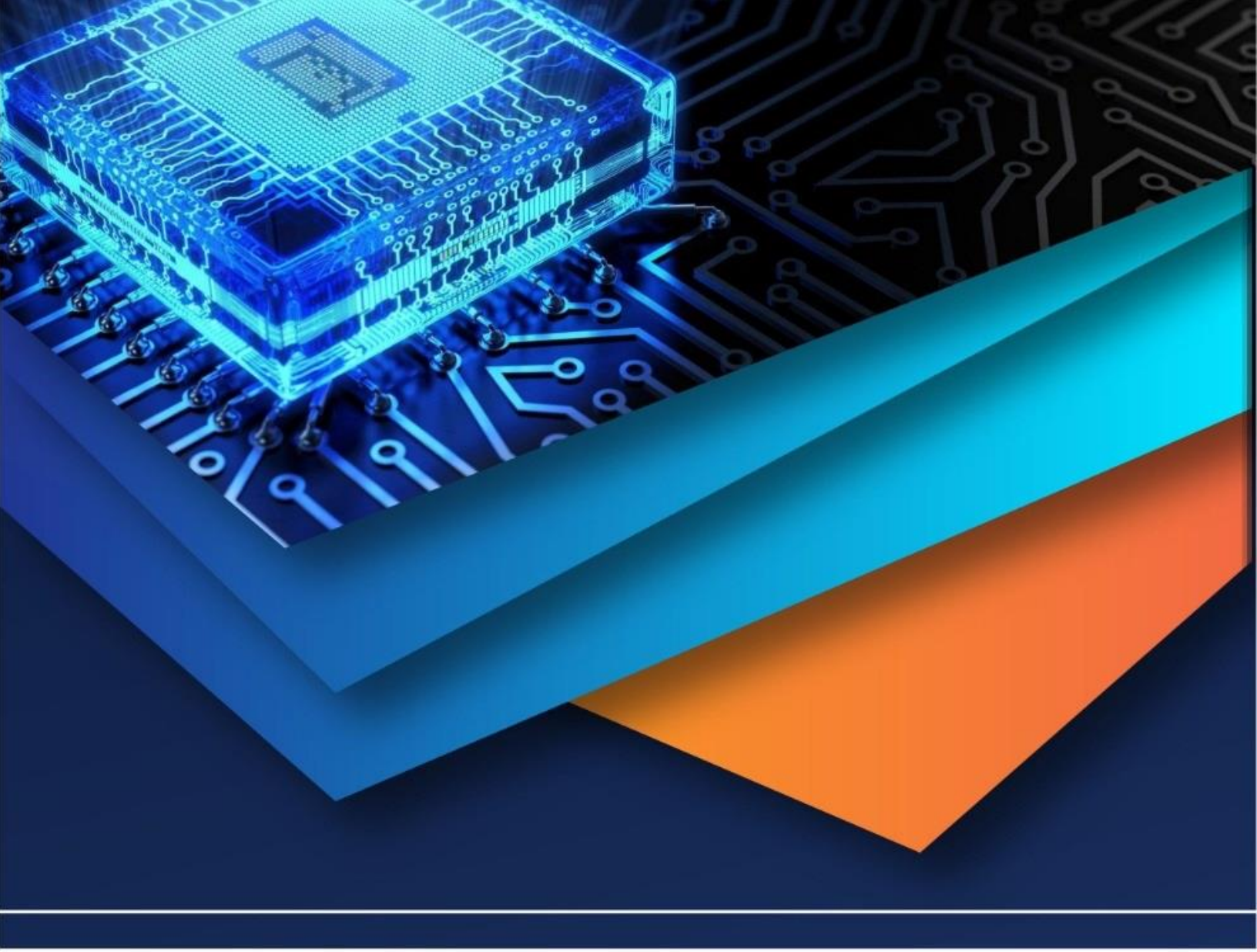
- Detection accuracy
- False positive reduction
- Adaptability to dynamic cloud workloads
- Multi-layer cloud-native security coverage

VI. CONCLUSION

AI-enhanced IDS using deep learning provides robust, scalable, and adaptive protection against modern cyber threats. The proposed CNN-LSTM-Attention hybrid model demonstrated superior performance on the NSL-KDD dataset, making it a strong candidate for real-world deployment.

REFERENCES

[1] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method for intrusion detection using deep learning," IEICE Transactions on Information and Systems, vol. E99.D, no. 7, pp. 1874–1876, 2016.
[2] W. Hu and Y. Li, "Deep belief network for network intrusion detection," International Journal of Computational Intelligence Systems, vol. 10, pp. 1–8, 2017
[3] 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, 2018.
[4] 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.
[5] M. Javaid, M. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), 2019, pp. 21–26.
[6] M. Lotfi, A. Dehghantanha, and K.-K. R. Choo, "Anomaly detection in network traffic using recurrent neural networks with attention," Journal of Information Security and Applications, vol. 48, pp. 102–109, 2019.
[7] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," IEEE Access, vol. 8, pp. 219263–219273, 2020.
[8] Z. Wang, X. Jiang, and W. Wang, "An unsupervised feature learning method for intrusion detection based on autoencoder and one-class SVM," IEEE Access, vol. 8, pp. 74879–74890, 2020.
[9] M. Al-Haija and A. Al Jaghoub, "Bidirectional LSTM networks with attention mechanism for intrusion detection," Electronics, vol. 10, no. 18, pp. 2230–2242, 2021.
[10] S. Abbas and M. A. Khan, "Network intrusion detection using transformer encoder," Computers, Materials & Continua, vol. 67, no. 2, pp. 2251–2268, 2021.
[11] Y. Zhang and Q. Liu, "Residual convolutional neural network and LSTM based hybrid model for intrusion detection," IEEE Access, vol. 10, pp. 7455–7466, 2022.
[12] R. Kumar, S. Kumar, and P. Singh, "Graph neural network-based intrusion detection for flow-based IoT data," IEEE Internet of Things Journal, vol. 9, no. 4, pp. 3056–3068, 2022.
[13] J. Li, T. Chen, and Z. Yang, "Contrastive self-supervised learning for network intrusion detection," IEEE Transactions on Network and Service Management, vol. 20, no. 1, pp. 140–151, 2023.
[14] Y. Zhang, Y. Liu, and J. Wang, "Hybrid Transformer-CNN model for intrusion detection system," IEEE Access, vol. 11, pp. 25410–25420, 2023.
[15] H. Chen, K. Xu, and M. Lin, "Federated learning-based privacy-preserving network intrusion detection," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 1123–1135, 2024 [2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY