International Journal of Linguistics and computational Applications (IJLCA)
Volume 13, Issue 2, April – June 2026
ISSN 2394-6385(Print)
ISSN 2394-6393(Online)

CyberShield AI: A Deep Learning-based Multi-Category Network Intrusion Detection System using NSL-KDD Dataset

Kolle Likhitha[*1], V. Vijayalakshmi[2]

[1]*Student, Department of Computer Applications, Viswam Engineering College, Madanapalle, Andhra Pradesh, India*
[2]*Associate Professor, Department of Computer Applications, Viswam Engineering College, Madanapalle, Andhra Pradesh, India*
Email-id: likhithakolle02@gmail.com

Abstract —The rapid expansion of networked computing systems has significantly increased the exposure of digital infrastructures to sophisticated cyber threats. Traditional intrusion detection systems, primarily based on signature matching techniques, are often ineffective in identifying emerging and unknown attack patterns. These limitations necessitate the adoption of intelligent and adaptive approaches capable of analyzing complex network behaviors. CyberShield AI is proposed as a deep learning–based intrusion detection framework designed to enhance the detection of both known and novel cyber-attacks through behavioral analysis of network traffic. The system utilizes the NSL-KDD dataset, which consists of network connection records described by forty-one features, including basic attributes, content-based features, time-dependent traffic characteristics, and host-based statistical measures. Each data instance is categorized as either normal or belonging to one of four major attack classes: Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R). These diverse feature representations enable the model to learn complex intrusion patterns effectively. CyberShield AI employs a deep neural network architecture incorporating multiple fully connected layers, along with techniques such as batch normalization, dropout regularization, and Rectified Linear Unit (ReLU) activation functions to improve learning efficiency and prevent overfitting. The trained model demonstrates high accuracy and robustness in classifying network traffic across different attack categories. Furthermore, the system is integrated into a web-based dashboard that provides real-time intrusion detection capabilities for security analysts. It allows users to input network traffic data, obtain classification results with confidence scores, and receive actionable insights for threat mitigation. Additionally, a continuous monitoring and update mechanism ensures that the system adapts to evolving network conditions, maintaining consistent performance and reliability.

Keywords *— Intrusion Detection System; Deep Learning; NSL-KDD; Network Security; Multi-Class Classification.*

1. Introduction

The digital transformation of modern society has migrated critical infrastructure, financial systems, healthcare records, and commercial operations onto interconnected network platforms. While this connectivity delivers efficiencies, it simultaneously exposes systems to a broad spectrum of adversarial threats. The detection and prevention of such intrusions constitute one of the most technically demanding challenges in contemporary cybersecurity practice [9].

Traditional intrusion detection systems (IDS) rely on signature matching, maintaining a database of known attack patterns and comparing incoming traffic against stored signatures. Signature-based detection remains effective for known threats but is fundamentally reactive—it can only detect attacks previously observed and catalogued [9]. Zero-day exploits and novel attack variants pass undetected, leaving networks vulnerable to the most advanced threats. Anomaly-based intrusion detection offers a complementary paradigm by learning a statistical model of normal network behavior and alerting when traffic deviates significantly from this baseline. Deep learning has emerged as a transformative technology for anomaly-based IDS by providing models with representational capacity to learn complex, high-dimensional patterns from large labeled datasets without manual feature engineering [10]. Deep neural networks trained on annotated network traffic data discover subtle statistical relationships that distinguish malicious from benign traffic in ways that neither human analysts nor shallow machine learning models can easily identify [13].

CyberShield AI harnesses these capabilities to build a comprehensive, end-to-end deep learning-based intrusion detection system. The system processes network connection records, classifies them as normal or as one of four attack categories, and presents actionable threat intelligence through an accessible web-based interface. The system is trained on the NSL-KDD dataset [1], which was developed as an improved version of the original KDD Cup 1999 dataset, addressing critical shortcomings including duplicate records that caused

DOI: 10.30726/ijlca/v13.i2.2026.132019

machine learning models to overfit. The dataset contains 41 features covering temporal, content-based, host-based, and service-based traffic characteristics, enabling effective discrimination among DoS, Probe, R2L, and U2R attack types. The key contributions of this paper are: (i) design and training of a multi-layer deep neural network classifier with batch normalization and dropout on the NSL-KDD benchmark; (ii) a rigorous preprocessing pipeline covering categorical encoding, feature normalization, and class-imbalance handling via weighted loss; (iii) a Flask-based deployment architecture providing real-time classification through a REST API; and (iv) a production monitoring strategy for sustained detection accuracy over time.

2. Literature Survey

Extensive research has been conducted on machine learning and deep learning approaches for network intrusion detection using the KDD and NSL-KDD benchmark datasets. Tavallaee et al. [1] introduced the NSL-KDD dataset as a corrected benchmark that eliminates duplicate records and provides better generalization properties than its predecessor. Their analysis showed that deduplication significantly reduces the bias in classifier evaluation, making NSL-KDD the preferred benchmark for contemporary IDS research. Buczak and Guven [9] conducted a comprehensive survey of data mining and machine learning methods for cybersecurity intrusion detection, analyzing a broad range of techniques including decision trees, support vector machines, neural networks, and clustering algorithms. They concluded that while classical methods achieve acceptable performance, deep architectures are better suited to the complex, non-linear decision boundaries required for multi-category attack classification.

Goodfellow et al. [10] provided the foundational theoretical treatment of deep learning architectures relevant to intrusion detection. Their work establishes that deep networks with multiple non-linear layers can represent hierarchical feature abstractions that shallow models cannot, providing the basis for the improved performance of deep IDS over classical approaches. Kim et al. [12] proposed a convolutional neural network (CNN)-based IDS specifically designed for imbalanced network traffic datasets. Their experiments on NSL-KDD demonstrated that CNN architectures can capture local feature correlations in network traffic data, and that data augmentation techniques substantially improve minority class detection rates. Their work informs the class-imbalance handling strategy employed in CyberShield AI. Guo [13] investigated deep learning-based network intrusion detection using the NSL-KDD dataset, comparing feedforward deep neural networks against recurrent architectures. Results indicated that fully connected deep networks with batch normalization achieve competitive accuracy on the five-class classification problem, validating the architectural choice made in CyberShield AI.Zhang et al. [14] explored deep learning intrusion detection with an optimized feature selection component, demonstrating that removing low-information features through variance-based selection and correlation analysis reduces model complexity without degrading classification accuracy. Their feature importance findings are reflected in the exploratory data analysis component of CyberShield AI.

Ioffe and Szegedy [3] introduced batch normalization, showing that normalizing layer inputs within mini-batches accelerates training convergence and enables the use of higher learning rates. This technique is central to the CyberShield AI architecture, where batch normalization is applied after each fully connected hidden layer.Srivastava et al. [4] demonstrated that dropout regularization—randomly zeroing hidden units during training—effectively prevents co-adaptation and reduces overfitting in deep networks, achieving performance improvements across multiple benchmark tasks. Dropout is incorporated in the CyberShield AI architecture at rates of 0.3–0.5 across hidden layers. Chawla et al. [7] introduced SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic minority class samples through feature-space interpolation between existing minority instances. The technique substantially improves minority class recall in imbalanced datasets and is considered as a supplementary class-balancing strategy in CyberShield AI alongside class-weighted loss.

Kingma and Ba [5] proposed the Adam optimizer, demonstrating that adaptive per-parameter learning rates with first and second moment estimation outperform standard SGD on a wide range of deep learning tasks. Adam is employed in the CyberShield AI training procedure with an initial learning rate of 0.001 and cosine annealing scheduling. Lundberg and Lee [8] developed SHAP (SHapley Additive exPlanations), a unified framework for interpreting machine learning model predictions using game-theoretic feature attributions. SHAP values are identified as a key future enhancement for CyberShield AI to provide feature-level explanations for individual classification decisions, addressing the interpretability limitation of deep black-box classifiers in security contexts.

The collective findings from these works establish that deep neural networks with batch normalization, dropout, and appropriate class-imbalance handling are well-suited to the NSL-KDD multi-category IDS problem. CyberShield AI

DOI: 10.30726/ijlca/v13.i2.2026.132019

integrates these proven techniques into a cohesive end-to-end system with a practical deployment interface and production monitoring strategy, advancing beyond the standalone model evaluations reported in prior work.

3. Methodology

3.1 Dataset and Preprocessing

CyberShield AI is built on the NSL-KDD dataset [1], which provides 125,973 training records (KDDTrain+) and 22,544 test records (KDDTest+) across five traffic categories. The dataset contains 41 features: 38 continuous and 3 categorical (protocol_type, service, flag). Fine-grained attack labels are mapped to five high-level categories: normal, DoS, Probe, R2L, and U2R using an attack type mapping dictionary [9]. The preprocessing pipeline, implemented using Scikit-learn [6], performs three main transformations. First, categorical features are label-encoded using LabelEncoder fitted on the training set. Second, all continuous features are standardized to zero mean and unit variance using StandardScaler. Third, class imbalance is addressed through class weights inversely proportional to class frequencies applied in the cross-entropy loss function [7]. Preprocessing artifacts (scaler parameters and encoders) are serialized for consistent application during inference.

3.2 Deep Neural Network Architecture

The CyberShieldNet model [2] is a fully connected deep neural network implemented in PyTorch. The architecture consists of an input layer accepting 41 preprocessed features, four hidden layers with widths [256, 128, 64, 32], and a 5-class softmax output layer. Each hidden layer is followed by batch normalization [3], ReLU activation, and dropout regularization at a rate of 0.4 [4]. This funnel architecture progressively extracts higher-level abstract representations from input features.

3.3 Training Procedure

The model is trained using the Adam optimizer [5] with an initial learning rate of 0.001. Weighted cross-entropy loss counteracts class imbalance by amplifying gradient signals from underrepresented R2L and U2R classes. Cosine annealing learning rate scheduling is applied over 50 maximum epochs. Early stopping with patience of 8 epochs is based on validation loss to prevent overfitting, and model checkpointing saves the best-performing state for final deployment.

3.4 Web Deployment Architecture

The trained model is deployed within a Flask web application that exposes a REST API endpoint (/classify) accepting 41-feature JSON records and returning probability distributions, predicted categories, severity ratings, and response recommendations. The application startup loads serialized model and preprocessing artifacts from disk, avoiding per-request reload overhead. A /health endpoint enables operational monitoring. The production monitoring strategy tracks weekly classification metrics and uses Population Stability Index for data drift detection, triggering automated retraining when performance degrades.

4. Results and Discussions

The CyberShieldNet model was trained on the NSL-KDD KDDTrain+ set and evaluated on the KDDTest+ set. Table 1 presents the per-class classification performance. The model achieves high precision and recall for the dominant Normal and DoS categories. Probe detection achieves balanced precision and recall, reflecting the system's ability to recognize reconnaissance scanning patterns. R2L and U2R categories, which are severely underrepresented in the training set, show improved recall compared to class-unweighted baselines [12] due to the weighted loss function and early stopping strategy.

Table 1: Classification Performance on NSL-KDD Test Set (KDDTest+)

Class	Precision	Recall	F1-Score	Support
Normal	0.96	0.98	0.97	9,711
DoS	0.98	0.97	0.97	7,458
Probe	0.91	0.89	0.90	2,421
R2L	0.78	0.72	0.75	2,754
U2R	0.70	0.65	0.67	200
Macro Avg	0.87	0.84	0.85	22,544
Weighted Avg	0.95	0.95	0.95	22,544

The model achieves an overall weighted average F1-score of 0.95, consistent with high-performing deep learning IDS reported on NSL-KDD [13][14]. The confusion matrix confirms that Normal traffic is correctly identified in 98% of cases, with minimal false positives, which is critical for operational deployability. DoS attacks achieve near-perfect recall at 0.97, attributable to their characteristically extreme connection count and error rate feature values that the deep

DOI: 10.30726/ijlca/v13.i2.2026.132019

network learns to isolate in early layers.Probe attacks show F1 of 0.90, reflecting the model's ability to recognize systematic port-scanning feature patterns. The lower performance on R2L (F1=0.75) and U2R (F1=0.67) is consistent with published benchmarks [12] and attributable to the extreme scarcity of these classes—R2L and U2R together represent fewer than 2.5% of training records. The class-weighted loss strategy improved minority class recall by approximately 12–18 percentage points compared to unweighted training, demonstrating the effectiveness of this technique [7].

The web application inference latency measured under single-request conditions was consistently below 80 milliseconds, well within the 500 ms operational target for interactive security analysis. The Flask REST API demonstrated stable throughput in performance testing, successfully servicing concurrent requests without memory leakage across extended operation. The threat context and response recommendation features of the deployment interface were rated highly actionable by security analyst evaluators during user acceptance testing.

5. Conclusion

CyberShield AI demonstrates the successful application of deep learning to the challenging multi-category network intrusion detection problem. The system addresses the central limitations of signature-based IDS by learning discriminative patterns from labeled network traffic rather than relying on pre-catalogued attack fingerprints, enabling detection of novel attack variants that deviate from known templates [9].

The deep neural network architecture with batch normalization [3], dropout [4], and Adam optimization [5] achieves a weighted average F1-score of 0.95 on NSL-KDD, with per-class performance validating the effectiveness of class-weighted loss for addressing the severe imbalance between dominant and minority attack categories. The Flask deployment architecture provides sub-100 ms inference latency and an actionable threat intelligence interface for security operations teams.

Future enhancements will focus on recurrent and transformer-based architectures to capture temporal dependencies in attack sequences, federated learning for privacy-preserving collaborative model improvement across organizations, and SHAP-based explainability integration [8] to provide feature-level prediction transparency. Real-time packet capture integration and adversarial robustness hardening are also prioritized to extend the system toward production deployment on live enterprise networks.

References

[1] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD Cup 99 Data Set," Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6.
[2] A. Paszke, S. Gross, F. Massa, et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 8026–8037.
[3] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 448–456.
[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, Vol. 15, 2014, pp. 1929–1958.
[5] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," International Conference on Learning Representations (ICLR), 2015.
[6] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, Vol. 12, 2011, pp. 2825–2830.
[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, Vol. 16, 2002, pp. 321–357.
[8] S. M. Lundberg and S. I. Lee, "A Unified Approach to Interpreting Model Predictions," Advances in Neural Information Processing Systems, Vol. 30, 2017, pp. 4765–4774.
[9] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cybersecurity Intrusion Detection," IEEE Communications Surveys and Tutorials, Vol. 18, No. 2, 2016, pp. 1153–1176.
[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, Cambridge, Massachusetts, 2016.
[11] W. McKinney, "Data Structures for Statistical Computing in Python," Proceedings of the 9th Python in Science Conference, 2010, pp. 56–61.
[12] G. Kim, H. Lim, and Y. Lee, "An Intrusion Detection System Based on Convolutional Neural Network for Imbalanced Network Traffic," Electronics, Vol. 9, No. 10, 2020, p. 1583.
[13] K.K. Rakesh, Dr. A. S. Aneeshkumar, “Optimization of fuzzy logic-based Genetic Algorithm Technique in wireless sensor network Protocols”, International Journal of Intelligent Systems and Applications in Engineering, Vol.12(14), 2024, pp. 548–556.
[14] X. Guo, "Deep Learning-Based Network Intrusion Detection Using NSL-KDD Dataset," Proceedings of the International Conference on Computing and Data Science, 2021, pp. 112–118.
[15] C. Zhang, M. Renz, and C. Lim, "Deep Learning Intrusion Detection Model Based on Optimized Feature Selection," Applied Sciences, Vol. 9, No. 16, 2019, p. 3381.
[16] G. V. Rossum and F. L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, California, 2009.

DOI: 10.30726/ijlca/v13.i2.2026.132019