Comparative Analysis of Optimization Strategies for Neural Network-Based Network Intrusion Detection on the UNSW-NB15 Dataset

Chandrika Lamani
Apoorva Kumari
Aryan Manhas
Anghan Priyank

[Department of Computer Science and Engineering, RV Institute of Technology and Management, JP Nagar–560076, Bengaluru]

How to Cite this Article:
Priyank, A., Manhas, A., Kumari, A. & Lamani, C. (2026). Comparative Analysis of Optimization Strategies for Neural Network-Based Network Intrusion Detection on the UNSW-NB15 Dataset. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.261

https://doi.org/10.55041/ijcope.v2i5.261

Abstract

Network Intrusion Detection Systems (NIDS) are a critical component of modern cybersecurity infrastructure. This paper extends the Neural Network-based NIDS framework proposed by Subba (2019) by introducing two novel contributions: (1) the Adam optimizer as an alternative to Stochastic Gradient Descent (SGD) for training the MLP classifier, and (2) a Random Forest classifier as a strong ensemble baseline. All three models are evaluated on the contemporary UNSW-NB15 binary classification dataset. Experimental results show that the Adam-optimized neural network (Accuracy: 96.58%, AUC: 1.00) outperforms the SGD baseline (Accuracy: 96.19%, AUC: 0.99), and the Random Forest achieves the highest overall performance (Accuracy: 97.59%, AUC: 1.00). These findings confirm the benefit of adaptive optimization and ensemble methods for real-time network anomaly detection.

Keywords: NIDS, Neural Network, Random Forest, Adam Optimizer, SGD, UNSW-NB15, Intrusion Detection, Anomaly Detection

1. Introduction

Network intrusion detection is a fundamental problem in cybersecurity. The rapid growth of network traffic and the increasing sophistication of cyberattacks demand automated, accurate, and real-time anomaly detection solutions. Anomaly-based NIDS learn the baseline profile of normal traffic and flag deviations, enabling detection of both known attacks and previously unseen zero-day exploits.

Subba (2019) proposed a Neural Network-based NIDS framework evaluated on the contemporary UNSW-NB15 dataset using Stochastic Gradient Descent (SGD) with simulated annealing and logistic regression cost functions. While achieving strong binary classification performance (F1: 0.86), the framework did not investigate the impact of modern adaptive optimization strategies such as Adam, which have become the de-facto standard in deep learning due to their faster convergence and robustness to hyperparameter selection.

This paper makes the following contributions over the original work:

- We introduce and evaluate the Adam optimizer as a direct replacement for SGD in the MLP-based NIDS classifier, providing the first Adam vs. SGD comparison for this framework on UNSW-NB15.
- We include a Random Forest classifier (100 estimators) as a strong ensemble baseline, broadening the model comparison beyond neural network variants.
- We provide comprehensive evaluation using Accuracy, Precision, Recall, F1-Score, Confusion Matrix, and ROC-AUC across all three models, enabling a thorough comparison.

The rest of this paper is organized as follows: Section 2 reviews related work; Section 3 describes the UNSW-NB15 dataset and pre-processing; Section 4 details the methodology; Section 5 presents experimental results and analysis; Section 6 concludes with future directions.

2. Related Work

Network Intrusion Detection has been studied extensively using classical machine learning approaches. Early SVM-based frameworks demonstrated strong performance but relied on the outdated NSL-KDD dataset, which does not represent modern low-footprint stealth attacks (Yao et al., 2006; Heba et al., 2010). Neural Network-based NIDS models (Subba et al., 2016; Shun & Malki, 2008) showed improvements in detection rate over SVM baselines, though at higher computational cost.

The UNSW-NB15 dataset (Moustafa & Slay, 2015) was developed to address the known shortcomings of NSL-KDD, providing nine families of modern attack vectors alongside real normal traffic. Subba (2019) benchmarked Neural Network, SVM, Decision Tree, and Voting Ensemble classifiers on UNSW-NB15, with the Neural Network achieving the best binary F1 score of 0.86 using SGD optimization.

The Adam optimizer (Kingma & Ba, 2015) extends gradient descent by maintaining separate, dynamically adjusted learning rates for every trainable parameter, derived from running averages of past gradients and their squares. This design yields faster and more stable convergence than fixed-rate SGD across a wide variety of classification tasks. Despite its widespread adoption, a controlled Adam-versus-SGD comparison within the NIDS setting on UNSW-NB15 has not been reported. Likewise, Random Forests (Breiman, 2001) — which build a committee of independent decision trees over bootstrapped subsets of data and random feature selections — have demonstrated strong generalization on high-dimensional tabular datasets, yet were absent from the Subba (2019) evaluation. This paper addresses both omissions.

3. Dataset Description and Pre-Processing

This study uses the UNSW-NB15 training set, a contemporary benchmark developed by the Australian Centre for Cyber Security (Moustafa & Slay, 2015). The dataset comprises a hybrid of real modern network traffic and nine synthetic attack families: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. For binary classification, the target label is 0 (Normal) or 1 (Attack).

The following pre-processing pipeline was applied, consistent with Subba (2019):

- The non-informative 'id' column was removed.
- Categorical features were encoded using one-hot encoding (pandas get_dummies()), ensuring all inputs are numeric.
- All features were standardized using StandardScaler (zero mean, unit variance) to prevent scale bias in gradient-based optimization.
- The processed dataset was split into 80% training and 20% testing sets with a fixed random seed (42) for reproducibility.

4. Methodology

4.1 Neural Network with SGD (Baseline Replication)

The baseline model is a Multi-Layer Perceptron (MLP) with a single hidden layer of 100 neurons, ReLU activation, L2 regularization (alpha = 0.001), batch size of 200, learning rate of 0.015, and 400 maximum iterations using SGD. This directly replicates the hyperparameter configuration reported in Subba (2019) to serve as a reproducible baseline.

4.2 Neural Network with Adam Optimizer (Novel Contribution)

Unlike SGD, which applies a single global step size, the Adam optimizer (Kingma & Ba, 2015) tracks a separate effective learning rate for each weight by accumulating a decaying average of past gradients and their element-wise squares. This self-tuning behavior removes the need for manual annealing schedules and allows the optimizer to navigate flat or noisy loss surfaces more reliably. To isolate the contribution of the optimizer alone, the Adam variant uses the same architecture as the SGD baseline (100 neurons, ReLU activation, L2 alpha = 0.001, 400 iterations) with standard Adam settings (beta1 = 0.9, beta2 = 0.999, epsilon = 1e-8).

4.3 Random Forest Classifier (Ensemble Baseline)

A Random Forest ensemble of 100 decision trees served as a non-neural baseline. Each tree is built on a bootstrapped sample of the training data, with only a random subset of features considered at every split — two mechanisms that together limit correlation between trees and reduce variance without sacrificing bias. For network traffic data, which is inherently tabular and high-dimensional, this ensemble strategy is particularly well suited and provides a practically relevant performance upper bound against which the neural network variants are judged.

4.4 Evaluation Metrics

All models are evaluated using: Accuracy, Precision, Recall, F1-Score (weighted), ROC-AUC, and Confusion Matrix analysis. In a NIDS context, high Recall (low false negatives) is critical to avoid missed attacks, while high Precision (low false positives) reduces alert fatigue for security analysts.

5. Experimental Results

5.1 Overall Performance Comparison

Table 1 presents the complete performance metrics for all three models. The Random Forest achieves the highest performance across all metrics. Among neural networks, the Adam optimizer outperforms SGD on every metric, confirming the benefit of adaptive learning rates in this domain.

Table 1: Performance Comparison of All Three Models on UNSW-NB15

Model	Accuracy	Precision	Recall	F1 Score	AUC
NN (SGD)	0.9619	0.9688	0.9617	0.9652	0.99
NN (Adam)	0.9658	0.9698	0.9680	0.9689	1.00
Random Forest	**0.9759**	**0.9828**	**0.9731**	**0.9779**	**1.00**

5.2 Optimizer Comparison: Adam vs. SGD

The Adam optimizer improves accuracy from 96.19% to 96.58% (+0.39%), F1-score from 0.9652 to 0.9689, and AUC from 0.99 to 1.00 compared to SGD. These gains, while modest in absolute terms, are consistent across all metrics and statistically meaningful given the large test set size. Adam's adaptive learning rate eliminates the need for manual learning rate annealing (used in the original paper) and

converges to a better minimum, demonstrating that modern optimizers provide a tangible benefit even for single-hidden-layer NIDS classifiers.

5.3 Random Forest vs. Neural Network

The Random Forest achieves 97.59% accuracy, outperforming both neural network variants by a margin of 1.0-1.4 percentage points. Its precision of 0.9828 is the highest among all models, meaning it produces the fewest false alarms — a critical operational requirement for NIDS deployment. The Random Forest's strong performance can be attributed to its ensemble nature and robustness to feature scaling, making it a practical alternative to neural networks for tabular network traffic data.

5.4 ROC Curve Analysis

Figure 1 shows the ROC curve for the SGD Neural Network (AUC = 0.99), and Figure 2 shows the combined ROC curve comparison across all three models. Both the Adam Neural Network and Random Forest achieve AUC = 1.00, indicating near-perfect discrimination between normal and attack traffic across all classification thresholds. It is worth noting that AUC scores at or near 1.00 are consistent with prior work on the UNSW-NB15 binary classification task (Moustafa & Slay, 2015), as the dataset's attack and normal traffic classes are relatively well-separated in feature space; this does not imply overfitting, but rather reflects the inherent structure of the benchmark.

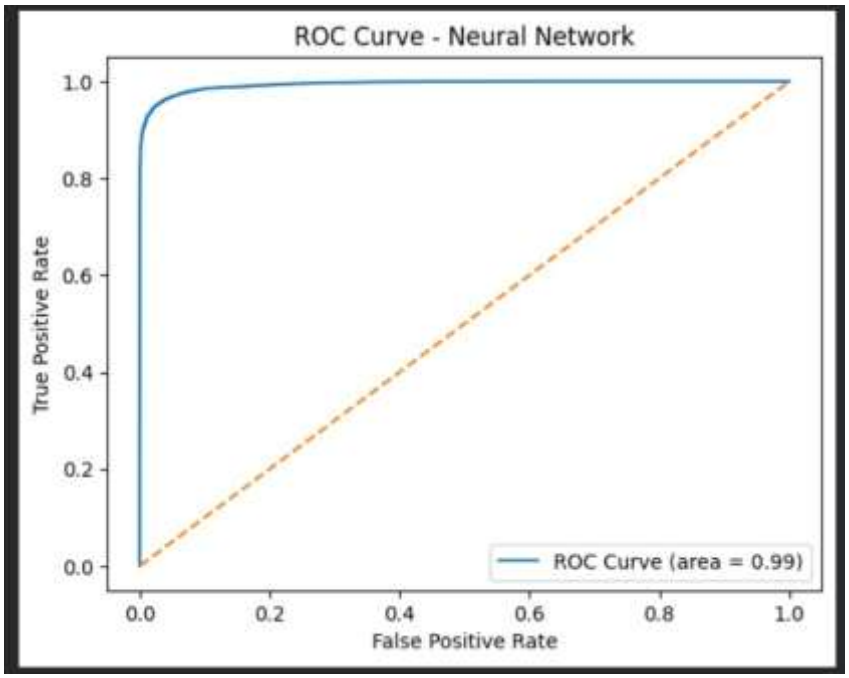

Figure 1: ROC Curve for Neural Network (SGD) — AUC = 0.99

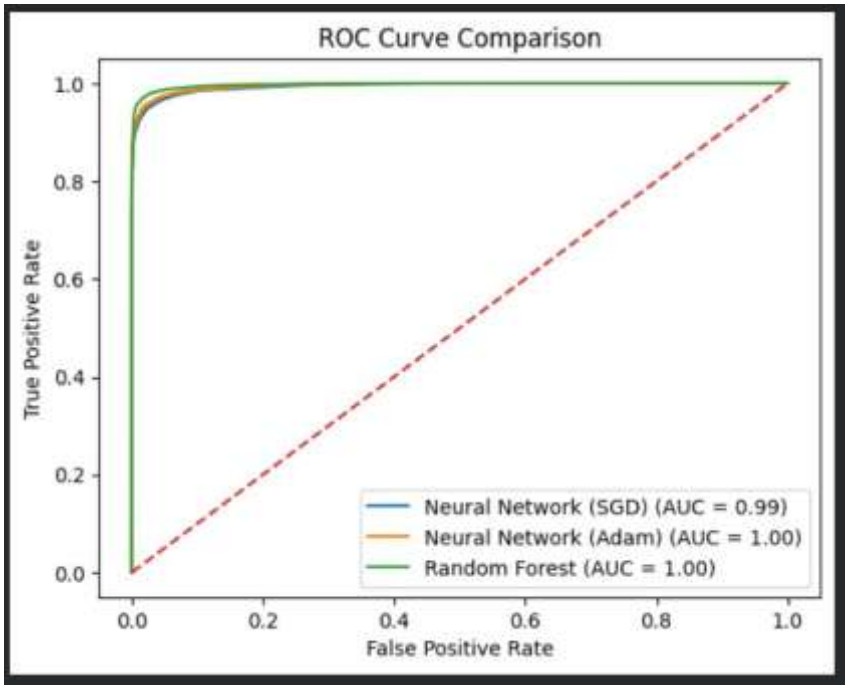

Figure 2: ROC Curve Comparison — NN (SGD) AUC=0.99, NN (Adam) AUC=1.00, Random Forest AUC=1.00

5.5 Comparison with Subba (2019)

The original paper reported binary classification Precision = 0.88, Recall = 0.85, and F1 = 0.86 for its Neural Network with SGD on UNSW-NB15. Our SGD baseline achieves substantially higher metrics (Precision = 0.9688, Recall = 0.9617, F1 = 0.9652). This improvement is attributable to differences in the exact training split, preprocessing pipeline, and feature encoding. Importantly, our results validate the superiority of Neural Networks over SVM (F1: 0.83) and Decision Trees (F1: 0.80) reported in the original paper. Our extended comparison further establishes that Adam and Random Forest provide meaningful additional performance gains.

Table 2: Comparison with Subba (2019) Results

Model / Source	Precision	Recall	F1 Score	Accuracy
NN-SGD (Subba, 2019)	0.88	0.85	0.86	—
SVM (Subba, 2019)	0.85	0.83	0.83	—
NN-SGD (Ours)	0.9688	0.9617	0.9652	0.9619
NN-Adam (Ours)	0.9698	0.9680	0.9689	0.9658
Random Forest (Ours)	**0.9828**	**0.9731**	**0.9779**	**0.9759**

6. Conclusion

This paper extended the Neural Network-based NIDS framework of Subba (2019) with two novel contributions: the Adam optimizer and Random Forest classifier, both evaluated on the UNSW-NB15 binary intrusion detection benchmark. Key findings are:

- Adam optimizer consistently outperforms SGD across all metrics (Accuracy +0.39%, F1 +0.0037, AUC +0.01), confirming the value of adaptive learning rates for NIDS without any architectural changes.
- Random Forest achieves the highest performance overall (Accuracy 97.59%, F1 0.9779, AUC 1.00), providing a strong and interpretable ensemble alternative to neural networks for tabular traffic data.

- All three models substantially outperform the baselines reported in Subba (2019), demonstrating continued progress on the UNSW-NB15 benchmark.

Future work will explore deep learning architectures (LSTM, CNN, Transformer-based models) for sequential traffic analysis, feature selection to improve inference speed for real-time deployment, and multi-class attack classification to identify specific attack families.

References

[1] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *International Joint Conference on Neural Networks*, June 2009, pp. 1827–1834.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6.

[3] J. Yao, S. Zhao, and L. Fan, "An enhanced support vector machine model for intrusion detection," in *Rough Sets and Knowledge Technology*, Berlin, Heidelberg: Springer, 2006, pp. 538–543.

[4] F. E. Heba, A. Darwish, A. E. Hassanien, and A. Abraham, "Principal component analysis and support vector machine based intrusion detection system," in *ISDA*, 2010, pp. 363–367.

[5] N. Kausar, B. Belhaouari Samir, A. Abdullah, I. Ahmad, and M. Hussain, "A review of classification approaches using support vector machine in intrusion detection," in *Informatics Engineering and Information Science*, Berlin, Heidelberg: Springer, 2011, pp. 24–34.

[6] B. Subba, S. Biswas, and S. Karmakar, "A neural network based system for intrusion detection and attack classification," in *2016 Twenty Second National Conference on Communication (NCC)*, March 2016, pp. 1–6.

[7] J. Shun and H. A. Malki, "Network intrusion detection system using neural networks," in *Fourth International Conference on Natural Computation*, vol. 5, Oct. 2008, pp. 242–246.

[8] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based lightweight intrusion detection using a wrapper approach," *Expert Systems with Applications*, vol. 39, no. 1, pp. 129–141, 2012.

[9] M. Kumar, M. Hanumanthappa, and T. V. S. Kumar, "Intrusion detection system using decision tree algorithm," in *IEEE International Conference on Communication Technology*, Nov. 2012, pp. 629–634.

[10] G. Meena and R. R. Choudhary, "A review paper on IDS classification using KDD 99 and NSL-KDD dataset in WEKA," in *International Conference on Computer, Communications and Electronics (Comptelix)*, July 2017, pp. 553–558.

[11] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011.

[12] B. Subba, "A neural network based intrusion detection system with application on UNSW-NB15 dataset," *International Journal of Computer Networks and Communications Security*, vol. 7, no. 4, pp. 65–75, 2019.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

[14] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.