Comparative Evaluation of Machine Learning Models for Network Intrusion Detection Using UNSW-NB15

Natalie Santo

Abstract

As technology advances, network intrusion detection systems are increasingly relying on machine learning to identify malicious activity within network traffic. This study uses the UNSW-NB15 dataset to evaluate the performance of four machine learning models: Logistic Regression, Random Forest, XGBoost, and a Neural Network. This dataset was chosen due to its large volume of network samples, roughly 2.54 million, and relevance to modern network trafficking. Due to significant class imbalance within the dataset, SMOTE (Synthetic Minority Oversampling Technique) was applied to equal the amount of benign and malicious samples to ensure fair model evaluation. Each model was assessed using F1 score, ROC-AUC, precision-recall curves, confusion matrices, and training runtime. After analyzing the results, it was determined that Random Forest achieved the highest overall detection performance, while XGBoost demonstrated the most efficient training time with a similar level of accuracy. The findings of this research highlights key tradeoffs between computational cost and performance in machine learning systems with a security focus.

Introduction

Each day, networked systems continue to expand in both scale and complexity. Protecting digital infrastructure from malicious activity has become increasingly challenging, as traditional rule based intrusion-detection systems struggle to generalize to modern attack patterns. Machine learning (ML) based approaches offer a unique alternative by learning patterns directly from network traffic data.

However, intrusion detection presents challenges for ML systems. Real network datasets are typically highly imbalanced, with benign traffic greatly outnumbering malicious activity. Additionally, network data often contains noisy or incomplete observations. These factors motivate a careful evaluation of model choice, preprocessing strategies, and evaluation metrics.

This work investigates the effectiveness of multiple ML and deep learning models on the UNSW-NB15 dataset, a widely used benchmark for intrusion detection research. The goal is to identify which modeling approach offers the best balance of detection accuracy, robustness, and computational efficiency.

Methodology

Dataset and Preprocessing

The UNSW-NB15 dataset consists of approximately 2.54 million modern synthetic network traffic samples. Each of these samples represents a network flow, and is described by 49 features which capture packet statistics, timing characteristics, content-based attributes, and includes a binary label to indicate whether the traffic is benign or malicious.

A significant class imbalance exists in the original dataset, as benign samples outnumber malicious samples by approximately 7:1. To address this imbalance, SMOTE was applied to the training split after standardization. This application synthesized new minority class samples (malicious) to ensure fair model evaluation.

```
sm = SMOTE(random_state=42)
X_train_res, y_train_res = sm.fit_resample(X_train_scaled, y_train)
print("Before SMOTE:", y_train.value_counts())
print("After SMOTE:", pd.Series(y_train_res).value_counts())
```

Figure 1: Instruction to print benign samples and malicious samples.

```
Before SMOTE: is_attack
0    1775011
1     257026
Name: count, dtype: int64
After SMOTE: is_attack
0    1775011
1    1775011
```

Figure 2: The output of Figure 1, counts of benign and malicious samples before and after SMOTE was applied, demonstrating equal sample sizes.

Raw CSV files were standardized and aggregated using UNSW-NB15 official column definitions. All non-numeric features were excluded to ensure compatibility with numerical ML models, any remaining feature was coerced to a numeric representation, and missing values were filled using zero.

Model Architecture

An artificial intelligence model is a combination of both algorithms and the data that is needed to train those algorithms to perform correctly. As defined by Oracle, AI model training is "the process of feeding curated data to selected algorithms to help the system refine itself to produce accurate responses to queries." Four different models were trained to represent a range of learning paradigms:

Logistic Regression:

A linear probabilistic classifier commonly used as a baseline for binary classification tasks.

Random Forest:

A collection of decision trees that include non-linear patterns and reduce overfitting through averaging.

XGBoost:

A gradient boosted decision tree framework that performs well on structured tabular data using sequential learning.

Neural Network:

A deep learning model consisting of multiple layers which is capable of learning complex interactions but requires more data and training time than average.

Evaluation Metrics

Model performance was assessed using multiple complementary metrics:

F1 Score:

A metric that offers a fair method of assessing a model by combining recall (the number of actual attacks the model successfully identified) and precision (the number of samples the model classified as attacks were actually attacks). The score itself is calculated by finding the mean of precision and recall, guaranteeing that the model cannot score well unless it excels in both precision and recall.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

ROC:

A diagnostic tool that assesses and shows how well a model can distinguish between benign and attack samples. It calculates the trade-off between the true positive rate (meaning how many attacks were correctly identified) and the false positive rate (meaning the number of benign samples that were mistakenly reported as attacks). A model whose ROC curve bends strongly toward the top left corner demonstrates a good ability to separate attacks from regular behavior.

Precision-Recall Curve:

This curve demonstrates how the models precision and recall change relative to one another when being tested across variome decision thresholds. The curve shows whether the model is consistently maintaining high precision without simultaneously sacrificing recall, and vice versa.

Confusion Matrix:

A matrix which shows the number of true positives (attacks correctly detected), false negatives (attacks missed), false positives (benign marked as attack), and true negatives (benign correctly detected), and uses these to calculate an accuracy score.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Runtime:

The total amount of training time measured to evaluate computational efficiency.

Experimental Results

F1 Score:

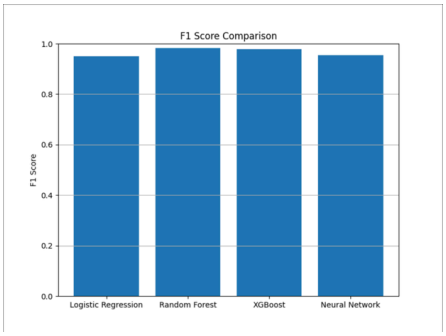


Figure 3: The F1 scores of the four models displayed as bar graphs.

Logistic Regression F1 Score: 0.9487112
Random Forest F1 Score: 0.9831118
XGBoost F1 Score: 0.9789111
Neural Network F1 Score: 0.9599813

Random Forest achieved the highest F1 score (0.983), followed closely by XGBoost (0.979). Logistic Regression and the Neural Network performed comparably but demonstrated weaker recall under certain thresholds.

ROC:

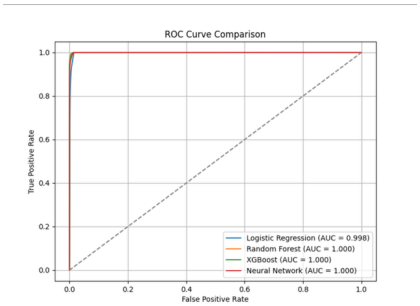
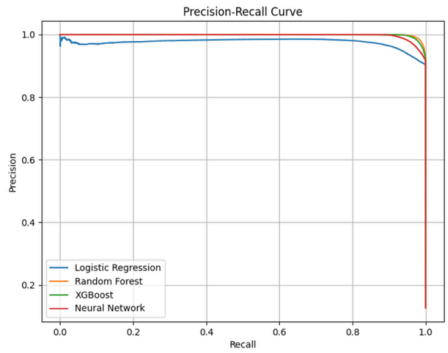


Figure 4: The ROC curves of the four models.

Logistic Regression AUC: 0.998
Random Forest AUC: 1.000
XGBoost AUC: 1.000
Neural Network AUC: 1.000

Random Forest, XGBoost, and the Neural Network achieved near perfect ROC-AUC scores (~1.0), while Logistic Regression showed slightly reduced discriminative capacity.

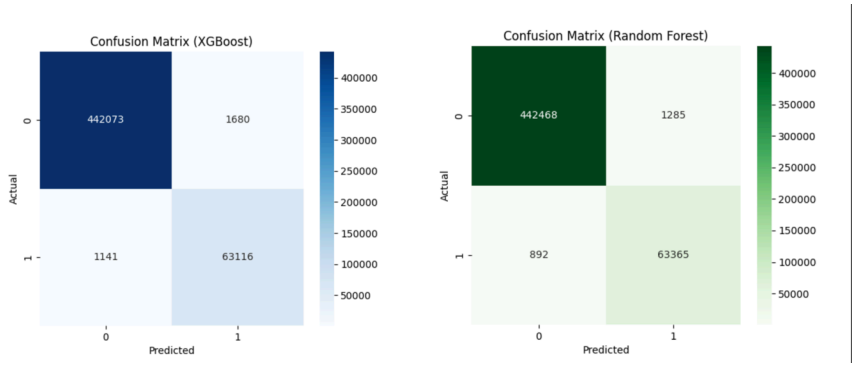
Precision-Recall Curve:



Random Forest shows the best precision recall curve, indicating it is capable of catching almost every attack while rarely incorrectly identifying benign samples. XGBoost is a close second, while Logistic Regression and Neural network perform worst.

Confusion Matrix:

The F1 Scores are calculated by using values found during the confusion matrices experiment. In other words, confusion matrices and F1 scores are directly related. The two confusion matrices shown below are of the two models with the highest F1 Scores, Random Forest and XGBoost, as it implies that they will have the highest accuracy scores.



Figures 5 and 6: The confusion matrices of XGBoost and Random Forest.

Using the formula:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Random Forest Confusion Matrix Score: (505833)/(508010) = 99.571%
XGBoost Confusion Matrix Score: (505189)/(508010) = 99.444%

Although both models show an incredibly high accuracy, Random Forest emerges as the best performer. This score of 99.571% shows that the model can accurately identify data, benign and malicious.

Runtime:

```
Training times: {'Logistic Regression': 26.591126203536987,
 'Random Forest': 103.9288501739502, 'XGBoost': 10.152359962463379,
 'Neural Network': 137.25953793525696}
```

Figure 7: The runtimes of the four models in seconds.

The above runtimes are in seconds.

Logistic Regression Runtime: ~26.6 seconds
Random Forest Runtime: ~104 seconds
XGBoost Runtime: ~10.2 seconds
Neural Network Runtime: ~137.3 seconds

XGBoost demonstrated the fastest training time, while the Neural Network incurred the highest computational cost due to its layered architecture.

Conclusion

The results above indicate that ensemble-based tree models (Random Forest, XGBoost) are well-suited for intrusion detection tasks involving structured network flow sample data. Random Forest's performance advantage can be determined due to its ability to model diverse attack signatures and simultaneously remain resistant to overfitting.

Deep learning models may offer theoretical advantages for other representation learning, however, their benefits were not utilized in this tabular setting due, especially given the computational overhead.

Limitations

This research solely focuses on binary classification. Future work could explore fine-grained attack categorization.

References

"UNSW-NB15 Dataset." Kaggle, n.d.,

https://www.kaggle.com/datasets/mrIllsdavid/unsw-nb15.

"LogisticRegression." Scikit-Learn, n.d.,

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

Jain, Abhishek. “Understanding Random Forest.” Towards Data Science, n.d., https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

Brownlee, Jason. “A Gentle Introduction to XGBoost for Applied Machine Learning.” Machine Learning Mastery, n.d., https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/.

“Sequential Model.” TensorFlow, n.d., https://www.tensorflow.org/guide/keras/sequential_model.

“SMOTE for Imbalanced Classification with Python.” GeeksforGeeks, n.d., https://www.geeksforgeeks.org/machine-learning/smote-for-imbalanced-classification-with-python/.

“Confusion Matrix in Machine Learning.” GeeksforGeeks, n.d., https://www.geeksforgeeks.org/machine-learning/confmeion-matrix-machine-learning/.

Sol Chen. “CSV Files: mee Cases, Benefits, and Limitations.” *OneSchema*, 27 Jan. 2025, www.oneschema.co/blog/csv-files

“sklearn.metrics — scikit-learn API Reference.” *scikit-learn*, sklearn.org/stable/api/sklearn.metrics.html.