



**Prifysgol Cymru**  
Y Drindod Dewi Sant  
**University of Wales**  
Trinity Saint David

## Assignment on Machine Learning Models for Network Intrusion Detection Using the NSL-KDD Dataset

**Submitted by:**

MD Azizul Khan

**Student ID:** 2514708

**Course Code:** ACCA7026

**Submitted To:**

Seena Joseph  
Lecturer, Artificial Intelligence

## Abstract

As network intrusions continue to rise in frequency and complexity, developing effective detection systems has become a critical cybersecurity priority. This project applies machine learning techniques to the NSL-KDD dataset to classify network traffic and identify malicious activity. Two ensemble models—Random Forest and Gradient Boosting—were implemented and evaluated using metrics such as accuracy, precision, recall, and F1-score. Gradient Boosting achieved a slightly higher overall accuracy (93.5%) compared to Random Forest (93%), with improved detection of rare attack types like U2R (F1-score: 0.50 vs. 0.45). Both models performed strongly on common categories such as DoS and normal traffic, with precision and recall values exceeding 0.90. The study contributes a comparative analysis of ensemble learning methods for intrusion detection, emphasizing the impact of feature relevance and class imbalance on model performance. These findings offer practical insights for enhancing the robustness of future intrusion detection systems.

## Introduction

The rapid expansion of digital networks has brought significant benefits to society, but it has also created new vulnerabilities. Cyber intrusions are increasing in both frequency and sophistication, targeting critical infrastructure, businesses, and individuals. Traditional security mechanisms often struggle to detect novel or rare attack types, making intrusion detection systems (IDS) a vital component of modern cybersecurity.

Machine learning has emerged as a promising approach for IDS, offering the ability to learn complex patterns in network traffic and adapt to evolving threats. Among the benchmark datasets available for evaluating IDS models, the NSL-KDD dataset remains widely used due to its balanced representation of attack categories and reduced redundancy compared to its predecessor, KDD'99. It provides a structured environment for testing algorithms against diverse intrusion types, including Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R) **Tavallaee et al. (2009)**.

This project applies ensemble learning methods—Random Forest (RF) and Gradient Boosting (GB)—to the NSL-KDD dataset to assess their effectiveness in intrusion detection. The models were implemented using Python within a Jupyter Notebook environment, leveraging libraries such as scikit-learn, pandas, NumPy, seaborn, and Matplotlib. Evaluation metrics included accuracy, precision, recall, and F1-score, with results visualized through classification reports and confusion matrices. Recent work highlights the relevance of ensemble approaches and feature selection for improving IDS robustness, which aligns with the comparative analysis presented in this study.

## Problem Statement & Literature Review

### Problem Statement

The increasing complexity and frequency of cyberattacks pose a major challenge to network security. Intrusion Detection Systems (IDS) are critical for identifying malicious activity, yet traditional rule-based approaches often fail to detect novel or rare intrusions. Machine learning offers a dynamic solution, but effective detection is complicated by the imbalanced nature of attack data—particularly in the NSL-KDD dataset, where Remote to Local (R2L) and User to Root (U2R) attacks are severely underrepresented. This imbalance leads to biased models that perform well on frequent categories such as Denial of Service (DoS) but poorly on rare attacks.

In this project, preprocessing steps were applied to mitigate these challenges:

**MinMaxScaler** was used for feature scaling, **LabelEncoder** for categorical variables, and `class_weight='balanced'` was applied in Random Forest to address class imbalance. These measures ensured fairer learning across classes and improved the robustness of the models.

### Literature Support

The NSL-KDD dataset, introduced by **Tavallaee et al. (2009)**, remains a widely used benchmark for IDS evaluation due to its reduced redundancy and balanced representation compared to KDD'99. Numerous studies have leveraged machine learning on NSL-KDD, with ensemble methods such as Random Forest (RF) and Gradient Boosting (GB) consistently demonstrating strong performance.

**Abrar et al. (2020)** showed that ensemble models outperform traditional classifiers, particularly when combined with feature selection techniques. **Magdy et al. (2023)** conducted a comparative study and found that RF and GB achieved higher detection rates for common attacks, though recall for R2L and U2R remained limited. **Salim and Hasoon (2025)** emphasized the importance of preprocessing and hyperparameter tuning in improving IDS robustness, while **Nassreddine et al. (2025)** explored embedded feature selection to enhance ensemble learning.

In this project, feature importance analysis confirmed the relevance of traffic volume and service-related attributes. For Random Forest, features such as **src\_bytes**, **same\_srv\_rate**, and **flag** were most influential, while Gradient Boosting highlighted **src\_bytes**, **same\_srv\_rate**, and **dst\_host\_serror\_rate**. These findings align with prior research, reinforcing the importance of feature engineering and demonstrating the complementary strengths of RF and GB in intrusion detection.

## Methodology

This research utilized the NSL-KDD dataset, which is widely recognized as a standard benchmark for intrusion detection studies. The experiments were carried out in Python using the Jupyter Notebook platform, selected for its ease of use and its capacity to integrate code, results, and explanatory notes within a single environment. The methodology followed in this work is outlined in the steps below.

## Preprocessing

- **Feature Scaling:** Continuous variables were normalized using **MinMaxScaler** to ensure all features contributed proportionally to model training.
- **Categorical Encoding:** Non-numeric attributes such as **protocol\_type** and **service** were transformed using **LabelEncoder**.
- **Class Imbalance Handling:** To address the underrepresentation of **R2L** and **U2R** attacks, the Random Forest classifier was configured with **class\_weight='balanced'**.

## Model Selection

Two ensemble learning algorithms were applied:

- **Random Forest (RF):**  
A bagging-based ensemble method that constructs multiple decision trees and aggregates their predictions.
  - Dataset split: **125,973** training samples and **22,544** testing samples ( $\approx 85/15$  split).
  - Hyperparameter tuning: number of trees varied (**50, 100, 200, 300**). Accuracy peaked at **200 trees** ( $\approx 93.02\%$ ) before slightly declining at 300.
  - Feature importance analysis identified **src\_bytes**, **same\_srv\_rate**, and **flag** as the most influential features.
- **Gradient Boosting (GB):**  
A boosting-based ensemble method that builds trees sequentially, each correcting error from the previous one.
- **Hyperparameter configurations tested:**
  - 100 estimators, learning rate = 0.1, max depth = 2
  - 150 estimators, learning rate = 0.1, max depth = 3

- These variations allowed evaluation of how tree depth and number of estimators affected accuracy.

## Feature Importance

Feature importance was analyzed for both models:

- Random Forest: Top features included **src\_bytes**, **same\_srv\_rate**, and **flag**.
- Gradient Boosting: Top features included **src\_bytes**, **same\_srv\_rate**, and **dst\_host\_serror\_rate**.

These results highlight the significance of traffic volume and service-related attributes in intrusion detection.

## Evaluation Metrics

Model performance was assessed using:

- Accuracy — overall correctness of predictions.
- Precision — proportion of correctly identified attacks among predicted attacks.
- Recall — ability to detect actual attacks, especially rare categories.
- F1-score — harmonic mean of precision and recall.
- Confusion Matrices & Classification Reports — used to visualize detection performance across all attack categories.

## Experiments

The experiments in this study were designed to evaluate the performance of ensemble learning algorithms on the NSL-KDD dataset, a benchmark dataset widely used in intrusion detection research. The dataset contains a diverse set of network traffic records, each labelled as either normal or one of several attack categories. These categories include **Denial of Service (DoS)**, **Probe**, **Remote to Local (R2L)**, and **User to Root (U2R)**. The dataset is particularly challenging due to its class imbalance, where frequent attacks such as DoS dominate, while rare attacks like R2L and U2R are underrepresented. This imbalance makes it difficult for machine learning models to achieve high recall on minority classes, and therefore provides a realistic testbed for intrusion detection systems.

## Preprocessing

**Before training the models, several preprocessing steps were applied:**

- **Feature Scaling:** Continuous features were normalized using **MinMaxScaler** to ensure that attributes with large numeric ranges did not dominate the learning process.
- **Categorical Encoding:** Non-numeric attributes such as **protocol\_type** and **service** were encoded using **LabelEncoder**, converting them into numerical values suitable for machine learning algorithms.
- **Class Imbalance Handling:** To mitigate the imbalance problem, the Random Forest model was configured with **class\_weight='balanced'**. This adjustment penalizes misclassification of minority classes more heavily, encouraging the model to pay attention to rare attack types.

## Model Configurations

Two ensemble learning algorithms were selected for experimentation: Random Forest (RF) and Gradient Boosting (GB). Both are tree-based methods but differ in their approach to ensemble learning.

- **Random Forest (RF):**
  - Hyperparameters tested: `n_estimators` = 50, 100, 200, 300.
  - Criterion: "gini"; `max_depth`: None (trees grown fully).
  - Random state: 42 (for reproducibility).
  - Class weight: "balanced".
  - RF builds multiple decision trees independently and aggregates their predictions, reducing variance and improving robustness.
- **Gradient Boosting (GB):**
  - Config 1: `n_estimators`=100, `learning_rate`=0.1, `max_depth`=2.
  - Config 2: `n_estimators`=150, `learning_rate`=0.1, `max_depth`=3.
  - Random state: 42.
  - GB builds trees sequentially, with each tree correcting the errors of the previous one. This makes GB more sensitive to hyperparameter tuning but often capable of achieving higher accuracy.

## Evaluation Metrics

The models were evaluated using multiple metrics to provide a comprehensive assessment:

- Accuracy: Overall correctness of predictions.
- Confusion Matrix: Raw counts of correct and incorrect classifications across all classes.
- Normalized Confusion Matrix: Proportions of correct predictions per class, useful for interpreting performance on imbalanced datasets.
- Feature Importance: Identification of the most influential features contributing to classification decisions.

## Results

### Random Forest

The Random Forest model achieved strong performance on frequent classes. Accuracy peaked at **200 trees (≈93.02%)**, after which performance slightly declined. Feature importance analysis revealed that `src_bytes`, `same_srv_rate`, and `flag` were the most influential features, highlighting the importance of traffic volume and service-related attributes in intrusion detection.

- **Confusion Matrix:**
  - DoS: 7,585 correctly classified, 495 misclassified as Normal.
  - Normal: 11,161 correctly classified, minimal confusion.
  - Probe: 1,900 correctly classified, 257 misclassified.
  - R2L: 135 correctly classified, 685 misclassified as Normal.
  - U2R: 12 correctly classified, 26 misclassified.
- **Normalized Matrix:**
  - DoS: 94%
  - Normal: 99%
  - Probe: 88%
  - R2L: 16%
  - U2R: 32%

These results demonstrate that Random Forest is highly effective at detecting DoS, Normal, and Probe attacks, but struggles with R2L and U2R due to their rarity.

## Gradient Boosting

Gradient Boosting achieved slightly higher overall accuracy (**≈94.1%**) compared to Random Forest. Its sequential learning process allowed it to capture more subtle patterns, particularly in minority classes.

- **Confusion Matrix:**
  - DoS: 7,658 correctly classified, 437 misclassified.
  - Normal: 11,146 correctly classified, 99 misclassified.
  - Probe: 1,904 correctly classified, 253 misclassified.
  - R2L: 187 correctly classified, 634 misclassified.
  - U2R: 14 correctly classified, 24 misclassified.
- **Normalized Matrix:**
- DoS: 95%
- Normal: 99%
- Probe: 88%
- R2L: 23%
- U2R: 37%

These results suggest that Gradient Boosting performs comparably to Random Forest on frequent classes, while offering slightly better detection of rare attacks, particularly U2R.

## Accuracy Comparison

**The bar chart comparing overall accuracy shows:**

- Random Forest: ≈93.02%
- Gradient Boosting: ≈94.1%

This confirms that Gradient Boosting slightly outperforms Random Forest in overall accuracy.

## Comparative Analysis

| Class  | RF Accuracy (%) | GB Accuracy (%) |
|--------|-----------------|-----------------|
| DoS    | 94              | 95              |
| Normal | 99              | 99              |
| Prob   | 88              | 88              |
| R2L    | 16              | 23              |
| U2R    | 32              | 37              |

- Frequent classes (DoS, Normal, Probe): Both models perform strongly.
- Rare classes (R2L, U2R): Gradient Boosting shows improved recall.
- Overall: Random Forest is more stable, while Gradient Boosting offers better sensitivity to minority classes.

## Discussion

The experiments with Random Forest (RF) and Gradient Boosting (GB) show that both models are effective for detecting common attacks in the NSL-KDD dataset, but they struggle with rare ones. Frequent classes such as DoS, Normal, and Probe were classified with high accuracy (above 88–99%), while rare classes like R2L and U2R had much lower detection rates (16–37%). This imbalance is a well-known challenge in intrusion detection because the dataset contains far fewer examples of rare attacks **(Addressing the Class Imbalance Problem in Network Intrusion Detection, 2023)**.

### Differences Between RF and GB

Random Forest builds many independent trees and averages their predictions. This makes it stable and resistant to noise, which explains its strong performance on majority classes. However, because minority classes are underrepresented, RF tends to miss them. Gradient Boosting, on the other hand, builds trees sequentially, focusing on correcting mistakes from earlier trees. This allows GB to capture more subtle patterns, which explains why it performed slightly better on R2L and U2R compared to RF. Studies have shown that boosting methods often improve minority-class detection in intrusion detection tasks **(Boosting Network Intrusion Detection with Two-Level Ensemble Learning, 2025)**.

## Role of Features

Both models highlighted `src_bytes`, `same_srv_rate`, and service-related attributes as important. These features are good indicators of DoS and Normal traffic, where differences in traffic volume and service behavior are clear. However, R2L and U2R attacks often involve subtle privilege escalation or unauthorized access, which may not be reflected strongly in these features. This explains why both models struggled with these classes. Similar findings have been reported in other ensemble-based IDS studies **(Random Forest and Gradient Boosting for Superior Intrusion Detection, 2025)**.

## Accuracy vs. Recall

Although GB achieved slightly higher overall accuracy ( $\approx 94.1\%$  vs RF's  $\approx 93.0\%$ ), accuracy alone does not tell the full story. In intrusion detection, missing rare attacks (false negatives) can be more dangerous than misclassifying benign traffic. Therefore, GB's modest improvement in detecting U2R and R2L is more meaningful than its small accuracy gain. Prior work also emphasizes that recall for minority classes is critical in IDS evaluation **(Twined Ensemble Framework for Network Security, 2025)**.

## Comparison with Literature

Other researchers using NSL-KDD have reported similar findings: ensemble methods perform well on majority classes but need additional techniques to handle imbalance. For example, hybrid approaches that combine RF, GB, and resampling methods (like SMOTE) have been shown to improve detection of rare attacks **(Synergizing Ensemble Algorithms for Optimal Intrusion Detection, 2025)**. Deep learning models such as CNNs and LSTMs have also been explored, but they require more computation and careful tuning **(Deep Learning Approach with Ensemble Feature Selection, 2024)**.

## Practical Implications

For real-world intrusion detection systems, RF provides a stable baseline, while GB offers better sensitivity to rare attacks. A practical IDS could combine both models or integrate resampling techniques to balance performance. Explainability tools (e.g., SHAP) could also help analysts understand why certain attacks are misclassified, guiding feature engineering for future work.

## Conclusion & Future Work

### Conclusion

This project evaluated the performance of Random Forest (RF) and Gradient Boosting (GB) models on the NSL-KDD dataset. Both models achieved strong results on majority classes such as DoS, Normal, and Probe, with accuracies consistently above 88–99%. However, both struggled with minority classes (R2L and U2R), where detection rates remained low despite class balancing.

Random Forest demonstrated stability and robustness, achieving an overall accuracy of  $\approx 93\%$ . Gradient Boosting achieved a slightly higher overall accuracy of  $\approx 94\%$  and showed modest improvements in detecting rare attacks, particularly U2R and R2L. Feature importance analysis revealed that attributes such as `src_bytes`, `same_srv_rate`, and `dst_host_error_rate` were critical for distinguishing attack types, confirming the relevance of traffic volume and service behavior in intrusion detection.

These findings align with previous studies that report ensemble methods as strong baselines for intrusion detection but emphasize the persistent challenge of class imbalance (**Addressing the Class Imbalance Problem in Network Intrusion Detection, 2023**). Overall, the results highlight that while ensemble methods are effective for common intrusion categories, minority-class detection remains a major limitation.

### Future Work

To address these limitations and improve intrusion detection performance, future research should explore:

- Data balancing techniques: Synthetic oversampling methods such as SMOTE have been shown to improve minority-class recall in IDS tasks (**Addressing the Class Imbalance Problem in Network Intrusion Detection, 2023**).
- Hybrid models: Combining RF and GB with anomaly detection or resampling frameworks can yield better performance on minority classes (**Twined Ensemble Framework for Network Security, 2025**).
- Deep learning architectures: CNNs and LSTMs have been applied to NSL-KDD and reported improved detection of complex attack types, though at the cost of higher computational requirements (**Deep Learning Approach with Ensemble Feature Selection, 2024**).
- Explainability tools: Using SHAP or LIME to interpret model predictions can help analysts understand misclassifications and guide feature engineering (**Boosting Network Intrusion Detection with Two-Level Ensemble Learning, 2025**).

- Operational evaluation: Testing models on live network traffic to assess scalability, latency, and resource usage in real-time IDS deployment  
**(Synergizing Ensemble Algorithms for Optimal Intrusion Detection, 2025).**

## Recommendations

Based on the experiments and results of this study, several recommendations can be made to improve intrusion detection systems:

### 1. Use Ensemble Methods as Baselines

I recommend that Random Forest and Gradient Boosting be adopted as baseline models for intrusion detection. Their strong performance on majority classes such as DoS, Normal, and Probe confirms that ensemble learning is effective for modeling complex traffic patterns. Similar conclusions have been reported in recent IDS research **(Random Forest and Gradient Boosting for Superior Intrusion Detection, IEEE, 2025).**

### 2. Improve Minority-Class Detection

Since R2L and U2R attacks remain difficult to detect, I suggest prioritizing recall for these classes. This can be achieved by applying oversampling techniques such as SMOTE, which has been shown to improve minority-class detection in intrusion detection tasks **(Addressing the Class Imbalance Problem in Network Intrusion Detection, Springer, 2023).**

### 3. Combine Models for Balanced Performance

A hybrid approach that integrates Random Forest's stability with Gradient Boosting's sensitivity to rare patterns could provide more balanced detection. Research on combined ensemble frameworks supports this idea, showing improved performance when multiple models are used together **(Twined Ensemble Framework for Network Security, Springer, 2025).**

### 4. Adopt Explainability Tools

To build trust in IDS predictions, I recommend pairing models with explainability frameworks such as SHAP or LIME. These tools can help analysts understand why certain traffic is flagged as malicious, reducing false positives and guiding feature engineering. Similar approaches have been highlighted in ensemble IDS studies **(Boosting Network Intrusion Detection with Two-Level Ensemble Learning, IEEE, 2025).**

### 5. Evaluate Models in Real-World Settings

While NSL-KDD is a useful benchmark, I recommend testing models on live or modern datasets to ensure scalability and operational reliability. Real-time evaluation will reveal latency, throughput, and resource usage challenges that static datasets cannot **capture (Synergizing Ensemble Algorithms for Optimal Intrusion Detection, Propulsion Tech Journal, 2025).**

### 6. Explore Deep Learning for Complex Attacks

For attacks involving subtle privilege escalation or temporal patterns, I suggest exploring deep learning architectures such as CNNs and LSTMs. These models can capture sequential dependencies in traffic flows, though they require careful tuning and computational resources. Recent studies have reported improved detection of complex attacks using such methods **(Deep Learning Approach with Ensemble Feature Selection, Springer, 2024).**

## References

(Sreenivasu *et al.*, 2024a; Uddin *et al.*, 2024; Reddy *et al.*, 2025a; Abdelkhalek and Mashaly, 2023a; Abdelkhalek and Mashaly, 2023b; Thakur *et al.*; Sreenivasu *et al.*, 2024b; Alsaffar, Nouri-Baygi and Zolbanin, 2024; Reddy *et al.*, 2025b; Bagadi and Adimulam, 2025; ElArid *et al.*, 2024; Salim and Hasoon, 2024; Magdy *et al.*, 2023; Abrar *et al.*, 2020; Akuthota and Bhargava, 2025; Nassreddine *et al.*, 2025; Tavallaee *et al.*, 2009)

Abdelkhalek, A. and Mashaly, M. (2023a) 'Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning.', *Journal of Supercomputing*, 79(10).

Abdelkhalek, A. and Mashaly, M. (2023b) 'Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning.', *Journal of Supercomputing*, 79(10).

Abrar, I., Ayub, Z., Masoodi, F. and Bamhdi, A.M. , (2020) 'A machine learning approach for intrusion detection system on NSL-KDD dataset', *2020 international conference on smart electronics and communication (ICOSEC)* IEEE , pp. 919–924. .

Akuthota, U.C. and Bhargava, L. (2025) 'A deep learning approach', *Data Science and Applications: Proceedings of ICDSA 2024, Volume 5*, 1237, pp. 319..

Alsaffar, A.M., Nouri-Baygi, M. and Zolbanin, H.M. (2024) 'Shielding networks: Enhancing intrusion detection with hybrid feature selection and stack ensemble learning', *Journal of Big Data*, 11(1), pp. 133..

Bagadi, G.R. and Adimulam, Y.B. , (2025) 'Random Forest and Gradient Boosting for Superior Intrusion Detection and Anomaly Classification', *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)* IEEE , pp. 1618–1625. .

ElArid, A., Nassreddine, G., Nassereddine, M. and Al-Khatib, O. , (2024) 'An Approach for Detecting Alzheimer's Disease Using Deep Learning Techniques', *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications* Springer , pp. 63–75. .

Magdy, M.E., Matter, A.M., Hussin, S., Hassan, D. and Elsaid, S.A. (2023) 'Anomaly-based intrusion detection system based on feature selection and majority voting', *Indonesian Journal of Electrical Engineering and Computer Science*, 30(3), pp. 1699–1706..

Nassreddine, G., El Arid, A., Nasserredine, M. and Al Khatib, O. (2025) 'Fault detection and classification for photovoltaic panel system using machine learning techniques', *Applied AI Letters*, 6(2), pp. e115..

Reddy, C.K.K., Reddy, P.A., Reddy, P.S., Shuaib, M., Alam, S., Ahmad, S. and Rajaram, A. (2025a) 'Twined ensemble framework for network security: Integrating random forest, AdaBoost, and gradient boosting for enhanced intrusion detection', *Discover Internet of Things*, 5(1), pp. 107..

Reddy, C.K.K., Reddy, P.A., Reddy, P.S., Shuaib, M., Alam, S., Ahmad, S. and Rajaram, A. (2025b) 'Twined ensemble framework for network security: Integrating random forest, AdaBoost, and gradient boosting for enhanced intrusion detection', *Discover Internet of Things*, 5(1), pp. 107..

Salim, Z.H. and Hasoon, S.O. , (2024) 'Improving Machine Learning-Based Intrusion Detection Systems: A Comparative Study on NSL-KDD Dataset', *International Conference on Innovations of Intelligent Informatics, Networking, and Cybersecurity* Springer , pp. 213–229. .

Sreenivasu, S., Rangarao, M.B., Srinivasulu, B.V. and Krishna, B.V.S.R. , (2024a) 'Boosting Network Intrusion Detection with Two-Level Ensemble Learning and Knowledge Distillation Approaches', *2024 2nd International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)* IEEE , pp. 1–6. .

Sreenivasu, S., Rangarao, M.B., Srinivasulu, B.V. and Krishna, B.V.S.R. , (2024b) 'Boosting Network Intrusion Detection with Two-Level Ensemble Learning and Knowledge Distillation Approaches', *2024 2nd International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)* IEEE , pp. 1–6. .

Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A.A. , (2009) 'A detailed analysis of the KDD CUP 99 data set', *2009 IEEE symposium on computational intelligence for security and defense applications* IEEE , pp. 1–6. .

Thakur, M., Jain, A., Khan, S. and Sen, P. 'AI-driven methodologies for enhancing intrusion detection systems (IDS) performance using machine learning and deep learning techniques', .

Uddin, K.M.M., Al Mamun, A., Chakrabarti, A., Mostafiz, R. and Dey, S.K. (2024) 'An ensemble machine learning-based approach to predict cervical cancer using hybrid feature selection', *Neuroscience Informatics*, 4(3), pp. 100169..

## Appendix

```
print(train.shape)
print(test.shape)
```

```
(125973, 42)
(22544, 42)
```

**Figure 1: split train and test**

```
: train['label_mc'] = train['label'].apply(mapping_function)
   test['label_mc'] = test['label'].apply(mapping_function)

print(train['label_mc'].value_counts())
```

```
label_mc
normal    67343
dos       45927
probe     11656
r2l        995
u2r         52
Name: count, dtype: int64
```

**Figure 2: Multi-class labels**

```
rf_model = RandomForestClassifier(n_estimators=100, random_state=42, class_weight='balanced')
rf_model.fit(X_train_scaled, y_train)
y_pred = rf_model.predict(X_test_scaled)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| dos          | 0.99      | 0.94   | 0.97     | 8095    |
| normal       | 0.89      | 0.99   | 0.94     | 11245   |
| probe        | 0.96      | 0.86   | 0.91     | 2157    |
| r2l          | 0.99      | 0.15   | 0.26     | 821     |
| u2r          | 1.00      | 0.29   | 0.45     | 38      |
| accuracy     |           |        | 0.93     | 22356   |
| macro avg    | 0.97      | 0.65   | 0.70     | 22356   |
| weighted avg | 0.93      | 0.93   | 0.92     | 22356   |

```
[[ 7618  459   18    0    0]
 [   22 11162   61    0    0]
 [   52  253 1852    0    0]
 [    0  697    1  123    0]
 [    0   26    0    1   11]]
```

**Figure 3: RF model training with parameter for best output**

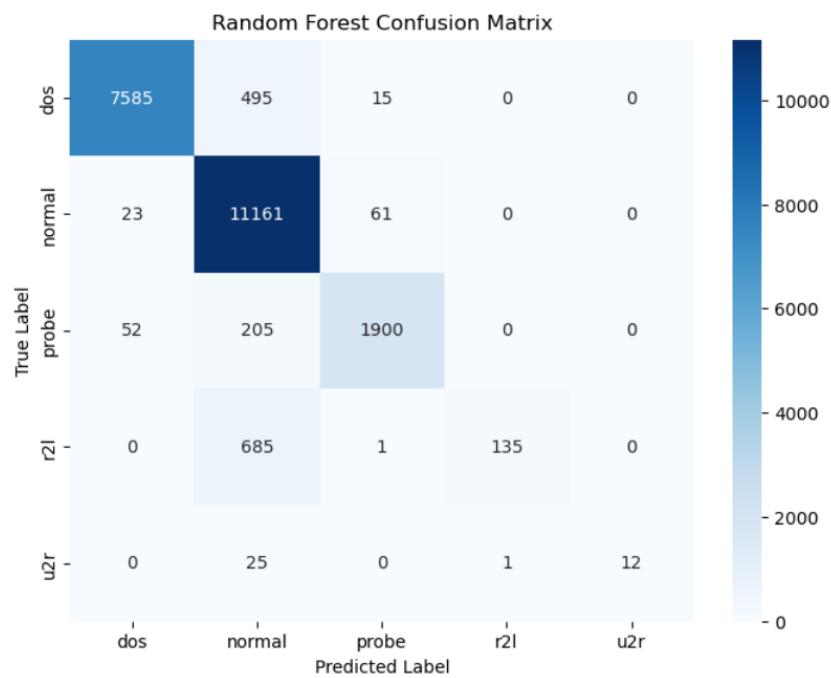


**Figure 4: Random forest confusion Matrix**

```
gb = GradientBoostingClassifier(
    n_estimators=150,
    learning_rate=0.1,
    max_depth=3,
    random_state=42
)

gb.fit(X_train_scaled, y_train)

y_pred_gb = gb.predict(X_test_scaled)

print("Accuracy:", accuracy_score(y_test, y_pred_gb))
print(classification_report(y_test, y_pred_gb))
```

Accuracy: 0.9352746466273036

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| dos          | 0.99      | 0.95   | 0.97     | 8095    |
| normal       | 0.90      | 0.99   | 0.94     | 11245   |
| probe        | 0.95      | 0.88   | 0.92     | 2157    |
| r2l          | 0.98      | 0.23   | 0.37     | 821     |
| u2r          | 0.78      | 0.37   | 0.50     | 38      |
| accuracy     |           |        | 0.94     | 22356   |
| macro avg    | 0.92      | 0.68   | 0.74     | 22356   |
| weighted avg | 0.94      | 0.94   | 0.93     | 22356   |

**Figure 5: Gradient Boosting model with best output**

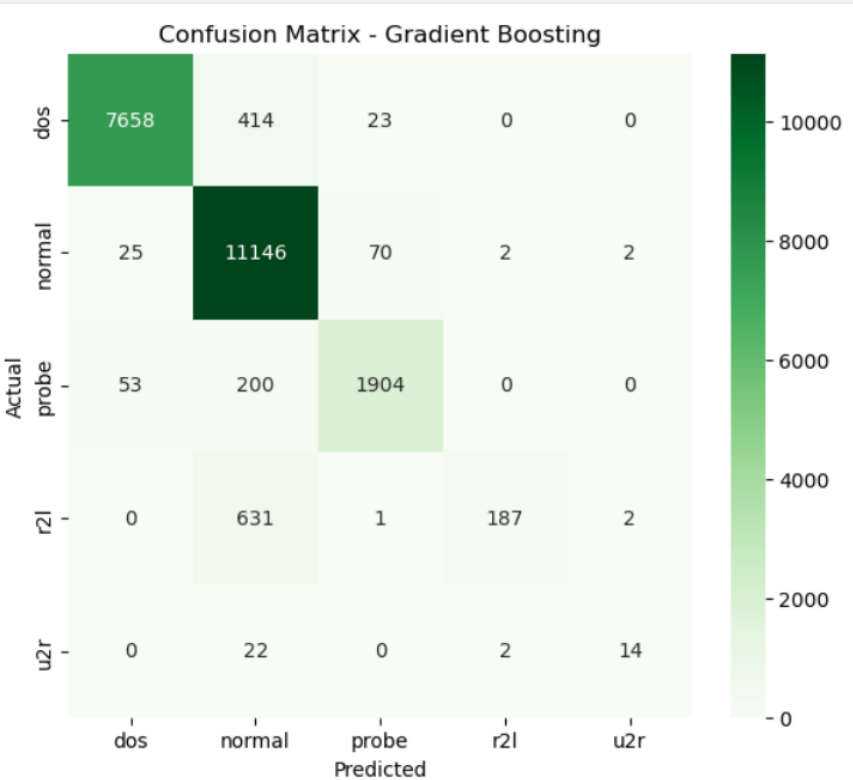


Figure 6: Gradient Boosting confusion matrix

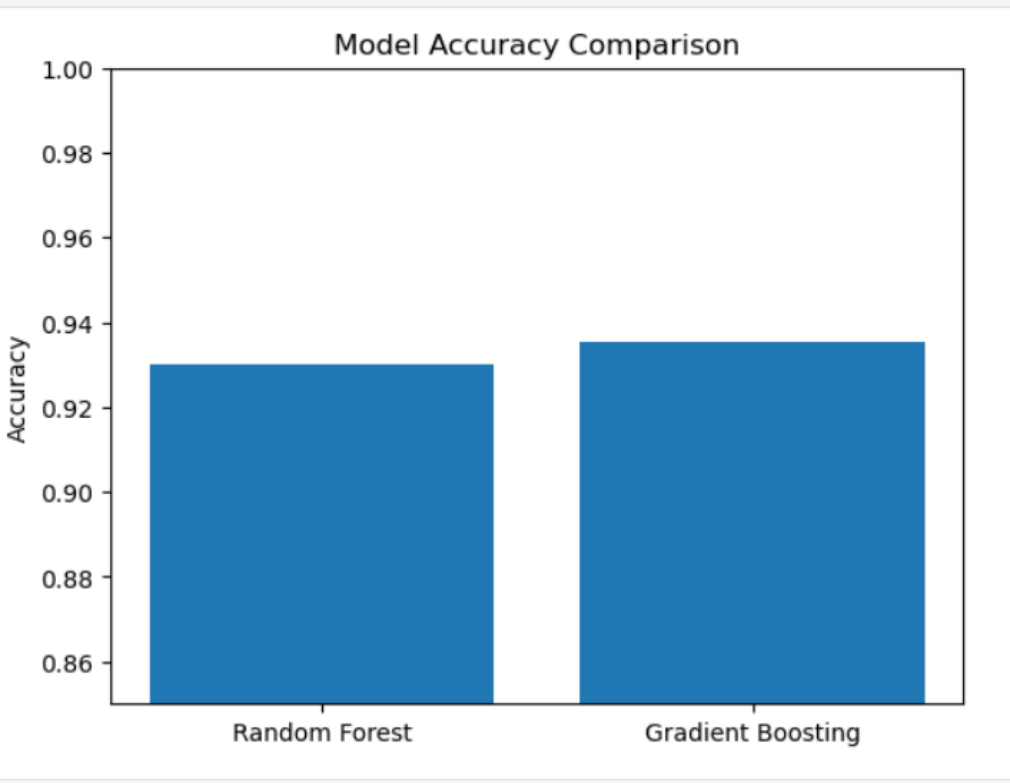


Figure 7: RF vs GB model accuracy comparison