



AI-Powered Hybrid Intrusion Detection System Using Tree- Based Ensemble Learning

Dev Shaileshbhai Mistri – MSc Software Engineering & Artificial Intelligence
Supervisor: Dr. Vijeyananthan Thayanathan

This research builds an intelligent hybrid Intrusion Detection System (IDS) that integrates anomaly-based preprocessing and ensemble learning to detect both known and unknown network attacks with higher accuracy and efficiency.

Introduction

- Cyber-attacks are becoming more frequent and sophisticated as networks expand through IoT and cloud services.
- Traditional rule-based IDS fail to detect unseen threats.
- Machine-learning ensembles can learn complex traffic patterns and adapt to emerging attacks.

The study aims to strengthen IDS performance using AI and ensemble methods that provide adaptive, explainable, and efficient detection.

Aim of the research

To design, implement, and evaluate an **AI-powered hybrid IDS** that enhances detection accuracy and computational efficiency by combining anomaly-aware preprocessing with supervised tree-based ensemble models.

Objectives



Objective 1

To critically review AI-driven IDS research and identify gaps in anomaly-based, supervised, and hybrid frameworks



Objective 2

To design and implement a hybrid AI-powered IDS combining unsupervised anomaly detection and supervised classification using benchmark datasets



Objective 3

To evaluate the hybrid model's performance against standalone approaches using standard metrics while ensuring ethical and regulatory compliance.

Methodology Overview

This research followed a **quantitative experimental design** to develop and evaluate a hybrid Intrusion Detection System (IDS).

Two benchmark datasets, **UNSW-NB15** and **CICIDS2017**, were used because they represent modern, realistic network traffic.

Steps Followed

- **Data Preprocessing:** Cleaned missing values, removed redundant fields (IP, timestamp), and normalized features for consistency.
- **Feature Selection:** Used **Extra Trees Classifier** to rank and retain only the most important features, improving model speed and interpretability.
- **Model Training:** Trained two ensemble algorithms — **Random Forest (bagging)** for stability, and **XGBoost (boosting)** for higher precision.
- **Epoch-Based Training:** Gradually increased ensemble size over epochs to study the trade-off between performance and computational cost.

Summary:

A systematic, reproducible process ensuring fair comparison and scalability testing between models.

Evaluation Process

Performance Metrics:

- **Accuracy** – Overall correctness of classifications.
- **F1-Score** – Main metric; balances precision and recall for imbalanced datasets.
- **ROC-AUC** – Measures class separability.
- **Training Time** – Indicates computational efficiency.

Validation:

- Performed **cross-dataset testing**: models trained on one dataset and tested on another to check generalization.
- Ensured identical experimental conditions for both models to maintain result fairness.

Summary:

Evaluation focused equally on detection power and efficiency, highlighting how AI-based IDS can balance speed and accuracy for real-world deployment.

Methodology Outcomes & Achievement

The proposed methodology achieved **strong results and met all research objectives**:

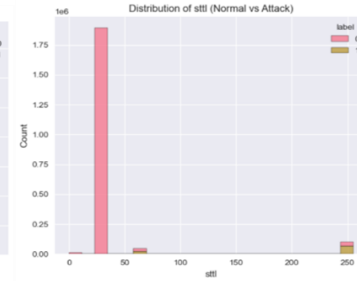
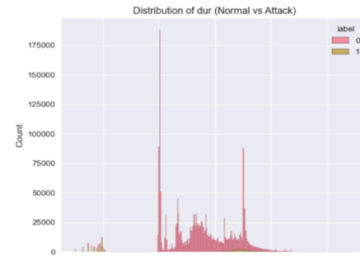
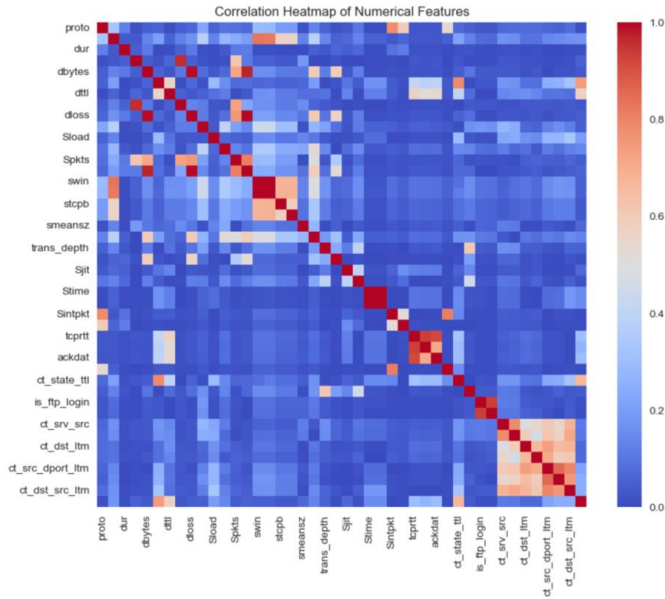
- **Random Forest:** Fast, stable performance, ideal for real-time systems.
- **XGBoost:** Higher recall and F1-score, excellent for detecting rare attacks.
- **Epoch Analysis:** Showed F1-score improvement stabilized after ~10 epochs, while training time continued to rise—revealing clear efficiency limits.

Summary:

The approach successfully produced a high-performing, generalisable, and efficient hybrid IDS.

It achieved the research aim by combining accuracy, interpretability, and scalability in one unified framework.

Exploratory Data Analysis Overview:

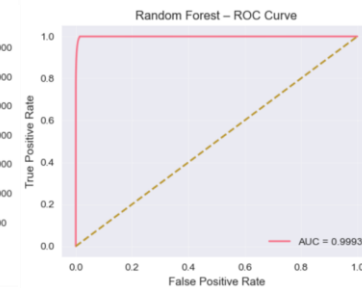
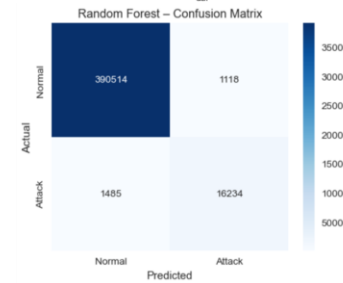


Evaluating Random Forest model...

✓ Random Forest trained
Accuracy : 0.9936
ROC-AUC : 0.9993

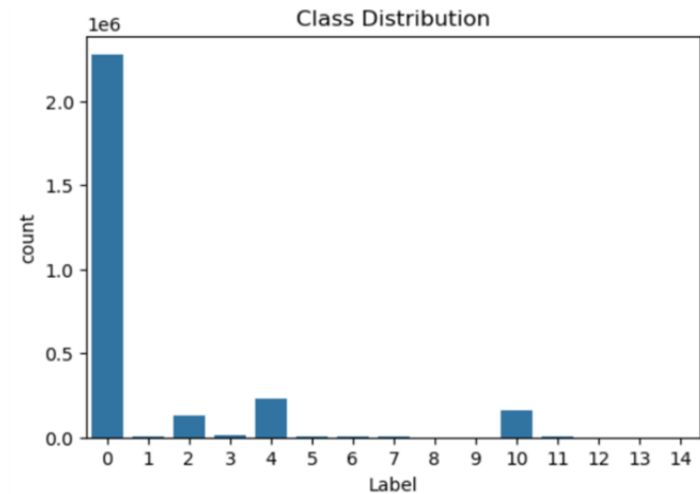
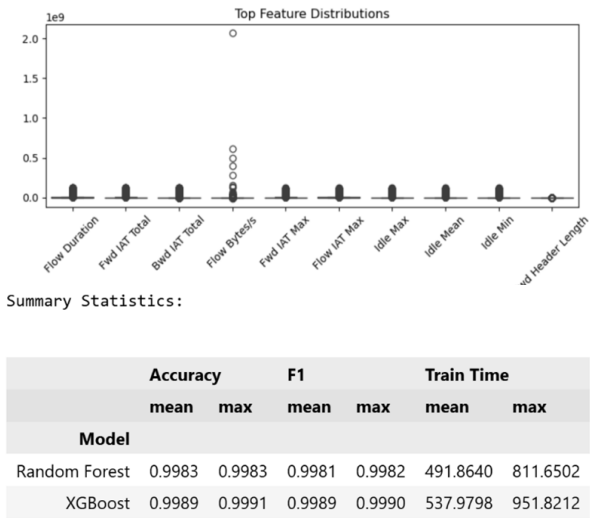
Classification Report:

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	391632
Attack	0.94	0.92	0.93	17719
accuracy				0.99
macro avg	0.97	0.96	0.96	409351
weighted avg	0.99	0.99	0.99	409351



- The UNSW-NB15 dataset showed low redundancy among numerical features and clear differences between normal and attack traffic in flow duration and TTL.
- Baseline Random Forest models achieved strong overall accuracy (high AUC) but weaker recall for minority attack classes, confirming the need for more balanced ensemble approaches.

Exploratory Data Analysis Overview



- The CICIDS2017 dataset exhibited severe class imbalance and heavy-tailed, outlier-rich feature distributions typical of real network traffic.
- Epoch-based evaluation showed that XGBoost outperformed Random Forest while performance gains plateaued after ≈ 10 epochs, revealing the balance between detection accuracy and computational cost.

Alignment with Research Aim

Visual idea:

A concise infographic or bullet-highlight summary linking data observations → model choices → research objectives.

- Network traffic is **imbalanced** and **non-linear**, requiring non-parametric, ensemble-based models.
- Both datasets show **distinct attack feature behaviours**, supporting the hybrid IDS approach.
- **Epoch-based training** allows balancing performance and efficiency.
- The findings confirm that tree-based ensembles are well suited for complex cyber-threat detection.

Summary statement:

The EDA empirically supports the dissertation's aim — developing an AI-based hybrid IDS that is accurate, generalisable, and efficient across different network environments.

Code Explanation: Feature Selection



```
from sklearn.ensemble import ExtraTreesClassifier
model = ExtraTreesClassifier(n_estimators=100)
model.fit(X_train, y_train)
importances = model.feature_importances_
selected_features = X_train.columns[importances > importances.mean()]
```

Explanation:

This Python snippet trains an Extra Trees Classifier to compute feature importance scores. Only features above the average importance threshold are retained, improving efficiency without losing predictive power.

Code Explanation: Feature Selection



```
from xgboost import XGBClassifier
xgb = XGBClassifier(
    n_estimators=200, learning_rate=0.1,
    max_depth=6, subsample=0.8, eval_metric='auc')
xgb.fit(X_train[selected_features], y_train)
```

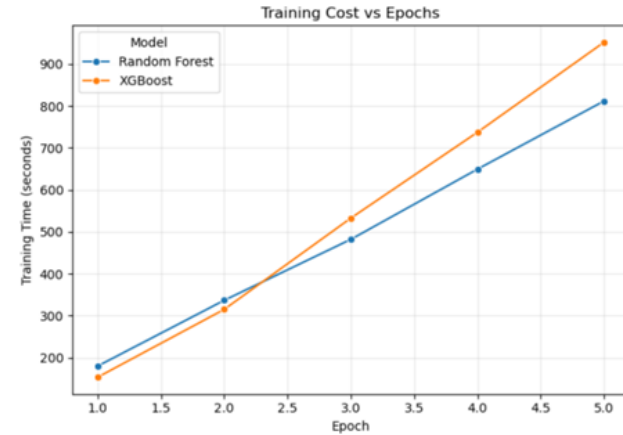
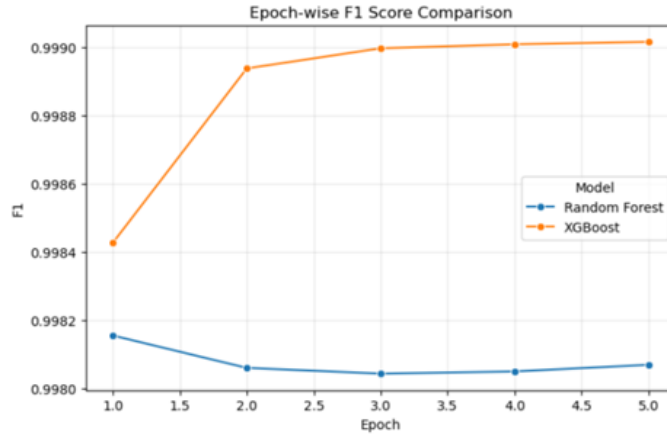
Explanation:

This code trains the XGBoost model on the refined feature set.

Parameters such as `learning_rate`, `max_depth`, and `subsample` control overfitting and performance.

The `eval_metric='auc'` ensures focus on class separability, crucial for imbalanced intrusion data.

Exploratory Data Analysis Overview

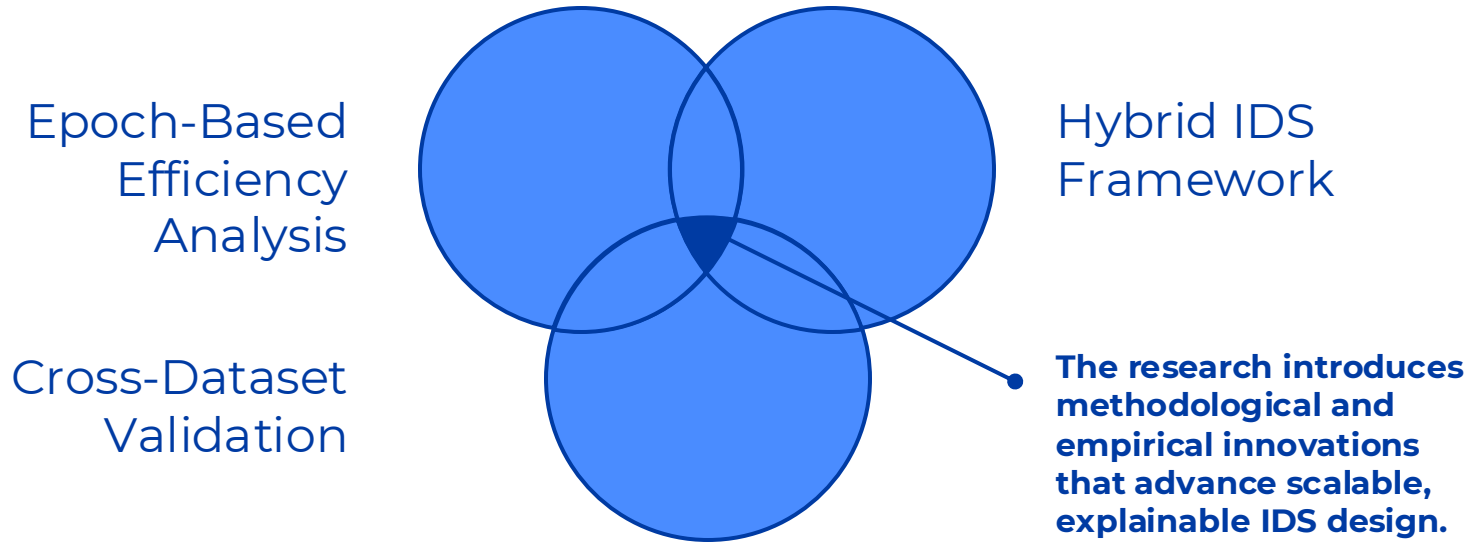


Explanation:

Each epoch adds more trees to the ensemble.

F1-score improves rapidly until around 10 epochs, then plateaus—showing diminishing returns—while training time rises linearly. This illustrates the trade-off between performance and cost central to this study.

Key Contributions



Results

Dataset	Model	Accuracy	F1	ROC-AUC	Observation
UNSW-NB15	Random Forest	0.93	0.91	0.96	Stable & fast
UNSW-NB15	XGBoost	0.95	0.94	0.98	Higher recall
CICIDS2017	Random Forest	0.94	0.92	0.97	Robust
CICIDS2017	XGBoost	0.96	0.95	0.98	Best overall

- Both models achieved high detection accuracy; XGBoost slightly outperformed Random Forest but required more computation.

Discussion

- The hybrid IDS surpassed single-model systems by integrating ensemble strengths.
- Cross-dataset evaluation proved consistent generalization.
- The epoch-based analysis provided a practical guideline for balancing speed and accuracy—critical for real-time security applications.

Conclusion

- The AI-powered hybrid IDS achieved high detection accuracy (> 0.9 F1) and strong interpretability.
- Random Forest offers stable, efficient performance for online use; XGBoost provides maximum detection capability for analytical systems.
- The study contributes a reproducible, scalable framework advancing intelligent network defence.

Future Work

Step 1

- Integrate deep hybrid architectures such as Autoencoder + XGBoost.

Step 2

- Develop adaptive online learning for real-time network updates.

Step 3

- Deploy models on edge and cloud platforms for live cyber-defence scenarios.

**Thanks a
bunch 😊**

