Intrusion Detection System on System Call sequences

Group 18
Amber Gupta 2024AIM1001
Prashant Rawat 2024CSM1015
Yash Narnaware 2024AIM1011

April 27, 2025

1. Executive Summary

This proposal outlines the design and implementation of a Machine Learning-Based Intrusion Detection System (IDS) using the ADFA-LD dataset. The key objectives include developing classifiers using models such as LSTM, GRU, Attention Based Transformer,ANN, Random Forest, Support Vector Machine (SVM) to detect anomalous behaviors effectively. Expected outcomes include achieving high detection accuracy, comparing various algorithms, and analyzing their performance. The methodology involves data preparation, model training, evaluation, and comparison with different algorithms.

2. Introduction

Traditional Intrusion Detection Systems (IDS) rely on predefined signatures, which are inadequate for detecting novel attacks. Therefore, machine learning techniques, particularly models like LSTM, GRU, Attention Based Transformer, ANN, Random Forest, Support Vector Machine (SVM), offer promising solutions for anomaly detection by learning from data patterns. This project aims to enhance IDS using these machine learning models applied to the ADFA-LD dataset.

3. Objectives and Scope

Primary Objectives

- Develop multi-class classifiers using models such as LSTM, GRU, Attention Based Transformer, Random Forest, ANN, Support Vector Machine (SVM) to differentiate between normal and anomalous system call sequences.
- Compare the performance of these models for multi-class classification tasks.

Secondary Objectives

- Provide comparative analysis against different models.

Scope

The project focuses on host-based intrusion detection using the ADFA-LD dataset. Limitations include dataset-specific results and challenges related to class imbalance.

4. Literature Review

Research has demonstrated the efficacy of various machine learning models for IDS. Algorithms like Naive Bayes, Random Forest, SVM, and Ensemble Learning are commonly employed due to their robustness and generalization capabilities. Previous studies have highlighted limitations of traditional approaches and showcased the potential of these models. However, gaps remain in achieving higher accuracy, particularly for multiclass classification.

5. Resource Used

- **Hardware:** High-performance computing systems (GPUs).
- **Software:** Python libraries such as Scikit-Learn, TensorFlow.

6. Methodology

The dataset used to build the Intrusion Detection System (IDS) is ADFA-LD. It contains system call sequences recorded during everyday tasks on a Linux-based system, along with sequences collected during simulated attacks. The ADFA-LD dataset is designed for anomaly detection, and our objective is to classify whether a given system call sequence is normal or anomalous.

The dataset consists of three main folders:

- **Attack Data Master**: Contains system call sequences during various attacks.
- **Training Data Master**: Contains normal system call sequences used for training.
- **Validation Data Master**: Contains normal system call sequences used for validation.

Data Type	**Trace Count**
Normal Training Data	833 Traces
Normal Validation Data	4373 Traces
Attack Data	746 Traces

Table 1: Trace count in dataset

The Attack Data Master folder is further divided into six subfolders corresponding to different types of attacks. The dataset simulates a real-world scenario by containing significantly more normal system call data than attack data.

Table 2: Distribution of Attack Types in the Dataset

Attack Name	**Label**	**Number of Traces**
Normal (not attack)	0	833
Web Shell	1	118
Meterpreter	2	75
Hydra SSH	3	176
Hydra FTP	4	162
Adduser	5	91
Java Meterpreter	6	124

Table 3: Distribution of Attack Types in the Balanced Dataset

Attack Name	**Label**	**Number of Traces**
Normal (not attack)	0	833
Web Shell	1	800
Meterpreter	2	800
Hydra SSH	3	800
Hydra FTP	4	800
Adduser	5	800
Java Meterpreter	6	800

Notably, the validation dataset includes only normal system calls. This aligns with our goal of anomaly detection—learning the patterns of normal behavior and flagging any deviation from it as a potential intrusion, without needing to identify the specific attack type.

- **Data Preparation:** Cleaning, preprocessing, and feature extraction.
- **Model Architecture:** Implementation of models such as LSTM, GRU, Attention Based Transformer, Random Forest, Support Vector Machine (SVM), ANN multi-class classification.
- **Evaluation:** Comparison of performance metrics such as confusion- matrix, accuracy, precision, recall, and F1-score among different models.
- **Risk Mitigation:** Addressing issues related to overfitting, class imbalance, and model generalization.

7. Results

Results for Original(Imbalanced) Dataset

- SVM

Table 4: SVM

Class	Precision	Recall	F1-Score	Support
0	0.92	0.99	0.95	178
1	0.12	0.29	0.17	7
2	0.43	0.40	0.42	25
3	0.50	0.15	0.23	20
4	0.81	0.57	0.67	37
5	0.52	0.64	0.57	25
6	0.55	0.50	0.52	24
Accuracy	0.76 (on 316 samples)			
Macro Avg	0.55	0.50	0.50	316
Weighted Avg	0.76	0.76	0.75	316

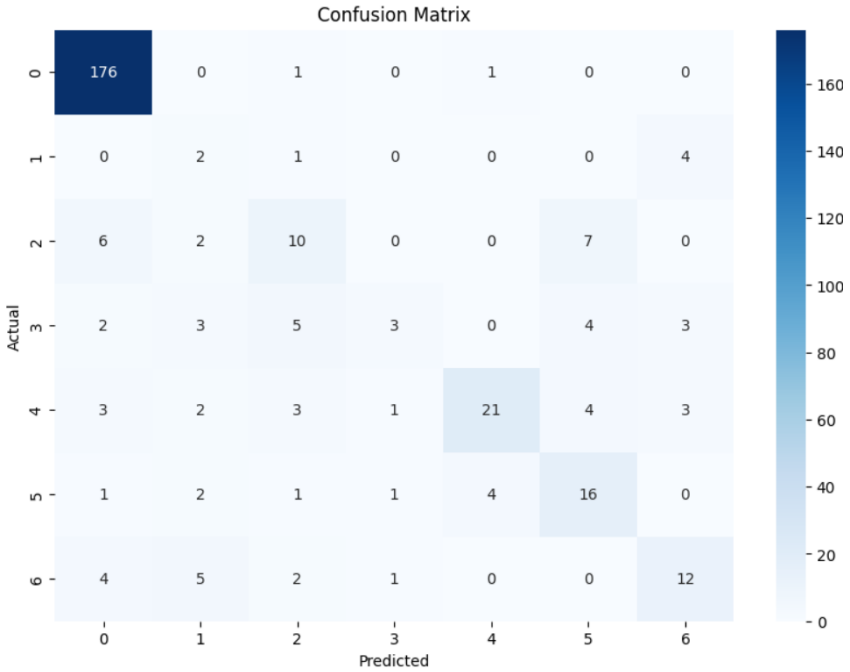

Figure 1: Classification report visualization for the SVM model

- Random Forest Classifier

Table 5: Random Forest Classifier

Class	Precision	Recall	F1-Score	Support
0	0.93	0.99	0.96	178
1	0.25	0.29	0.27	7
2	0.68	0.68	0.68	25
3	0.42	0.25	0.31	20
4	0.88	0.76	0.81	37
5	0.64	0.72	0.68	25
6	0.67	0.58	0.62	24
Accuracy	0.83 (on 316 samples)			
Macro Avg	0.64	0.61	0.62	316
Weighted Avg	0.81	0.83	0.82	316

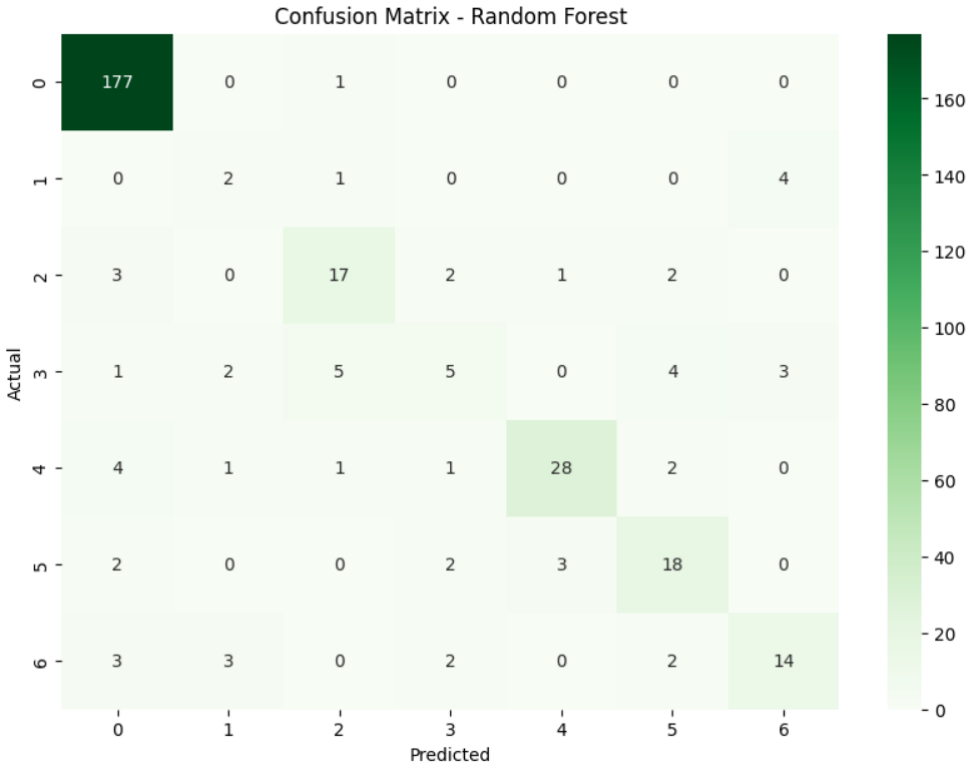

Figure 2: Classification report visualization for the Random Forest model

- ANN

Table 6: ANN

Class	Precision	Recall	F1-Score	Support
0	0.97	0.96	0.96	178
1	0.25	0.57	0.35	7
2	0.69	0.44	0.54	25
3	0.53	0.45	0.49	20
4	0.83	0.78	0.81	37
5	0.45	0.76	0.57	25
6	0.79	0.46	0.58	24
Accuracy	0.80 (on 316 samples)			
Macro Avg	0.64	0.63	0.61	316
Weighted Avg	0.83	0.80	0.81	316

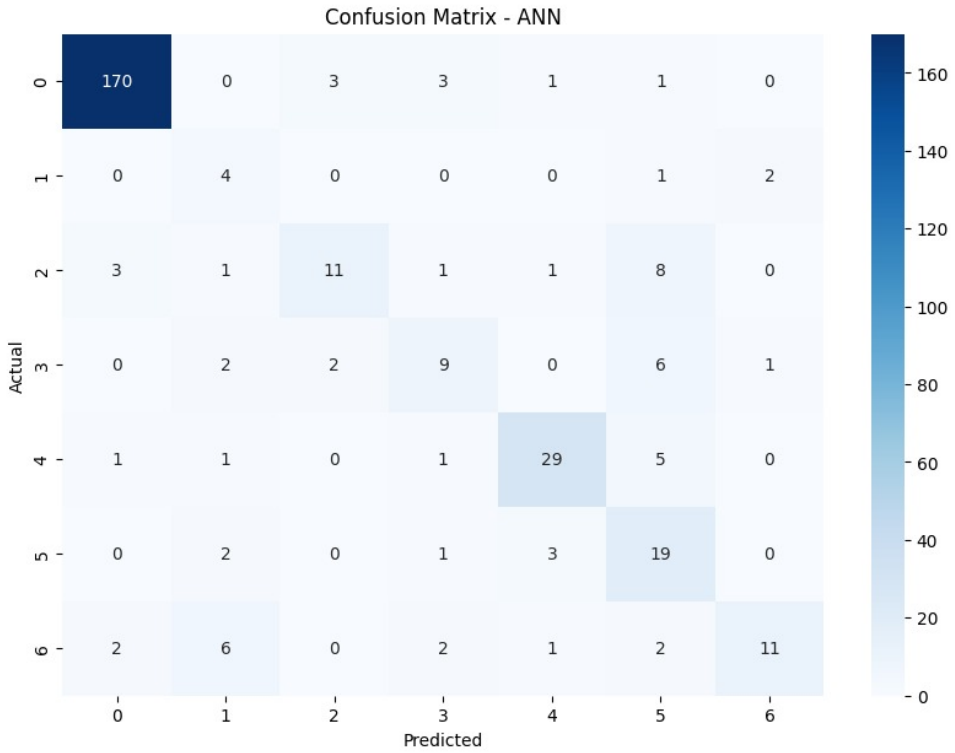

Figure 3: Classification report visualization for the ANN model

- LSTM

Table 7: LSTM Classifier

Class	Precision	Recall	F1-Score	Support
0	0.93	0.97	0.95	178
1	0.14	0.43	0.21	7
2	0.55	0.44	0.49	25
3	0.17	0.05	0.08	20
4	0.72	0.62	0.67	37
5	0.53	0.64	0.58	25
6	0.50	0.42	0.45	24
Accuracy	0.75 (on 316 samples)			
Macro Avg	0.51	0.51	0.49	316
Weighted Avg	0.75	0.75	0.74	316

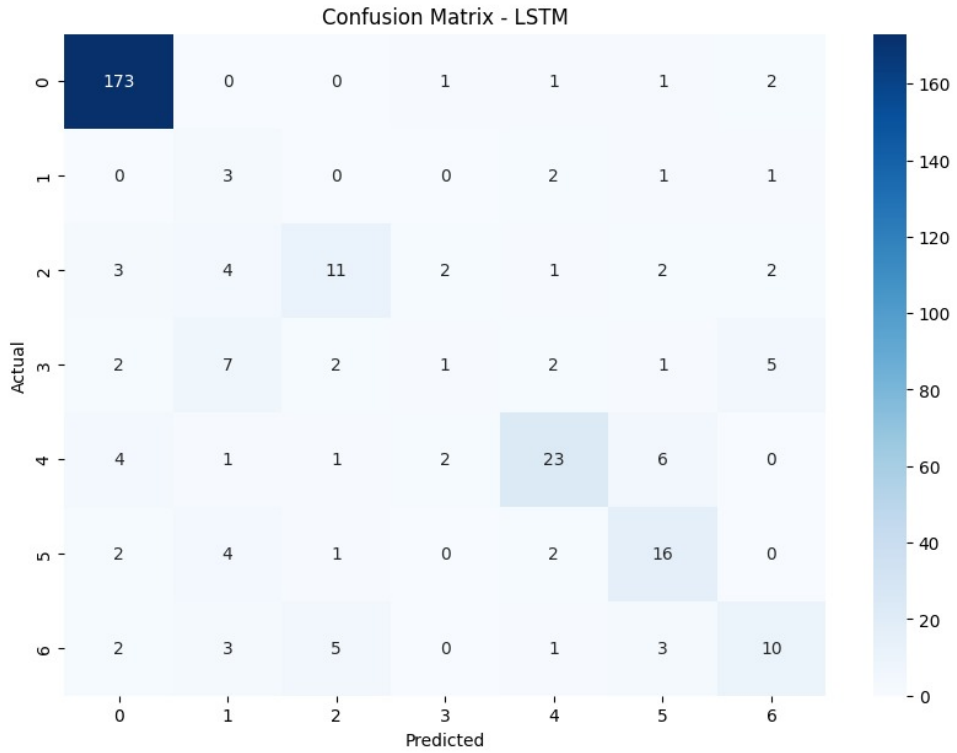

Figure 4: Classification report visualization for the LSTM model

- GRU

Table 8: GRU Classifier

Class	Precision	Recall	F1-Score	Support
0	0.98	0.90	0.94	178
1	0.13	0.57	0.21	7
2	0.57	0.52	0.54	25
3	0.54	0.70	0.61	20
4	0.57	0.65	0.61	37
5	0.62	0.60	0.61	25
6	1.00	0.25	0.40	24
Accuracy	0.75 (on 316 samples)			
Macro Avg	0.63	0.60	0.56	316
Weighted Avg	0.82	0.75	0.76	316

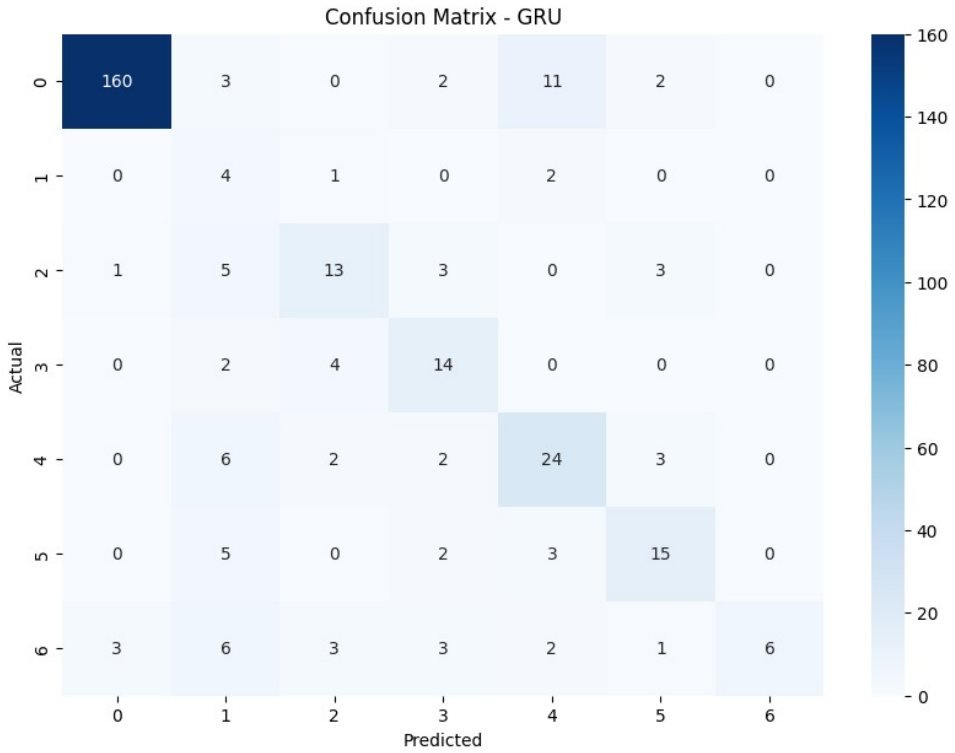

Figure 5: Classification report visualization for the GRU model

- Attention Based Transformer

Table 9: Attention-Based Transformer Classifier

Class	Precision	Recall	F1-Score	Support
0	0.96	0.95	0.95	178
1	0.20	0.14	0.17	7
2	0.54	0.52	0.53	25
3	0.42	0.40	0.41	20
4	0.44	0.68	0.53	37
5	0.57	0.52	0.54	25
6	0.42	0.21	0.28	24
Accuracy	0.74 (on 316 samples)			
Macro Avg	0.51	0.49	0.49	316
Weighted Avg	0.74	0.74	0.74	316

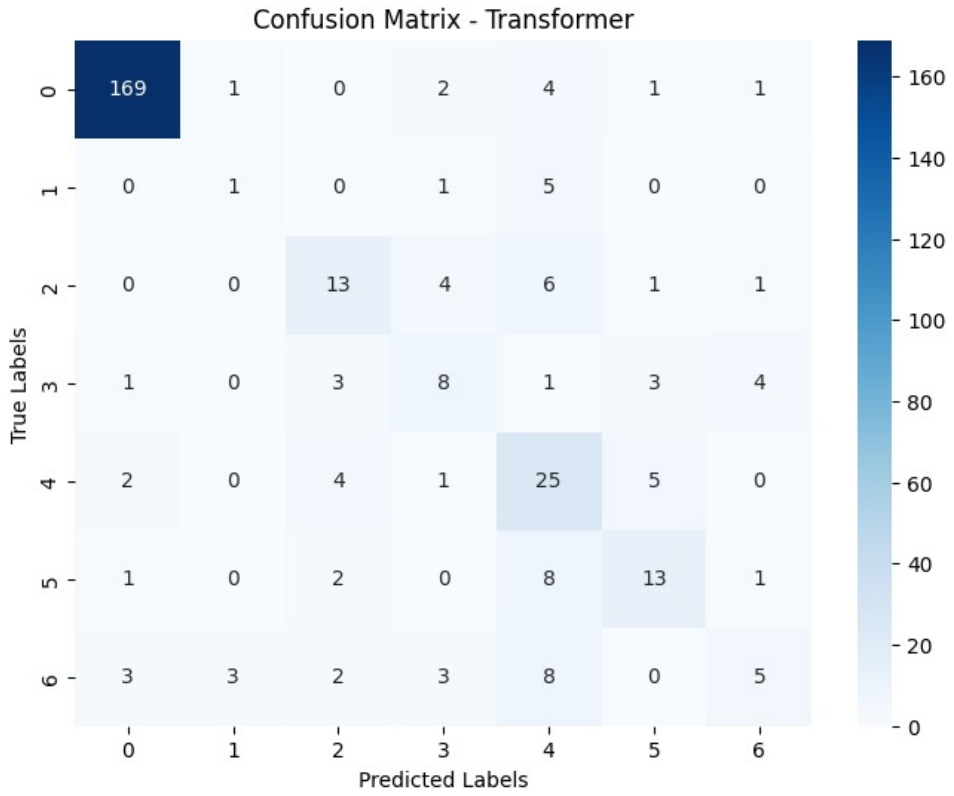

Figure 6: Classification report visualization for the Attention Based Transformer

Results for Balanced Dataset

- SVM

Table 10: SVM Classifier (Balanced)

Class	Precision	Recall	F1-Score	Support
0	0.97	0.99	0.98	182
1	0.59	0.89	0.71	159
2	0.73	0.85	0.78	172
3	0.61	0.69	0.65	168
4	0.99	0.80	0.89	147
5	0.95	0.68	0.79	136
6	0.89	0.51	0.65	163
Accuracy	0.78 (on 1127 samples)			
Macro Avg	0.82	0.77	0.78	1127
Weighted Avg	0.81	0.78	0.78	1127

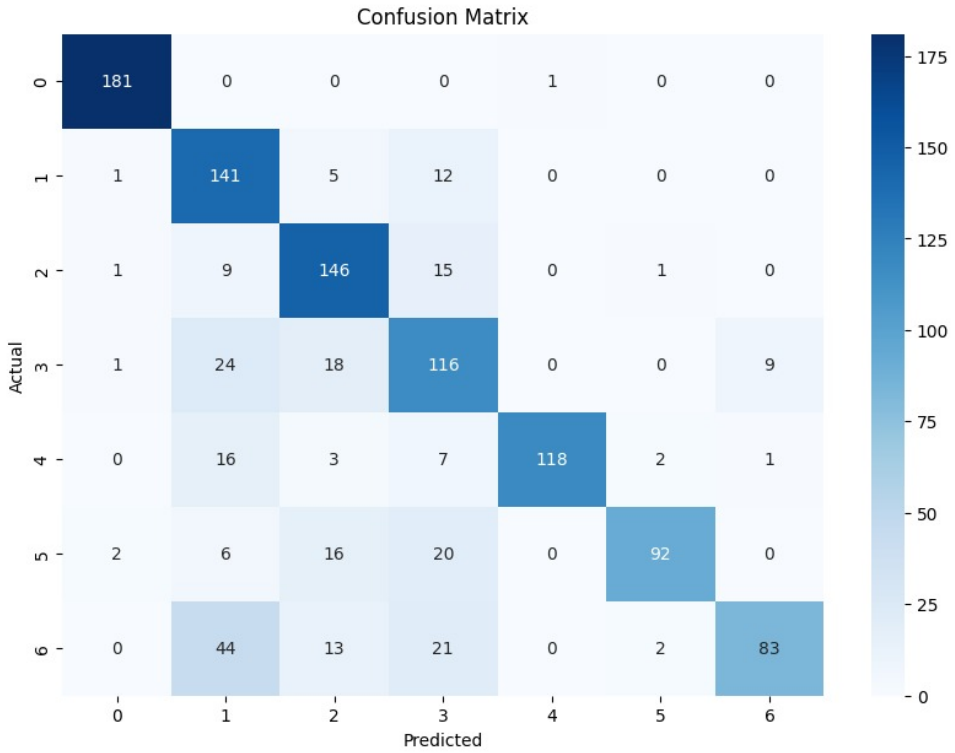

Figure 7: Classification report visualization for the SVM

- Random Forest Classifier

Table 11: Random Forest Classifier (Balanced)

Class	Precision	Recall	F1-Score	Support
0	0.98	0.99	0.99	182
1	0.86	0.94	0.90	159
2	0.91	0.95	0.93	172
3	0.86	0.88	0.87	168
4	0.95	0.91	0.93	147
5	0.95	0.89	0.92	136
6	0.89	0.82	0.86	163
Accuracy	0.91 (on 1127 samples)			
Macro Avg	0.92	0.91	0.91	1127
Weighted Avg	0.91	0.91	0.91	1127

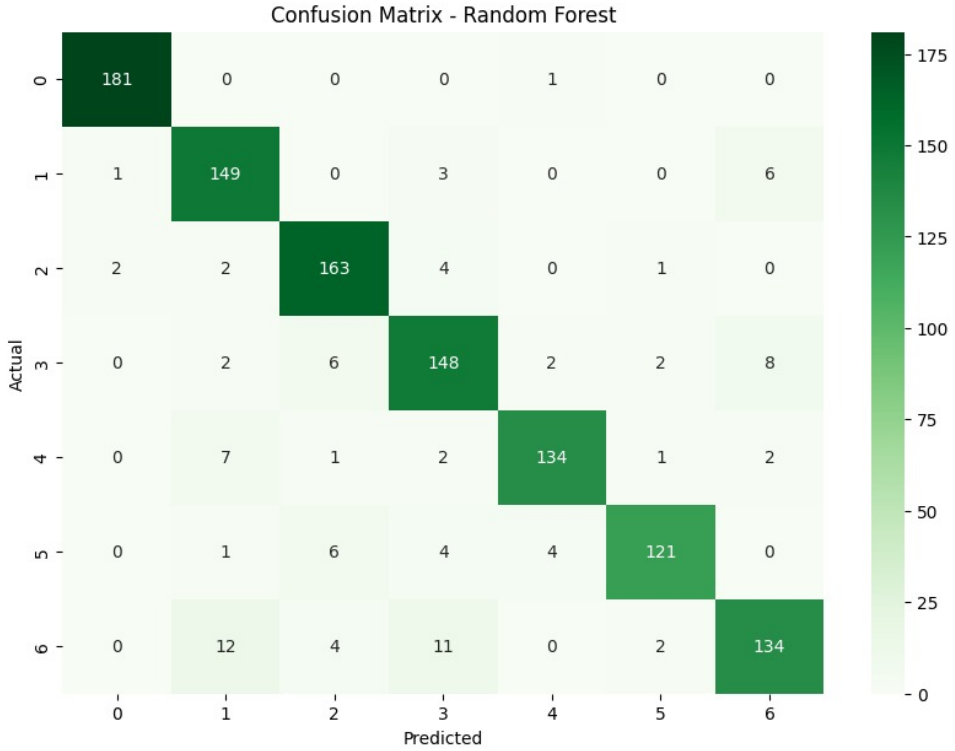

Figure 8: Classification report visualization for the Random Forest

- ANN

Table 12: ANN Classifier (Balanced)

Class	Precision	Recall	F1-Score	Support
0	0.97	0.97	0.97	182
1	0.63	0.87	0.73	159
2	0.70	0.83	0.76	172
3	0.65	0.71	0.68	168
4	0.96	0.88	0.92	147
5	0.97	0.69	0.81	136
6	0.84	0.56	0.68	163
Accuracy	0.79 (on 1127 samples)			
Macro Avg	0.82	0.79	0.79	1127
Weighted Avg	0.81	0.79	0.79	1127

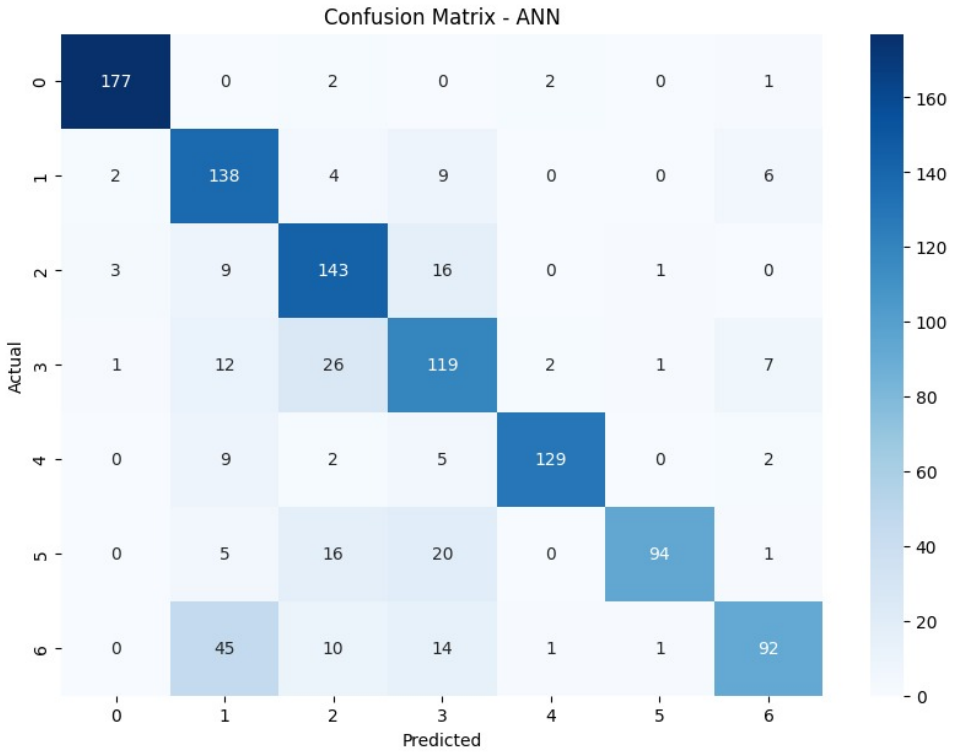

Figure 9: Classification report visualization for the ANN

- Attention Based Transformer

Table 13: Attention Based Transformer Encoder (Balanced)

Class	Precision	Recall	F1-Score	Support
0	0.94	0.97	0.95	182
1	0.53	0.83	0.64	159
2	0.75	0.70	0.72	172
3	0.57	0.64	0.60	168
4	0.95	0.52	0.68	147
5	0.71	0.76	0.73	136
6	0.71	0.48	0.58	163
Accuracy	0.71 (on 1127 samples)			
Macro Avg	0.74	0.70	0.70	1127
Weighted Avg	0.74	0.71	0.70	1127

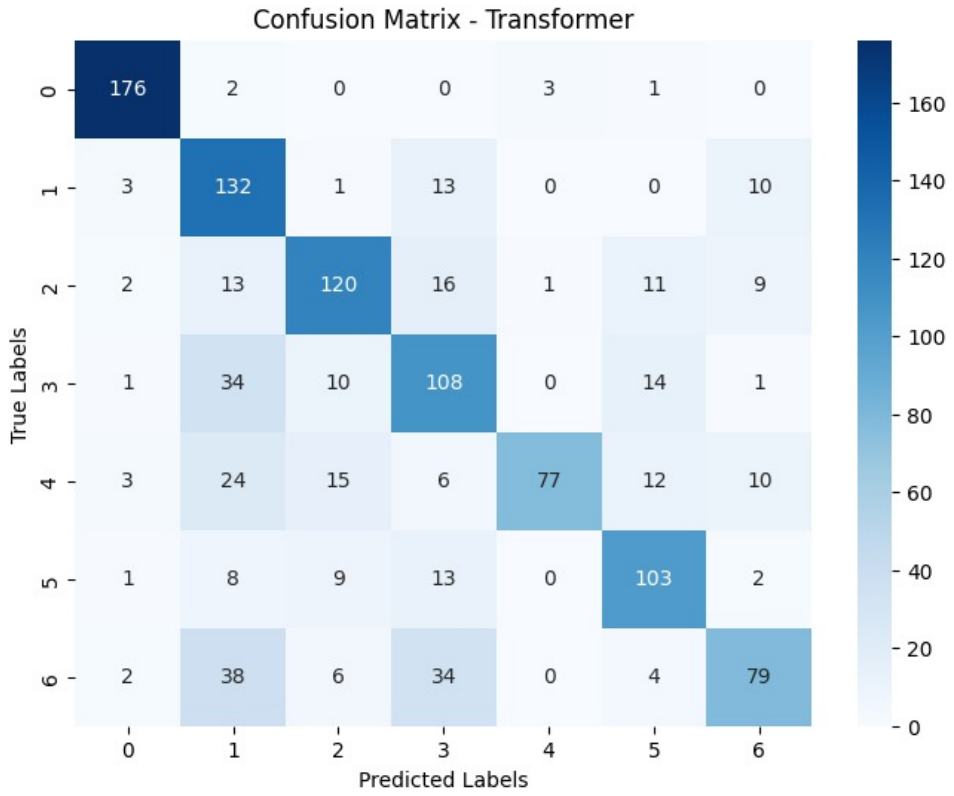

Figure 10: Classification report visualization for the Attention Based Transformer

8. Analysis

- Original Dataset

Model Name	Accuracy
SVM	0.76
Random Forest (RF)	0.83
Artificial Neural Network (ANN)	0.80
LSTM	0.75
GRU	0.75
Attention-based Transformer	0.74

Table 14: Comparitive analysis of different model based on Accuracy

- For Balanced dataset

Model Name	Accuracy
SVM	0.78
Random Forest (RF)	0.91
Artificial Neural Network (ANN)	0.79
Attention-based Transformer	0.71

Table 15: Comparitive analysis of different model based on Accuracy

- **The LSTM and GRU models could not be trained on the balanced dataset due to limited computational resources available on Google Colab, Kaggle, and the IIT server.**

9. Future Work

1. Analysis of these models on ADFA-WD dataset
2. Work on collecting more attack system call traces to increase the ADFA-LD dataset.
3. System call trace collection for different operating systems like Android,MacOS,etc.

10. Conclusion

The proposed project aims to enhance IDS capabilities using various machine learning models. Evaluating different models on the same dataset highlights their capabilities, while preprocessing techniques such as using trigrams can significantly enhance model performance. Tested the different models on ADFA LD to see the behavior and effectiveness of these models.

11. References

1. G. Creech and J. Hu, "Generation of a new ids test dataset: Time to retire the KDD collection," IEEE Wireless Communications and Networking Conference (WCNC), pp. 4487–4492, 2013.
2. G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns." Computers, IEEE Transactions on, p. (99):11, 2013.
3. G. Creech, "Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks," 2014.

12. Team Contributions

1. Prashant Rawat: Implemented SVM, GRU, Worked on Project Report.
2. Amber Gupta: Implemented Attention Based Transformer, Worked on Project Report, Handling Data imbalance.
3. Yash Narnaware: Implemented LSTM,Random Forest Classifier, ANN, Handling Data imbalance, Worked on Project Report.