Sanket Suryawanshi – code.sanket@gmail.com

Network Intrusion Detection System (NIDS) - Analysis & Findings Report

Dataset Used: CICIDS2017 (or similar public intrusion detection dataset)
Objective: To classify network traffic into **Normal** or **Attack** categories using **Machine Learning**.

Step 1: Data Preprocessing & Cleaning

Dataset Overview:

- The dataset consists of categorical and numerical features related to network traffic.
- Important features: totalSourceBytes, totalDestinationBytes, sourcePort, destinationPort, protocolName, etc.
- The target variable is Label (Normal/Attack).

Initial Data Analysis & Cleaning

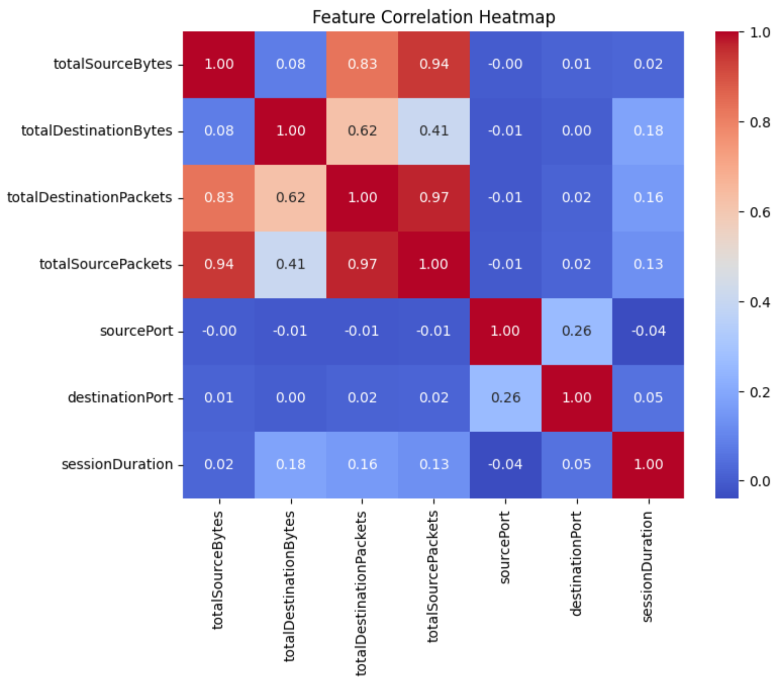
Removed unnecessary columns:

- Non-numeric & redundant features like appName, protocolName, startDateTime, stopDateTime.
  Handled missing values & normalized numerical features.
  Encoded categorical labels (Normal = 0, Attack = 1).

Step 2: Feature Selection & Correlation Analysis

Feature Correlation Heatmap

This heatmap shows the relationships between different numerical features.



Key Observations:

✓ **Highly Correlated Features**

- totalSourcePackets & totalDestinationPackets (**0.97**)
- totalSourceBytes & totalSourcePackets (**0.94**)
- totalDestinationPackets & totalSourceBytes (**0.83**)

✓ **Low Correlation Features**

- sourcePort, destinationPort, and sessionDuration show weak correlations with other features.

✓ **Negative Correlation**

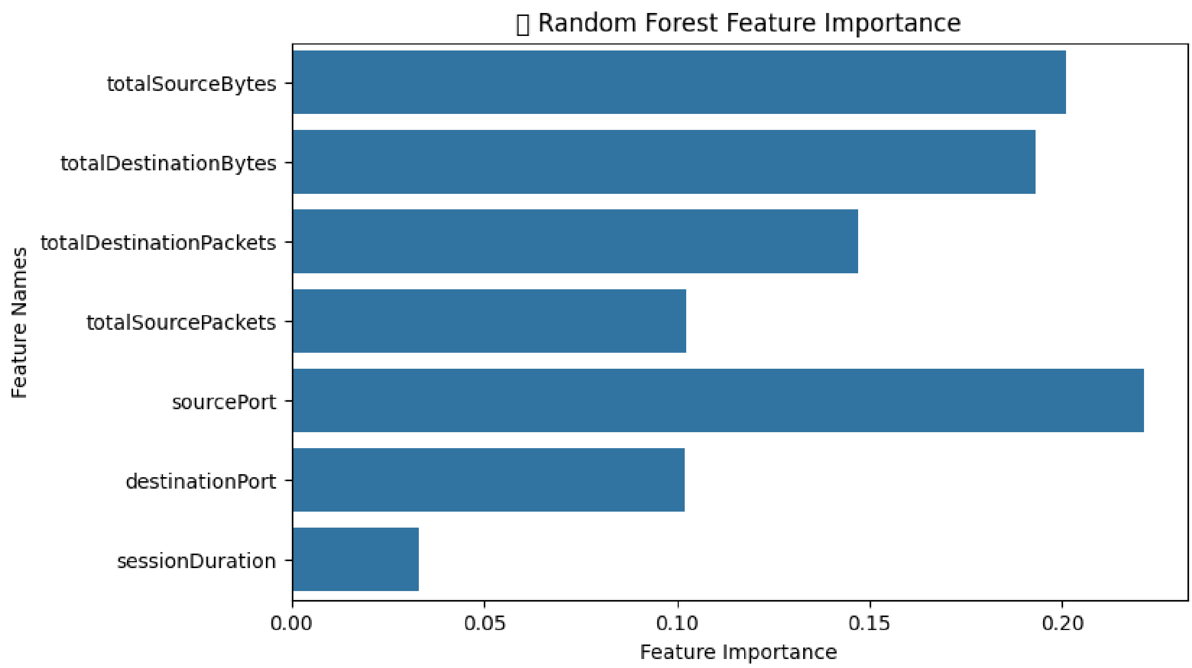
- sessionDuration & sourcePort have a slight negative correlation.

Action Taken:
Feature selection applied : Highly correlated features reduced to prevent redundancy.

Step 3: Model Training & Performance Evaluation

Model 1: Random Forest Classifier

Feature Importance (Random Forest)

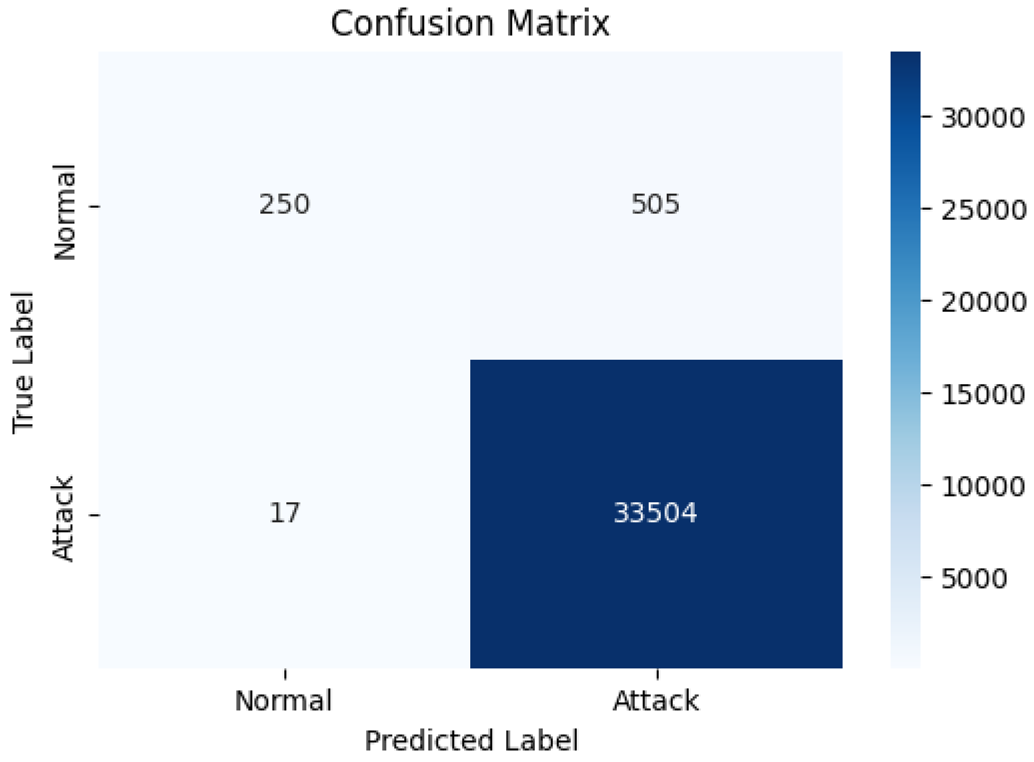


Classification Report (Random Forest)

Class	Precision	Recall	F1-Score	Support
Attack	0.98	0.97	0.98	755
Normal	1.00	1.00	1.00	33,521
Accuracy	-	-	**1.00**	34,276
Macro Avg	0.99	0.99	0.99	34,276
Weighted Avg	1.00	1.00	1.00	34,276

Random Forest Accuracy:99.91%

Confusion Matrix - Random Forest



Interpretation:
High Accuracy (99.91%) : Model performs exceptionally well in classifying normal vs. attack traffic.
Misclassifications:

- 505 false positives (Normal classified as Attack).
- 17 false negatives (Attack classified as Normal).

Feature Importance Ranking:

- totalSourceBytes & sourcePort are most significant.
- sessionDuration contributes the least.

Model 2: Artificial Neural Network (ANN)

Classification Report (ANN)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 (Normal)	0.94	0.33	0.49	755
1 (Attack)	0.99	1.00	0.99	33,521
Accuracy	-	-	**0.98**	34,276
Macro Avg	0.96	0.67	0.74	34,276
Weighted Avg	0.98	0.98	0.98	34,276

ANN Accuracy: 98.0%

Interpretation:
ANN has high accuracy (98%) but struggles with detecting **minority class (Normal traffic)**.
Recall for **Normal traffic is only 33%**, meaning the model fails to identify a significant portion of normal traffic.

Step 4: Final Comparison & Conclusion

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Observations**
Random Forest	**99.91%**	**1.00 (Normal), 0.98 (Attack)**	**1.00 (Normal), 0.97 (Attack)**	**1.00 (Normal), 0.98 (Attack)**	**Performs best, high recall, minimal false negatives**
ANN	**98.0%**	**0.99 (Attack), 0.94 (Normal)**	**1.00 (Attack), 0.33 (Normal)**	**0.99 (Attack), 0.49 (Normal)**	**Struggles with minority class (Normal traffic)**

Final Observations & Recommendations

Best Model: Random Forest (99.91% Accuracy)
Key Issues with ANN:

- Fails to detect **Normal traffic effectively**.
- Neural networks might require **more fine-tuning (hyperparameter optimization, more layers, different activation functions)**.

Feature Importance & Optimization:

- Some features like sessionDuration were **less important**, which could be removed for model optimization.

Final Takeaways

Random Forest is the best model for this dataset due to its high accuracy & recall.
Feature selection & correlation analysis helped in optimizing the dataset.
ANN struggled due to class imbalance, requiring further optimization.
Further work should focus on class imbalance handling & model efficiency improvement.