Methodology, Results, and Conclusions

Introduction

The report gives the methodology and results also conclusion of a machine learning project that inlcludes many models for network intrusion detection that use the UNSW-NB15 dataset the dataset include nine attack categories and also the modern network activities making it an best choice for evaluating machine learning model in a real world cases.

Methodology

Data Collection:

The UNSW-NB15 dataset is collected and used comprising the both normal activities and nine synthetic attack categories the dataset was split into training and testing subset for model development and also the evaluation.

Data Preprocessing

- **Handling Missing Values**: Missing numeric value were fill with the mean and while categorical value were replaced with the mode.
- **Removing Duplicates**: Duplicate are found and can be removed to ensure data integrity.
- **Normalization**: Numerical feature were normalized using the feature Min Max scaling.
- **Feature Engineering**: New features such as Source to Destination Packet Ratio and Total Bytes were created to improve model performance.

Exploratory Data Analysis (EDA)

- Distribution of target labels and attack categories was analyzed.
- Correlation matrices and pair plots were generated to identify relationships between features.
- Advanced visualizations like heatmaps, scatter plots, and violin plots were used to understand feature distributions and relationships.

Model Development

Four models were implemented:

- Decision Tree
- Random Forest
- Support Vector Machine

- Neural Network (MLP)

Each model was trained using a stratified train-test split 70% training & 30% testing.

Evaluation Metrics

Models were evaluated using:

- Accuracy
- Precision
- Recall
- F1 Score
- ROC-AUC Cross-validation was performed to ensure robustness.

Hyperparameter Tuning

- **Decision Tree**: Grid Search was used to optimize criteria, depth, and sample splits.
- **Random Forest**: Randomized Search with reduced folds and iterations was applied.
- **SVM**: A pipeline with scaling and linear kernel tuning was employed.
- **Neural Network**: Grid Search focused on activation functions, learning rates, and layer sizes.

Analysis and Comparison of Model Performance

The best performance of four machine learning models Decision Tree, Random Forest, Support Vector Machine and Neural Network (MLP) has been evaluated based on several metrics. Below is a detailed analysis of the result before and after hyperparameter tuning.

Evaluation Metrics Overview

Key Metrics:

- **Accuracy**: Correctly classified samples.
- **Precision**: Ratio of true positive to all the predicted positive.
- **Recall**: Ratio of true positive to all the actual positive.
- **F1 Score**: Harmonic mean of the precision and the recall.
- **ROC-AUC:** Area under the receiver and operating characteristic curve.

Comparison of Performance Before Hyperparameter Tuning

Model	Accuracy	Precision	Recall	F1 Score	ROC - AUC
Decision Tree	0.9046	0.9058	0.9011	0.9034	0.9056
Random Forest	0.9210	0.9066	0.9369	0.9215	0.9819
SVM	0.6300	0.7800	0.9900	0.5700	0.8238
Neural Networks (MLP)	0.8555	0.7799	0.9866	0.8711	0.8931

Observations:

Decision Tree:

- Good overall performance balance precision and recall but slightly lower than Random Forest.

Random Forest:

- The best performance overall high accuracy and precision and ROC-AUC show a robust model.

SVM:

- Poor accuracy and F1 score because of high recall and low precision likely overfitting or struggle with the imbalanced data.

Neural Networks:

- Strong recall but relatively lower precision showing potential overfitting to the positive class.

Cross Validation Results Before Hyperparameter Tuning

Model	Mean CV Accuracy	Standard Deviation
Decision Tree	0.8557	1.1441
Random Forest	0.8630	0.1666
SVM	0.6293	0.0703
Neural Network (MLP)	0.7180	0.1363

Observations:

Decision Tree:

- High variance show the model is sensitive to data split and may lack robustness.

Random Forest:

- Similar to Decision Tree but with also the better mean accuracy and robustness.

SVM:

- Poor mean accuracy and low variance show that consistent underperformance.

Neural Networks:

- Moderate accuracy but high variance show that the sensitivity to data splits.

Results After Hyperparameter Tuning

Model	Best Cross Validation Accuracy
Decision Tree	0.9088
Random Forest	0.9212
SVM	0.8677
Neural Networks (MLP)	0.8363

Observations:

Decision Tree:

- Improve performance with optimized parameters.

Random Forest:

- The best model overall that shows all the benefits of fine tuning.

SVM:

- ❖ Significant improvement on sample dataset but still underperform compared to Random Forest and Decision Tree.

Neural Networks:

- ❖ Improvement in performance but still less effective than Decision Tree and Random Forest.

Insights and Final Recommendation

Best Model: **Random Forest**

- **Highest Accuracy**: Post tuning accuracy of 92.12%.
- **Robustness**: High ROC-AUC 0.9819 that shows strong partial capability.
- **Balanced Metrics**: Excellent balance across precision, recall, and F1 score.

Other Observations:

- ❖ **Decision Tree**: Close performance to Random Forest after tuning but less accurate and robust.
- ❖ **SVM**: Poor baseline performance improves post tuning but still has an issue in competitiveness due to its less accuracy and sensitivity to imbalanced data.
- ❖ **Neural Network**: While good recall was achieved it underperformed compared to tree based models in terms of precision and overall accuracy.

Final Recommendation

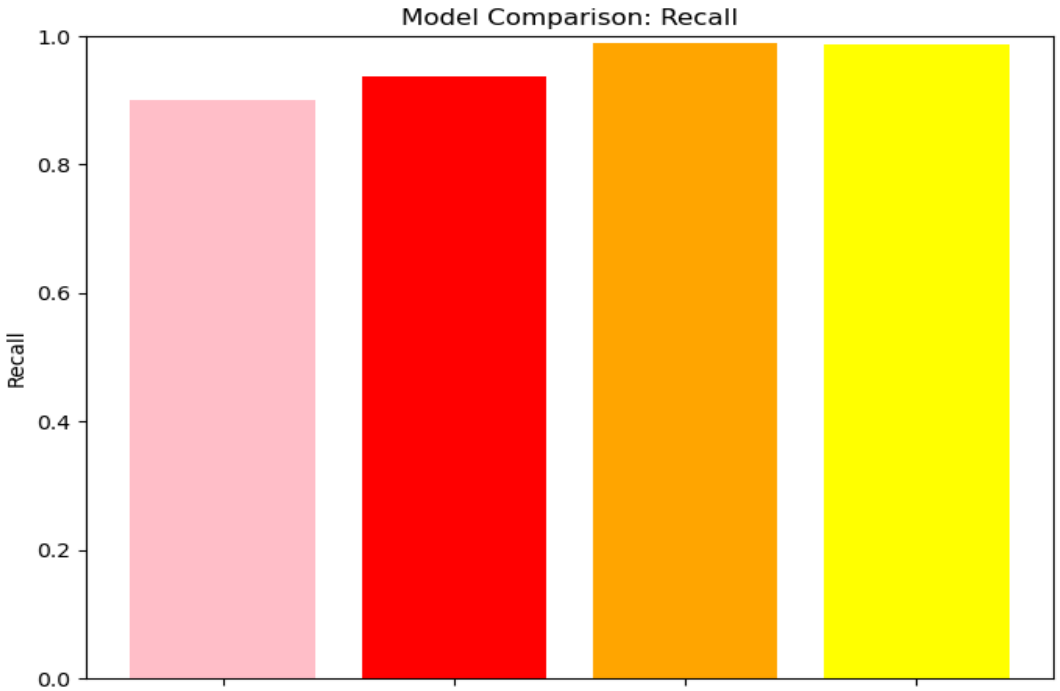
The **Random Forest model** is the best performing model based on accuracy, ROC-AUC, and overall balance metrics it should be used for the task as it provides both high precision and recall while maintaining robustness across data splits.

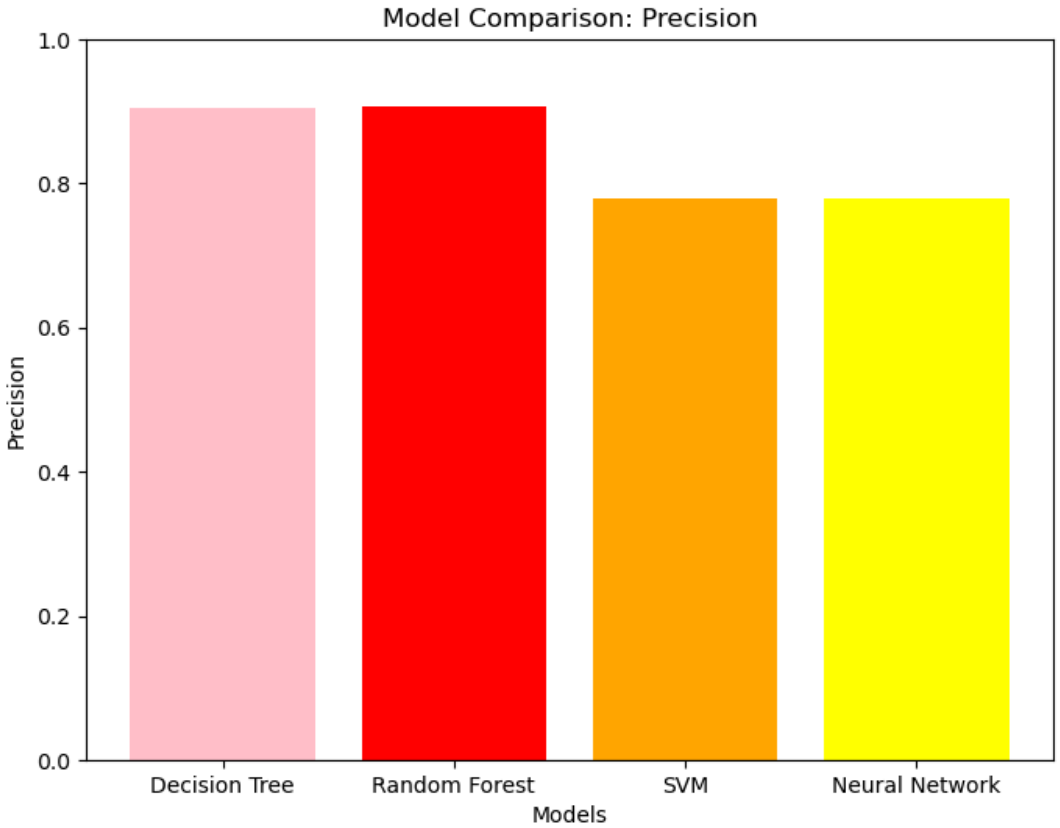
Conclusions

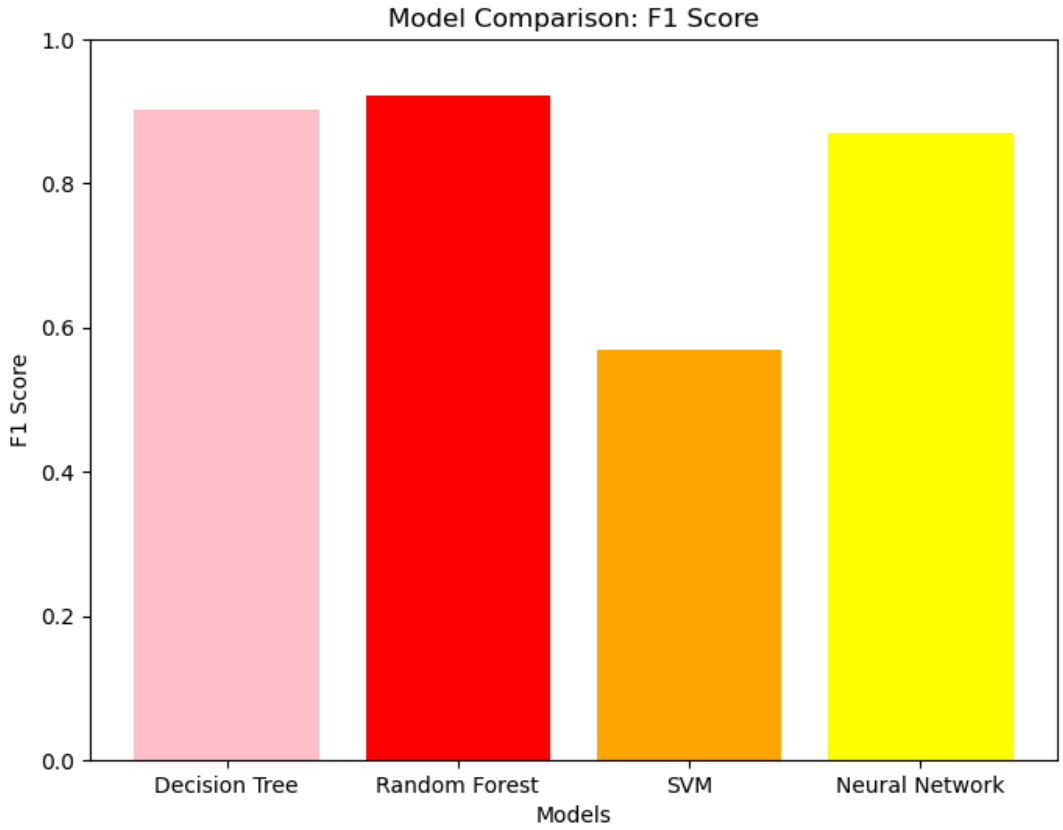
- ❖ The **Random Forest** model is shown as the best performing model due to its superior accuracy, balanced metrics, and high ROC-AUC score.

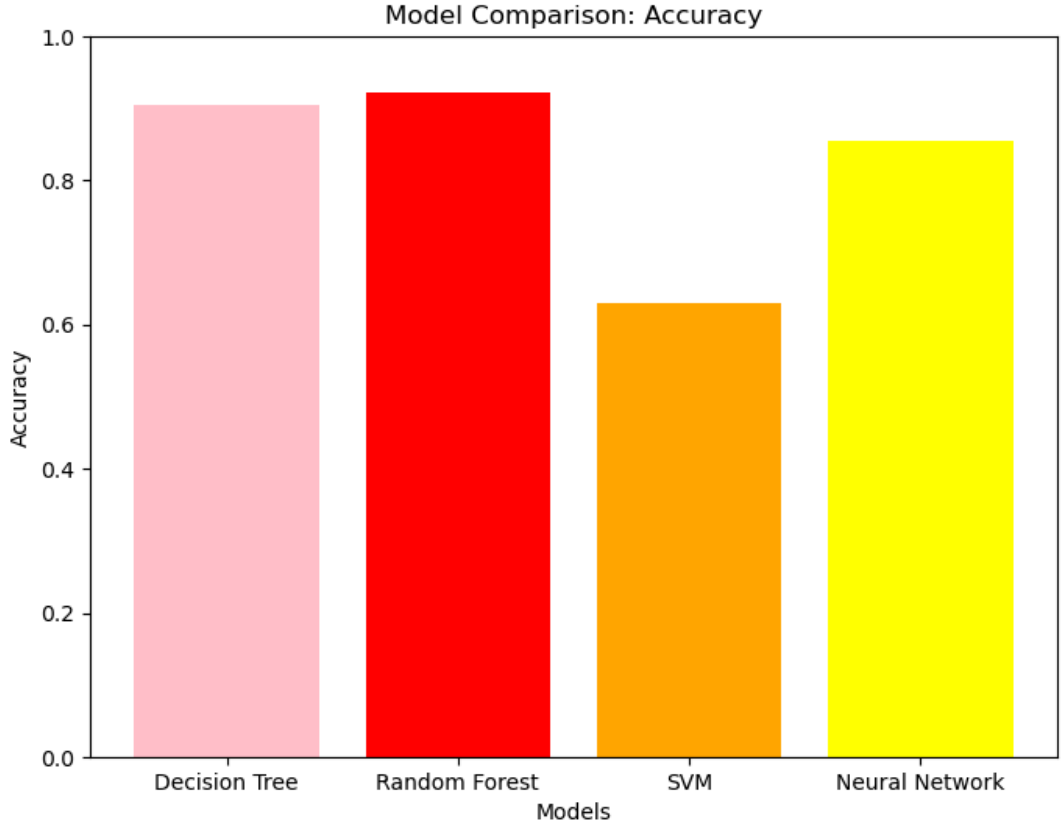
- **Decision Tree** serves as an effective alternative when computational efficiency is a priority.
- **SVM** and **Neural Network** show confidence that they require further optimization and adjustment to address specific challenges like overfitting and also the imbalance.

Model Performance Visualization (Accuracy, Precision, Recall, F1-Score)

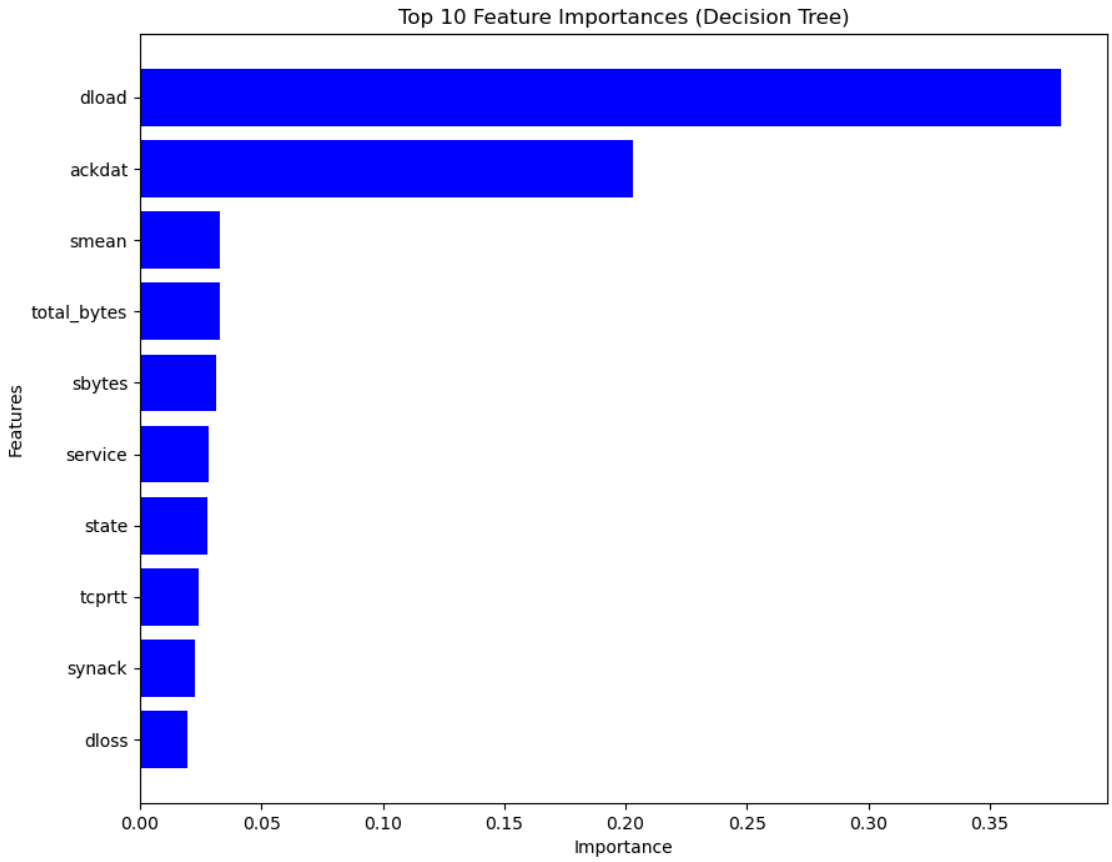


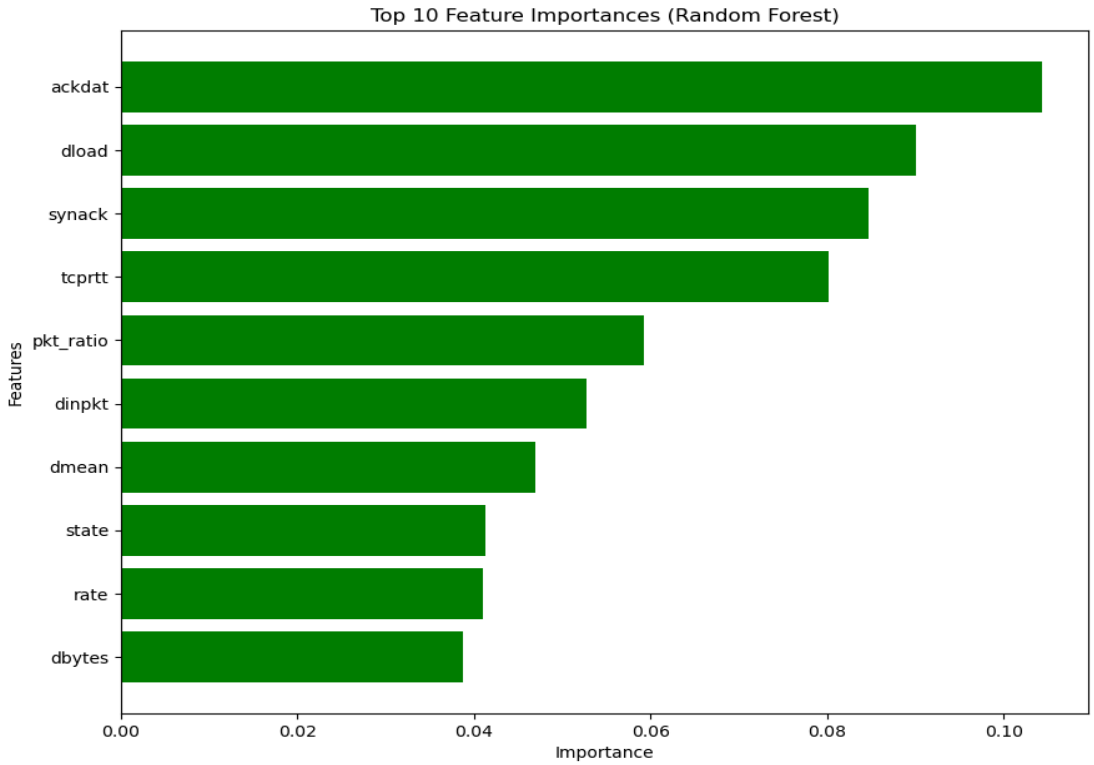



Model Comparison: F1 Score
F1 Score
1.0
0.8
0.6
0.4
0.2
0.0
Decision Tree
Random Forest
SVM
Neural Network
Models

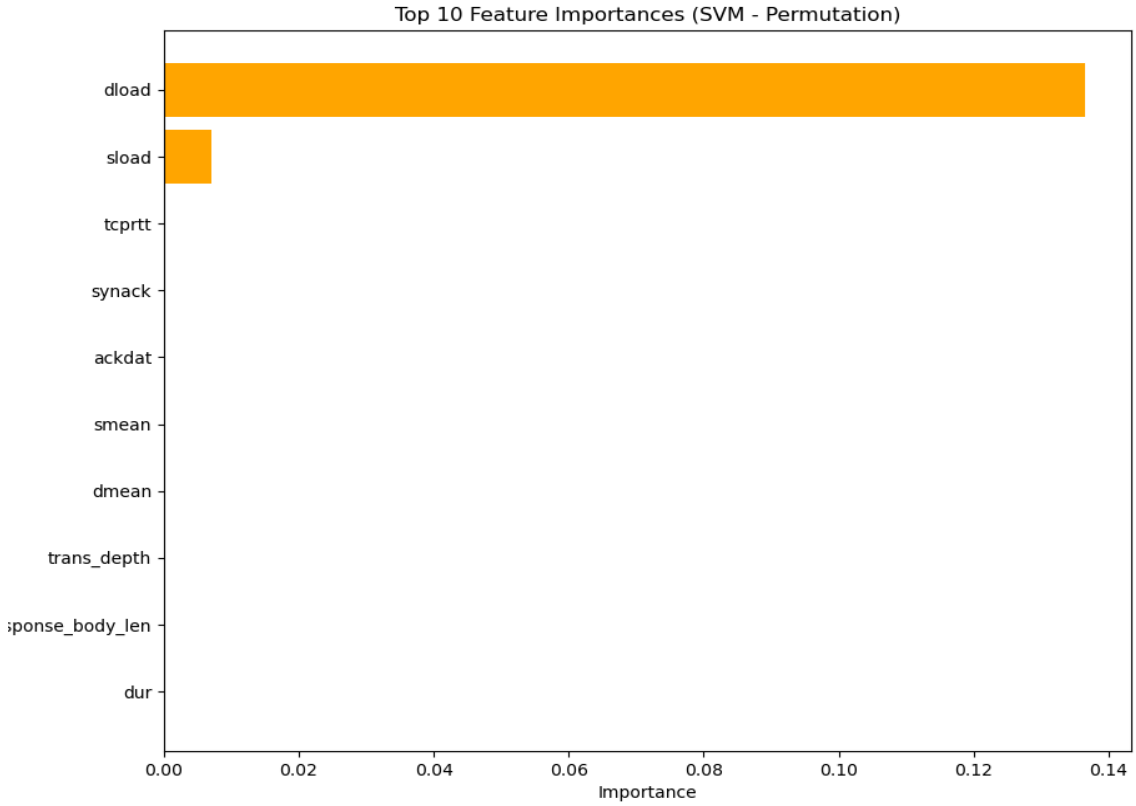

Model Comparison: Accuracy
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Decision Tree
Random Forest
SVM
Neural Network
Models

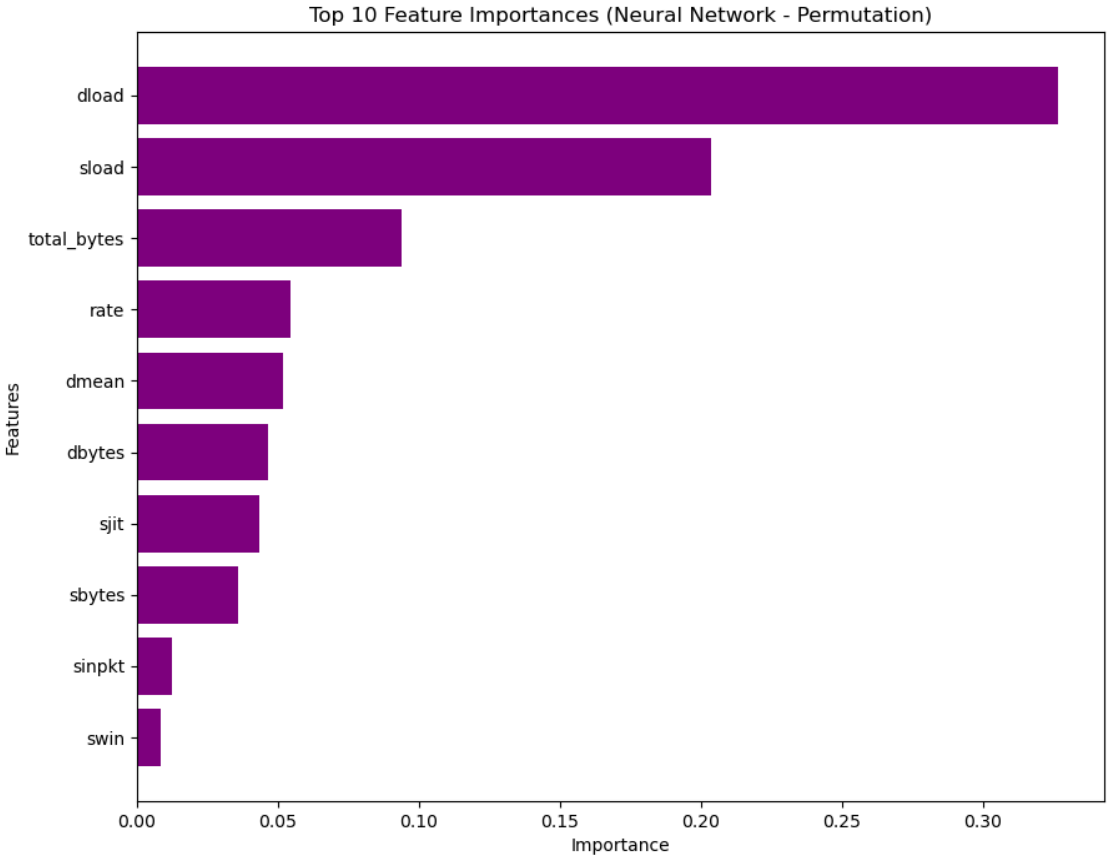
Feature Importance for Decision Tree and Random Forest





Feature Importance for SVM and Neural Network Using Permutation Importance




Top 10 Feature Importances (Neural Network - Permutation)
dload
sload
total_bytes
rate
dmean
dbytes
sjit
sbytes
sinpkt
swin
Features
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance