Intrusion Detection System using Random Forest

1. Dataset: KDDTrain+_20Percent.arff from the KDD Cup 1999 dataset.

This dataset is specifically designed for evaluating IDS models as it contains a wide variety of network intrusion scenarios, including both normal and anomalous activities making it suitable for our needs.

The dataset has almost the same number of normal and anomaly cases, with only a small difference between them. This means the data is not heavily unbalanced, making it good for training and testing machine learning models without needing to fix class imbalances.

2. Data preprocessing:

- Our dataset consists of 25,192 rows and 42 columns.
- Target: **class** which has 2 labels (normal and anomaly).

1. First, we converted the categorical values of the class into numerical values which are 0 & 1. This is important for model compatibility.

2. Encoding categorical features which are (protocol_type, service', flag, land, logged_in, is_host_login, is_guest_login).

3. Data scaling in a range between 0 & 1 to prevent any single feature with a large range from dominating the model.

A sample of data before:

	duration	protocol_type	service	flag	src_bytes	...	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	class
0	0.0	tcp	ftp_data	SF	491.0	...	0.00	0.00	0.05	0.00	normal
1	0.0	udp	other	SF	146.0	...	0.00	0.00	0.00	0.00	normal
2	0.0	tcp	private	S0	0.0	...	1.00	1.00	0.00	0.00	anomaly
3	0.0	tcp	http	SF	232.0	...	0.03	0.01	0.00	0.01	normal
4	0.0	tcp	http	SF	199.0	...	0.00	0.00	0.00	0.00	normal

[5 rows x 42 columns]

A sample of data after:

	duration	protocol_type	service	flag	src_bytes	...	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	class
0	0.0	0.5	0.292308	0.9	1.286320e-06	...	0.00	0.00	0.05	0.00	1
1	0.0	1.0	0.630769	0.9	3.824902e-07	...	0.00	0.00	0.00	0.00	1
2	0.0	0.5	0.707692	0.5	0.000000e+00	...	1.00	1.00	0.00	0.00	0
3	0.0	0.5	0.338462	0.9	6.077927e-07	...	0.03	0.01	0.00	0.01	1
4	0.0	0.5	0.338462	0.9	5.213394e-07	...	0.00	0.00	0.00	0.00	1

3. Data splitting:

Data split into 80% training and 20% testing.

4. Choosing suitable algorithm:

We have chosen **Random Forest** which is a Machine learning algorithm that can **detect unknown attacks** and **adapt to new types of threats**.

It also handles high-dimensional data effectively, robust to overfitting due to averaging of tree predictions and provides feature importance.

5. Model evaluation:

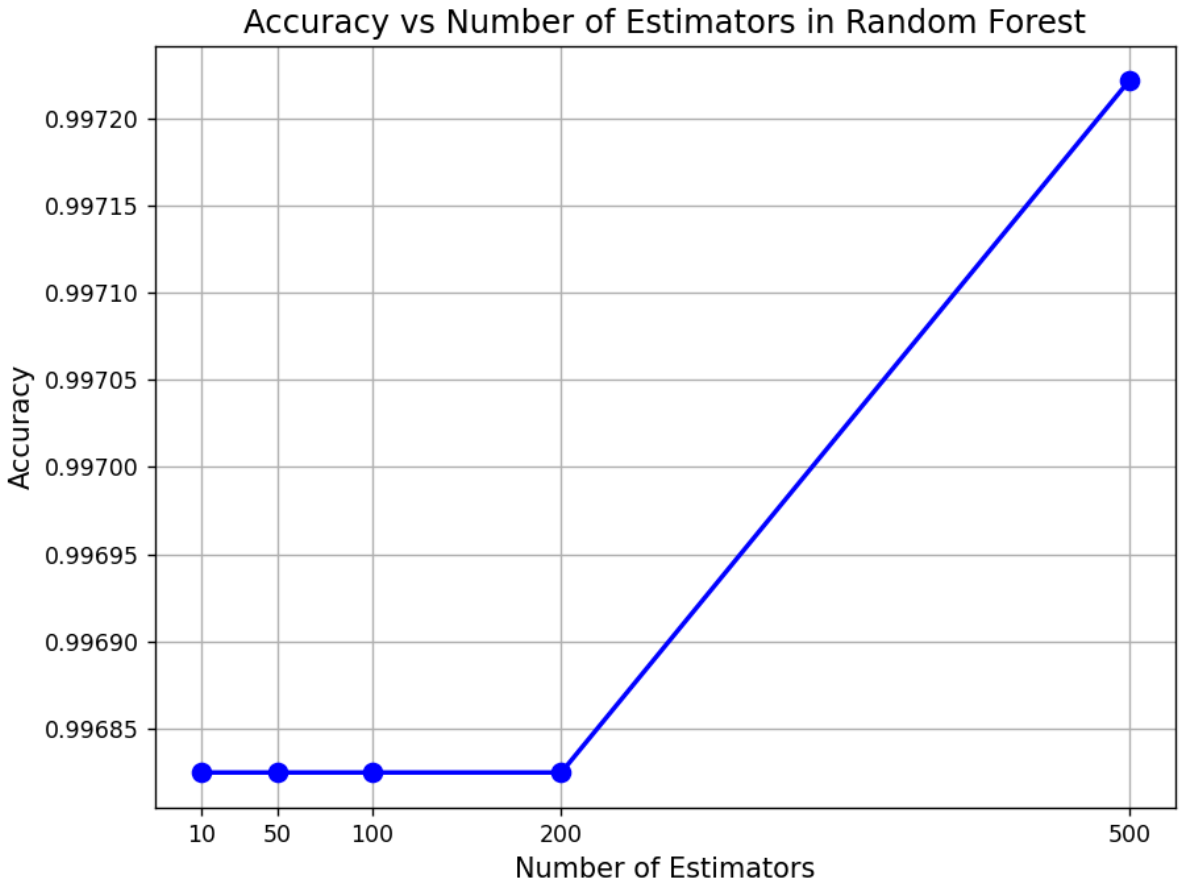
In this step we applied the algorithm and tested different values of number of estimators (trees) [10, 50, 100, 200, 500] to see how it impacts the model's performance.

For each value of estimators, we trained the model on the training data then used the model to make predictions on the test set.

We calculated the accuracy of the model, which is the percentage of correct predictions on the test data. After that, we generated a classification report to evaluate other metrics like precision, recall, and F1-score.

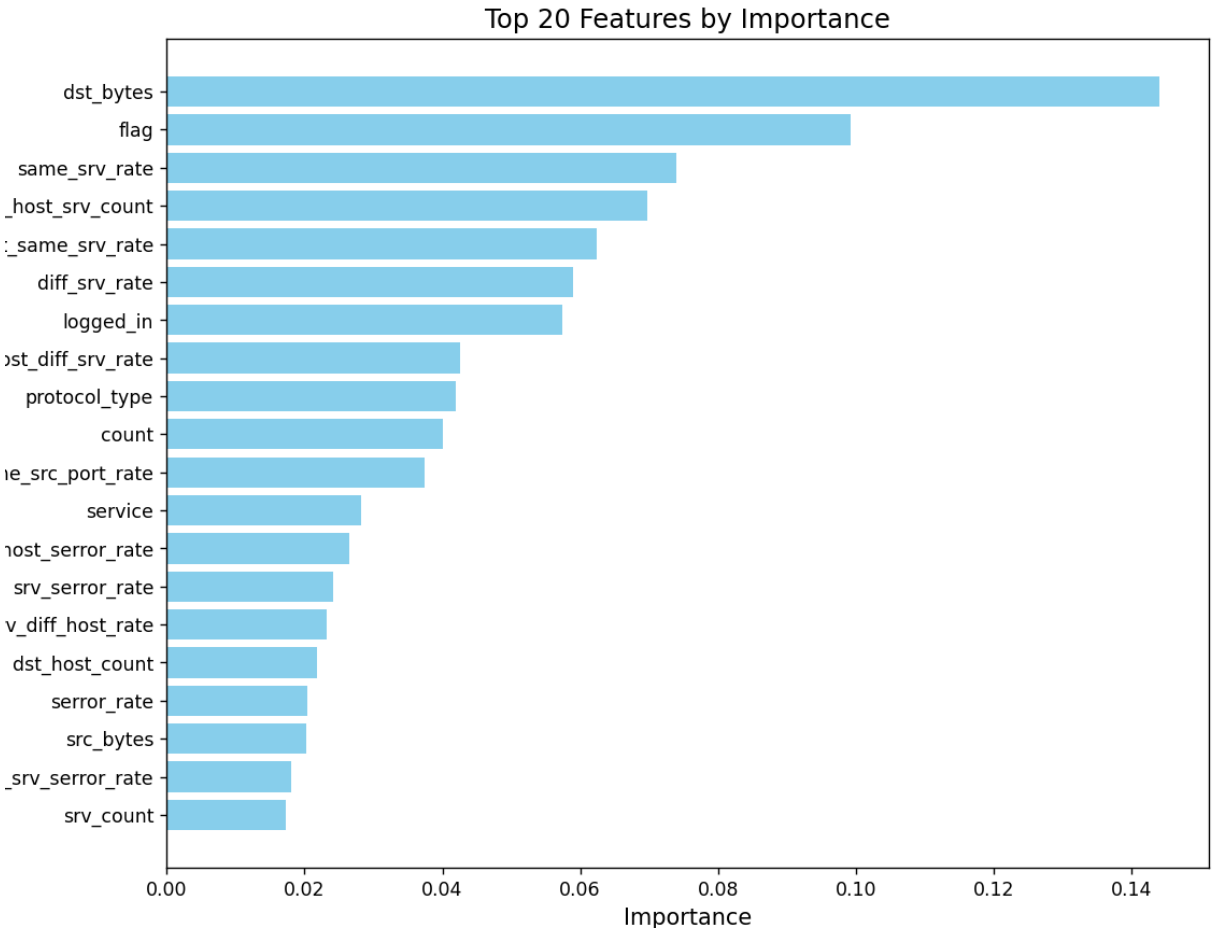
The **best model** was identified by comparing the accuracy for each number of estimators then selecting the one with **highest accuracy**.

```
Best Model:
Best Accuracy: 0.99722 with 500 Estimators
Best Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2349
           1       1.00      1.00      1.00      2690

    accuracy                           1.00      5039
   macro avg       1.00      1.00      1.00      5039
weighted avg       1.00      1.00      1.00      5039
```



6. Feature importance:

We analyzed the feature importance of the best model to see each one's impact on the model's decision. The features were sorted by importance and the top 20 were selected for further analysis. This helps identify the most impactful features for detecting anomalies.



By : Malak Mohamed Osman 2205059, Fadya Hesham ElOraby 2205177, Abdelrahman Ayman Saad 2205033.