DNS Anomaly Detection using Random Forest

1. Synthetic Data generation and Feature Extraction:

- DNS Query Simulation

DNS queries are constructed using the **scapy** library.
Domain names are randomly selected from a predefined list and sent to various DNS servers.
Each DNS query includes essential protocol layers: IP, UDP, DNS, and DNSQR.

- Anomaly Simulation

40% of the DNS queries are altered to simulate anomalies.
Key attributes modified:
Time-To-Live (TTL): Very low or very high values.
Packet Size: Abnormal sizes.
Response Time: Delayed responses.
Response Codes: Errors such as SERVFAIL, NXDOMAIN, or REFUSED.

- Feature Extraction

Key metrics collected from each DNS query and response (TTL, Transaction ID, Response Code (rcode), Query Length, Response Length, Response Time (RTT)).

Derived features include:

Query Volume: Frequency of a domain being queried.
Geographic Region: Based on the DNS server queried.
Response Code Ratio: Ratio of error responses to successful ones.

- Dataset Structure

The dataset consists of 19 columns containing 18 features and the anomaly label column that marks queries as either **"Normal"** or **"Anomalous"** based on the presence of anomalies.

- Dataset Output

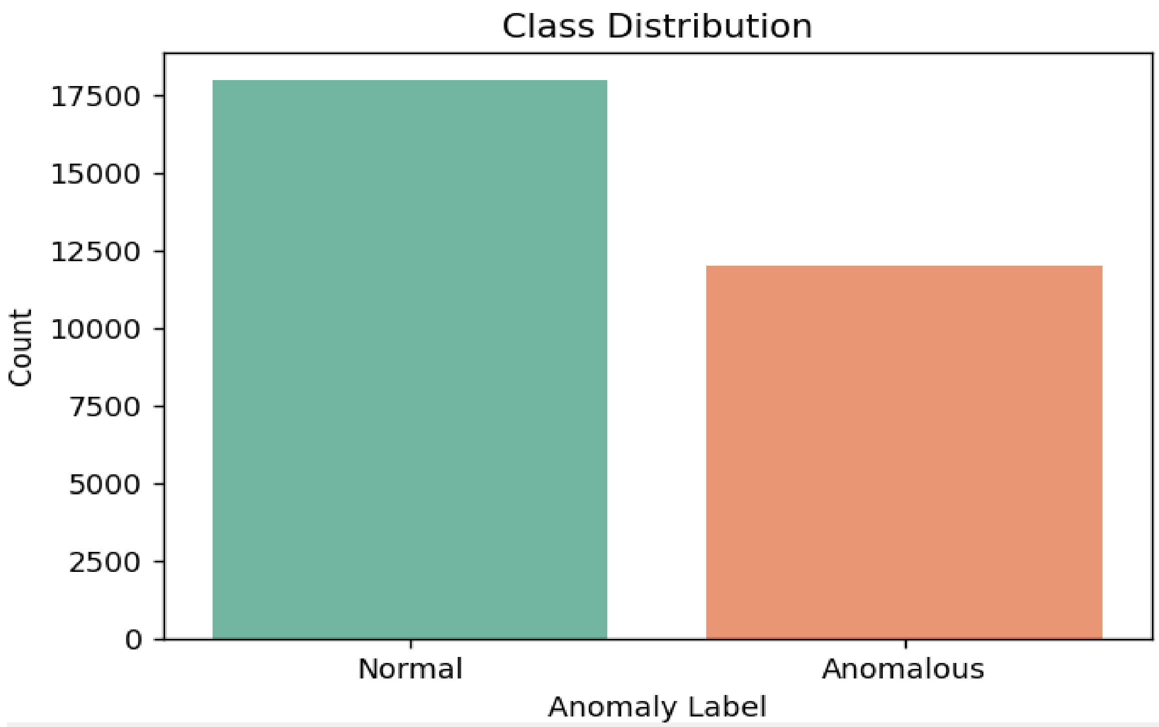
30,000 DNS queries with a mix of normal and anomalous traffic.

The final dataset is saved as `dns_data2.csv` for analysis and anomaly detection.

2. Data preprocessing:

- Our dataset consists of 30000 and 19 columns.
- Target: **anomaly_label** which has 2 labels (Normal and Anomalous).

Number of occurrences for each class label.



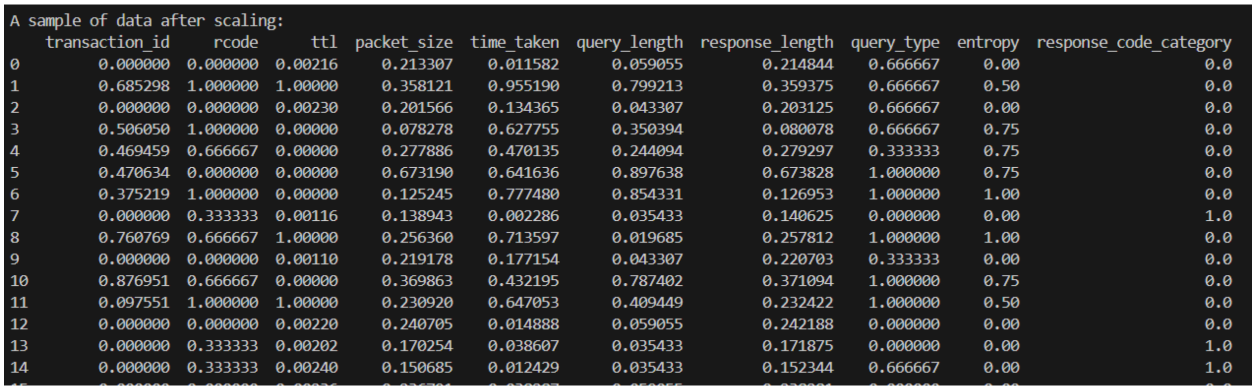
1. First, we converted the categorical values of the anomaly_label into numerical values which are 0 for Normal and 1 for Anomalous.

2. All categorical columns are encoded into numerical values to ensure compatibility.

3. A correlation matrix is generated to evaluate the relationship between each feature and the target variable (anomaly_label).

Columns with weak correlation (src_ip, dst_ip, query_name, src_port, dst_port, geo_region, response_code_ratio) are dropped to reduce noise as they will not affect the model's decision.

4. 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 after scaling:
    transaction_id    rcode      ttl  packet_size  time_taken  query_length  response_length  query_type  entropy  response_code_category
0         0.000000 0.000000  0.00216     0.213307    0.011582      0.059055         0.214844    0.666667     0.00                     0.0
1         0.685298 1.000000  1.00000     0.358121    0.955190      0.799213         0.359375    0.666667     0.50                     0.0
2         0.000000 0.000000  0.00230     0.201566    0.134365      0.043307         0.203125    0.666667     0.00                     0.0
3         0.506050 1.000000  0.00000     0.078278    0.627755      0.350394         0.080078    0.666667     0.75                     0.0
4         0.469459 0.666667  0.00000     0.277886    0.470135      0.244094         0.279297    0.333333     0.75                     0.0
5         0.470634 0.000000  0.00000     0.673190    0.641636      0.897638         0.673828    1.000000     0.75                     0.0
6         0.375219 1.000000  0.00000     0.125245    0.777480      0.854331         0.126953    1.000000     1.00                     0.0
7         0.000000 0.333333  0.00116     0.138943    0.002286      0.035433         0.140625    0.000000     0.00                     1.0
8         0.760769 0.666667  1.00000     0.256360    0.713597      0.019685         0.257812    1.000000     1.00                     0.0
9         0.000000 0.000000  0.00110     0.219178    0.177154      0.043307         0.220703    0.333333     0.00                     0.0
10        0.876951 0.666667  0.00000     0.369863    0.432195      0.787402         0.371094    1.000000     0.75                     0.0
11        0.097551 1.000000  1.00000     0.230920    0.647053      0.409449         0.232422    1.000000     0.50                     0.0
12        0.000000 0.000000  0.00220     0.240705    0.014888      0.059055         0.242188    0.000000     0.00                     0.0
13        0.000000 0.333333  0.00202     0.170254    0.038607      0.035433         0.171875    0.000000     0.00                     1.0
14        0.000000 0.333333  0.00240     0.150685    0.012429      0.035433         0.152344    0.666667     0.00                     1.0
```

3. Data splitting:

Data split into 80% training and 20% testing.

4. Choosing suitable algorithm:

We have chosen **Random Forest**, a machine learning algorithm that is suitable for DNS anomaly detection problems. It effectively identifies patterns in DNS traffic to detect unknown anomalies 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.

Initial model parameters:

- n_estimators = 100
- max_depth = 10
- min_samples_split = 20

The initial model training resulted in overfitting, so to address this issue, we introduced label noise. This approach helps in reducing the overfitting by making the model less likely to memorize the training data and instead focus on general patterns.

5. Introducing Noise to Labels

This is done by flipping 4% of the labels in the dataset. A random subset of the labels is flipped between 0 (Normal) and 1 (Anomalous).

6. Hyperparameter tuning

GridSearchCV is used to tune model's hyperparameters to reduce overfitting.

The following hyperparameters are explored: (27 combinations were tested)

n_estimators: [100, 500, 1000]

max_depth: [10, 20, 30]

min_samples_split: [2, 10, 20]

Cross-Validation:

A 5-fold cross-validation is used to enhance the performance of a machine learning model and ensure that it generalizes well to unseen data.

The dataset is split into 5 equal folds. The model is trained on 4 of the folds and tested on the remaining one.

7. Final Model Evaluation

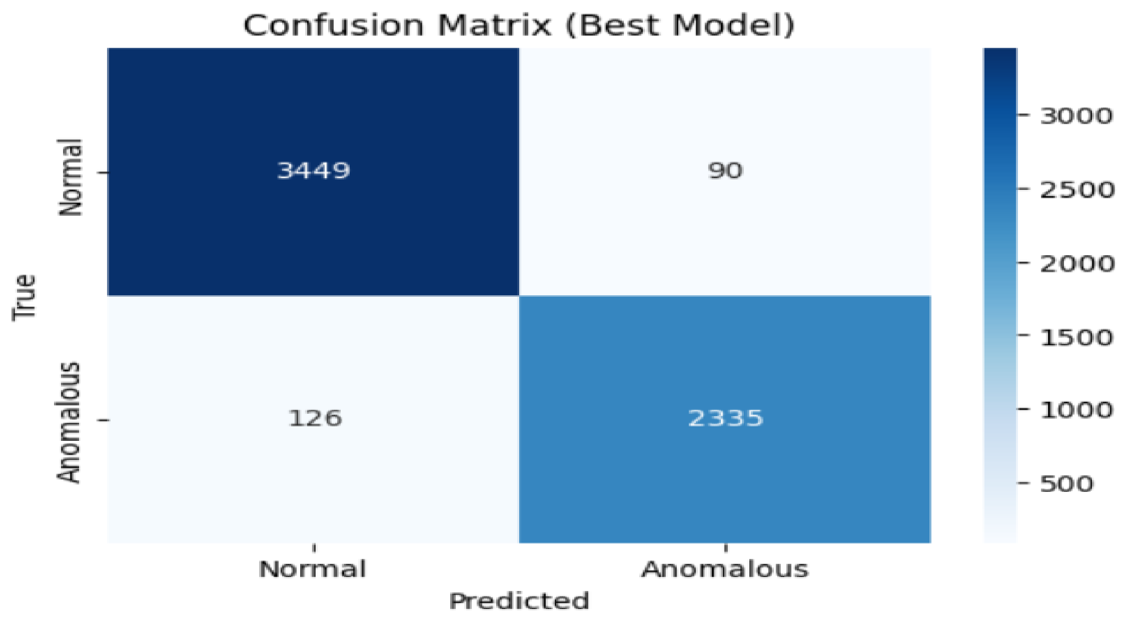
The optimized model is evaluated on the noisy test data to ensure it can handle imperfections.

It is selected based on the highest accuracy.

```
Best Model Accuracy: 0.9640
Best Hyperparameters:  {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 100}
Classification Report (Best Model):
              precision    recall  f1-score   support

           0       0.96      0.97      0.97      3539
           1       0.96      0.95      0.96      2461

    accuracy                           0.96      6000
   macro avg       0.96      0.96      0.96      6000
weighted avg       0.96      0.96      0.96      6000
```



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