Methodology for Tweet Classification

Cybertroll Dataset by Data-Turk used for this

1. Data Preprocessing

The tweets were preprocessed using the following steps:

- **Expanding Contractions**: All contractions (e.g., “can't”) were expanded to their full forms (e.g., “cannot”).
- **Lowercasing**: All text was converted to lowercase to ensure uniformity.
- **Removing URLs**: Any URLs present in the text were removed.
- **Removing Special Characters and Punctuation**: Special characters and punctuation were eliminated to retain only meaningful words.
- **Tokenization**: The text was split into individual tokens (words).
- **Lemmatization**: Tokens were lemmatized to reduce them to their base forms.
- **Named Entity Recognition (NER)**: Entities such as names, locations, and organizations were identified and retained as additional features.

2. Balancing the Corpus

To address class imbalance, the **RandomOverSampler** technique was applied to the dataset, ensuring a balanced distribution of classes.

3. Train-Test Split

The dataset was split into training and testing sets while maintaining a representative distribution of classes.

4. Fine-Tuning DistilBERT

The DistilBERT model was fine-tuned on the training dataset with the following configurations:

```
training_args = TrainingArguments(
    output_dir="./results",
    num_train_epochs=10,
    per_device_train_batch_size=16,
    warmup_steps=1000,
    weight_decay=0.0004,
    logging_dir="./logs",
    logging_steps=10,
    save_total_limit=1,
    save_strategy="epoch"
)
```

The optimizer used was **AdamW**.

5. Feature Extraction

- **BERT Embeddings**: The fine-tuned DistilBERT model was used to extract word embeddings for the training and testing datasets.
- **FastText Embeddings**: Pretrained FastText embeddings were also extracted for the datasets.
- **Concatenation**: The BERT and FastText embeddings were concatenated to form a combined feature representation.

6. Multi-Scale CNN for Feature Extraction

The combined embeddings were passed through a multi-scale CNN with filter sizes of 3, 5, and 7, each having 64 filters, to extract meaningful features. The resulting CNN embeddings were concatenated with the original embeddings.

7. Feature Mapping with Feedforward Neural Network (FFNN)

The concatenated embeddings were passed through a two-layer FFNN with the following structure:

- **Linear Layer**: Input = Combined Embeddings, Output = 64
- **Activation**: ReLU

This produced a feature map of dimension 64.

8. Feature Reduction

Dimensionality reduction was performed on the feature map using the following techniques:

- **Eigenvalue Decomposition (EIG)**: Extracted significant features.
- **Ant Colony Optimization (ACO)**: Further refined the feature set.
- **Pearson Correlation Coefficient (PCC)**: Reduced the features to the final 10 dimensions.

```
--- Feature Selection Pipeline for Training Data ---
EIG: Reduced 64 features to 30 based on mutual information.
Selected features after EIG: [11 50  8 60 19 63 26 23 24 41 55 42 38 30 47 15 56 54 21  6 18 28 29 32
 27 58 53 16  9 44]
ACO iterations: 100%|██████████| 10/10 [00:00<00:00, 68.44it/s]
Iteration 1/10: Best solution score = 1.5767
Iteration 2/10: Best solution score = 1.6149
Iteration 3/10: Best solution score = 1.6532
Iteration 4/10: Best solution score = 1.6733
Iteration 5/10: Best solution score = 1.7169
Iteration 6/10: Best solution score = 1.7169
Iteration 7/10: Best solution score = 1.6890
Iteration 8/10: Best solution score = 1.7489
Iteration 9/10: Best solution score = 1.7558
Iteration 10/10: Best solution score = 1.7493
ACO: Reduced 30 features to 15 based on ant colony optimization.
Selected features after ACO: [10 27  9 11  3 26 21 28 16 22 23  0 14  4 15]
PCC: Reduced 15 features to 10 based on Pearson Correlation Coefficient.
Selected features after PCC: [ 7  1  8 12  2 13  5  6 14  9]

Training Data: Reduced from 64 to 10 features.
```

9. Classifier Module

The reduced features were passed through an attention-based neural network. The process included:

- Applying a linear layer.
- Computing attention weights using the softmax function.
- Weighting features by multiplying them with attention weights.
- Passing the weighted feature vectors to an ensemble classifier.

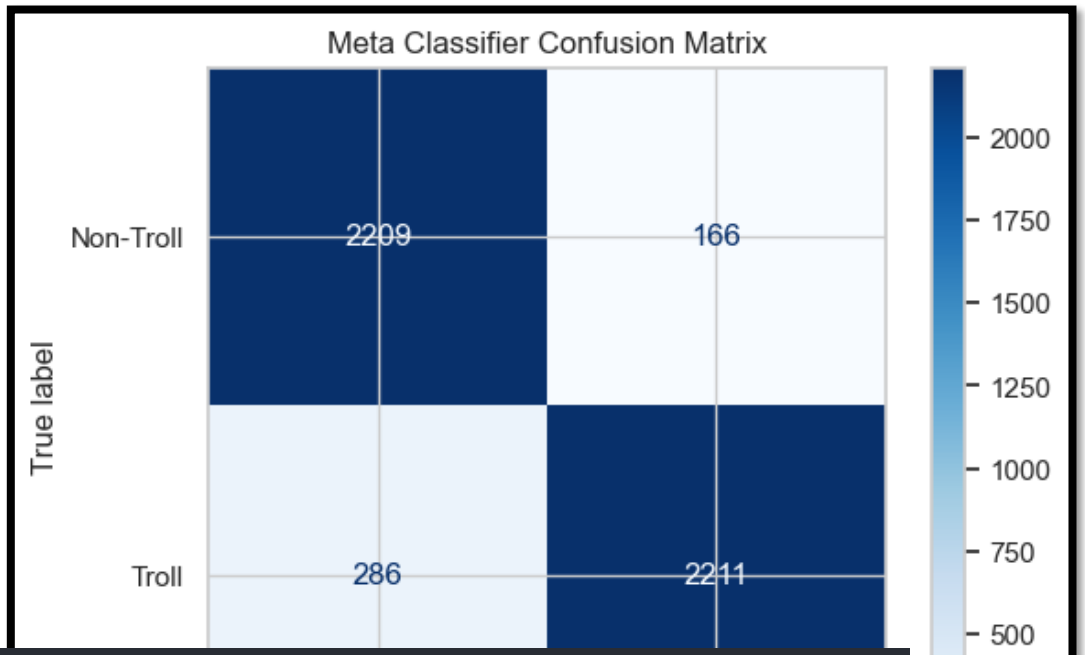
Ensemble Classifier

The ensemble classifier consisted of:

- **XGBoost**
- **Support Vector Classifier (SVC)**
- **Random Forest Classifier (RFC)**
- **Simple Neural Network (Feed Forward with Adam Optimizer, LR = 0,01)**:
    - Linear Layer (Input = 10, Output = 64) → ReLU
    - Linear Layer (Input = 64, Output = 32) → ReLU
    - Linear Layer (Input = 32, Output = 1) → Sigmoid

10. Evaluation

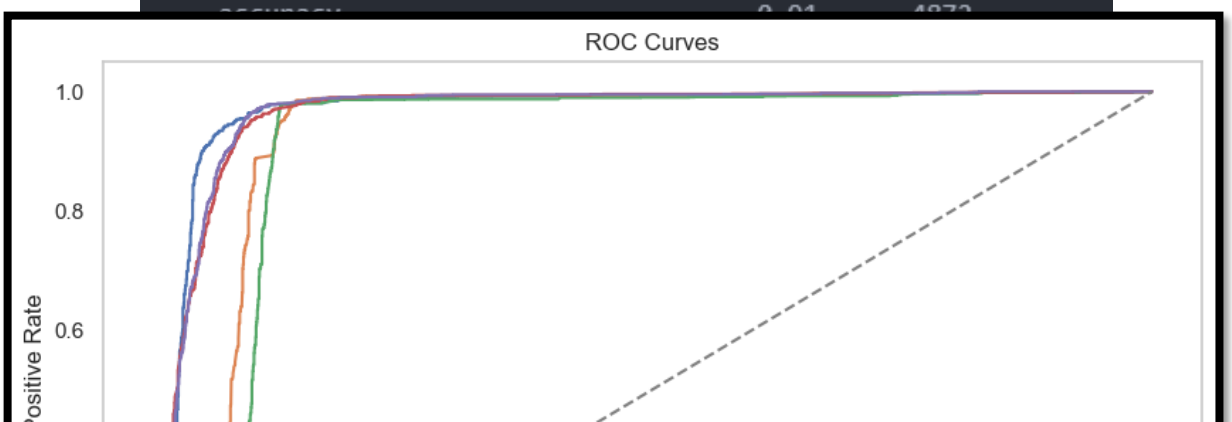
The trained ensemble model was evaluated on the test dataset using various metrics:



Neural Network Classification Report:

	precision	recall	f1-score	support
0	0.97	0.88	0.93	2375
1	0.90	0.98	0.94	2497
accuracy			0.93	4872
macro avg	0.93	0.93	0.93	4872
weighted avg	0.93	0.93	0.93	4872

```
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.86      0.91      2375
           1       0.88      0.98      0.93      2497

    accuracy                           0.92      4872
   macro avg       0.93      0.92      0.92      4872
weighted avg       0.93      0.92      0.92      4872

XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.94      0.89      2375
           1       0.93      0.85      0.89      2497

    accuracy                           0.89      4872
   macro avg       0.89      0.89      0.89      4872
weighted avg       0.89      0.89      0.89      4872

Meta Classifier Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.93      0.91      2375
           1       0.93      0.89      0.91      2497
```



```
Neural Network - Accuracy: 0.9308, F1-Score: 0.9353, ROC-AUC: 0.9650
Random Forest - Accuracy: 0.4949, F1-Score: 0.0660, ROC-AUC: 0.9168
SVM - Accuracy: 0.9206, F1-Score: 0.9267, ROC-AUC: 0.8991
XGBoost - Accuracy: 0.8894, F1-Score: 0.8868, ROC-AUC: 0.9621
Meta Classifier - Accuracy: 0.9072, F1-Score: 0.9073, ROC-AUC: 0.9618
```