Information Security Assignment 2: Classifying Malicious Links

Sarim Aeyzaz (i21-0328), Ehtsham Walidad (i21-0260)

1 Dataset Preprocessing

We merged two URL datasets into one, ensuring consistency by removing duplicate entries and filling missing values. To handle class imbalance, we applied **undersampling**, leading to a balanced dataset distribution.

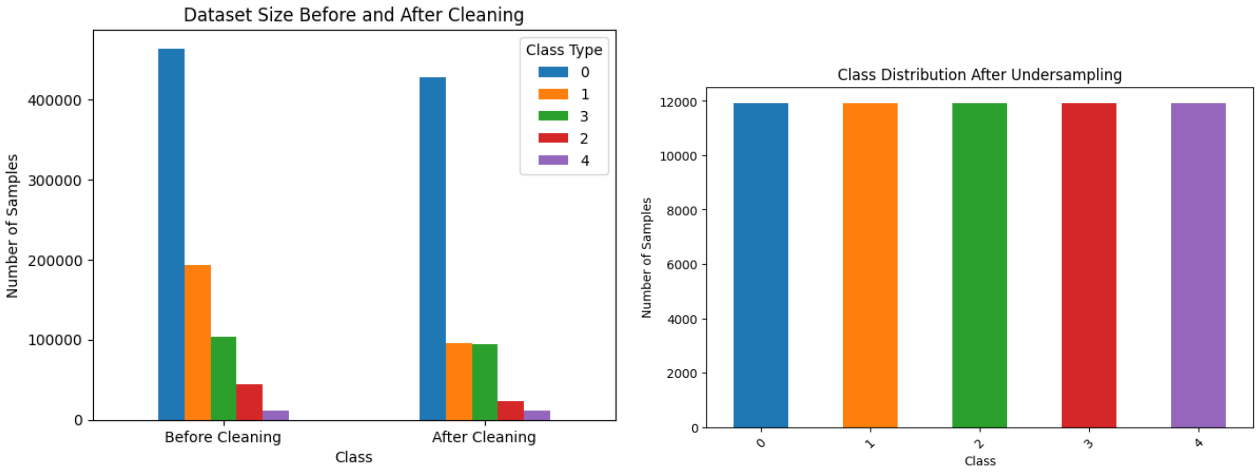


Figure 1: Dataset before and after cleaning (left), undersampled class distribution (right).

2 Feature Extraction

We extracted two sets of features:

- **BERT-based uncased embeddings** of the URL.
- **Lexical features** including URL length, domain length, path length, entropy measures, and symbol counts, and others listed below:

```
{ "url_len", "domain_len", "path_len", "domain_entropy", "suffix_entropy",
  "continuity_rate", "symbol_count", "token_count", "digit_count",
  "path_url_ratio", "domain_url_ratio" }
```

3 Exploratory Data Analysis

We performed an exploratory data analysis (EDA) to understand the distribution of the data set.

Observations:

(a) Spam URLs have more outliers in terms of length.

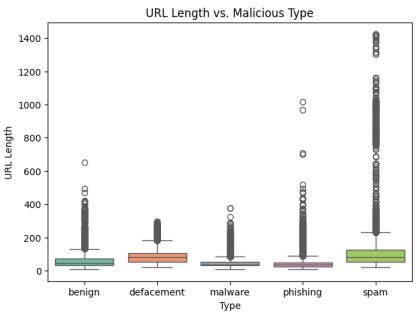
(b) Domain entropy shows distinct violin plot patterns for malware, phishing, and spam.

(c) Symbol count, digit count, path URL ratio, and path length strongly correlate with type.

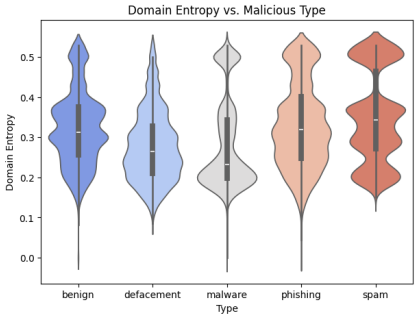
(d) ".exe" and ".m" are the 2nd and 3rd most frequent TLDs.

(e) Defacement and spam URLs have higher symbol counts.

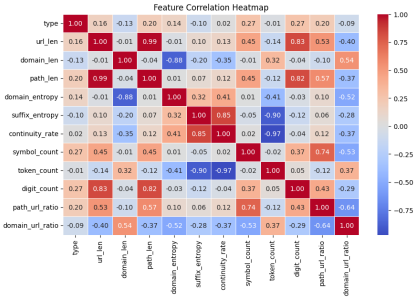
(f) Phishing URLs exhibit a spikier domain entropy distribution.



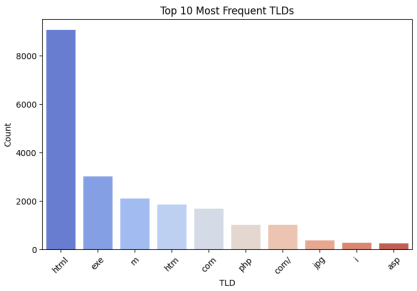
(a) URL length vs Type



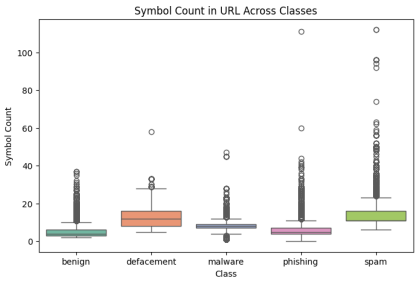
(b) Domain entropy vs Type



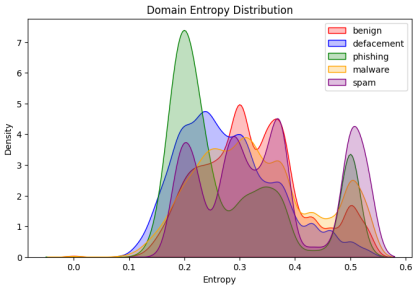
(c) Lexical Feature Heatmap



(d) Most Frequent TLDs



(e) Symbol Count vs Type

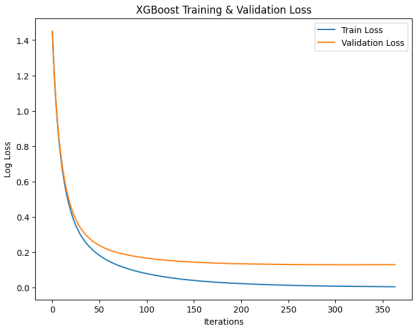


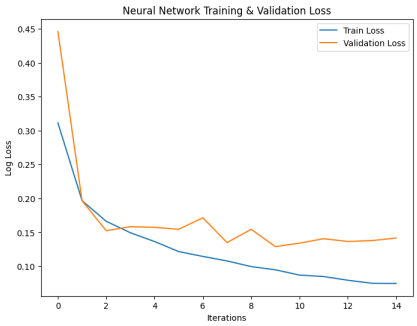
(f) Entropy Distribution

Figure 2: Key EDA visualizations

4 Model Training and Evaluation

We trained three models: XGBoost, a Neural Network, and a Transformer-based model. Their performance metrics are compared below:





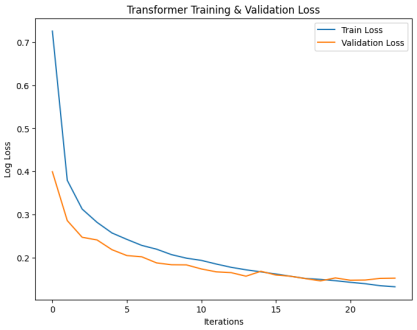


Figure 3: Loss curves for all models

Model	Train F1	Validation F1	Test F1
XGBoost	**1.00**	**0.96**	**0.96**
Neural Network	0.98	**0.96**	**0.96**
Transformer	0.96	0.95	0.94

Table 1: Performance comparison of models (F1)

Model	Train Acc	Validation Acc	Test Acc
XGBoost	**1.00**	**0.96**	**0.96**
Neural Network	0.98	**0.96**	**0.96**
Transformer	0.96	0.95	0.94

Table 2: Performance comparison of models (Accuracy)

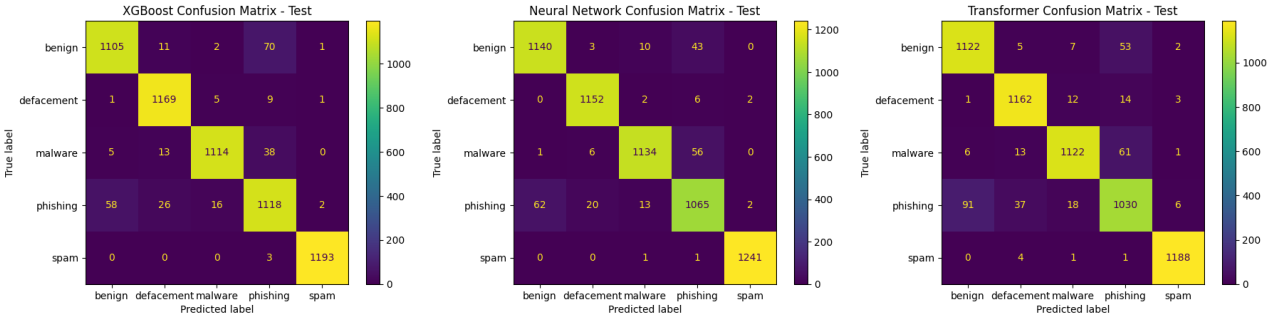


Figure 4: Confusion matrices on test set

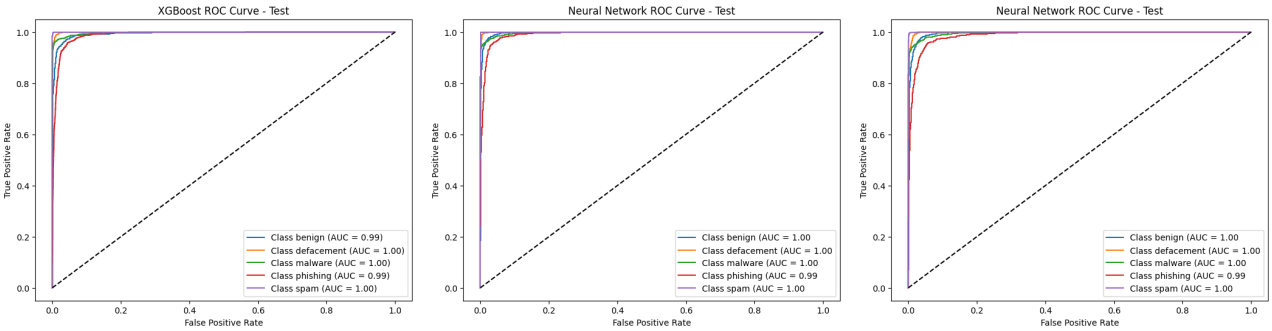


Figure 5: ROC curves on test set

5 Analysis and Discussion

From our evaluation, XGBoost outperformed both the Neural Network and Transformer models. The key reasons behind its success include:

- Strong handling of structured features with tabular data.
- Robust performance on smaller datasets compared to deep learning approaches.
- Effective internal feature selection and importance ranking.

6 Challenges and Future Improvements

One major challenge was extracting meaningful embeddings from URLs. While BERT embeddings provided some useful information, they might not be the best representation of URL semantics. Future work could explore improved embedding techniques tailored for URL-based data.