STAT 576 – DATA INFORMATICS

STAT 576 FALL 2024: UCI PHISHING URL PROJECT

Authored By: *Cory Suzuki, Nate Talampas, Richard Diazdeleon*

1. Motivation and Data Summary

In today's interconnected digital world, phishing attacks have become a pervasive threat to individuals and organizations, exploiting users' trust to steal sensitive information and cause significant financial harm. The increasing sophistication of phishing schemes necessitates advanced analytical methods to effectively identify and mitigate these attacks. Understanding the patterns and characteristics of phishing websites is a critical step towards developing robust detection and prevention mechanisms.

To address this challenge, we utilize the Phishing Website URL Dataset from the UCI Machine Learning Repository. Below is

- Summarization of the Phishing Website URL Dataset:
  - **Number of Observations:** 235,795 instances.
  - **Number of Features:** 55 features.
  - **Feature Description:**
    - Presence of IP addresses in URLs.
    - URL length.
    - Use of "@" symbols.
    - SSL certificate status.
  - **Purpose:** To support the development and evaluation of machine learning models for distinguishing legitimate and phishing websites.
  - **Source:** UCI Machine Learning Repository.
  - **Year Compiled:** 2015 by Rami Mohammad and Lee McCluskey.

The dataset, with 54 diverse features spanning URL characteristics, domain properties, website structure, and security indicators, presents a high-dimensional feature space that necessitates dimensionality reduction to simplify analysis, reduce redundancy, and enhance computational efficiency. Techniques like PCA or t-SNE are employed to preserve essential patterns while projecting the data into a lower-dimensional space. In addition, feature selection by correlation between input features and variance threshold is utilized to remove the presence of any features that are too highly correlated with each other

or that cannot distinguish the target well. Clustering is applied to uncover inherent groupings within the data, enabling the identification of phishing behavior patterns and their comparison against labeled categories. K-means clustering is used as a baseline due to its simplicity and efficiency, while advanced approaches, such as hierarchical clustering or combining dimensionality reduction with clustering (K-means), are explored to achieve more accurate groupings and deeper insights into phishing URL detection.

Clustering Objective

By applying clustering algorithms and dimensionality reduction techniques, this project aims to identify patterns within the data that differentiate phishing websites from legitimate ones. Specifically, clustering techniques are employed to uncover inherent groupings within the dataset, enabling us to analyze phishing behaviors in the absence of explicit labels. In this project, we aim to provide promising clusterings generated from KMeans, Agglomerative Clustering, MiniBatch KMeans, and DBSCAN. Our results concluded that KMeans outperformed all other algorithms by analyzing visual separability and the Adjusted Rand Index (ARI) score.

MiniBatch KMeans is similar to the baseline KMeans model with the exception that the model is trained on the training set in incremental batches to accelerate training computational runtime. Agglomerative Clustering clusters data points in a hierarchal fashion and DBSCAN is a density-based algorithm that clusters based on deterministic choices of the epsilon (distance radius) and minimum sample points hyperparameters, in which we introduce a semi-supervised learning DBSCAN algorithm to aid in our analysis.

2. Descriptive Statistics

2.1 URLCharProb Distribution

- The distribution is unimodal and skewed left, concentrated around 0.06. This suggests that a high probability of specific URL characteristics are legitimate.
- Outliers are observed on both ends of the distribution, indicating some URLs deviate significantly from the central tendency.

2.2 DomainLength Distribution

- Most domain lengths fall between 15 and 30 characters.
- The distribution is right-skewed, showing that longer URL domains (greater than 30 characters) are rare.

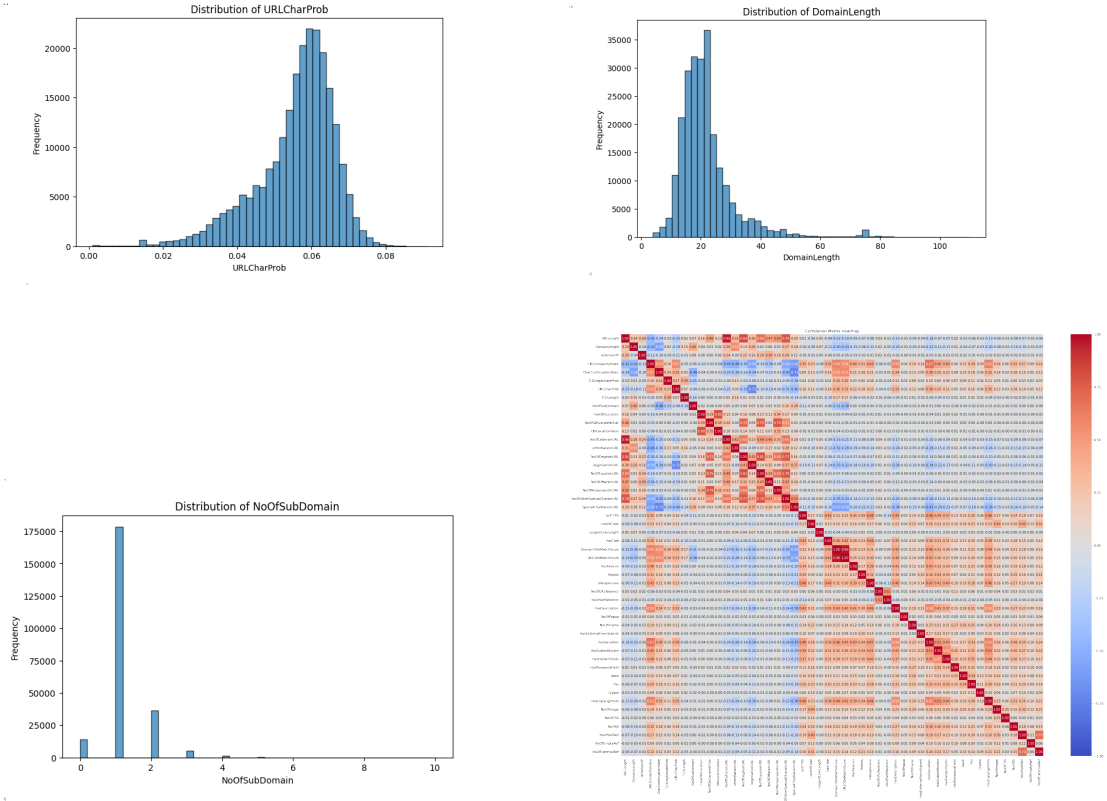

Figure 1: Histogram and Correlation Heatmap

2.3 NoOfSubDomain Distribution

- The majority of URLs have 1 or 2 subdomains.
- Subdomains beyond 4 are rare, indicating that most domains have a simple hierarchical structure.

2.4 Correlation Heatmap

- **Strong Positive Correlations:** `URLLength` and `NoOfLettersInURL` exhibit a strong positive correlation. This relationship aligns with the structural composition of URLs. Similarly, `DomainTitleMatchScore` and `URLTitleMatchScore` also exhibit a strong positive correlation. To prevent collinearity, features `NoOfLettersInURL` and `URLLength` were dropped.

2.5 Variance Threshold

- A variance threshold of 0.1 was employed to remove low-variance features that could not distinguish the target variable well. After implementing this feature

selection method, 36 features remained.

- The remaining features are:

```
URLLength, DomainLength, URLSimilarityIndex, TLDLength,
 NoOfSubDomain, NoOfObfuscatedChar, NoOfLettersInURL,
 NoOfDegitsInURL, NoOfEqualsInURL, NoOfAmpersandInURL,
 NoOfOtherSpecialCharsInURL, IsHTTPS, LineOfCode,
 LargestLineLength, HasTitle, DomainTitleMatchScore,
 URLTitleMatchScore, HasFavicon, Robots, IsResponsive,
 NoOfURLRedirect, HasDescription, NoOfPopup, NoOfiFrame,
 HasSocialNet, HasSubmitButton, HasHiddenFields, Bank, Pay,
 HasCopyrightInfo, NoOfImage, NoOfCSS, NoOfJS, NoOfSelfRef,
 NoOfEmptyRef, NoOfExternalRef
```

Overall Descriptive Summary

- The dataset primarily consists of short to moderately long URLs with simple subdomain structures.
- Probabilistic features such as `URLCharProb` exhibit a strong central tendency with some notable outliers. Due to the length limitations of the report, we strongly suggest that the reader analyzes the visualizations of each distribution in the accompanying code file titled "Phishing_Final_576.ipynb".
- Feature selection by correlation between input features was employed, removing highly correlated features with a correlation coefficient above the deterministic threshold of 0.85. This step ensured that redundant features with correlations close to 1 were excluded, minimizing multicollinearity and simplifying the feature set for downstream predictive modeling tasks.
- Feature selection using a variance threshold of 0.1 was also used to remove low-variance features that could not distinguish the target variable effectively. Additionally, we removed `URL` and `Domain`, as they were deemed unique identifiers and provided little value for generalization.

3. Final Model Description and Results

3.1 Dimensionality Reduction Method and Defaults

The dataset consists of 235,795 observations with 34 features after dimensionality reduction, along with a binary target variable (`label`) indicating phishing (`1`) or non-phishing (`0`) websites. The target class proportions show a slight imbalance, with approximately

57.2% phishing and 42.8% non-phishing, which is preserved after sampling 5,000 observations (phishing: 56.8%, non-phishing: 43.2%). The feature set includes URL characteristics, domain properties, obfuscation metrics, and security indicators. Dimensionality reduction is applied to simplify analysis, retain essential information, and reduce computational costs for clustering. Clustering is employed to group similar instances and identify patterns indicative of phishing behavior, particularly valuable when dealing with ambiguous or incomplete labels.

3.2 Dimensionality Reduction Methods

Linear Methods

Principal Component Analysis (PCA)

PCA generates a covariance matrix from the centered data, which emphasizes the covariance between the features. The process involves performing eigendecomposition or singular value decomposition (SVD) to extract eigenvalues and eigenvectors. The principal components correspond to the eigenvectors associated with the k largest eigenvalues.

Multidimensional Scaling (MDS)

MDS aims to find a low-dimensional representation of the data while preserving the pairwise distance between points. The process involves generating a pairwise distance matrix from the data and converting it into a Gram matrix or inner product matrix. By performing eigendecomposition on this matrix, we obtain the eigenvalues and eigenvectors and select the k largest eigenvalues to construct the low-dimensional embedding.

Limitations of Linear Approaches

Linear methods often struggle to capture complex, nonlinear relationships in high-dimensional data, which prompts the need for nonlinear techniques.

Nonlinear Methods

Isomap

Isomap aims to find the optimal lower-dimensional representations that preserve the global geodesic distances between the data points on a manifold. Unlike MDS, which utilizes Euclidean distance, Isomap relies on geodesic distances, computed using a k-nearest-neighbor graph. This approach holds the assumption that the data is sufficiently large and well-distributed to reveal the underlying manifold structure. This algorithm is effective for manifold-based data but may struggle with outliers or manifolds with high curvature or large holes.

Locally Linear Embedding (LLE)

LLE assumes that the data points and their neighbors are in a local linear space. This method approximates each data point as a linear combination of its neighbors, using weights that minimize the reconstruction error. The process involves calculating the weight from neighbors:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{k} w_{ij} \mathbf{x}_j \right\|^2,$$

After fixing the weights, the low-dimensional vectors $\mathbf{z}_i$ are determined by minimizing the following cost function:

$$E(\mathbf{Z}) = \sum_{i=1}^{n} \left\| \mathbf{z}_i - \sum_{j=1}^{k} \mathbf{W}^* \cdot \mathbf{z}_j \right\|^2$$

which can then be utilized to map the embedded coordinates into a lower-dimensional space.

This process has limitations, such as the lack of an out-of-sample extension and limited theoretical guidance about the number of intrinsic dimensions.

t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is designed to visualize high-dimensional data by preserving local similarities, emphasizing neighborhoods and short-range interactions between points. The process involves calculating the probability that data point i selects data point j as its neighbor based on their Euclidean distance:

$$p_{ij} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}.$$

In the low-dimensional space, a similar probability q_{ij} is calculated using a t-distribution with one degree of freedom:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}.$$

The similarity between p_{ij} and q_{ij} is measured using the Kullback-Leibler (KL) divergence:

$$\mathrm{KL}(P \| Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

One notable issue with t-SNE is the *crowding problem*, where data points are crowded too close together when mapped to a lower-dimensional space. In the original high-dimensional space, neighbors may be relatively far apart. To address this issue, t-SNE uses a t-distribution instead of a Gaussian distribution, as the former's fatter tails allow for better modeling of distant points and potential outliers.

The formula for q_{ij} remains the same as above, while the p_{ij} formula is unchanged.

Findings

t-SNE visualized distinct groupings of the data. This result demonstrates that t-SNE's ability to preserve local similarities and emphasize the relationship between neighborhoods and short-range interactions between points is most significant when embedding the data into a lower-dimensional space. PCA was able to capture two dimensions that separated the two classes, but the data was not well-separated enough to create clusters. MDS, Isomap, and LLE failed to separate the classes (refer to data visualizations on page 8).

3.3 Suggested Clustering

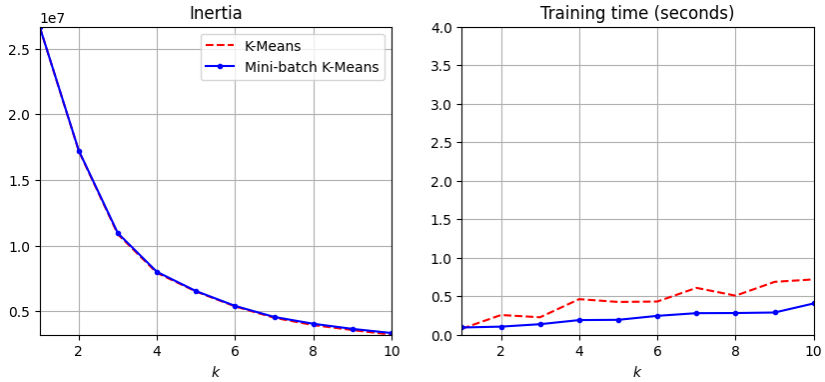

Figure 17: Silhoutte score using the elbow method

- **K-means vs Batch:** The left graph shows that inertia decreases significantly as k increases, starting from around 2.5×10^7 at $k = 2$ to less than 5×10^6 at $k = 10$, with diminishing returns after $k = 5$, indicating the potential elbow point. The right graph illustrates that training time grows as k increases, with K-Means taking nearly 1 second at $k = 10$ compared to Mini-Batch K-Means, which remains under 0.5 seconds throughout. Mini-Batch K-Means demonstrates consistently lower computational time while maintaining comparable inertia, making it more efficient for larger datasets.

3.4 Improvements

Model Selection Enhancements: The default K-Means model exhibited limitations, particularly with overlapping clusters. Mini-Batch K-Means was implemented as an alternative, demonstrating substantial improvements in training time without sacrificing clustering performance. Additionally, non-linear dimensionality reduction methods such as Isomap and Locally Linear Embedding (LLE) provided deeper insights into complex data patterns, uncovering relationships that linear methods like PCA were unable to capture.

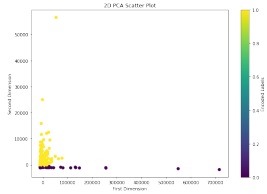

Figure 2: PCA

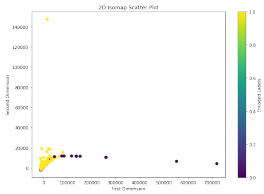

Figure 3: Isomap

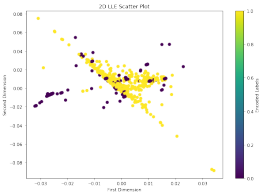

Figure 4: LLE

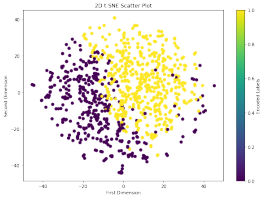

Figure 5: t-SNE

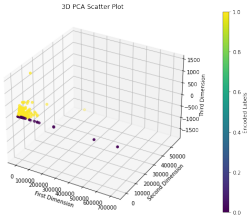

Figure 6: PCA 3D

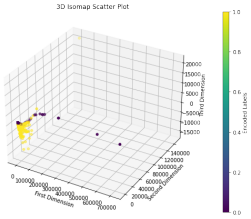

Figure 7: Isomap 3D

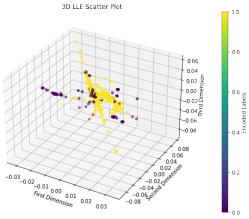

Figure 8: LLE 3D

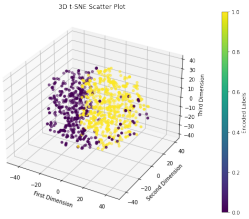

Figure 9: t-SNE 3D

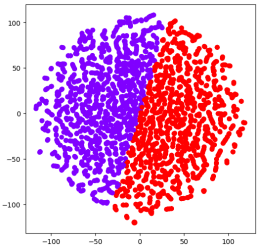

Figure 10: K-Means

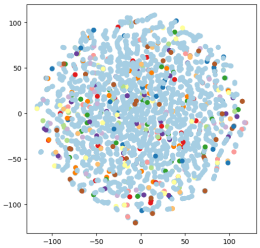

Figure 11: DBSCAN

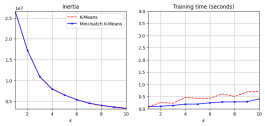

Figure 12: Silhoutte

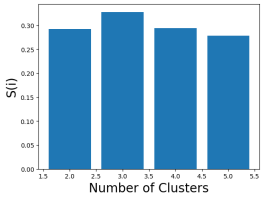

Figure 13: Clusters

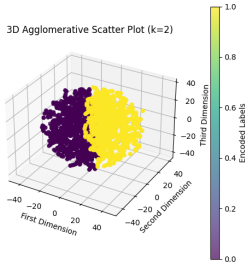

Figure 14: 3D Agglomerative

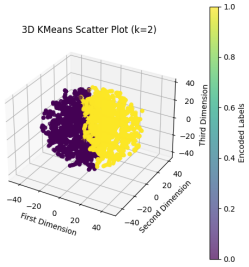

Figure 15: 3D KMeans

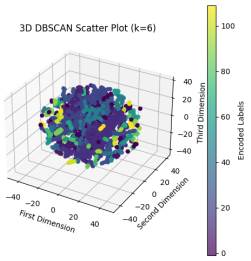

Figure 16: 3D DBSCAN

Comparison of dimensionality reduction and clustering methods: PCA, MDS, Isomap, LLE, T-SNE, and DBSCAN visualizations, along with silhouette and cluster bar graphs.

Algorithm Optimization: The adoption of Mini-Batch K-Means over traditional K-Means significantly reduced computational costs, as evident in training time comparisons. Mini-Batch K-Means leveraged stochastic updates to achieve scalability and efficiency, making it particularly suitable for large datasets. This optimization ensured the clustering process was both time-efficient and capable of maintaining robust performance. For DBSCAN, we also provide a glimpse into a more optimized DBSCAN result which still performs worse than KMeans and Agglomerative clustering. This tuned version uses the k-distance graph to estimate the epsilon hyperparameter based upon the KNN distances between points, thereby being a semi-supervised algorithm. The results are given below.

Table 1: Comparison of Clustering Methods in 2 Dimensions

Clustering Method	**ARI**	**Observations**
K-Means (Default)	0.2368	Moderate performance.
Mini-Batch K-Means	0.2368	Faster training in batches.
Agglomerative Clustering	0.1557	Handles hierarchical groupings and computationally efficient.
DBSCAN	0.0056	Struggled with sparse clusters, density-based.

Table 2: Comparison of Clustering Methods in 3 Dimensions

Clustering Method	**ARI**	**Observations**
K-Means (Default)	0.5346	Best overall performance.
Mini-Batch K-Means	0.5346	Faster training, best performance.
Agglomerative Clustering	0.0155	Performed worse than 2D space.
DBSCAN	0.0159	Struggled with clustering in 3D space.

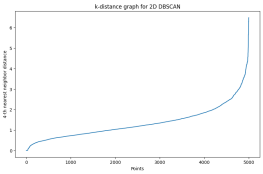

Figure 18: K-Distance (k=4)

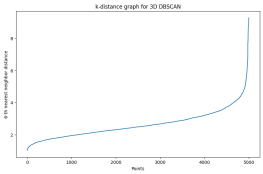

Figure 19: K-Distance (k=6)

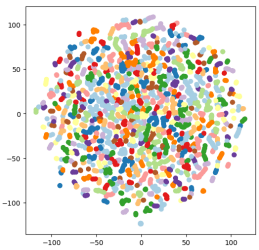

Figure 20: DBSCAN SS 2D

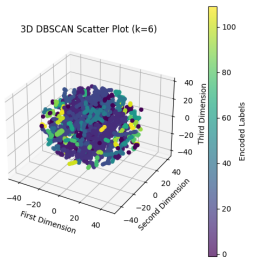

Figure 21: DBSCAN SS 3D

K-Distance Plots and Semi-Supervised DBSCAN visualizations.

4 Discussion and Analysis

- **Highlighting Improvements Over the Default Model:**
  The default clustering model (K-Means) showed limitations in training time and inertia for larger values of k. Mini-Batch K-Means provided faster convergence and comparable clustering accuracy, while Isomap and LLE allowed for better separation of phishing and legitimate labels in the feature space.

- **Comparison to Original Labels:**
  Clustering results were evaluated against the original labels (1 for phishing, 0 for legitimate). While DBSCAN struggled to effectively classify the two classes, K-Means produced the clearest separability between phishing and legitimate labels. Agglomerative Clustering also demonstrated reasonable performance, though it did not achieve the same level of clarity as K-Means.

- **Reasoning Behind Model Combinations:**
  The combination of dimensionality reduction techniques (PCA, Isomap, and LLE) and clustering algorithms (Mini-Batch K-Means) was chosen to balance computational efficiency and clustering accuracy. Non-linear dimensionality reduction methods were selected to capture underlying data structures that the linear methods could not identify. Agglomerative Clustering was explored due to its hierarchical nature, complementing K-Means, but DBSCAN's density-based approach proved less effective for this dataset.

- **Domain-Specific Findings:**
  Phishing URLs showed distinct patterns in features such as URL length, obfuscation ratios, and security indicators (e.g., HTTPS usage). Clustering models effectively identified groups of URLs with similar phishing-related attributes, providing actionable insights for improving cybersecurity measures. The better performance of K-Means highlights its suitability for this type of data, where cluster shapes align well with spherical boundaries, while DBSCAN's limitations suggest that the dataset's density distribution may not align with its assumptions, as more than 2 clusters were found.

5. Additional Remarks

Limitations

- **Dataset Representativeness:** The dataset was compiled in 2015, so it currently may not fully capture modern phishing techniques and evolving trends in cybersecurity which is a more time-dependent field and may require further analysis.

- **Clustering Assumptions:** Certain clustering methods, such as K-Means, rely on assumptions like spherical cluster shapes, which may not align with the actual data structure.
- **Memory Storage** While attempting to one-hot encode categorical features for dimensionality reduction, current memory storage was unable to process the resulting matrix which required over 51.7 GiB of memory.
- **Dimensionality Reduction Trade-offs:** Techniques like Principal Component Analysis (PCA) can reduce interpretability by obscuring the relationship between original features and clustering outcomes.

Prospective Future Work

For future extensive analysis on this particular dataset, one key improvement may be to include updated real-time data in order to verify the consistency of our models in dimensionality reducing the data and fitting clustering models. This may also reveal any additional hidden feature information that accompanies the addition of new, unseen data. Another direction we could move forward with is the potential of further optimizing our hyperparameters for the clustering methods and potentially explore deep-learning techniques to achieve more accurate and robust clustering results. To reiterate, many of the hyperparameters used are deterministic and were up to our discretion. To further this, we may need to experiment with different distance metrics as we had assumed the usage of the Euclidian metric as our default for measuring distances.

Summary of Significant Results

The project faced challenges such as the dataset's age, clustering assumptions, and the representative size of the sampled dataset from dimensionality reduction. Future work should focus on utilizing modern datasets, exploring more advanced clustering methods via semi-supervised learning techniques, and applying anomaly detection algorithms via tree-based methodologies to construct practical phishing URL detection systems. These efforts will enhance the robustness and applicability of the findings on phishing URL classification and further the effectiveness of these unsupervised learning algorithms on real-world data.

References

[1] DataCamp. *A Guide to the DBSCAN Algorithm: Determining the Epsilon Parameter*. Accessed 30 November 2024. https://www.datacamp.com/tutorial/dbscan-clustering-algorithm.

[2] Lee, Seungjoon. STAT 576 Lectures and Code Handouts. CSULB. Accessed Fall 2024.

[3] Prasad, A., & Chandra, S. (2023). *PhiUSIIL: A Diverse Security Profile Empowered Phishing URL Detection Framework Based on Similarity Index and Incremental Learning.* Computers & Security, 103545. doi: https://doi.org/10.1016/j.cose.2023.103545