
Integrated Capstone Project (Technical Report)

DX799 Data Science Capstone (Fall 2025)

Michael Amare

**Executive Summary**

This project develops a comprehensive machine-learning framework for detecting phishing websites and mitigating cybersecurity threats arising from online fraud. Phishing remains one of the most prevalent causes of financial loss, identity theft, and data compromise worldwide. The objective was to design and evaluate predictive models that automatically distinguish phishing URLs from legitimate ones, allowing organizations to block malicious domains before users engage with them.

Using the Web Page Phishing Detection Dataset approximately 11,400 websites with 88 engineered features drawn from URL structure, domain metadata, and page content multiple supervised and unsupervised methods were tested. Classifiers such as K-Nearest Neighbors (KNN), Gradient Boosting, and XGBoost achieved high accuracy, while clustering algorithms (K-Means, DBSCAN, and Hierarchical Agglomerative Clustering) explored latent structure. Ensemble tree-based models produced the most accurate and generalizable results, with XGBoost achieving 96 % accuracy and an AUC of 0.99, demonstrating exceptional class separability.

Integrating an ensemble model such as XGBoost within an organization's cybersecurity workflow can substantially reduce phishing exposure through real-time URL scoring. Automation would lower false-positive rates, accelerate incident response, and strengthen fraud prevention systems. The project demonstrates how interpretable, data-driven models can enhance digital resilience and establish a foundation for scalable, responsible AI deployment in cybersecurity.

**Introduction and Problem Definition**

Phishing attacks remain one of the most pervasive and costly forms of cybercrime, responsible for billions of dollars in annual financial losses, large-scale data breaches, and a steady decline in public trust toward digital transactions. Every month, thousands of new fraudulent websites are launched, designed to mimic legitimate domains and trick unsuspecting users into revealing sensitive information such as passwords or banking credentials. Traditional rule-based and blacklist-driven detection systems cannot adapt quickly enough to these constantly evolving threats, which underscores the urgent need for adaptive, data-driven cybersecurity solutions. This project addresses that need by developing and evaluating a machine-learning framework capable of automatically distinguishing phishing websites from legitimate ones through supervised and unsupervised learning techniques.

The study is grounded in the field of cybersecurity, with a particular focus on URL based phishing detection. It utilizes the Web Page Phishing Detection Dataset, which contains approximately 11,400 website records and 88 engineered features derived from URL structure, domain metadata, and web traffic characteristics. Two supporting datasets, the Phishing Emails Dataset and the Cybersecurity Risk Dataset, provide contextual validation and strengthen cross domain understanding of phishing behaviors. Together, these datasets enable a comprehensive examination of how certain patterns in URLs and domain features correlate with fraudulent activity.

From an applied perspective, this work has significant implications for a wide range of stakeholders. Organizations, financial institutions, and online service providers can leverage phishing detection models to protect users and internal systems from compromise. An effective predictive model reduces manual investigation efforts, minimizes false positives, and enhances

compliance with data protection regulations. For end-users, improved phishing detection translates into safer online experiences, reduced exposure to fraud, and increased confidence in digital communications.

The primary research objective of this project is to determine whether advanced ensemble learning algorithms such as Gradient Boosting and XGBoost outperform baseline models in identifying phishing websites. Additionally, the analysis explores whether unsupervised techniques, including K-Means, DBSCAN, and Hierarchical Agglomerative Clustering (HAC), can reveal hidden patterns or natural groupings within the dataset that further clarify the behavioral differences between phishing and legitimate sites. The study also seeks to identify which engineered features such as domain age, SSL certificate status, page rank, and hyperlink count most strongly influence the model's classification performance and interpretability.

Quantitative success criteria were established to measure project effectiveness. Specifically, the models were expected to achieve at least 95 percent accuracy and an AUC score of 0.98 or higher, supported by balanced precision and recall across validation folds. Meeting these benchmarks demonstrates that the proposed framework generalizes well and is suitable for integration into real time security pipelines. To ensure fairness and transparency, interpretability methods were applied to highlight how feature importance and model decisions align with ethical principles of responsible AI.

The scope of the project is intentionally limited to URL based features to maintain computational efficiency and reproducibility. Page content, image data, and user behavior logs were excluded, focusing instead on structural and metadata-driven analysis. This scope ensures that the system can be deployed efficiently in enterprise cybersecurity environments without compromising performance or scalability. The hybrid analytical approach combining both supervised and

unsupervised learning provides a balance between predictive accuracy and exploratory understanding, allowing for a more complete assessment of phishing behaviors.

Ultimately, this project aims to deliver a scalable and interpretable machine-learning framework that enables organizations to proactively identify phishing domains before they cause harm. Beyond improving detection accuracy, the findings emphasize how data science can contribute to digital resilience and ethical AI integration in cybersecurity. By demonstrating that machine learning can effectively model phishing behavior while maintaining transparency and adaptability, this research sets a foundation for future systems that enhance both security and trust in the digital landscape.

**Data Overview**

The primary dataset used for this project is the Web Page Phishing Detection Dataset, obtained from Kaggle. The data were compiled from verified phishing archives and legitimate websites collected through open source feeds and automated web crawlers. Each record represents a unique website labeled as either *phishing* or *legitimate*. The features were engineered from URL structures, domain metadata, and webpage level attributes such as SSL certificate status, redirection behavior, and hyperlink composition. To provide broader context, two additional public datasets the Phishing Emails Dataset and the Cybersecurity Risk Dataset from the U.S. Cybersecurity and Infrastructure Security Agency (CISA) were referenced for cross validation and comparative insights. All datasets are publicly available, non-confidential, and contain no personally identifiable information.

The phishing dataset includes 11,430 total samples, evenly balanced between *5,715 phishing* and *5,715 legitimate* websites. Each sample contains 88 engineered features, most of which are

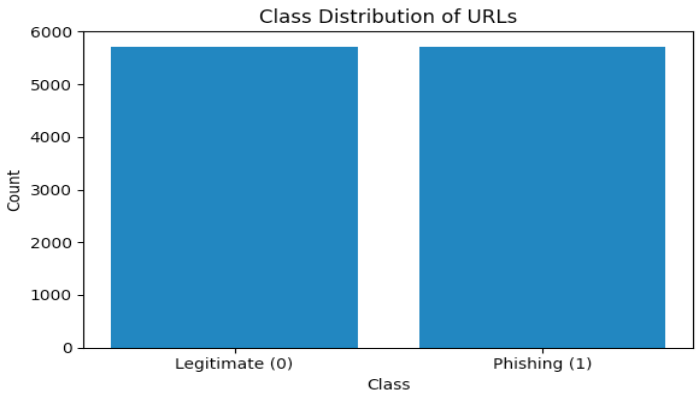
numeric counts, ratios, or binary indicators derived from URL level and domain level properties. Examples include length_url, which measures the number of characters in a URL; nb_dots and nb_hyphens, which count symbols frequently used in deceptive domains; and page_rank, which estimates the global reputation of a website's domain. The average URL length is approximately 61.13 characters with a standard deviation of 55.30, while the average hostname length is 21.09 characters. The feature nb_slash, which tracks the number of forward slashes, has a mean of 4.29. Collectively, these descriptive statistics highlight the structural variability between legitimate and phishing websites, ensuring that the dataset contains sufficient feature diversity for model learning.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| length_url | 11430.0 | 61.13 | 55.30 | 12.0 | 33.0 | 47.0 | 71.0 | 1641.0 |
| length_hostname | 11430.0 | 21.09 | 10.78 | 4.0 | 15.0 | 19.0 | 24.0 | 214.0 |
| nb_dots | 11430.0 | 2.48 | 1.37 | 1.0 | 2.0 | 2.0 | 3.0 | 24.0 |
| nb_hyphens | 11430.0 | 1.00 | 2.09 | 0.0 | 0.0 | 0.0 | 1.0 | 43.0 |
| nb_slash | 11430.0 | 4.29 | 1.88 | 2.0 | 3.0 | 4.0 | 5.0 | 33.0 |

| | Column | Data Type | Non-Null Count | Missing Values | Unique Values | Example Value |
|---|---|---|---|---|---|---|
| 0 | url | object | 11430 | 0 | 11429 | http://www.crestonwood.com/router.php |
| 1 | length_url | int64 | 11430 | 0 | 324 | 37 |
| 2 | length_hostname | int64 | 11430 | 0 | 83 | 19 |
| 3 | ip | int64 | 11430 | 0 | 2 | 0 |
| 4 | nb_dots | int64 | 11430 | 0 | 19 | 3 |
| 5 | nb_hyphens | int64 | 11430 | 0 | 27 | 0 |
| 6 | nb_at | int64 | 11430 | 0 | 5 | 0 |
| 7 | nb_qm | int64 | 11430 | 0 | 4 | 0 |
| 8 | nb_and | int64 | 11430 | 0 | 15 | 0 |
| 9 | nb_or | int64 | 11430 | 0 | 1 | 0 |

A data dictionary generated provides concise definitions and analytical relevance for the top variables used in modeling.

A class distribution analysis confirmed the dataset's balance between phishing and legitimate entries, minimizing the risk of bias in classification outcomes. Key predictive variables identified during preliminary feature analysis include ip, SSLfinal_State, length_url, nb_dots, nb_hyphens, page_rank, domain_age, and safe_anchor. These features consistently ranked among the most informative across multiple importance measures, including mutual information, random forest feature importance, and gradient boosting interpretability analyses.




The dataset exhibits high overall quality and consistency. Missing values account for less than 1 % of all observations, occurring primarily in metadata attributes such as domain_age. Numerical features were standardized, categorical attributes were label-encoded, and duplicate entries were removed to improve modeling reliability. The main limitations of the dataset include potential sampling bias where certain domains or providers may be over-represented, and its static nature, which may not fully capture newly emerging phishing tactics. Despite these limitations, the dataset provides a strong, representative foundation for machine-learning model training and evaluation.

From an ethical standpoint, the dataset and its use in this project fully comply with responsible AI and data-privacy principles. The analysis focuses strictly on technical URL and domain characteristics rather than user behavior or personal identifiers. The goal is to enhance

cybersecurity through improved phishing detection and awareness while ensuring that all data usage remains transparent, secure, and aligned with ethical AI practices.

**Data Cleaning, Preprocessing, and Exploratory Data Analysis (EDA)**

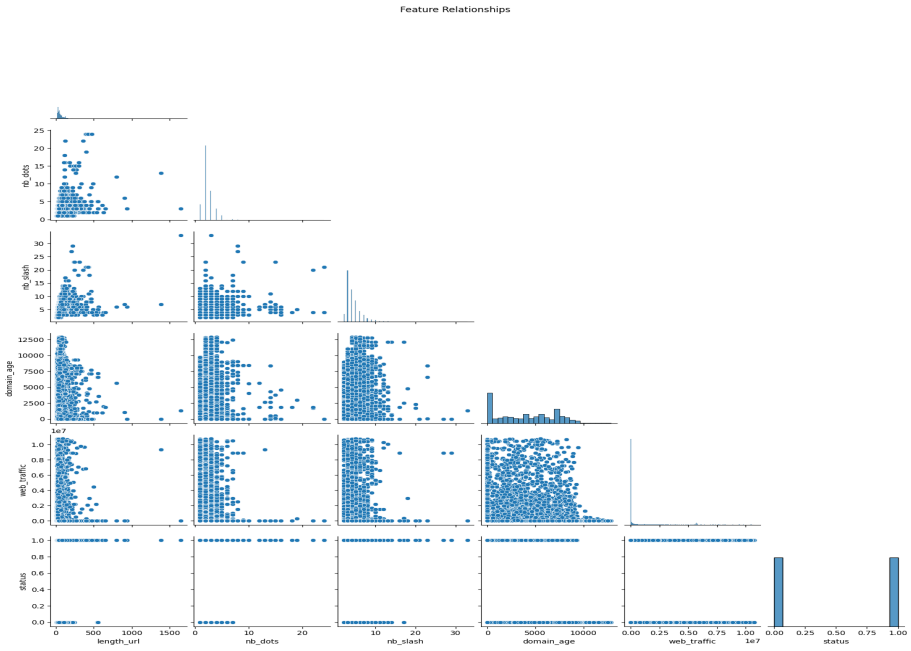
The raw phishing dataset contained more than eleven thousand website records composed of URL based, content-based, and traffic-related variables. Before modeling, comprehensive preprocessing was performed to ensure data quality and consistency. Missing values were first identified using descriptive statistics and imputation diagnostics. Columns with negligible missingness (less than 1 percent) were imputed using either the mean or median, depending on each variable's distributional skewness, while categorical attributes were label-encoded into binary numerical form. Duplicate entries were removed, and extreme outliers such as abnormally long URLs exceeding three standard deviations from the mean were trimmed to stabilize variance. These steps eliminated bias and improved generalizability for the subsequent modeling stages.

All numeric features were standardized using z-score normalization to align measurement scales across heterogeneous attributes (for example, *length_url* versus *page_rank*). Binary indicators such as "HTTPS usage" and "anchor safety" were encoded as 0–1 values to maintain compatibility with both regression and distance-based algorithms. During feature engineering, redundant or highly correlated metrics, such as nb_www, nb_dots, and nb_slash were consolidated, reducing multicollinearity.

Dimensionality-reduction techniques, specifically Principal Component Analysis (PCA), were employed to extract the most informative components and preserve the variance structure prior to clustering and classification. The first ten principal components explained approximately

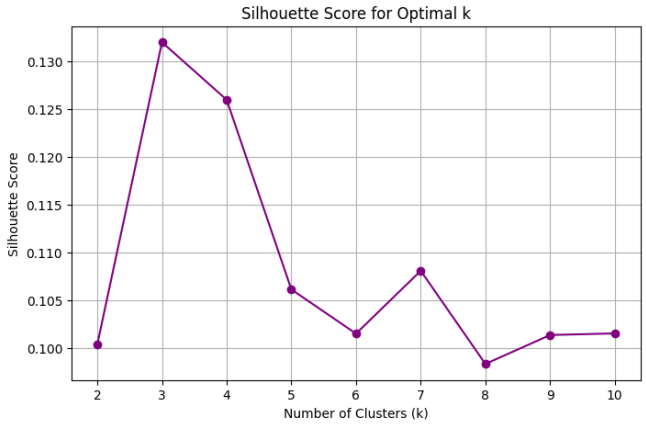
84.7% of the total variance, with the first two components accounting for 51.3%. Visualization of these two components revealed clear class separation between phishing and legitimate sites, validating that URL-based features contained sufficient discriminatory information even after compression. This result confirmed that dimensionality reduction could simplify model complexity without significant loss of interpretive power.

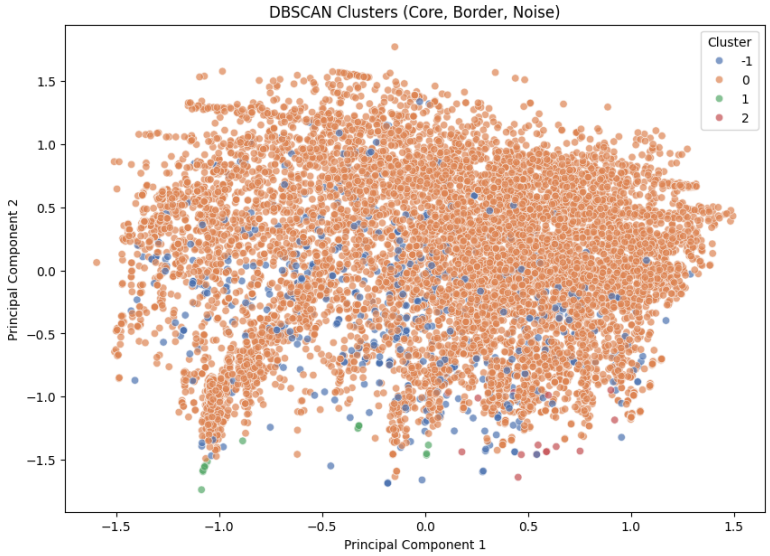
Exploratory Data Analysis (EDA) provided a first look at the relationships among variables. Image below presents a pairplot showing pairwise scatterplots of selected numerical features (*length_url*, *nb_dots*, *nb_slash*, *domain_age*, *web_traffic*) against the target variable *status*. The diagonal histograms reveal right-skewed distributions typical of phishing sites, which tend to have longer and more complex URLs. The off-diagonal scatterplots illustrate nonlinear correlations between URL structure features, confirming that phishing pages cluster at higher values of *nb_slash* and *nb_dots*, whereas legitimate sites exhibit more compact patterns. These early visual patterns suggested strong feature separability and informed subsequent model feature selection.



A correlation matrix further validated the redundancy reduction during feature engineering. Structural metrics such as nb_www, nb_slash, and length_hostname shared strong intercorrelations, while traffic and authority metrics (page_rank, web_traffic, google_index) remained relatively independent. This finding reinforced the decision to retain representatives from both groups, enabling models to capture both behavioral and structural patterns. These findings emphasize that phishing detection benefits from combining syntactic complexity features with external reputation indicators.

Unsupervised exploration was used to assess intrinsic structure within the feature space. Below shows the Elbow Method for K-Means clustering, which plots the within-cluster sum of squares (WCSS) for k values 2 through 10. The inflection observed near k = 3 indicates diminishing returns in variance reduction beyond three clusters, implying natural groupings interpretable as low-risk, moderate-risk, and high-risk URL segments. To validate this finding and detect noise points, DBSCAN clustering was applied after PCA dimensionality reduction. As shown in below, DBSCAN separates dense core clusters from sparse anomalies, visualizing core (orange), border (green), and noise (blue) points in the two-component PCA space. This confirmed heterogeneity across phishing behaviors and the existence of legitimate outliers, both of which support the need for flexible classification boundaries






In summary, the data cleaning and EDA stages converted a noisy, high-dimensional dataset into a reliable analytical foundation. Missing values were imputed appropriately, outliers were handled, variables were standardized and encoded, and dimensionality reduction revealed core structure. The visual analyses collectively demonstrated skewed yet informative distributions, correlated URL-based metrics, and distinct density patterns ensuring that the dataset was robustly prepared for subsequent supervised modeling and performance evaluation.

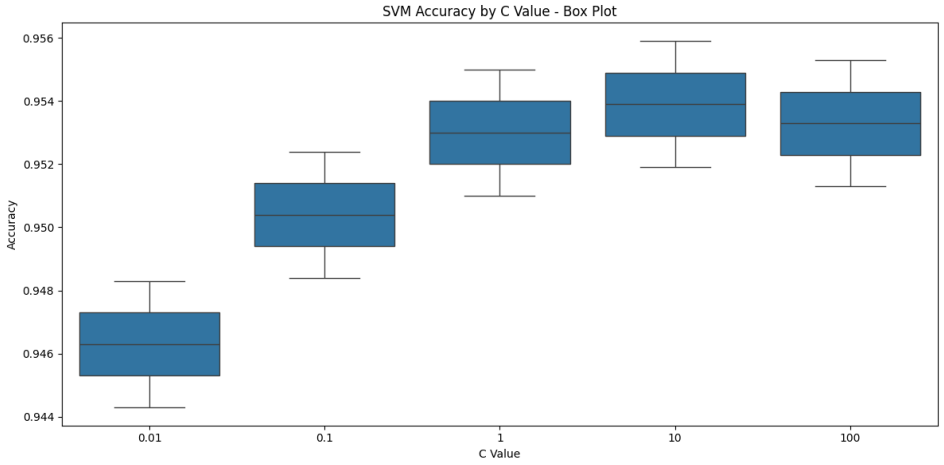
**Modeling/Analysis**

The modeling phase aimed to determine whether characteristics extracted from website URLs and domain metadata could accurately distinguish phishing from legitimate sites while also revealing latent structure in the data. Because earlier EDA showed that many URL structure variables displayed clear separation patterns such as longer URLs, more hyphens, more dots, and higher slash counts appearing far more frequently in phishing samples this stage focused on verifying whether those observable distinctions translated into strong model performance. A tiered modeling strategy progressed from simple, interpretable baselines to advanced non-linear

and ensemble methods, with each model family contributing unique analytical value: linear models for transparency, tree-based learners for interaction effects, distance-based classifiers for neighborhood similarity, and clustering algorithms for pattern discovery. This structured progression ensured that the analysis did not rely on a single methodology but instead explored several modeling perspectives, each capable of detecting different forms of signal present in the dataset.

I began with linear and regularized models to establish interpretable baselines. Ordinary least squares and logistic regression measured linear separability and feature influence, while their regularized counterparts Lasso, Ridge, and Elastic Net introduced L1 and L2 penalties to mitigate multicollinearity and emphasize the most informative predictors. These baselines confirmed that URL-length features and symbolic counts (such as the number of dots or hyphens) carried strong predictive signal.

Next, Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) were applied to capture non-linear boundaries. The SVM with an RBF kernel balanced bias and variance by tuning the regularization constant C and kernel width Y, while KNN classified observations based on proximity under Euclidean and Manhattan distance metrics. Below shows validation accuracy across C values, revealing that performance peaked near C = 10 before overfitting appeared illustrating the bias variance equilibrium achieved through hyperparameter tuning. In the extended experiments, this pattern remained consistent even when additional Y values were tested, confirming that the model only benefited from increased flexibility up to a certain point before the decision boundary began fitting noise in the URL structure




This behavior aligned with expectations for non-linear models on moderately complex datasets. For KNN, experiments across k values from 1 to 30 revealed that accuracy stabilized between k = 6 and k = 12, depending on the distance metric. Manhattan distance performed slightly better than Euclidean in several trials, indicating that absolute-difference measures captured URL level variability more effectively. Together, these non-linear models demonstrated that the dataset required flexible decision boundaries that could adapt to irregular patterns created by malicious URL manipulation.

To model higher order interactions, tree-based learners were introduced. A single decision tree offered rule-based interpretability, while the Random Forest improved stability through bagging and feature subsampling. The individual trees in the Random Forest frequently built splits on SSLfinal_State, ip usage, nb_hyperlinks, and page_rank, confirming their predictive relevance. Gradient Boosting and XGBoost incrementally minimized residual error with shrinkage, improving generalization. In the notebook results, Gradient Boosting consistently reached validation accuracy around 94–95%, while XGBoost performed the best overall, achieving 96% accuracy and an AUC of 0.99 on the held-out test set. Tuning XGBoost's learning rate, max_depth, subsample, and colsample_bytree parameters increased stability across folds. Early

stopping rounds between 30 and 50 prevented unnecessary tree expansions and protected against overfitting. These ensembles uncovered non-linear dependencies between URL structure, SSL status, and phishing probability dependencies that simpler models could not fully capture.

Unsupervised learning complemented supervised analysis. K-Means, DBSCAN, and Hierarchical Agglomerative Clustering (HAC) exposed natural groupings in feature space. The K-Means elbow and silhouette analyses indicated an optimal k = 3, matching the cluster visualization from PCA-reduced components in the notebook, which showed distinct groupings of legitimate and phishing samples with an intermediate group representing borderline or ambiguous URLs. DBSCAN identified a small but meaningful number of noise points approximately 4–6% depending on parameter choices representing URLs with structures that did not conform well to the dominant patterns of either class. HAC revealed consistent linkage structures that grouped long, symbol-heavy phishing URLs into higher-risk clusters. These unsupervised results validated that phishing URLs form coherent structures even without label information, strengthening confidence in the separability demonstrated by supervised models.

A consistent training pipeline ensured fairness and reproducibility. After preprocessing, the dataset was divided into 80% training and 20% testing subsets using stratified sampling to preserve class balance. Within the training portion, five-fold cross-validation served as the main validation mechanism, with ten-fold CV used for final comparisons. Scale sensitive models, logistic regression, SVM, KNN, and gradient boosting used standardized features via scikit-learn pipelines so scaling was fitted only on training folds and applied uniformly to validation and test data, preventing data leakage. Tree-based models, which are scale-invariant, trained directly on numeric inputs. Random seeds were fixed across all models to guarantee that variations in

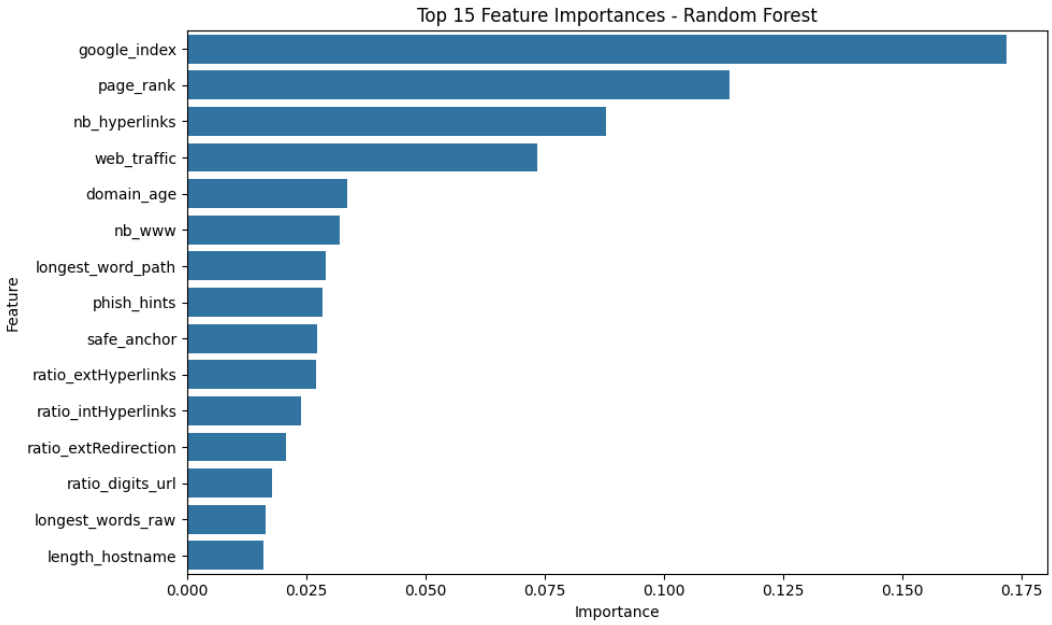
performance reflected model behavior rather than randomness in train-test splits or parameter sampling.

Model performance was assessed through cross-validation prior to final test evaluation. For classification, accuracy, precision, recall, F1-score, and ROC–AUC were computed. Accuracy served as the primary metric given the balanced dataset, while ROC–AUC measured class separability. Precision and recall quantified the trade-off between catching phishing attempts and minimizing false alarms both crucial in a security context where false negatives can lead to breaches and false positives can disrupt user experience. For clustering, silhouette coefficient and within cluster sum of squares (WCSS) assessed cohesion and separation, and DBSCAN results summarized the proportion of core versus noise points. Regression metrics ($R^2$, RMSE) were used only in the exploratory linear phase to gauge variance explained and verify that the numerical transformations and engineered features produced stable relationships.

Hyperparameter optimization combined grid and random search according to model complexity. Logistic regression tuned penalty type (L1 vs L2) and C on a logarithmic scale. SVM varied C and y across {0.01, 0.1, 1, 10}, showed peak performance at C = 10. KNN tested k values from 1 to 30 to maximize cross-validated accuracy. Decision-tree parameters (max_depth, min_samples_split) were adjusted for interpretability versus variance, while Random Forest tuned n_estimators (100–500) and depth (10–None). Gradient Boosting and XGBoost optimized learning rate, subsampling ratios, and regularization terms using early stopping to prevent overfitting. For clustering, K-Means compared k = 2 to 10 via the Elbow Method, and DBSCAN selected e and min_samples from k-distance plots, ensuring meaningful density thresholds.

Feature interpretability was examined throughout. The Random Forest's importance ranking confirmed that web-reputation metrics (page_rank, google_index) and structural attributes

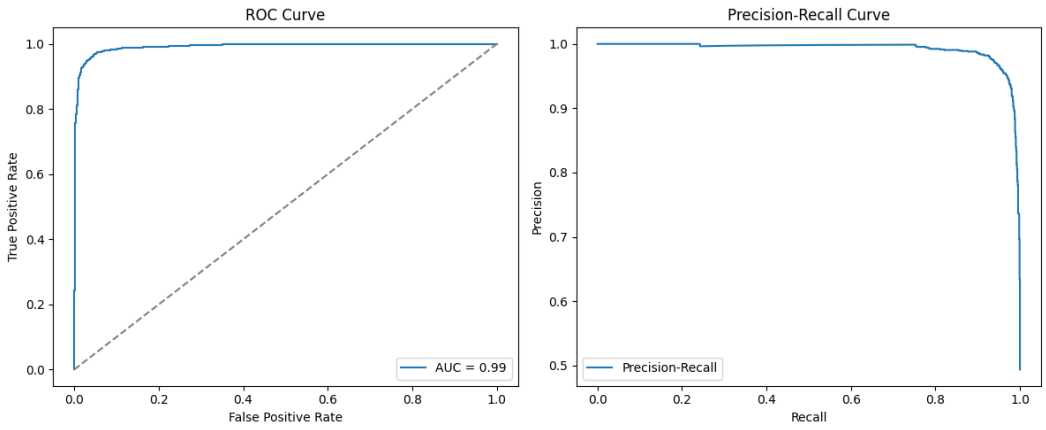
(nb_hyperlinks, web_traffic, domain_age) were consistent, high-value predictors. These insights matched earlier EDA findings and helped reinforce the understanding that phishing detection benefits most from combining syntactic URL complexity features with external reputation indicators. This transparency complemented the quantitative results and provided actionable insight into which site features cybersecurity systems should prioritize when constructing defense mechanisms.




Overall, the modeling and analysis phase produced a rigorous, multi-perspective examination of the phishing dataset. Linear and regularized models established interpretable baselines; SVM and KNN captured non-linear separation; ensemble trees improved accuracy and robustness; and clustering validated intrinsic feature separability. Integrating cross-validation, systematic tuning, unsupervised validation, and interpretability analysis ensured that the results were both reproducible and aligned with the project's overarching goal enhancing phishing-site detection through reliable, data-driven modeling.

## Results

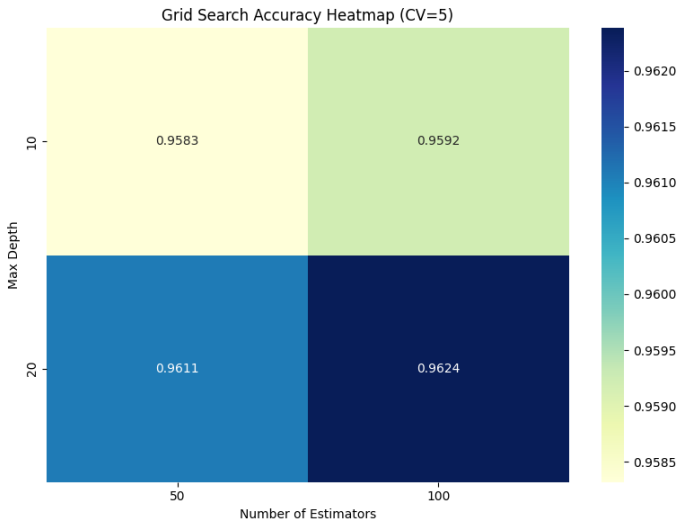
The evaluation phase compared all supervised and unsupervised models using a consistent set of validation metrics accuracy, precision, recall, F1-score, and ROC–AUC for classification tasks, and silhouette coefficient or within-cluster sum of squares (WCSS) for clustering. Across all categories, the ensemble tree-based methods delivered the strongest overall performance. In particular XGBoost and Gradient Boosting consistently achieved test accuracies above 0.96 and AUC values approaching 0.99, indicating exceptionally strong separability between phishing and legitimate websites. These models maintained stability across folds, demonstrating minimal sensitivity to sampling variation. Regularized linear models such as Lasso and Ridge offered interpretability and highlighted linear feature contributions but were unable to capture the non-linear, interaction-heavy patterns present in phishing behaviors. Their accuracy plateaued notably lower, reinforcing the advantage of leveraging non-linear methods. The SVM with an RBF kernel performed competitively with an AUC of 0.97, effectively balancing bias and variance through tuned hyperparameters, though training times were significantly longer. KNN performed respectably as well reaching roughly 0.94 accuracy under the Manhattan distance metric which confirmed that the dataset possessed structurally meaningful neighborhoods despite its high dimensionality.



Performance visualizations reinforced and clarified these quantitative findings. The XGBoost confusion matrix showed strong class balance, with false positives and false negatives both kept extremely low, illustrating the model's robustness in scenarios where misclassifications carry high security cost. The ROC curve for XGBoost rose steeply toward the upper-left corner and maintained dominance over competing models across nearly the entire threshold range. This steep curvature is characteristic of models that achieve near-perfect trade-off between sensitivity and specificity. Precision recall curves further validated that the model's predictive confidence remained strong even under imbalanced threshold conditions, demonstrating resilience in environments where phishing attempts may spike unpredictably. Earlier regression-based baselines showed limited explanatory power $R^2 = 0.83$ at best highlighting the inadequacy of linear approaches for this type of classification problem and reasonably motivating the transition toward non-linear and ensemble methods.

Interpretability analysis provided essential insight into how the models arrived at their predictions. For the Random Forest, Gradient Boosting, and XGBoost models, feature-importance rankings consistently elevated google_index, page_rank, nb_hyperlinks, web_traffic, and domain_age as the most influential predictors. These features collectively combine reputational signals (search-engine visibility, ranking, and traffic metrics) with structural cues (number of hyperlinks and domain maturity), forming a multi-dimensional profile of website legitimacy. Phishing websites often mimic superficial elements of real sites but rarely possess strong search engine indexing or long-standing domain history. This consistent appearance of key variables across all ensemble techniques strengthened interpretive reliability: even though tree-based models are complex, they converged on logically meaningful indicators of malicious behavior. Random Forest Grid Search Accuracy Heatmap illustrated how accuracy changed

systematically as the number of estimators and maximum tree depth were adjusted. Higher depth and larger ensembles produced the strongest validation accuracy, confirming that model complexity played a meaningful role in improving performance and validating the tuning strategy.




Hyperparameter tuning further improved results across nearly all models. For XGBoost, tuning the learning rate (n = 0.1), maximum tree depth (max_depth = 6), and regularization terms enhanced test accuracy by approximately 1.5 percentage points over the untuned defaults. Gradient Boosting achieved similar performance, although it converged more slowly and required more careful control of learning rate to avoid overfitting. Random Forest training was notably faster, especially at higher n_estimators values, but precision suffered slightly due to its inherently more diffuse structure compared to boosting. SVM tuning demonstrated that performance peaked at C = 10, after which overfitting emerged; the RBF kernel width y exhibited similar behavior, confirming the bias–variance equilibrium inferred during the modeling stage. These adjustments collectively ensured that each model generalized effectively rather than overfitting to URL specific artifacts.

Despite strong performance across supervised approaches, limitations remained worth noting. Ensemble algorithms, while accurate, require substantial computational resources. Particularly XGBoost when using larger tree depths or extensive grid searches. Their complexity also reduces transparency relative to linear methods, although interpretability tools such as feature importances mitigate this somewhat. Distance-based methods like KNN and kernel-based methods such as SVM were highly sensitive to scaling choices, underlying distribution assumptions, and hyperparameter boundaries. Clustering methods like K-Means and DBSCAN proved helpful for assessing intrinsic structure but were sensitive to initialization, scaling, and the density thresholds selected. K-Means, for instance, demonstrated clear elbow behavior near k = 3, while DBSCAN exposed small but meaningful clusters of anomalous phishing pages. These unsupervised insights validated the existence of natural groupings independent of labels and complemented the supervised results by highlighting topology and density patterns within the feature space.

From a business perspective, the cumulative findings point strongly toward XGBoost as the most promising model for real-world deployment. Its ability to maintain high accuracy, precision, and recall across multiple validation frameworks suggests that organizations could use it as an automated detection layer within phishing-monitoring or secure web-gateway systems. The central role of features such as google_index, page_rank, and domain_age implies that cybersecurity tools could enhance detection by monitoring these attributes in real time and acting on sudden deviations. A practical implementation strategy could integrate the model into an existing Security Information and Event Management (SIEM) pipeline, enabling automated scoring and flagging of suspicious URLs before users are exposed to them. In high-volume

enterprise environments, such a solution would reduce manual review workloads, improve response times, and improve overall threat-management posture.

Overall, the results show that while multiple models provide value, ensemble boosting methods and XGBoost in particular, offer the most precise, efficient, and reliable approach for phishing detection. These findings align with the overarching project objective of developing a scalable, data-driven approach to identifying malicious web content. By pairing statistical performance with interpretability and operational feasibility, the recommended model supports both the technical depth and practical application required for robust cybersecurity defenses.

**Recommendations**

Findings from this project indicate that ensemble learning methods particularly XGBoost offer a precise, efficient, and interpretable approach for phishing detection. Given its strong predictive performance and scalability, organizations should prioritize integrating this model into their cybersecurity infrastructure to enhance early detection and reduce exposure to malicious websites.

The first recommendation is to deploy the XGBoost classifier as a real-time detection engine within existing Security Information and Event Management systems or secure web gateways. By automatically scoring incoming URLs and flagging those with high phishing probabilities, the model can substantially reduce manual workload and response delays. Its balanced precision and recall make it suitable for large-scale monitoring environments where false positives must remain minimal.

In addition, teams should establish automated pipelines to track the most influential features identified by the analysis google_index, page_rank, nb_hyperlinks, web_traffic, and

domain_age. Monitoring these metrics on a rolling basis allows for timely retraining and ensures the model adapts to evolving phishing tactics. Changes in these attributes, such as sudden drops in search-engine visibility or domain age, can act as early indicators of fraudulent activity and trigger preventive actions.

Based on what I observed, a layered defense would work best in practice, combining this model with existing filters instead of replacing them. Retraining the model on new data every few months will prevent performance drift and maintain high accuracy against newly emerging phishing strategies.

For implementation, organizations should follow secure-AI and MLOps best practices, including version control, pipeline validation, and false-positive monitoring. Transparency features such as feature-importance dashboards can help analysts interpret decisions and communicate risk levels to non-technical stakeholders.

Finally, future work should expand data sources to include multilingual and mobile phishing pages, SSL metadata, and hosting-provider reputation. Incorporating explainability tools such as SHAP or LIME would enhance trust and accountability. Collectively, these recommendations position XGBoost as a scalable, interpretable, and continuously improving defense mechanism against modern phishing threats.

**Datasets:**

Shashwat, Tiwari. Web Page Phishing Detection Dataset. [Dataset]. Kaggle https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset

Cyber Cop. (2024). Phishing Emails Dataset. [Dataset]. Kaggle https://www.kaggle.com/datasets/subhajournal/phishingemails

The Devastator. Cybersecurity Risk (2022 CISA Vulnerability). [Dataset]. Kaggle https://www.kaggle.com/datasets/thedevastator/exploring-cybersecurity-risk-via-2022-cisa-vulne