International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Explainability Measurement of Machine Learning Model in Phishing Detection

Abdullah Fajar[1], Indra Budi[2], Setiadi Yazid*[3]

Submitted: 05/01/2025 **Revised:** 25/02/2025 **Accepted:** 08/03/2025

Abstract: Explainability in phishing detection models can enhance phishing assault mitigation by fostering confidence and elucidating the detection process. The essential requirements for facilitating human comprehension and assessment of the reasons a specific URL is deemed insecure for visitation. The aims of this study are to investigate some machine learning models in phishing detection which have abilities to fulfil the critical needs of explanation using explainability metric. This study applies a methodology starting with dataset collection of phishing and legitimate URL as the sources of various features. Then the models selected, which are often known have good quality in classification between phishing or legitimate label. The modeling results are processed using an explainer method to generate a comprehensive understanding of feature behaviors that influence model predictions. Instead of present accuracy metric results only, this study discusses how explainability metric shows how the features contribute to the model. The conclusion shows that some features have abilities to influence the model decision in general or specifically, then how the features contribute to the model in terms of stability and distribution behaviors. The study shows that some features that may be identified as key features of model behavior then can be applied practically to phishing detection systems such as firewall or SIEM (Security Information and Event Management).

Keywords: *Phishing Detection, Machine Learning, Explainability Metric, URL, Features*

1. Introduction

Detecting phishing attempts has become a vital task in cybersecurity, as these misleading tactics change and represent major hazards to individuals and companies. A practical alternative has emerged in the form of machine learning models to address this difficulty. Improving the precision and consistency of phishing detection, researchers are exploring different approaches [1].

One important part of this project is being able to explain the model's findings, which is especially important in areas where safety is paramount, like cybersecurity. Explainable AI has garnered significant attention as a solution to the "black box" issue prevalent in many machine learning models, which is the difficulty in understanding their decision-making process[2].

New studies have compared and contrasted white-box and black-box machine learning algorithms for phishing detection, highlighting their respective benefits and drawbacks[3]. In safety-critical domains like cybersecurity, explaining the model's predictions helps boost detection system trust and responsibility.[4]. Explainable artificial intelligence has attracted interest for addressing the "black box" nature of many machine learning models, in which the internal decision-making process is unclear. Generating explanations enhances the comprehension of the variables influencing the model's decisions, which is essential for validating its behavior, particularly in high-stakes applications such as phishing detection[5].

Previous studies have raised questions regarding the necessity of generating explanations in phishing detection. Charmet *et.al,* [6] in his work describe how to explain phishing attempts? This topic emphasizes the necessity for systems that can detect phishing attempts and explain attacker strategies. The key question is how to produce consistent and thorough natural language explanations for anti-phishing system judgments. [7]. These addresses Users disregarding warnings owing to insufficient information. Researchers are studying methods to provide insights into algorithm predictions to help humans determine why a URL is unsafe. [8]. This inquiry pertains to the overarching domain of Explainable AI (XAI) within cybersecurity. A critical inquiry pertains to the selection of the most salient features for phishing detection models while maintaining their interpretability. [9]. This entails reconciling the necessity for precision with the demand for elucidation.

In summary, AI-driven phishing detection systems have demonstrated remarkable efficacy in combating phishing efforts, employing sophisticated methods to identify and anticipate vulnerabilities. Integrating explainability elements into these systems improves their effectiveness by giving clients a better understanding of why phishing warnings are sent. This transparency enhances user comprehension and fosters trust in the detection method.

Also, study shows how important it is to give users explanations that are clear and to the point so that they can make smart choices about possible threats. Using user-centered design principles and

1 Information System Dept, Faculty of Industrial Engineering, Universitas Telkom - 40257, INDONESIA
ORCID ID : 0009-0004-7647-9166
2 Faculty of Computer Science, Universitas Indonesia-16424, INDONESIA
ORCID ID : 0000-0002-2107-6552
3 Faculty of Computer Science, Universitas Indonesia-16424, INDONESIA
ORCID ID : 0000-0002-5291-5144
** Corresponding Author Email: author@email.com*

Table 1 Dataset for Modelling

No.	Dataset	Sumber (www.kaggle.com)	Year	# Instance	#Features	Phis/Legit
1	ds_235795_54	/datasets/joebeachcapital/phiusiil-phishing-url	2012	235.795	54	43/57
2	ds_129K112	/datasets/michellevp/dataset-phishing-domain-detection-cybersecurity	2021	129.698	112	41/59
3	ds_100K20	datasets/danielfernandon/web-page-phishing-dataset	2020	100.000	20	36/64
4	ds_88K112	/datasets/ravirajkukade/phishingdomaindetection	2021	88.647	112	35/65
5	ds_11K89	/datasets/manishkc06/web-page-phishing-detection/data	2020	11.481	89	20/80
6	ds_11055	/datasets/akashkr/phishing-website-dataset	2017	11.055	32	44/56
7	ds_90K32	/datasets/rashazieni/zieni-dataset	2024	96.018	32	50/50
8	ds_10K50	/datasets/shashwatwork/phishing-dataset-for-machine-learning	2018	10.000	50	50/50
9	ds_10K18	/datasets/hasibur013/url-data-for-phishing-website-detection	2024	10.000	18	50/50
10	ds_600K11	/datasets/simaanjali/phising-detection-dataset/code	2024	662.591	11	15/85
11	ds_249750	/datasets/6tm2d6sz7p/1	2021	249750	41	51/49

making models that are strong and easy to understand is important for creating trustworthy anti-phishing solutions. This shows how important it is for phishing detection systems to not only be right, but also to give people a clear explanation of their choices. For users, this will help them understand, trust, and eventually stay safe online.

This study aims to investigate various machine learning models for phishing detection, which effectively address the essential requirements of explanation using explainability metrics. Attaining objectives. This study presents an approach that begins with the compilation of datasets comprising phishing and genuine URLs as sources for various attributes. The selected models are typically recognized for their efficacy in distinguishing between phishing and authentic labels. The modeling findings are further analyzed using an explanatory method to produce a thorough understanding of the feature behaviors influencing the predictions of the inference model. This study not only presents accuracy metric findings but also examines how the explainability metric illustrates the contribution of features to the model.

The remainder of this paper describes the results, providing further details about the methodology. It then goes on to describe and discuss the results, highlighting some findings and contributions. Lastly, the paper presents a conclusion about the achievement of the objectives and the contributions made. This part explains limitations and future work that may follow.

2. Methodology

The method that used in this research comprises of three parts, starting dataset collection and model preparation, followed out by machine learning modelling and the final step is explainability metric processing using explainer method. The result of explainer visualized and analyzed to answer the question and the objectives of this study.

2.1. Data Collection and Model Preparation

This study will use datasets from many sources, some of which have been used in related studies. The main source came from www.kaggle.com.

These datasets distribution are tested first to ascertain their normality. To determine the normality of the row and column distributions, the Shapiro-Wilk test is used with the null hypothesis that the data is normally distributed. The calculation results for the normality of the rows and columns are as follows

	Dataset
Rows W-Statistic	0.6940035820007324
Rows p-Value	0.0007381692412309349
H0: Rows W-Statistic>Rows p-Value	TRUE
	Rows is Normal Distributed

Columns W-Statistic	0.8618614673614502
Columns p-Value	0.05158619582653046
H0: Columns W-Statistic> Columns p-Value	TRUE
	Column is Normal Distributed

The dataset has undergone preprocessing and is prepared for modeling since it is determined that both rows and columns are statistically regularly distributed. Outliers, skewness, or excessive kurtosis may be present in non-normal datasets, which must be addressed during preprocessing.

The model preparation in short describes as follows:

1. The first step is loading the dataset, which is the main source of data used in the research.
2. To enable independent processing of inputs and outputs, the characteristics (independent variables) and target labels (dependent variable) are then separated. If the target label is categorical, it is modified to guarantee that it complies with the modeling specifications. To standardize the categorization process, the target labels are then transferred to binary values, usually 0 and 1. To guarantee that the model can be trained and assessed efficiently, the dataset is then divided into training and testing subsets.
3. To avoid bias in the machine learning model and guarantee accurate and equitable predictions, any imbalance in the distribution of target labels is finally fixed.

Table 2 XGBoost Model Result

Dataset Name	Accuracy	Precision	Recall	False Positive Rate	ROC AUC	Runtime (seconds)
ds_100K20_.csv	89,68%	88,15%	91,60%	12,23%	96,32%	296,88
ds_10K18.csv	99,85%	100,00%	99,70%	0,00%	100,00%	94,15
ds_10K50_rev.csv	98,95%	98,63%	99,31%	1,42%	99,91%	78,46
ds_11055.csv	97,16%	96,85%	97,62%	3,32%	99,68%	1,94
ds_11055_rev.csv	97,44%	97,24%	97,78%	2,91%	99,67%	1,63
ds_11K89.csv	98,65%	98,66%	98,58%	1,28%	99,71%	2,50
ds_129K112.csv	97,45%	97,26%	97,65%	2,76%	99,68%	21,93
ds_235795_54_rev.csv	99,99%	99,99%	100,00%	0,01%	100,00%	28,83
ds_247950_rev.csv	90,99%	93,23%	88,37%	6,39%	96,91%	23,32
ds_600K11_rev.csv	82,55%	80,63%	85,73%	20,65%	90,80%	9,16
ds_88K112.csv	97,59%	97,36%	97,79%	2,59%	99,67%	4,23
ds_90K32.csv	99,99%	100,00%	99,99%	0,00%	100,00%	0,78

2.2. Machine Learning Modeling Process

As far as Affenzeller et al.[10], in the field of machine learning, black-box models are complex systems that have internal workings that are either obscured or difficult to comprehend or challenging-to-understand. According to Rudin [5], these models are frequently employed in high-risk decision-making across a range of industries, including criminal justice and healthcare. With ramifications for numerous domains and high-risk choices, the argument over the use of Black-box models with explanations versus models that are intrinsically interpretable is still ongoing.

Transparency and interpretability are hallmarks of the White-box paradigm in general. This model generates a model that can be thoroughly examined and whose structure is not concealed [10].
It is crucial to remember, though, that in some situations, White-box models can be less accurate at making predictions than Black-box models [11]. The particular problem, the requirement for interpretability, and the significance of prediction accuracy are generally the deciding factors when choosing between White-box and Black-box models.

several popular algorithms such as:

1. Random Forest: Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs for prediction. They are known for their high predictive accuracy and their ability to handle complex relationships in the data. [10] explain that Random Forest is often used in various applications and has shown good performance in benchmark problems.
2. XGBoost is a scalable ensemble technique that has been shown to be a dependable and effective machine learning algorithm [12]. XGBoost is a member of the Gradient Boosted Decision Trees (GBDT) family and has been extensively utilized in numerous machine learning research projects and real-world applications[12], [13].
3. CatBoost is a newer addition to the Gradient boosting technique family, similar to XGBoost, introduced in 2017 by Ostroumova et al., [14]. This algorithm for processing categorical features, as concluded by Hancock & Khoshgoftaar [15], allows CatBoost to handle categorical features more effectively.
4. EBM, Explainable Boosting Machine, is an interpretable

Table 3 Explainable Boosting Machine Model Result

Dataset Name	Accuracy	Precision	Recall	False Positive Rate	ROC AUC	Runtime (seconds)
ds_100K20_.csv	88,97%	87,60%	90,69%	12,75%	95,99%	256,98
ds_10K18.csv	99,95%	100,00%	99,90%	0,00%	100,00%	4,58
ds_10K50_rev.csv	98,30%	97,85%	98,81%	2,23%	99,81%	12,14
ds_11055.csv	95,78%	95,44%	96,35%	4,82%	99,40%	9,89
ds_11055_rev.csv	94,76%	94,14%	95,71%	6,23%	99,11%	12,01
ds_11K89.csv	98,00%	97,87%	98,04%	2,04%	99,55%	26,58
ds_129K112.csv	97,65%	97,73%	97,58%	2,28%	99,67%	3.581,70
ds_235795_54_rev.csv	99,99%	99,99%	99,99%	7,40%	100,00%	314,90
ds_247950_rev.csv	89,22%	91,35%	86,62%	8,18%	95,81%	2.342,93
ds_600K11_rev.csv	79,68%	77,44%	83,82%	24,48%	87,60%	10.804,34
ds_88K112.csv	97,20%	97,08%	97,27%	2,87%	99,56%	453,78
ds_90K32.csv	99,99%	100,00%	99,99%	0,00%	100,00%	44,08

Based on rational explanations above, this study employs Machine Learning method that is an improvement over

the Generalized Additive Model [16]. What distinguishes EBM is its focus on interpretability. The algorithm provides visualizations of these functions, allowing users to understand how each variable affects the predictions. Additionally, EBM can capture interactions between variables and offer tools such as variable importance estimates and local explanations to further clarify the decision-making process. This makes EBM a valuable algorithm in situations where accuracy and understanding of the model's reasoning are paramount.

In short, the machine learning algorithm above employed in this process as follows:

guarantee that it is suitable for modeling processes.

3. Immediately after the data has been prepared, it is divided into training and testing sets in order to make the process of training and evaluating the model more manageable. A technique known as the Synthetic Minority Over-sampling Technique (SMOTE) is utilized to rectify any imbalances that may exist within the dataset.
4. After that, the model is trained, and then it is applied to prediction and assessment to evaluate its performance. The results of the evaluation are shown with the use of a confusion matrix, which offers insights into the accuracy of the model as well as the faults that it contains.

2.3. Analysis of the Explainability Metric Utilizing SHAP

Table 4 Random Forest Model Result

Dataset Name	Accuracy	Precision	Recall	False Positive Rate	ROC AUC	Runtime (Second)
ds_100K20_.csv	89,71%	89,00%	90,54%	11,11%	96,11%	30
ds_10K18.csv	100,00%	100,00%	100,00%	0,00%	100,00%	0,3
ds_10K50_rev.csv	98,20%	98,22%	98,22%	1,82%	99,87%	0,684
ds_11055.csv	97,16%	96,70%	97,78%	3,49%	99,49%	1,2
ds_11055_rev.csv	97,52%	97,24%	97,93%	2,91%	99,47%	1,08
ds_11K89.csv	97,95%	97,86%	97,95%	2,04%	99,67%	1,5
ds_129K112.csv	99,16%	99,06%	99,25%	0,94%	99,89%	29
ds_235795_54_rev.csv	99,99%	99,99%	99,99%	0,01%	100,00%	67,62
ds_247950_rev.csv	96,73%	97,32%	96,10%	2,64%	99,28%	55,29
ds_600K11_rev.csv	85,75%	83,10%	89,79%	18,31%	93,56%	185,747
ds_88K112.csv	97,67%	97,22%	98,10%	2,75%	99,67%	15,83
ds_90K32.csv	99,98%	99,98%	99,98%	0,02%	100,00%	2,99

1. Beginning with the loading of the dataset, which supplies the raw data for analysis, is the first step in the procedure.
2. The subsequent stage is known as data preparation, and it entails cleaning and preparing the dataset in order to

In the final stage of the workflow, a SHAP analysis, which is an abbreviation for SHapley Additive exPlanations, is carried out. This stage is essential for evaluating the model and acquiring a comprehensive understanding of the contribution and impact that each individual characteristic has on the predictions that the model

Table 5 . CatBoost Model Result

Dataset Name	Accuracy	Precision	Recall	False Positive Rate	ROC AUC	Runtime (seconds)
ds_100K20_.csv	89,74%	88,52%	91,24%	11,76%	96,40%	86,60
ds_10K50_rev.csv	98,50%	98,14%	98,91%	1,92%	99,87%	6,37
ds_11055.csv	97,28%	97,00%	97,70%	3,16%	99,71%	6,09
ds_11055_rev.csv	97,48%	97,09%	98,01%	3,07%	99,67%	5,67
ds_129K112.csv	97,73%	97,58%	97,90%	2,44%	99,70%	36,89
ds_247950_rev.csv	91,96%	94,00%	89,61%	5,70%	97,37%	46,88
ds_600K11_rev.csv	83,31%	80,99%	87,12%	20,50%	91,51%	100,12
ds_88K112.csv	97,64%	97,47%	97,78%	2,49%	99,67%	23,49
ds_10K18.csv	99,35%	99,19%	99,49%	0,79%	99,92%	27,10
ds_11K89.csv	98,56%	98,57%	98,49%	1,36%	99,78%	38,07
ds_235795_54_rev.csv	99,99%	99,99%	99,99%	0,01%	100,00%	246,70
ds_90K32.csv	99,99%	100,00%	99,98%	0,00%	100,00%	60,22

generates. The study provides a clear and interpretable explanation of how different qualities influence the anticipated outcome, both positively and negatively, by giving a Shapley value to each feature. This allows for a better understanding of how the features influence the outcome. This guarantees that the judgments made by the model are open and easy to comprehend, which in turn helps to cultivate trust in the results it generates. This is especially important in applications where explainability is of utmost importance, such as the healthcare, financial, or legal domains. When this interpretability analysis is finished, the process is finished, and all of the necessary procedures, ranging from the compilation of the data to the evaluation of the model, have been carried out in a thorough manner.

This process outlines the sequential procedure for employing SHAP (SHapley Additive exPlanations) to analyze and interpret predictions made by a machine learning model. Here as the process:

1. A SHAP explainer is utilized to calculate Shapley values. The explainer serves as an intermediary between the trained model and the SHAP framework, facilitating the decomposition of predictions into feature-level contributions.
2. SHAP values are calculated for every feature within the dataset. The values quantify each feature's contribution to the model's predictions for specific data points.

$$\Phi i = \sum_{S \subseteq N\{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (1)$$

Φi= The SHAP value

N = The Set of all features in the model

S = A subset of features excluding i i.e $S \sqsubseteq N\{i\}$

$|S|$ = The Number of features

$|N|$ = The total number of features in dataset.

$f(S)$ = the model's prediction when only the features in subset S are considered

$f(S \cup \{i\})$ = the model's prediction when feature I added to Subset S

$\frac{|S|!(|N|-|S|-1)!}{|N|!}$ = The Shapley weight, which ensures fair distribution of credit across all subset.

SHAP values calculate the marginal contribution of a feature i by comparing the model's output with and without i, across all possible subsets of features S.

3. Computed SHAP values are examined to understand the relationships between features and the model's output. This aids in identifying the features that significantly impact predictions.
4. Mean Absolute SHAP Value: Mean absolute SHAP values are computed to assess feature importance. This offers a quantitative assessment of the average contribution of each feature to the model's predictions.

$$Feature\ Importance = \frac{1}{n}\sum_{i=1}^{N} \left| SHAP\_Value_{ij} \right| \quad (2)$$

5. A visualization is generated, typically as a bar chart or summary plot, to illustrate feature importance according to the mean absolute SHAP values. This visualization aids in the interpretation of results and enhances effective communication.

This process is essential for enhancing model transparency and fostering trust in its predictions, especially in contexts where comprehending feature contributions is crucial.

A visalization that results in SHAP explainer are two types that will be used to analyzed, there are:

1. Feature Importance Plot

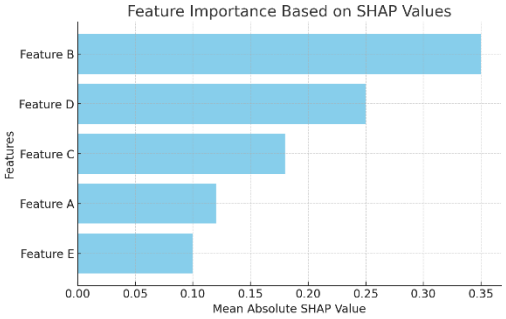

Fig 1 Feature Importance Plot

The figure above depicts the feature importance calculated from SHAP values. Each bar represents a feature, and its length indicates its average absolute SHAP value. This value measures the average magnitude of each feature's contribution to the model's predictions. The features are arranged in descending order of importance, with "Feature B" being the most influential and "Feature E" the least. This graphic helps comprehend which features significantly impact the model's predictions, aiding in interpretability and decision-making.

2. Causal Effect Plot using Swarm bee Plot

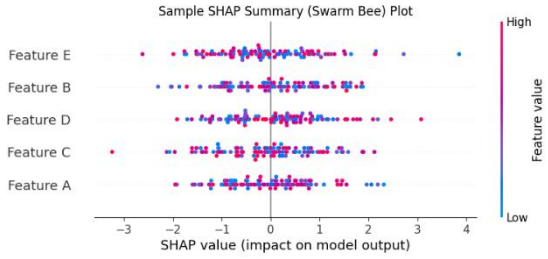

Fig 2 Swarm Bee Plot

A **swarm bee plot** is a summary visualization used to interpret the impact of individual features on the model's predictions. Each dot in the plot represents a single data point for a given feature.

- **Spread of Dots:** The spread of dots along the x-axis for a feature indicates the range of its impact. A wider spread implies the feature has a larger effect on the predictions for different data points.
- **Position of Dots:** The position of dots on the x-axis shows the direction of the impact:
  - Positive SHAP values indicate an increase in the prediction.
  - Negative SHAP values indicate a decrease in the prediction.
- **Color Patterns:** The color gradient of the dots shows how feature values correlate with the impact. For example, if red dots (high values) are mostly on the right (positive SHAP values), it suggests high feature values increase the prediction.

3. Results of Performance Metrics

These parts present the results of model and derivate from the

confusion matrix to explain each parameter as follow:

1. Precision: Precision measures how many of the predictions identified as phishing are phishing. It is essential in scenarios where reducing false alarms (false positives) is important.
2. Accuracy: Accuracy can be misleading in imbalanced datasets, where the majority class dominates. It should only be considered when the class distribution is balanced.
3. Recall: Recall measures how many actual phishing emails or websites are correctly detected. It ensures the model catches as many phishing attempts as possible, which is critical for security systems.
4. ROC-AUC: ROC-AUC evaluates the model's ability to distinguish between classes (phishing vs. non-phishing) across all thresholds. It is especially useful for comparing models.
5. False Positive Rate (FPR): The proportion of legitimate items that are incorrectly flagged as phishing. A low FPR is essential to avoid overwhelming users or systems with false alarms.

In general, the resume of each parameter result describes as follow:

1. Accuracy: Overall, all the models used generate average accuracy that is not statistically significant. Comparing the proportion of valid predictions.
2. Recall that Random Forest and XGBoost outperform EBM, but there is no meaningful difference between them and CatBoost.
3. Runtime: Random Forest outperformed EBM in terms of processing time. However, all models have different runtime variances, while the differences are not statistically significant.
4. Precision: Random Forest and XGBoost outperform the other algorithms in terms of precision.
5. False Positive: This measure should be considered in the context of phishing detection because it can have serious effects if it occurs. Between Random Forest and EBM, there is a considerable difference, with EBM having a greater false positive rate.
6. AUC_ROC: The goal of this statistic is to graphically represent classification performance. Overall, Random Forest performs better than the others.

According to the overall result, interesting finding identified as follow:

1. Large dataset has a negative influence on accuracy, precision, recall, and ROC AUC in all models, particularly XGBoost. This statement offers an initial conclusion to the research statement about the model's reliability against changes in huge datasets.
2. Feature Addition: Has a somewhat beneficial influence on accuracy, precision, and ROC AUC, with Random Forest and Explainable Boosting Machine benefiting the most.
3. Model Selection: XGBoost is suitable for quick runtime on large datasets but has worse precision and recall. Random Forest and Explainable Boosting Machine perform more consistently as the number of features changes. This statement can be used as a preliminary conclusion to address the model's quality in feature selection, which influences it.

4. Result of Explainability Metric

In this section, the explanation of the explanatory metric analysis consists of 2 parameters, namely the feature importance value and the causal effect, which explain the role of the feature in the model's class prediction, thus obtaining features that are consistently important in the model.

4.1. Feature Importance Measurement

In this section, the features that were utilized to model each dataset. The analysis focused on each model's primary feature features, including generalization capability and interpretability. The interpretation of each aspect is as follows:

1. The Main Feature Aspect illustrates the main characteristics in model prediction, as indicated by the distribution of differences with other features.
2. The Generalization Capability Aspect describes the model's ability to forecast new data patterns; the better the generalization capability, the more adaptable the model is to new information. This may be seen in the distribution of SHAP values for a model[17].
3. The Interpretability Aspect is defined as the ratio of features having SHAP values greater than zero to the total number of features in the dataset. The smaller the ratio, the greater the interpretability, because the model can be explained with fewer features, whereas the more features that explain it, the lower the interpretability[18].

The feature importance measurement result explains at **Table 6** Model's Feature Importance Explanation (see the appendix for plot location url)

4.2. Causal Effects Measurement

When analysing a SHAP swarm bee plot, several characteristics such as consistency, stability, actionability, and accuracy are essential for obtaining accurate and actionable insights from the model's predictions. Here's an overview of these characteristics and why they're important:

1. Consistency: Consistency ensures that the SHAP values align with the model's logic, meaning that when the importance of a feature increases in the model, the corresponding SHAP values also reflect that increase. Ensure that features with larger SHAP value distributions (longer x-axis ranges) correspond to their overall relevance in the model[19].
2. Stability: Stability ensures that the SHAP values remain consistent across different model runs or similar datasets, especially when data is resampled or slightly altered. Ensure that identical SHAP plots are achieved when testing with different data subsets or model versions[20].

Table 6 Model's Feature Importance Explanation (see the appendix for plot location url)

Dataset Name	Explanation
ds_100K20_.csv	The Explainable Boosting Machine (EBM) model highlights key findings, with URL, LineOfCode, and FILENAME identified as the top features influencing predictions. These features are easily interpretable and critical to the model's outcomes. The model demonstrates strong generalizability with universally applicable features, though low-contributing ones should be reviewed to avoid overfitting to specific datasets. EBM's findings are intuitive, with plots ranking features by importance, measuring their contributions, and presenting the results clearly. This simplicity enhances transparency and builds confidence in the model.
ds_10K50_rev.csv	Among critical characteristics, `PctExtNullSelfRedirectHyperlinksRT`, `PctExtHyperlinks`, and `FrequentDomainNameMismatch` have the highest SHAP values, suggesting their considerable contributions to model predictions. Low contributions to generalization capabilities have little effect on predictions and may increase model complexity. Keeping such traits may hinder the model's generalization to fresh data. Features such as `PctExtHyperlinks` (percentage of external hyperlinks) and `FrequentDomainNameMismatch` (frequent domain name mismatches) are easily interpretable as phishing indications. User trust in the model and its predictions increases due to their logical relationship to phishing dangers.
ds_11055.csv	The dominant features across models are URL_of_anchor, SSLfinal_State, and web_traffic, with additional contributions from Prefix_suffix and Having_subdomain. High-impact features like URL_of_anchor, SSLfinal_State, and web_traffic effectively identify phishing attempts, while low-contribution features such as poUpWindow and HTTPS_token reflect less common behaviors. Models clearly explain phishing mechanisms using these high-contribution features, though context-specific traits introduce variability, making interpretability more complex.
ds_11055_rev.csv	Key features like URL_of_Anchor and SSLfinal_State significantly enhance phishing detection, while low-contribution features such as age_of_domain and Google_Index risk increasing complexity without value, potentially reducing generalization as phishing tactics evolve. Intuitive features like SSLfinal_State and URL_of_Anchor are easily understood by non-technical users, improving trust and usability.
ds_129K112.csv	Except for XGBoost, `time_domain_activation` dominates important attributes across multiple models. In terms of generalization, `time_domain_activation` can detect phishing by recognizing new domains. However, characteristics like `qty_space_file` only impact certain phishing instances and have little impact on the model. The interpretability of phishing models is enhanced by high-contribution characteristics including `time_domain_activation`, `directory_length`, `URL_length`, and `qty_dot_domain`. Although XGBoost differs, its main feature is `qty_dot_directory`. Variable feature relevance across models represents varied phishing activity patterns, increasing interpretability difficulty.
ds_247950_rev.csv	Url_length dominates critical features across models, with average_subdomain_length and domain_length being important, except for the Random Forest model. Url_length is a key phishing signal, displaying excellent generalizability. XGBoost contributes to less features than EBM, which contributes evenly. Finally, url_length and number_of_subdomain help most models describe phishing. In certain cases, other, less consistent indicators suggest unique phishing tendencies, complicating interpretability.
ds_600K11_rev.csv	The SHAP summary plots highlight NumDots, URLLength, and PathLength as the most dominant features across models, demonstrating strong contributions to phishing detection and excellent generalization capability. These features capture universal phishing patterns, while lower-impact features like HttpsInHostname or AtSymbol are context-specific and may add unnecessary complexity. The high-impact features are intuitive and easily interpretable, providing clear insights into phishing mechanisms. Simplifying the models by focusing on these dominant features can enhance both performance and usability.
ds_88K112.csv	High SHAP values for crucial features like `time_domain_activation`, `qty_dot_domain`, and `directory_length` greatly impact model predictions, highlighting their significance in detecting phishing efforts. The model's capacity to generalize to new datasets is shown by features such as `time_domain_activation`, which often indicates domain reliability. Although `directory_length` and `length_url` are linked to URL length, they may create dependencies that hinder the model's performance on datasets with different distributions. Features such as `time_domain_activation` and `qty_dot_domain` are easily interpreted by non-technical users, indicating phishing or unusual activity. However, features such as `qty_hyphen_file` and `qty_slash_url` may need further explanation to determine their significance in the model. These factors emphasize balanced feature selection and model insight communication.
ds_10K18.csv	High-SHAP-value features like `URL_Length`, `URL_Depth`, and `Prefix/Suffix` dominate model predictions, except in the EBM model, where `domain` is most influential. All models have high generalization potential and can be simplified by deleting low-contribution characteristics to enhance efficiency without compromising accuracy. Phishing mechanisms are clearly explained by models utilizing high-contribution characteristics such as `URL_Length`, `URL_Depth`, and `Prefix/Suffix`, except for EBM, where `domain` is the dominant explanatory feature. These findings show the models' phishing detection and transparency/usability strengths.
ds_11K89.csv	For key features, google_index, page_rank, and nb_www have the highest SHAP values, making them significant contributors to predictions. Regarding generalization, these features reflect common phishing patterns, such as Google indexing and page ranking, enhancing model applicability to new datasets. In terms of interpretability, features like google_index and page_rank are intuitive and easily explainable, while features like phish_hints and nb_www may require additional clarification to ensure user understanding.
ds_235795_54_rev.csv	The Explainable Boosting Machine (EBM) model shows substantial feature dominance, generalization, and interpretability findings. For key characteristics, the model highlights `URL`, `LineOfCode`, and `FILENAME` as the top contributors, making them easy to comprehend as primary predictors. The EBM model has great generalizability, with important traits that are relevant across datasets. Features with smaller contributions should be carefully reviewed to ensure they are not too dataset-specific. Finally, EBM plots are easy to understand and use because to their obvious feature ranks, measurable contributions, and straightforward representations.
ds_90K32.csv	For **key features**, DNSRecordType, Domain, and NumericSequence have the highest average SHAP values, making them the most influential in model predictions. Regarding **generalization**, these features effectively capture common phishing patterns, enhancing the model's ability to perform well across datasets. In terms of **interpretability**, features like DNSRecordType and Domain are intuitive and easy to explain to non-technical users, such as unusual DNS types or suspicious domains serving as clear phishing indicators.

consistently identified as significant contributors. The discussion

Table 7 Feature 's Causal Effect from All Models (see the appendix for plot location url)

Dataset Name	Explanation
ds_100K20_.csv	Features like url_length, n_slash, and n_dots consistently show high SHAP values, aligning with domain intuition that phishing URLs often have complex structures. While features like n_hyphens and n_redirection exhibit variability, they still provide actionable insights, such as identifying excessive redirections or suspicious patterns. These features offer clear rules for detecting phishing attempts, enhancing both model consistency and practical application.
ds_10K50_rev.csv	FrequentDomainNameMismatch, PctExtNullSelfRedirectHyperlinksRT, and InsecureForms display similar SHAP patterns, enabling reliable predictions across the dataset. These traits have stable SHAP distributions, while others with symmetrical distributions around zero indicate lesser, sample-dependent contributions. Key features like `PctExtHyperlinks` and `InsecureForms` help detect phishing, as phishing URLs commonly contain suspicious external hyperlinks and unsecured forms. Take advantage of `InsecureForms` to identify URLs with high external hyperlinks or unsafe forms, and `NumDash` to prioritize URLs with many dashes for examination.
ds_11055.csv	Features like `SSLfinal_State`, `URL_of_Anchor`, and `web_traffic` show consistent SHAP patterns, with high values predicting phishing danger and low ones reducing it. Features with broad but targeted SHAP value distributions contribute steadily, while `popUpWindow` and `Statistical_report` have lesser, sample-dependent impacts. SSL final state is important for indicating validity, while URL_of_Anchor and web_traffic reveal user interactions with dubious URLs. URLs lacking SSL certificates and minimal web traffic, which are often linked to phishing, should be investigated.
ds_11055_rev.csv	Features such as `URL_of_Anchor`, `SSLfinal_State`, and `Prefix_Suffix` consistently contribute to phishing detection using SHAP patterns. The characteristics `URL_of_Anchor` and `SSLfinal_State` exhibit steady SHAP distributions, indicating sustained impact on predictions, while `Shortining_Service` and `Redirect` have lesser, sample-dependent impacts. Key aspects like `URL_of_Anchor` and `SSLfinal_State` are indicative of phishing, as suspicious anchors and incorrect SSL certificates are common signs. Use `SSLfinal_State` to identify URLs without valid SSL certificates, and `web_traffic` and `age_of_domain` to prioritize examination of low-traffic or new domains.
ds_129K112.csv	Features like `time_domain_activation`, `directory_length`, and `length_url` follow typical SHAP patterns, with high values boosting phishing risk estimates and low values decreasing them. `Time_domain_activation` shows stable SHAP value distributions, whereas `directory_length` and `qty_dot_domain` contribute consistently across datasets. Key characteristics like `time_domain_activation` are important, as newly formed domains are typically linked to phishing. Similarly, `length_url` and `directory_length` support the idea that complicated and lengthy URLs are more likely to be phishing. Actionable insights include prioritizing new domains using `time_domain_activation` and establishing URL length and directory structure thresholds to detect suspicious patterns.
ds_247950_rev.csv	Key features such as `url_length`, `domain_length`, and `number_of_dots_in_url` exhibit consistent trends, with high values increasing phishing risk predictions and low values decreasing them. Features like `url_length`, `average_subdomain_length`, and `domain_length` are stable across SHAP value distributions, whereas `entropy_of_url` and `number_of_special_characters_in_domain` have tiny, context-specific contributions. Longer URLs or those with numerous dots are suspicious, therefore these aspects improve phishing detection. Features such as `number_of_subdomains`, `url_length`, and `entropy_of_domain` can help mitigate risk by limiting URL length or subdomain counts, and identifying patterns with special characters in domains.
ds_600K11_rev.csv	SHAP distributions are evident in features such as `NumDots`, `UrlLength`, and `PathLength`, with high values indicating higher phishing risk estimates and low. These features are reliable in model predictions, with steady contributions across datasets, but `HttpsInHostname` and `IpAddress` have inconsistent influences based on sample. Phishing is commonly associated with excessive dots or long URLs, therefore key features like `NumDots` and `UrlLength` correspond with domain understanding. Actionable insights from `NumDots`, `PathLength`, and `UrlLength` can aid in phishing mitigation by identifying suspicious patterns through URL complexity and path depth thresholds.
ds_88K112.csv	Features such as `time_domain_activation`, `length_url`, and `directory_length` have consistent SHAP distributions, indicating their dependable phishing detection contributions. These broad but concentrated SHAP value distributions indicate strong stability and significant impact across phishing scenarios. Features with near-symmetrical zero distributions are less stable and influential. Features such as `time_domain_activation` and `length_url` are crucial, as newly created domains or lengthy URLs are typically suspect. To gain actionable information, use `time_domain_activation` to prioritize new domains and `length_url` to set suspicious URL length thresholds.
ds_10K18.csv	Features such as `URL_Length`, `iFrame`, and `Web_Traffic` follow consistent SHAP patterns, whereas `Domain_Age` and `Domain_End` support the idea that older domains are more reliable. `URL_Length`, `URL_Depth`, and `Web_Traffic` exhibit consistent SHAP distributions, contributing to model predictions, while `Have_At` and `Right_Click` have lower, sample-dependent impacts. Key features such as `URL_Length` and `Web_Traffic` are crucial for phishing detection, as long URLs or low web traffic frequently indicate suspicious behavior. Actionable insights include URL length thresholds and phishing risk analysis of low-traffic URLs.
ds_11K89.csv	Features such as `page_rank`, `google_index`, and `nb_www` have consistent SHAP patterns, indicating phishing detection reliability. Page_rank, google_index, and web_traffic exhibit steady and significant SHAP distributions, suggesting their importance in phishing scenarios. Domain_in_brand and longest_word_path have more scattered contributions, indicating sample-specific relevance. Relevant attributes include `page_rank` and `google_index`, as phishing URLs sometimes have low rankings or are not indexed by Google. Features such as `nb_hyperlinks` and `web_traffic` support the idea that URLs with high external links and low traffic are suspect. Using `google_index` to prioritize non-indexed URLs, `page_rank` to identify low-ranking sites, and `nb_hyperlinks` to define danger indicators for external link counts are actionable insights.
ds_235795_54_rev.csv	The constant SHAP distributions of LineOfCode and HasCopyrightInfo indicate their significant and dependable contributions to model predictions. URL, NoOfExternalRef, and FILENAME also show trends, with high values affecting predictions. URL and NoOfExternalRef have stable SHAP value distributions, but NoOfCSS and IsHTTPS have less consistent contributions across samples. URL and URLLength are linked to phishing since longer URLs indicate harmful action. URL and NoOfExternalRef insights can help set detection criteria like limiting URL or web document external references.
ds_90K32.csv	Features such as `DNSRecordType`, `Domain`, and `NumericSequence` consistently contribute to phishing detection through SHAP patterns. Features with large but focused SHAP value distributions are stable and significant, while features like `ConsonantSequence` and `VowelRatio` have scattered contributions, indicating lower and sample-specific relevance. Key elements like `DNSRecordType` and `DomainLength` can help identify phishing, as uncommon DNS types or lengthy domain names generally indicate phishing tendencies. Using `DNSRecordType` to identify suspicious DNS types and `DomainLength` and `SubdomainNumber` to create thresholds for domain length and subdomains might help identify phishing risks.

3. Actionability: Actionability ensures that the SHAP values provide insights that can be translated into meaningful actions or decisions. Concentrate on the most important attributes (those at the top of the y-axis) and examine how the insights can help drive practical decisions[21].
4. Accuracy Explanation: Accuracy means that the SHAP values correctly capture the contribution of each feature to the prediction, without over- or under-representing their impact. Cross-validate to ensure that the distribution and positioning of SHAP values accurately reflect the model's logic and feature relationships[22].

The result of Causal effect measurement explains at **Table 7** Feature 's Causal Effect from All Models (see the appendix for plot location url)

5. Discussion

Phishing detection relies on identifying key features that consistently contribute to accurate classification of phishing attempts. Understanding the significance and behavior of these features is essential for building effective, interpretable, and generalizable models. Analyzing the dominant features, their ability to generalize, and their impact on model interpretability provides actionable insights for model optimization. To enhance he interpretability, reliability, and operational value of such odels, it is critical to analyze key aspects of feature behavior and their contributions. This analysis, often supported by SHAP (SHapley Additive exPlanations), provides insights into the significance, stability, and actionability of individual features.

5.1. Feature Importance Analysis

The following sections detail the key aspects and findings of phishing detection models. Specifically, it highlights the importance of features such as url_length, n_slash, n_dots, SSLfinal_State, and URL_of_anchor, which have been also addresses the generalization capability of dominant features, contrasting them with minor, context-dependent features that show limited applicability across datasets[23].

Finally, the interpretability of models relying on these major traits is explored, emphasizing the importance of simplifying the model by removing low-contribution features without compromising its performance[24]. This analysis aims to provide a foundation for refining phishing detection models and ensuring their adaptability and reliability in diverse scenarios.

Key Aspects and Findings Dominant Features:

1. Various models consistently show that url_length, n_slash, n_dots, SSLfinal_State, and URL_of_anchor are the most important features. These variables have a significant and consistent impact on phishing detection, making them essential to the model's decision-making process.
2. Generalization Capability:
   Because of their universal relevance in phishing identification, dominant features such as url_length and SSLfinal_State have significant cross-dataset generalization. Minor characteristics, on the other hand, are more context-dependent and have less influence when applied to new or previously unexplored data sets.
3. Interpretability:
   Models based on these major traits have excellent interpretability since their impact on predictions is evident and consistent. Features with low contribution values should be considered for removal to simplify the model without compromising performance.

Phishing detection model optimization and deployment affect performance, usability, and adaptability[25]. Keeping only the most important features streamlines the model, lowering computational complexity and keeping accuracy[26]. For implementation in dynamic situations, the model must generalize well on varied datasets and unknown scenarios. Data on regional and contextual phishing tendencies must be validated.

The model's predictions must be trusted by stakeholders like security teams and end-users; therefore interpretability is crucial[27]. Transparent models that highlight feature contributions, like SHAP explanations, build confidence and deliver actionable insights. For real-time phishing detection, a model tuned for low latency and scalability must analyze huge amounts of data efficiently[4]. Combining machine learning predictions with rule-based systems enhances robustness by using insights like url_length criteria and SSL certificate legitimacy to enforce security standards.

Phishing methods change quickly, thus continuous development is essential. Retraining the model with new data guarantees it can handle new threats[28]. Monitoring precision, recall, and false positive rates helps maintain operating standards and identify improvements. User and administrator feedback can help reduce false positives and improve the model. Regulatory norms include data privacy regulations and ethical principles ensure the implementation meets legal and organizational requirements[29]. Customizing deployment tactics for use cases like high recall in high-risk industries improves the model's impact and applicability. These factors ensure the phishing detection model is effective, dependable, and adaptive to real-world situations[30].

5.2. Causal Effect Analysis

The following sections outline the primary findings from the evaluation of key features in phishing detection models. These findings cover aspects such as feature consistency, stability, explanation accuracy, and actionability, as well as the potential negative effects of less significant features[31]. Additionally, the role of high-impact features in enhancing detection is highlighted, offering actionable insights for optimizing and deploying phishing detection systems[32]. The goal is to refine the model's design while ensuring it remains effective, interpretable, and scalable across diverse scenarios.

1. Consistency: The model depends on reliable and predictable features like url_length, n_slash, and SSLfinal_State to detect phishing. Consistency emphasizes their model strength and importance.
2. Stability, Variability: URL_length and SSLfinal_State are reliable phishing indicators since they are stable across datasets and contexts. Contextual factors like poUpWindow vary more, suggesting they may be less generalizable across datasets. Features like HTTPS_token and Google_Index, which contribute little to the model's predictions, may reduce its ability to generalize effectively. Their inclusion could introduce noise, potentially leading to overfitting or less reliable predictions.
3. Accuracy of explanation: The model is more credible when features like url_length and number_of_dots_in_url match domain intuition. Phishing URLs are long and have many dots, which have

high SHAP values.

4. Actionability: Rule-based interventions like limiting maximum URL lengths or identifying URLs with strange slash patterns can use url_length and n_slash. SSLfinal_State validates SSL certificates in URLs, a frequent phishing signal, to deliver actionable insights..

Several realistic implementation options can be used to turn the findings into phishing detection system stages. Leveraging Feature For accurate predictions, consistency is essential. Since they consistently help detect phishing, url_length, n_slash, and SSLfinal_State should be prioritized during model training and evaluation. Adding feature-specific limits like a maximum URL length can improve detection. These features should also govern ensemble model feature selection, and weight them uniformly.

Validating stable features like url_length and SSLfinal_State across varied datasets helps maintain stability and manage variability. To promote generalization, model training should omit or downweight context-dependent variables like poUpWindow, which contribute variablely. New datasets can be checked before deployment to detect features with inconsistent affects and alter criteria dynamically. Filter away low-impact features like HTTPS_token and Google_Index to avoid overfitting and enhance model performance.

Another important stage is improving explanation accuracy. Show that features match domain intuition using explainability methods like SHAP values. URL_length and number_of_dots_in_url should show phishing trends. Model predictions can be explained clearly with SHAP-based visuals in monitoring dashboards. Security teams should also learn to interpret these justifications to make educated flagged case choices.

Actionability can be achieved with rule-based and machine learning hybrid systems. The model detects more nuanced than rule-based limits like url_length and n_slash. SSLfinal_State automates SSL validation, distinguishing real from phishing URLs. Security operations centers (SOCs) can use these hybrid systems for real-time alerts, and actionable insights can help design browser extensions that block suspicious URLs.

6. Conclusion

Phishing detection models rely heavily on identifying and prioritizing key features that consistently contribute to accurate predictions. Features like url_length, n_slash, and SSLfinal_State have been shown to provide significant and stable contributions, making them essential components in detecting phishing attempts across various contexts. These features not only enhance the model's interpretability but also offer actionable insights for rule-based interventions, such as setting thresholds for URL length or validating SSL certificates. By leveraging explainability tools like SHAP, the models ensure alignment with domain intuition, reinforcing trust in their predictions and enabling stakeholders to make informed decisions.

To optimize and deploy phishing detection systems effectively, it is crucial to focus on maintaining generalizability across diverse datasets while managing the variability of context-dependent features[33]. Continuous monitoring and retraining help adapt to evolving phishing strategies, ensuring the model remains relevant and effective. Combining machine learning predictions with rule-based systems enhances robustness, allowing for real-time detection and operational scalability. These measures collectively ensure that phishing detection models are accurate, interpretable, and adaptable, meeting the demands of real-world applications.

To advance phishing detection systems, several future directions and improvements should be considered. Enhancing feature engineering by identifying new features derived from evolving phishing tactics, such as advanced domain reputation metrics or contextual signals like time-based activities, can significantly improve adaptability. Additionally, integrating multi-modal data sources, including email content, URL structures, and network logs, can create a more comprehensive detection system. Multi-modal models capable of analyzing structured and unstructured data simultaneously will enhance accuracy in complex scenarios.

Expanding the use of advanced explainability tools, such as real-time visualization enhancements to frameworks like SHAP, will provide immediate and actionable insights. Efforts to reduce false positives and negatives should include cost-sensitive learning techniques and adaptive thresholding tailored to specific operational contexts, balancing precision and recall effectively..

Appendix

All dataset and plots files located at github repository, the url is https://github.com/abdfajar/Hasil-Pemodelan

Acknowledgements

This research was partially supported by Universitas Indonesia and Universitas Telkom. We thank our colleagues from Waditra Reka Cipta who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper. We thank Falahah from Universitas Telkom for comments that greatly improved the manuscript.

Author contributions

Indra Budi: Conceptualization, Methodology, Software, Field study **Setiadi Yazid:** Conceptualization, Methodology, Field study

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] M. Das, S. Saraswathi, R. Panda, A. Mishra, and ..., "Exquisite analysis of popular machine learning–based phishing detection techniques for cyber systems," *Journal of Applied …*, no. Query date: 2023-03-02 08:19:27, 2021, doi: 10.1080/19361610.2020.1816440.

[2] W. Saeed and C. Omlin, "Explainable AI (XAI): A Systematic Meta-Survey of Current Challenges and Future Opportunities," *Knowl. Based Syst.*, vol. 263, p. 110273, 2021, doi: 10.1016/j.knosys.2023.110273.

[3] A. Nadeem, D. Vos, C. Cao, L. Pajola, and ..., "Sok: Explainable machine learning for computer security applications," *2023 IEEE 8th …*, 2023, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10190524/

[4] A. Warnecke, D. J. Arp, C. Wressnegger, and K. Rieck, "Don't Paint It Black: White-Box Explanations for Deep Learning in Computer Security.," *Cornell University*, Jun. 2019, [Online]. Available: https://arxiv.org/abs/1906.02108v1

[5] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable Machine Learning: Fundamental

Principles and 10 Grand Challenges," *Cornell University*. Jan. 2021. doi: 10.48550/arXiv.2103.

[6] F. Charmet, T. Morikawa, A. Tanaka, and T. Takahashi, "VORTEX : Visual phishing detectiOns aRe Through EXplanations," *ACM Trans. Internet Technol.*, vol. 24, no. 2, pp. 1–24, May 2024, doi: 10.1145/3654665.

[7] G. Ramesh, "Identification of phishing webpages and its target domains by analyzing the feign relationship," *Journal of Information Security and Applications*, vol. 35, no. Query date: 2024-02-17 23:38:53, pp. 75–84, 2017, doi: 10.1016/j.jisa.2017.06.001.

[8] S. Mittal, "Explaining URL Phishing Detection by Glass Box Models," *Proceedings of the 2023 Fifteenth International Conference on Contemporary Computing*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:263147290

[9] M. C. Calzarossa, P. Giudici, and R. Zieni, "Explainable Machine Learning for Bag of Words-Based Phishing Detection," *... on Explainable Artificial Intelligence*, 2023, doi: 10.1007/978-3-031-44064-9_28.

[10] M. Affenzeller *et al.*, "White Box vs. Black Box Modeling: On the Performance of Deep Learning, Random Forests, and Symbolic Regression in Solving Regression Problems," in *International Conference/Workshop on Computer Aided Systems Theory*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:215791513

[11] O. Loyola-González, "Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View," *IEEE Access*, vol. 7, pp. 154096–154113, 2019.

[12] C. Bentéjac, A. Csörgo, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2019.

[13] E. M. Kenny, C. Ford, M. Quinn, and M. T. Keane, "Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies," *Artificial Intelligence*. Elsevier, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370221000102

[14] L. Ostroumova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Neural Information Processing Systems*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:5044218

[15] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *Journal of Big Data*, vol. 7, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:226254770

[16] A. E. Maxwell, M. Sharma, and K. A. Donaldson, "Explainable Boosting Machines for Slope Failure Spatial Predictive Modeling," *Remote Sensing*, vol. 13, no. 24, p. 4991, Dec. 2021.

[17] R. Massafra *et al.*, "Analyzing breast cancer invasive disease event classification through explainable artificial intelligence," *Frontiers Media*, vol. 10, Feb. 2023, doi: 10.3389/fmed.2023.1116354.

[18] H. Kaneko, "Interpretation of Machine Learning Models for Data Sets with Many Features Using Feature Importance," *American Chemical Society*, vol. 8, no. 25, pp. 23218–23225, Jun. 2023, doi: 10.1021/acsomega.3c03722.

[19] K. Aas, M. Jullum, and A. Løland, "Explaining individual predictions when features are dependent: More accurate approximations to Shapley values," *Elsevier BV*, vol. 298, pp. 103502–103502, Mar. 2021, doi: 10.1016/j.artint.2021.103502.

[20] M. Philipp, T. Rusch, K. Hornik, and C. Strobl, "Measuring the Stability of Results From Supervised Statistical Learning," *Taylor & Francis*, vol. 27, no. 4, pp. 685–700, May 2018, doi: 10.1080/10618600.2018.1473779.

[21] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Definitions, methods, and applications in interpretable machine learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 44, pp. 22071–22080, 2019, doi: 10.1073/pnas.1900654116.

[22] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Cornell University*. Jan. 2017. doi: 10.48550/arXiv.1705.

[23] S. R. Sharma, "A Feature Selection Comparative Study for Web Phishing Datasets," *Proceedings of CONECCT 2020 - 6th IEEE International Conference on Electronics, Computing and Communication Technologies*, no. Query date: 2024-02-17 23:37:56, 2020, doi: 10.1109/CONECCT50063.2020.9198349.

[24] I. Covert, S. Lundberg, and S.-I. Lee, "Understanding Global Feature Contributions With Additive Importance Measures," *Cornell University*. Jan. 2020. doi: 10.48550/arXiv.2004.

[25] H. Faris and S. Yazid, "Phishing Web Page Detection Methods: URL and HTML Features Detection," *... IEEE International Conference on Internet of ...*, no. Query date: 2023-03-02 08:19:27, 2021, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9359694/

[26] L. F. Gutiérrez and A. S. Namin, "Generating Interpretable Features for Context-Aware Document Clustering: A Cybersecurity Case Study," *... Conference on Big Data (Big Data)*, 2022, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10021049/

[27] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining Explanations: An Overview of Interpretability of Machine Learning," *Cornell University*. Jan. 2018. doi: 10.48550/arxiv.1806.00069.

[28] P. Maneriker, J. Stokes, E. Lazo, and ..., "URLTran: Improving phishing URL detection using transformers," *MILCOM 2021-2021 ...*, no. Query date: 2023-03-02 08:19:27, 2021, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9653028/

[29] S. Dalvi, G. Gressel, and K. Achuthan, "Tuning the false positive rate/false negative rate with phishing detection models," *Int. J. Eng. Adv. Technol*, no. Query date: 2023-03-02 08:19:27, 2019, [

[30] F. S. Bidabadi and S. Wang, "A new weighted ensemble model for phishing detection based on feature selection," *Cornell University*. Jan. 2022. doi: 10.48550/arxiv.2212.11125.

[31] H. Yuan, "Detecting Phishing Websites and Targets Based on URLs and Webpage Links," *Proceedings - International Conference on Pattern Recognition*, vol. 2018, no. Query date:

2024-02-17 23:38:53, pp. 3669–3674, 2018, doi: 10.1109/ICPR.2018.8546262.

[32] H. Zuhair, "New hybrid features for phish website prediction," *International Journal of Advances in Soft Computing and its Applications*, vol. 8, no. 1, pp. 28–43, 2016.

[33] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techni ques," *Telecommun Syst*, vol. 76, no. 1, pp. 139–154, Oct. 2020, doi: 10.1007/s11235-020-00733-2.