JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Investigating a SMOTE-Tomek Boosted Stacked Learning Scheme for Phishing Website Detection: A Pilot Study

Eferhire Valentine Ugbotu [1,*], Frances Uchechukwu Emordi [2,*], Emeke Ugboh [3], Kizito Eluemunor Anazia [4], Christopher Chukwufunaya Odiakaose [2], Paul Avweresuo Onoma [5], Rebecca Okeoghene Idama [4], Arnold Adimabua Ojugo [5,*], Victor Ochuko Geteloma [5], Amanda Enaodona Oweimieotu [6], Tabitha Chukwudi Aghaunor [7], Amaka Patience Binitie [3], Anne Odoh [8], Chris Chukwudi Onochie [3], Peace Oguguo Ezzeh [3], Andrew Okonji Eboka [3], Joy Agboi [9], and Patrick Ogholuwarami Ejeh [2]

[1] Faculty of Science, Engineering and Environment, University of Salford, Manchester M54WT, United Kingdom; e-mail : eferhire.ugbotu@gmail.com
[2] Faculty of Computing, Dennis Osadebay University, Asaba, Delta State 320212, Nigeria; e-mail : emordi.frances@dou.edu.ng, osegalaxy@gmail.com, patrick.ejeh@dou.edu.ng
[3] School of Science Education, Federal College of Education (Technical), Asaba, Delta State 320212, Nigeria; e-mail : andrew.eboka@fcetasaba.edu.nh; ugboh1972@fcetasaba.edu.ng, amaka.binitie@fcetasaba.edu.ng, xtoline2@gmail.com, peace.ezzeh@fcetasaba.edu.ng, ebokaandrew@gmail.com
[4] Faculty of Computing, Southern Delta University, Ozoro, Delta State 334111, Nigeria; e-mail : anaziake@dsust.edu.ng, idamaro@dsust.edu.ng
[5] College of Computing, Federal University of Petroleum Resources, Effurun, Delta State 330102, Nigeria; e-mail : kenbridge14@gmail.com, ojugo.arnold@fupre.edu.ng; geteloma.victor@fupre.edu.ng
[6] Faculty of Science, Edwin Clark University, Kiagbodo, Delta State 333116, Nigeria; e-mail : oweimieotuamanda@edwinclarkuniversity.edu.ng
[7] School of Data Intelligence and Technology, Robert Morris University, Pittsburgh, PA 15108, United States of America; e-mail : tabitha.aghaunor@gmail.com
[8] School of Media and Communications, Pan-Atlantic University, Lekki, Lagos State 332109, Nigeria; e-mail : aodoh@pau.edu.ng
[9] Faculty of Computing, Delta State University, Abraka, Delta State 330105, Nigeria; e-mail : agboijoy0@gmail.com
***** Corresponding Author : Eferhire Valentine Ugbotu, Kizito Eluemunor Anazia and Arnold Adimabua Ojugo

Abstract: The daily exchange of informatics over the Internet has both eased the widespread proliferation of resources to ease accessibility, availability and interoperability of accompanying devices. In addition, the recent widespread proliferation of smartphones alongside other computing devices has continued to advance features such as miniaturization, portability, data access ease, mobility, and other merits. It has also birthed adversarial attacks targeted at network infrastructures and aimed at exploiting interconnected cum shared resources. These exploits seek to compromise an unsuspecting user device cum unit. Increased susceptibility and success rate of these attacks have been traced to user's personality traits and behaviours, which renders them repeatedly vulnerable to such exploits especially those rippled across spoofed websites as malicious contents. Our study posits a stacked, transfer learning approach that seeks to classify malicious contents as explored by adversaries over a spoofed, phishing websites. Our stacked approach explores 3-base classifiers namely Cultural Genetic Algorithm, Random Forest, and Korhonen Modular Neural Network – whose output is utilized as input for XGBoost meta-learner. A major challenge with learning scheme(s) is the flexibility with the selection of appropriate features for estimation, and the imbalanced nature of the explored dataset for which the target class often lags behind. Our study resolved dataset imbalance challenge using the SMOTE-Tomek mode; while, the selected predictors was resolved using the relief rank feature selection. Results shows that our hybrid yields F1 0.995, Accuracy 0.997, Recall 0.998, Precision 1.000, AUC-ROC 0.997, and Specificity 1.000 – to accurately classify all 2,764 cases of its held-out test dataset. Results affirm that it outperformed benchmark ensembles. Result shows the proposed model explored UCI Phishing Website dataset, and effectively classified phishing (cues and lures) contents on websites.

Keywords: Ensemble learning; Feature selection; Imbalanced dataset; Machine learning; Phishing detection; SMOTE-Tomek; Stacked ensemble; XGBoost.

Received: August, 26th 2025
Revised: September, 25th 2025
Accepted: September, 27th 2025
Published: October, 1st 2025

1. Introduction

The digital revolution has ushered in a plethora of tools and processes that aim to facilitate efficient knowledge exchange among users [1], [2]. The devices ease data processing while offering the benefits of flexibility in the shared resource cum enhanced user-connectivity [3]. With security a major issue, such advances have continued to ignite the interest of adversaries [4]. The proliferation of smartphones with their processing capacities has further eased it as invasive targets, with protocols made more possible with emergent tools [5], [6]. An adversary uses the penetrative tools like malware (spam) to bolster socially-engineered threats that explore subterfuge mode to coordinate their attack at unsuspecting devices in their bid to compromise network infrastructure and resources [7], [8]. The attack ensures data exchange is targeted to exploit a user's social need [9], desires and insatiable trait [10]. Today's businesses are reshaped via fusion of informatics [11] – as a channel to deliver high-end values to consumers, who receive services as rendered. This exchange has today become a trillion-dollar war [12], as businesses must seek new frontiers to curb phishing attacks amongst other issues [13], as failure to safeguard these exchanges ushers in the need for cross-cutting research [14].

The success of many of these adversarial attacks hinges on user personality traits, which include online presence, emotional seclusion, insatiable desires, and trust issues [15]. An adversary masks their intent as a trusted ally to exploit a compromised resource, providing the attacker with pivot access for further exploits on the infrastructure [16]. The consequent rise in the adoption of smartphones has further eased these attacks and compromises considerably. Phishing simply redirects a user's request to a spoofed website, rippled with malicious content that seeks to expose a targeted user [17] or device without their knowledge [18]. Phishing consists of three elements: (a) a lure masks an attacker as a genuine user, targeting a user's empathy, fear and curiosity [19], (b) a hook is an embedded link in a message [20], and (c) a catch obtains an exposed device's private data. Its success is hinged on its frequency and diversity [21] with unrealistic demands that seek to intimidate a user's psyche with petty gifts [22], [23]. Vulnerability to scam can be due to demographics (i.e. age, gender, status, etc) [24], [25]. As illustrated in Figure 1–3, phishing susceptibility varies significantly across gender, social status, and age range. For instance, girls between 24–42 years were the most phished due to media presence or social seclusion [26]; there was also the factor of educational status cum societal approval [27]; and users between 18- 29 years were also phished more due to behavioural traits [28], [29].

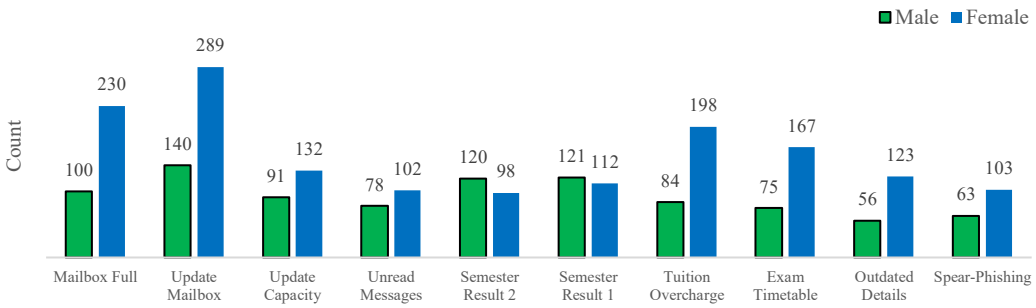

Figure 1. Phishing susceptibility by gender

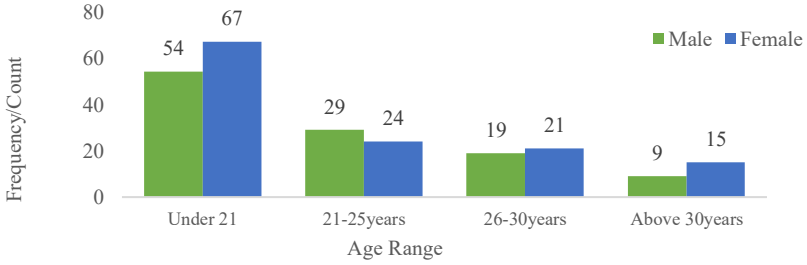

Figure 2. Phishing susceptibility by Status

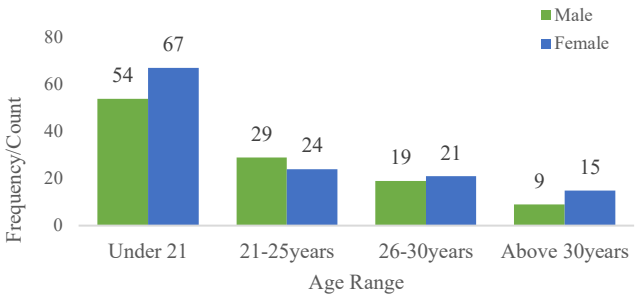

Figure 3. Phishing susceptibility by Age

Victimization impacts website's contents and its structure with greater probability an unsuspecting user will fall prey [30]. To identify malicious contents, we must eliminate gaps as follows [31], [32]: (a) identify lures that increases believability in a user [33], and (b) assess the undetectability and potency of cues to unsuspecting users [34], [35]. Learning models are successfully used to identify attacks, and detect cues and lures that leave users as susceptible. They identify data anomalies via learned outliers in a dataset [36], [37] as accomplished via vote, bagging, boosting, and stacked models/schemes [38], [39].

MLs are veritable tools to identify attacks. A trained MLs can detect anomalous patterns – even with its dynamic predictors. Learning schemes are grouped into: machine learning (ML) [40], deep learning (DL) [41], and ensemble learning (EL) [42]. ML's flexibility and robustness help it to learn intrinsic patterns and decode predictors that fastens model design, and ease outliers' identification. Its pitfalls are imbalanced dataset and the feature selection mode used. DLs utilize recurrent neural networks to capture chaotic, high-dimensional data patterns [43]. Its poor generalization due to the vanishing gradient problem, restricts its use. But, its variant overcomes this via its gates to control its input [44], and eases its adaptability to learned changes as long-term dependency [45], [46]. Its inability to handle larger dataset and longer training time required implies the quest for better alternative [47]. Lastly, ELs fuses ML with DL into a stronger learner to enhance performance [35]. It must resolve conflicts of structure and data-encode; while, leveraging the merits of both ML and DL to avoid model overfit as birthed by the underlying models [48], [49]. Thus, we explore the XGBoost to achieve such predictive abilities, leveraging its base, weak learners to enhance itself [48], [50]. It will improve its performance via error reduction on its weak (base) learner, and reduce its overall variance and bias in the dataset to improve generalization. It benefits from the comprehensive knowledge of its weaker base learners, to improve its generalization by exploiting the XGBoost scheme. With degraded performance due to an imbalanced data [51], [52], we explore the variant SMOTE-Tomek balancing. Our study wishes to: (a) identify phishing lures content on spoofed website, (b) resolve data imbalance via SMOTE-Tomek, and (c) select predictors concerning the target class via the relief rank feature selection mode.

Resolving data imbalance via oversampling has become imperative in ML, as it accounts for the minor class as crucial [53]. It is opposed to under-samplers that often reduces or ignore as meaningless, the minor class in a dataset. Thus, we use the synthetic minority oversample technique (SMOTE) [54], or its variants SMOTE-Tomek and SMOTEEN [55]. Our study is structured as follows: Section 1 introduces the subject and highlights the research gaps that motivate this work. Section 2 presents the proposed methodology, covering data collection, preprocessing, dataset balancing through SMOTE-Tomek, feature selection, and the construction of the stacked ensemble with XGBoost training and validation. Section 3 discusses the experimental results and provides a broader contextual analysis of the proposed model's performance on the UCI phishing website dataset. Section 4 presents the results and discussion in detail, while Section 5 concludes the study with key findings and implications.

2. Related Literatures

Various studies have been espoused on phishing website detection recently. For example, Li et al. [56] integrated feature selection approach with tree-based learning ensemble that aimed at improved accuracy. The study compared a variety of models and showed that their proposed AdaBoost achieved an accuracy of up to 93.2% to outperform benchmark models with accuracy between 70-to-91.5%. Ojugo and Eboka [13] explored a variety of model to

assess and compare the performance of both predictor selection by omitting redundant features (FSOR), and by filtering method (FSFM). FSOR yields 22-features while FSFM yields 11-features, respectively. They evaluated phishing detection performance using RF, MLP, and SVM. The results showed that Random Forest (RF)optimized with FSOR achieved highest performance with accuracies of 95% for RF, 94.7% for MLP, and 91% for SVM with efficient processing times. Pujara and Chaudhari [57] utilized several ML schemes to detect phishing websites including SVM, Logistic Regression (LR), RF, and Neural Networks (NNN). They sought to improve model accuracy via comprehensive feature selection. Their model achieved significant improvements with an accuracy of 96.7%. Setiadi et al. [58] explored BiGRU on feature selection by omitting redundant (FSOR) method to detect phishing websites. Their approach advanced a deep learning model to analyze websites features, improving phishing detection accuracy and robustness. They reported accuracy, F1 and AUC of 1.00. Ejeh et al. [59] proposed three meta-learner models based on the cost-penalty attribute (CostPA), which assigns weight to features used to build efficient decision trees, resulting in high accuracy and low false alarm rates. This approach achieved accuracies ranging from 95-to-97.6%.

A cursory look at the reviewed works shows the utilization of Phishing Websites dataset from the UCI Machine Learning Repository. It implies there is a consistent baseline for comparing various approaches in phishing website detection. While previous studies have demonstrated the effectiveness of various ML-and-DL approaches, including feature selection flexibility and imbalanced dataset resolution, these have necessitated the need to explore more advanced architectures such as transfer learning using stacked approach with meta-learners to improve phishing detection. Our research proposes using 3-base classifiers (i.e. Cultural Genetic Algorithm, RF and Korhonen Modular Neural Network) with the XGBoost meta-learner as combined with the relief ranking feature selection techniques with SMOTE-Tomek data balancing scheme. This approach aims to leverage temporal dependencies in the data more effectively, potentially offering superior performance in detecting phishing websites compared to existing methods.

3. Material and Method

The proposed transfer learning approach is shown in Figure 4.

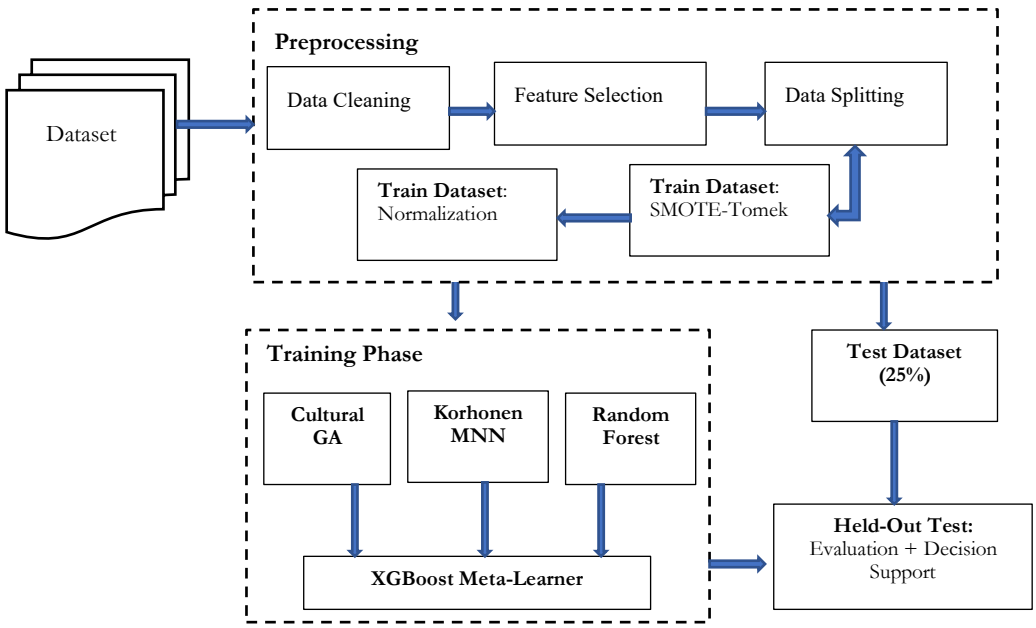

Figure 4. Proposed stacking ensemble with boosted learner

3.1. Data Collection

We explore the UCI phishing dataset, which consists of 11,055 records distributed into 5,180 cases in the genuine class, and 5,875 cases in the phishing class [60]. The original dataset is plotted as in Figure 5.

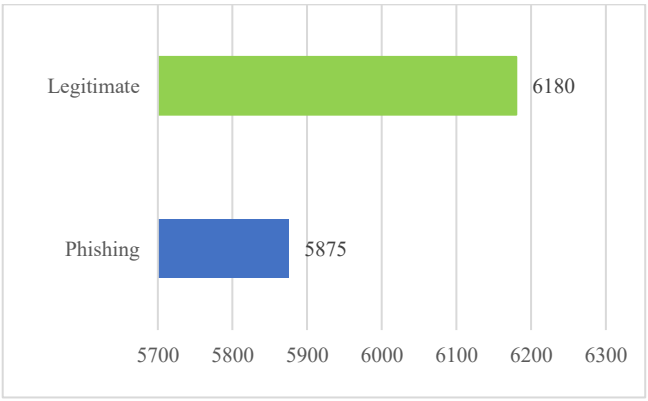

Figure 5. Original Dataset plot

3.2. Pre-processing

Cleans up the dataset by expunging redundancies to yield integrity, and removes missing values to yield quality. Optimized data is then encoded using the one-hot mode that transforms categorical data into its equivalent binary forms [61], [62]. Figure 6 shows the optimized dataset.

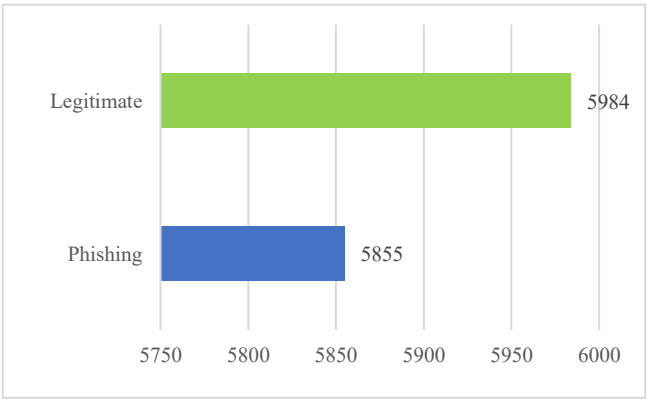

Figure 6. Optimized Dataset plot

3.3. Relief Rank Feature Selection

We select strictly, only the predictors of relevance to expunge all docile feats and reduce dataset dimensionality, to aid fastened model construction [63]. The relief rank: (a) assumes all features have same weight and influence on accuracy, (b) identifies the nearest sample from the same class as the nearest hit, and the nearest sample from a varying class as the nearest miss, and (c) uses feature value of nearest neighbour to update its weight(s) [64]. It assesses the correlation of all predictors for ground-truth as in Equation (1) [65]. With a threshold of 8.321 as computed, algorithm 1 ranked features to choose a total of 20 predictors as in Table 1, from the original UCI dataset with the initial 30 features.

$$Y = 100 * \sum |(x_1^2 - x_2^2)^2 + (1 - x_1^2)^2| \quad (1)$$

Algorithm 1. Relief Ranking Feature Selection Approach

INPUT: N, A, V // N is training instance, A is vectorAttribute, V is classValues
OUTPUT: W // W is vectorWeights

1: load dataset with training samples, predictors, weights and initialize predictor weights
2: for all weights → randomly select target predictor R && compute nearest hit = H && nearest miss = M
3: find W[A] = W[A] – *diff*(A,R,H)/m + *diff*(A,R,M)/m
4: return computed predictorScore for W
5: end

Table 1. Phishing dataset with relief ranking feature selection for predictors

Parameters	Description	Data Type	Selected
shortening_service	Whether a URL shortening service like bit.ly is used (1=Yes, -1=No)	char	Yes
double_slash_redirect	Presence of "//" in the URL path (1: Yes, -1: No)	char	Yes
having_IP_Address	Whether URL has IP Address instead of a domain name (1=Yes, 1=No)	alphanumeric	No
having_At_Symbol	Presence of "@" symbol in the URL (1: Yes, -1: No)	char	No
having_Sub_Domain	Number of subdomains in the URL (1: More than one, 0: One, -1: None)	char	Yes
URL_lenght	Length of the URL (1=long, 0=medium, -1=short)	integer	Yes
domain_reg_length	Length of time domain has been registered (1: over a year, -1: Less than a year)	integer	Yes
Prefix_Suffix	Presence of "-" in the domain part of the URL (1: Yes, -1: No)	char	No
SSLfinal_State	Whether the website uses HTTPS with a valid SSL certificate (1: Yes, -1: No)	char	Yes
Favicon	Whether the favicon is loaded from the same domain (1: Yes, -1: No)	char	Yes
port	Use of non-standard ports (1: Yes, -1: No)	alphanumeric	No
HTTPS_token	Presence of "HTTPS" token in the URL (1: Yes, -1: No)	char	Yes
Request_URL	Percentage of external links in the source code of the website (1: High, -1: Low)	alphanumeric	No
URL_of_Anchor	Percentage of external anchor links on the website (1: High, -1: Low)	char	Yes
Links_in_tags	Percentage of external links in tags (e.g., meta, script) (1: High, -1: Low)	char	Yes
SFH	Form Handler, where form data is submitted (1: External, 0: Internal, -1: Same)	alphanumeric	Yes
Submitting_to_email	Whether the form submits data to an email address (1: Yes, -1: No)	alphanumeric	Yes
Abnormal_URL	Whether the URL is abnormal (1: Yes, -1: No)	alphanumeric	No
Redirect	Number of redirects (1: More than one, -1: Less than one)	alphanumeric	No
on_mouseover	Whether changing status bar content on mouseover (1: Yes, -1: No)	char	No
RightClick	Whether right-click is turned off on the website (1: Yes, -1: No)	char	Yes
popUpWindow	Whether pop-up windows are present (1: Yes, -1: No)	char	Yes
Iframe	Whether iframe is used on the website (1: Yes, -1: No)	char	No
age_of_domain	Age of the domain (1: More than 6 months, -1: Less than 6 months)	integer	No
DNSRecord	Whether the DNS record exists (1: Yes, -1: No)	boolean	Yes
web_traffic	Web traffic rank (1: High, 0: Medium, -1: Low)	alphanumeric	Yes
Page_Rank	Google PageRank (1: High, -1: Low)	integer	Yes
Google_Index	Whether Google indexes the site (1: Yes, -1: No)	integer	Yes
Links_point_to_page	Number of links pointing to the page (1: High, 0: Medium, -1: Low)	alphanumeric	Yes
Statistical_report	Whether the website is reported as a phishing site (1: Yes, -1: No)	integer	Yes

3.4. Data Split/Balance

First, dataset is split into train (75% or 8,291-label), and test (25%, or 2,764-label). Balancing resample data, interpolating its nearest neighbour to create synthetic data to repopulate a pool, or removing data from train dataset to create a balanced dataset. Using SMOTE-Tomek [66], we fused the SMOTE oversampler with Tomek-links undersampler as in algorithm 2 [32] with Figure 7 as plot using the SMOTE-Tomek scheme.

Algorithm 2. SMOTE-Tomek's Links data balancing approach

INPUT: X, y, samplingStrategy=auto, k_neighbours
OUTPUT: X_resampled, y_resampled
1: load dataset with trainTestSplit → partition trainset (75%) and testSet (25%) via stratifyShuffleSplit
2: for trainset, use 5-fold split with randomState = 42
3: choose random point from minorClass
4: for each selectedData → compute: relativeDistance && kNearestNeighbour
5: choose randomValue [0,1] && compute randomValue * relativeDistance
6: update minorClassNew && repeat till setThreshold is reached for minorClassNew
7: select randomMinorClass: compute kNearestNeighbor(randomized_data)
8: with selected minorClassNew → evaluate newPool with TomekLink function
9: end TomekLink

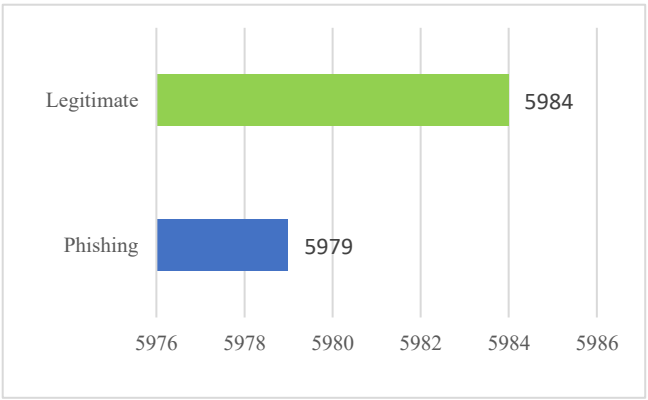

Figure 7. SMOTE-Tomek data balancing

After applying SMOTE-Tomek, the dataset yielded 5984 legitimate and 5979 phishing instances (Figure 7). Although not perfectly balanced, this slight difference results from the Tomek-link undersampling step, which removes borderline instances to improve class separability and reduce noise.

The choice of data splitting depends largely on the tradeoff between the need for a more robust model favoring the 75%:25% train-and-test ratio mode, or it can be poised towards the need for improved performance as guided by model complexity, larger dataset size and other feats so as to favor the 80%:20% mode. Here, our choice of the 75%:25% ratio leans on the small nature of the explored dataset with 11,055 records so that we can ultimately have a more robust evaluation on diverse unseen held-out (test) dataset, address the concerns of flexibility in feature selection, and proffer a more adaptive assessment with more accurate and less bias generalization of the model. In addition – with the train-subset still unbalanced, we performed data normalization using the z-score normalization as in Equation (2).

3.5. Stacked-Ensemble

In this step, we fuse 3-base learners with the XGB meta-learner, explained as:

3.5.1. Cultural Genetic Algorithm (CGA)

CGA uses belief spaces to guide the evolutionary search: (a) normative values to which predictors are bound, (b) domain knowledge that equips predictors with task-specific information, (c) temporal components that ensure predictors retain knowledge of past solutions,

and (d) spatial components that provide topological structure for solution exploration [66], The influence function sets the upper and lower bounds between (0,1), while Equations (3) and (4) allow knowledge transfer between the belief space and the population pool, modifying predictors to conform with the evolving knowledge base. Each chromosome gene $b_i \in \{1,0\}$ encodes a rule or feature representation, where the binary gene values denote the presence or absence of particular attributes [67]. During evolution, individuals are evaluated using a fitness function that measures their ability to discriminate between phishing and legitimate websites. The fittest individuals represent decision rules, and the majority voting of the evolved population determines the final class output of CGA. Table 2 is the CGA design.

$$f(x) = L_{lower} + x' \frac{L_{upper}}{2^N - 1} \tag{3}$$

$$x' = \sum_{i=0}^{N} b_i 2^i \tag{4}$$

Table 2. CGA design configuration.

Features	Value	Description
populationSize	120	The maximum number of individuals or candidate solutions in each generation
nosGenerations	30	Number of solutions in a generation
crossoverProbability	0.7	The likelihood of 2-parent individuals to create an offspring
selectionMode	int	1-rank, 2-elitism, 3-steadyState, 4-tourney, 5-stochasticUniversal-Sampling
req_fit_function	10	Minimal number of samples needed
offspring_created	int	Offspring: 1-crossover, 2-mutation
crossoverType	int	1-onePoint, 2-twoPoints, 3-uniform
mutationProbability	0.005	Controls the chance of random alteration in a candidate solution

3.5.2. Random Forest (RF)

RF successively grows its decision trees independently via a bootstrap sample, in bagging mode. It uses a binary split on its extra layer to extend the randomness on how its trees are constructed, so that its best nodes are selected randomly to capture intricate feats in the dataset. Its inability to handle diversity in categorical data results in its poor performance. Thus, we tune the hyperparameters to reduce model overfitting [68]. Expressed in Equation (5), with $normfi$ as normalized feature importance for i in tree j in Equation (6). T is the total number of trees, and fi_j is the importance of a feature i about ground-truth, and ni_j is nodal feature importance as in Equation (7) that yields Gini value. Table 3 shows the Random Forest design configuration.

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j} \tag{5}$$

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ features\ i} ni_j}{\sum_{k \in all\ nodes} ni_k} \tag{6}$$

$$ni_j = w_j c_j - w_{left(j)} c_{left(j)} - w_{right(j)} c_{right(j)} \tag{7}$$

3.5.3. Korhonen Modular Neural Network (KMNN)

KMNN yields a deep, modular learning model that computes its output using the tan-sigmoid function. It divides a network into smaller units for enhanced dependability and improved efficacy of its components [69]. This improves its computational efficiency, reduces time to convergence, and enables it to handle more tasks effectively in parallel. Its diversity grants each unit independent training, making KMNN more robust and flexible, with improved generalization. Table 4 details the KMNN design.

Table 3. Random Forest design configuration.

Features	Value	Description
nEstimators	250	Number of trees constructed
maxFeatures	log	Helps to control overfit when splitting a node
maxDepth	5	Max depth of each tree
minSampleSplit	10	Minimal samples needed to control tree size and complexity
random_state	25	The seeds for reproduction
eval_metric	error, logloss	Performance evaluation metrics
eval_set	x,val, y_val	Train data for evaluation
bootstrap	True	sets bootstrap aggregation used

Table 4. KMNN design configuration.

Features	Value	Description
eval_perf_set	MSE	Evaluation metrics at training
hidden_layers	10	Number of hidden layers adopted
training_percent	75	k-fold dataset used for training and cross-validation
transfer_hidden	tan-sigmoid	Transfer (activation) learning function
learning_rate	0.25	Step size learning to update the ensemble
number_layer	10	Minimal number of samples needed
data_division	stratified	k-fold dataset for construction
train_net_algo	LMBP	Training mode by a neural network
backpro_momentum	auto	Backpropagation-in-momentum learn

3.5.4. XGBoost

XGBoost meta-learner leans on the predictive output of its base models, expanding its objective function via a regularization term $\Omega(f_t)$ and loss function $l\left(Y_i^t, \hat{Y}_i^t\right)$ [70]. The objective is expressed in Equation (8), ensuring improved accuracy and generalization through hyperparameter tuning as shown in Table 5 [71].

$$obj^{(t)} = \sum_i l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i) + \Omega(f_t)\right) \tag{8}$$

3.6. Train and Cross Validation

This step is initialized with default configuration as in Tables (2)-(5) to tune hyperparameters. Each tree is iteratively constructed and trained to ensure the collective knowledge to identify intricate data. Training blends synthetic with original data to guarantee its comprehensive learning with improved adaptability to various configurations [72].

4. Results and Discussion

4.1. Ensemble Performance

For a comprehensive evaluation devoid of overfit, we use a 5-fold training partition on the 75% train-subset obtained via SMOTE-Tomek, and a final evaluation carried out via a held-out test (25%) subset as in Table 6. Proposed hybrid yields average accuracy 0.997, Recall 0.998, Precision 1.000, F1 0.995, Specificity 1.000 and AUC 0.997. Table 6 results in high value for MCC, and implies model accurately and consistently handles the minority class with data balancing performed; while the Specificity of 1.000 reached indicates that the model effectively recognizes phishing, malicious websites, and that no benign records were also misclassified. The held-out test (25%) assesses the model's generalization ability with unseen data. The results showed AUC value of 0.997, which implies that the model was able to differentiate between the benign and malignant records.

Table 5. XGBoost meta-learner design configuration.

Features	Value	Description
nEstimators	200	Number of trees constructed
learningRate	0.25	Step size learning to update XGBoost
maxDepth	5	Max depth of each tree
subSample	uniform (0.1)	Percent of rows used for each tree construction to prevent overfit
evalSet	x_val, y_val	Train dataset to evaluate performance
minSampleSplit	10	Minimal samples needed to control tree size and complexity
treeMethod	auto	Tree construction algorithm used in XGBoost
randomState	25	The seeds for reproduction

Table 6. XGBoost Meta-Learner Design Configuration.

Metrics	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Held-Out Test
Accuracy	0.991	0.981	0.997	0.998	1.000	0.997
Recall	0.981	1.000	0.975	0.976	1.000	0.998
Precision	1.000	0.984	1.000	0.996	1.000	1.000
F1	0.991	0.989	0.995	0.985	1.000	0.995
MCC	0.982	0.963	0.955	0.985	1.000	0.986
Specificity	1.000	1.000	0.985	0.998	1.000	1.000
AUC-ROC	0.999	0.999	0.986	0.996	1.000	0.997

Figure 8 is the AUC-ROC with a 0.997, and shows the model's capability to differentiate the negative and positive classes. The proposed model accurately identified all 2,764 of the test data. With only a misclassified case and no false positives recorded, its specificity of 1.000 implies that no phishing content was misclassified. This is critical to avoid misclassification when detecting phishing. Proposed model enhances phishing website detection performance on both the training data and the held-out test set.

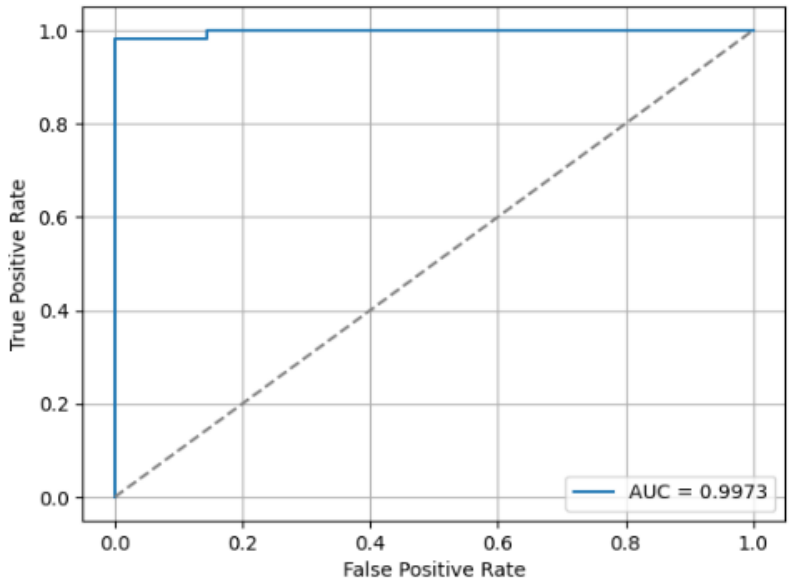

Figure 8. ROC result of the held-out test dataset

Figure 9 implies the ensemble correctly classified all test datasets with perfect accuracy. The utilization of both feature selection, SMOTE-Tomek balancing, and data normalization did not degrade performance [73]. Rather, it focuses on critical feats for model construction to successfully detect spoofed websites with reduced errors that will secure user(s) resources and enhance experience [74], [75].

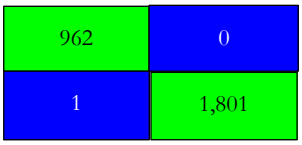

Figure 9. Confusion matrix

4.2. Ablation Studies with Benchmark Comparison

To assess the performance of our proposed ensemble, we focus on held-out test which offers a more realistic indication of the model's generalization capabilities. Summarized using the metric, Table 7 shows the ablation studies with performance of the base learners applied. Our hybrid ensemble yielded best result with F1 0.699, accuracy 0.697, precision and recall values of 0.685 and 0.684 respectively. Conversely, our benchmarks yield the F1 range [0.619, 0.639], accuracy range [0.609, 0.637], precision range [0.611, 0.64] and recall range [0.614, 0.64] respectively.

Table 7. Ablation results per components.

Models/ Components	Accuracy	Precision	Recall	F1
Kohonen Modular NN	0.609	0.611	0.614	0.619
Cultural Genetic Algorithm	0.619	0.632	0.634	0.611
Random Forest	0.627	0.642	0.653	0.631
XGBoost	0.637	0.640	0.640	0.639
Proposed	0.697	0.685	0.684	0.699

With the relief ranking feature selection strategy as applied. Table 8 shows performance of our hybrid versus the benchmark models. Results shows that our hybrid ensemble outperformed the benchmark with F1 0.857, accuracy 0.832, Precision and Recall values of 0.846 and 0.847. Conversely, our benchmarks yield the F1 range [0.793, 0.839], accuracy range [0.713, 0.826], precision range [0.704, 0.830] and recall range [0.771, 0.847] respectively.

Table 8. Performance with and without relief ranking feature selection.

Components	Without Relief Ranking				With Relief Ranking			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
KMNN	0.609	0.611	0.614	0.619	0.713	0.704	0.771	0.793
CGA	0.619	0.632	0.634	0.611	0.769	0.798	0.799	0.801
RF	0.627	0.642	0.653	0.631	0.819	0.842	0.822	0.842
XGBoost	0.637	0.640	0.640	0.639	0.826	0.830	0.845	0.839
Proposed	0.697	0.685	0.684	0.699	0.832	0.846	0.847	0.857

Table 9 shows our hybrid ensemble outperformed the benchmark with F1 0.995, accuracy 0.997, precision 1.000 and Recall 0.998 respectively. Conversely, our benchmarks show the various ranges for the various metric of performance as applied for the different benchmarks with F1 range [0.881, 0.955], accuracy range [0.899, 0.958], precision range [0.881, 0.951] and recall range [0.853, 0.952] respectively.

Table 9. Performance with and without SMOTE-Tomek data balancing.

Components	Without SMOTE-Tomek				With SMOTE-Tomek			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
KMNN	0.713	0.704	0.771	0.793	0.899	0.881	0.853	0.881
CGA	0.769	0.798	0.799	0.801	0.921	0.911	0.911	0.928
RF	0.819	0.842	0.822	0.842	0.928	0.944	0.944	0.938
XGBoost	0.826	0.830	0.845	0.839	0.958	0.951	0.952	0.955
Proposed	0.832	0.846	0.847	0.857	0.997	1.000	0.998	0.995

Table 10 shows that the proposed model underperforms against [58] due to its use of BiGRU with hybrid feature selection; However, other benchmark model underperformed in comparison to our proposed model, across metrics on the test dataset – achieving its high accuracy 0.997, precision 1.000, recall 0.998, specificity 1.000 and AUC-ROC 0.997 – showing best generalization with low false-positives, which is crucial in phishing detection especially with complex lures used by adversaries [76], [77] in their evolving exploit methods.

Table 10. Comparison with related works.

Metrics	SEM + DBN [17]	DHH + GRU [78]	BiGRU + FSOR [58]	LSTM +CNN [47]	GBM + PSO [79]	Ours
Accuracy	0.973	0.919	1.000	0.992	0.969	0.997
Recall	0.974	0.959	1.000	0.989	0.976	0.998
Precision	0.982	0.948	1.000	0.992	0.947	1.000
F1	0.976	0.973	1.000	0.985	0.974	0.995
Specificity	-	0.926	-	0.991	-	1.000
AUC-ROC	0.938	-	1.000	0.987	0.958	0.997

Models leverage deep learning capabilities – their performance can be seen to be slightly lower in metrics, and the lack thereof of specificity indicates that they are less robust; whereas, our model can be seen to maintain high sensitivity performance, even with its transfer learning architectures [80]. We used the SMOTE-Tomek scheme to address class imbalances.

5. Conclusions

The study affirms that our proposed stacked learning approach yields a strong learner potential with improved performance generalization by proffering a total of 60-rules using its CGA block with 18-of-such-rules found to yield a classification accuracy of 0.997. It implies that the rules as generated by proposed ensemble has 0.997 (i.e. 99.7%) rate to adequately identify phishing websites. Furthermore, the upper and lower bounds of the CGA ensure an elitist system for ground-truth is averted via the use of other base learners in the stacked ensemble. This increases its early detection rate at its training and validation so that model witnesses an increase in accuracy with decreased loss. Results suggest a robust and well-regularized model, whose success can be attributed to the effective combination of the balanced dataset, optimized weights, and a suitable learning. With top rules selected, ensemble yields accuracy 0.997, recall 0.998, precision 1.000, F1 0.995, specificity 1.000 and AUC 0.997 respectively. In addition, the proposed ensemble achieved high discriminative capability via statistically fused heuristics mode to successfully mitigate class-imbalance with enhanced evaluation scores for F1, accuracy, recall, specificity and AUC respectively. Study advances a lightweight yet effective framework that avoids complex training and validation that results in overfit or over-parameterization, effectively handles larger data complexities; while offering interpretability and high performance.

Author Contributions: Conceptualization: EVU, FUE and AAO; Methodology: EU, CCOd KEA, JA and CCOn; Software: PAO, KEA and AEO; Validation: APB, AOE, AO and POE; Formal Analysis: VOG, ROI, JA and CCOd; Investigation: AAO, TCA, ROI and POEz; Data Curation: POEj, POEz, CCOn and APB; Writing—original draft: EVU, KEA and FUE; Writing—review and editing: EU, JA and PAO; Visualization: AOE, VOG, POEj and TCA; Supervision: AAO, CCOn, ROI and CCOd and AO; Project administration: ROI, POEj, POEz, AEO and KEA; Funding acquisition: All. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is available online [web]: https://archive.ics.uci.edu/dataset/327/phishing+websites.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] S. Sinduja, "Efficient Phishing Website Detection using Machine Learning Algorithm," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 4, pp. 1442–1448, Apr. 2021, doi: 10.22214/ijraset.2021.33957.

[2] J. Hera, "Phishing Defense Mechanisms: Strategies for Effective Measurement and Cyber Threat Mitigation," *ResearchSquare*. 2024. doi: 10.13140/RG.2.2.27576.15364.

[3] H. Alamleh, A. A. S. AlQahtani, and B. Al Smadi, "Secure Mobile Payment Architecture Enabling Multi-factor Authentication," *arXiv*. Apr. 19, 2023. [Online]. Available: http://arxiv.org/abs/2304.09468

[4] F. O. Aghware *et al.*, "Enhancing the Random Forest Model via Synthetic Minority Oversampling Technique for Credit-Card Fraud Detection," *J. Comput. Theor. Appl.*, vol. 1, no. 4, pp. 407–420, Mar. 2024, doi: 10.62411/jcta.10323.

[5] L. De Kimpe, M. Walrave, W. Hardyns, L. Pauwels, and K. Ponnet, "You've got mail! Explaining individual differences in becoming a phishing target," *Telemat. Informatics*, vol. 35, no. 5, pp. 1277–1287, Aug. 2018, doi: 10.1016/j.tele.2018.02.009.

[6] R. Mohammad, T. L. Mccluskey, and F. Thabtah, "An Assessment of features related to phishing websites using an annotated technique," in *2012 International Conference for Internet Technology and Secured Transactions*, 2012, pp. 492–497. [Online]. Available: https://ieeexplore.ieee.org/document/6470857

[7] R. R. Atuduhor *et al.*, "StreamBoostE: A Hybrid Boosting-Collaborative Filter Scheme for Adaptive User-Item Recommender for Streaming Services," *Adv. Multidiscip. Sci. Res. J. Publ.*, vol. 10, no. 2, pp. 89–106, Jun. 2024, doi: 10.22624/AIMS/V10N2P8.

[8] E. A. Otorokpo *et al.*, "DaBO-BoostE: Enhanced Data Balancing via Oversampling Technique for a Boosting Ensemble in Card-Fraud Detection," *Adv. Multidiscip. Sci. Res. J. Publ.*, vol. 12, no. 2, pp. 45–66, 2024, doi: 10.22624/AIMS/MATHS/V12N2P4.

[9] N. Sudheer, B. Divya, P. Deepa, B. Geethika, and B. Balaji, "Detection of phishing websites using a machine learning alogrithm," 2025, p. 030059. doi: 10.1063/5.0247187.

[10] A. U. Z. Umar, "Simple but Smart: Against the Pursuit of Endless Complexity in Deep Learning Models for Detecting Phishing URLs," in *International Physical Sciences Conference*, 2025, no. August.

[11] G. Sasikala *et al.*, "An Innovative Sensing Machine Learning Technique to Detect Credit Card Frauds in Wireless Communications," *Wirel. Commun. Mob. Comput.*, vol. 2022, no. 1, pp. 1–12, Jan. 2022, doi: 10.1155/2022/2439205.

[12] M. R. Baker, T. Etem, K. H. Jihad, and S. Buyrukoğlu, "The Role of Hyperparameter Tuning in Phishing Website Classification: A Comparative Analysis of ML Models," in *Hybrid Intelligent Systems*, Springer, Cham, 2025, pp. 68–77. doi: 10.1007/978-3-031-78928-1_8.

[13] A. A. Ojugo and A. O. Eboka, "Comparative Evaluation for High Intelligent Performance Adaptive Model for Spam Phishing Detection," *Digit. Technol.*, vol. 3, no. 1, pp. 9–15, Nov. 2018, doi: 10.12691/dt-3-1-2.

[14] C. Li, N. Ding, H. Dong, and Y. Zhai, "Application of Credit Card Fraud Detection Based on CS-SVM," *Int. J. Mach. Learn. Comput.*, vol. 11, no. 1, pp. 34–39, Jan. 2021, doi: 10.18178/ijmlc.2021.11.1.1011.

[15] D. Huang, Y. Lin, Z. Weng, and J. Xiong, "Decision Analysis and Prediction Based on Credit Card Fraud Data," in *The 2nd European Symposium on Computer and Communications*, Apr. 2021, pp. 20–26. doi: 10.1145/3478301.3478305.

[16] I. A. Anderson and W. Wood, "Habits and the electronic herd: The psychology behind social media's successes and failures," *Consum. Psychol. Rev.*, vol. 4, no. 1, pp. 83–99, Jan. 2021, doi: 10.1002/arcp.1063.

[17] S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," *Appl. Sci.*, vol. 13, no. 8, p. 4649, Apr. 2023, doi: 10.3390/app13084649.

[18] W. Li, S. Manickam, Y. Chong, and S. Karuppayah, "Talking Like a Phisher: LLM-Based Attacks on Voice Phishing Classifiers," *arXiv*, no. July. Jul. 22, 2025. [Online]. Available: http://arxiv.org/abs/2507.16291

[19] A. Maureen, O. Anthonia, E. Omede, J. P. A. . Hampo, J. Anenechukwu, and C. Hampo, "Use of Adaptive Boosting Algorithm to Estimate User ' s Trust in the Utilization o f Virtual Assistant Systems," *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 1, pp. 502–507, 2023.

[20] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "A review of spam email detection: analysis of spammer strategies and the dataset shift problem," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 1145–1173, Feb. 2023, doi: 10.1007/s10462-022-10195-4.

[21] M. Ahmed, K. Ansar, C. B. Muckley, A. Khan, A. Anjum, and M. Talha, "A semantic rule based digital fraud detection," *PeerJ Comput. Sci.*, vol. 7, no. 1, p. e649, Aug. 2021, doi: 10.7717/peerj-cs.649.

[22] M. A. Haque *et al.*, "Cybersecurity in Universities: An Evaluation Model," *SN Comput. Sci.*, vol. 4, no. 5, p. 569, Jul. 2023, doi: 10.1007/s42979-023-01984-x.

[23] T. Sahmoud and M. Mikki, "Spam Detection Using BERT," *arXiv*. Jun. 07, 2022. [Online]. Available: http://arxiv.org/abs/2206.02443

[24] A. A. Ojugo, M. I. Akazue, P. O. Ejeh, C. C. Odiakaose, and F. U. Emordi, "DeGATraMoNN: Deep Learning Memetic Ensemble to Detect Spam Threats via a Content-Based Processing," *Kongzhi yu Juece/Control Decis.*, vol. 38, no. 1, pp. 667–678, 2023.

[25] C. Iuga, J. R. C. Nurse, and A. Erola, "Baiting the hook: factors impacting susceptibility to phishing attacks," *Human-centric Comput. Inf. Sci.*, vol. 6, no. 1, p. 8, Dec. 2016, doi: 10.1186/s13673-016-0065-2.

[26] R. De', N. Pandey, and A. Pal, "Impact of digital surge during Covid-19 pandemic: A viewpoint on research and practice," *Int. J. Inf. Manage.*, vol. 55, no. June, p. 102171, Dec. 2020, doi: 10.1016/j.ijinfomgt.2020.102171.

[27] M. D. Okpor *et al.*, "Pilot Study on Enhanced Detection of Cues over Malicious Sites Using Data Balancing on the Random Forest Ensemble," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 109–123, Sep. 2024, doi: 10.62411/faith.2024-14.

[28] N. Tabassum, F. F. Neha, M. S. Hossain, and H. S. Narman, "A Hybrid Machine Learning based Phishing Website Detection Technique through Dimensionality Reduction," in *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, May 2021, pp. 1–6. doi: 10.1109/BlackSeaCom52164.2021.9527806.

[29] P. Justice and N. Dumka, "An Improved Model for Detecting Uniform Resource Locator (URL) using Deep Learning," *IJARCCE*, vol. 10, no. 11, Nov. 2021, doi: 10.17148/IJARCCE.2021.101107.

[30] N. Roja, "Phishing Website Detection Using GUI Implementation," *Int. J. Sci. Res. Eng. Manag.*, vol. 09, no. 07, pp. 1–9, Jul. 2025, doi: 10.55041/IJSREM51200.
[31] W. Li, S. Manickam, Y.-W. Chong, W. Leng, and P. Nanda, "A State-of-the-Art Review on Phishing Website Detection Techniques," *IEEE Access*, vol. 12, no. December, pp. 187976–188012, 2024, doi: 10.1109/ACCESS.2024.3514972.
[32] M. Mia, D. Derakhshan, and M. M. A. Pritom, "Can Features for Phishing URL Detection Be Trusted Across Diverse Datasets? A Case Study with Explainable AI," in *Proceedings of the 11th International Conference on Networking, Systems, and Security*, Dec. 2024, pp. 137–145. doi: 10.1145/3704522.3704532.
[33] W. Li, S. U. A. Laghari, S. Manickam, Y. W. Chong, and B. Li, "Machine Learning-Enabled Attacks on Anti-Phishing Blacklists," *IEEE Access*, vol. 12, no. December, pp. 191586–191602, 2024, doi: 10.1109/ACCESS.2024.3516754.
[34] M. I. Akazue, I. A. Debekeme, A. E. Edje, C. Asuai, and U. J. Osame, "UNMASKING FRAUDSTERS: Ensemble Features Selection to Enhance Random Forest Fraud Detection," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 201–211, Dec. 2023, doi: 10.33633/jcta.v1i2.9462.
[35] G. S. Nayak, B. Muniyal, and M. C. Belavagi, "Enhancing Phishing Detection: A Machine Learning Approach With Feature Selection and Deep Learning Models," *IEEE Access*, vol. 13, pp. 33308–33320, 2025, doi: 10.1109/ACCESS.2025.3543738.
[36] D. R. I. M. Setiadi, A. R. Muslikh, S. W. Irianananda, W. Warto, J. Gondohanindijo, and A. A. Ojugo, "Outlier Detection Using Gaussian Mixture Model Clustering to Optimize XGBoost for Credit Approval Prediction," *J. Comput. Theor. Appl.*, vol. 2, no. 2, pp. 244–255, Nov. 2024, doi: 10.62411/jcta.11638.
[37] R. J. van Geest, G. Cascavilla, J. Hulstijn, and N. Zannone, "The applicability of a hybrid framework for automated phishing detection," *Comput. Secur.*, vol. 139, no. January, p. 103736, Apr. 2024, doi: 10.1016/j.cose.2024.103736.
[38] J. Yao, C. Wang, C. Hu, and X. Huang, "Chinese Spam Detection Using a Hybrid BiGRU-CNN Network with Joint Textual and Phonetic Embedding," *Electronics*, vol. 11, no. 15, p. 2418, Aug. 2022, doi: 10.3390/electronics11152418.
[39] A. A. Ojugo *et al.*, "Forging a User-Trust Memetic Modular Neural Network Card Fraud Detection Ensemble: A Pilot Study," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 50–60, Oct. 2023, doi: 10.33633/jcta.v1i2.9259.
[40] P. A. Onoma *et al.*, "Investigating an Anomaly-based Intrusion Detection via Tree-based Adaptive Boosting Ensemble," *J. Fuzzy Syst. Control*, vol. 3, no. 1, pp. 90–97, Mar. 2025, doi: 10.59247/jfsc.v3i1.279.
[41] A. Varun Kumar, A. Prathiba, A. Ashritha, N. Harish Reddy, and D. X. S. Asha Shiny, "Phishing Website Detection Based on URL Features," *Int. J. Sci. Res. Eng. Technol.*, pp. 73–78, Apr. 2025, doi: 10.59256/ijsreat.20250502011.
[42] A. P. Binitie *et al.*, "Stacked Learning Anomaly Detection Scheme with Data Augmentation for Spatiotemporal Traffic Flow," *J. Fuzzy Syst. Control*, vol. 2, no. 3, pp. 203–214, Oct. 2024, doi: 10.59247/jfsc.v2i3.267.
[43] B. O. Malasowe, F. O. Aghware, M. D. Okpor, B. E. Edim, R. E. Ako, and A. A. Ojugo, "Techniques and Best Practices for Handling Cybersecurity Risks in Educational Technology Environment (EdTech)," *J. Sci. Technol. Res.*, vol. 6, no. 2, pp. 293–311, 2024, doi: 10.5281/zenodo.12617068.
[44] C. L. Kumar *et al.*, "Metaparameter optimized hybrid deep learning model for next generation cybersecurity in software defined networking environment," *Sci. Rep.*, vol. 15, no. 1, p. 14166, Apr. 2025, doi: 10.1038/s41598-025-96153-w.
[45] M. S. Ataa, E. E. Sanad, and R. A. El-khoribi, "Intrusion detection in software defined network using deep learning approaches," *Sci. Rep.*, vol. 14, no. 1, p. 29159, Nov. 2024, doi: 10.1038/s41598-024-79001-1.
[46] S. Kayathri, T. Harikrishnan, K. Kb, and K. Kirubanithi, "Mechanism for Identifying Appropriate Phishing Websites Utilizing Machine Learning," *Int. Res. J. Mod. Eng. Technol. Sci.*, no. January, Apr. 2024, doi: 10.56726/IRJMETS52165.
[47] Y. A. Alsariera, A. V. Elijah, and A. O. Balogun, "Phishing Website Detection: Forest by Penalizing Attributes Algorithm and Its Enhanced Variations," *Arab. J. Sci. Eng.*, vol. 45, no. 12, pp. 10459–10470, Dec. 2020, doi: 10.1007/s13369-020-04802-1.
[48] R. Mahajan and I. Siddavatam, "Phishing Website Detection using Machine Learning Algorithms," *Int. J. Comput. Appl.*, vol. 181, no. 23, pp. 45–47, Oct. 2018, doi: 10.5120/ijca2018918026.
[49] C. C. Odiakaose *et al.*, "Hypertension Detection via Tree-Based Stack Ensemble with SMOTE-Tomek Data Balance and XGBoost Meta-Learner," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 3, pp. 269–283, Dec. 2024, doi: 10.62411/faith.3048-3719-43.
[50] P. J, P. G. James, A. P, R. B, A. S, and K. N, "Phishing Website Detection Using Machine Learning," in *2024 2nd International Conference on Networking and Communications (ICNWC)*, Apr. 2024, vol. 2, no. 2, pp. 1–5. doi: 10.1109/ICNWC60771.2024.10537279.
[51] N. Islam *et al.*, "Towards Machine Learning Based Intrusion Detection in IoT Networks," *Comput. Mater. Contin.*, vol. 69, no. 2, pp. 1801–1821, 2021, doi: 10.32604/cmc.2021.018466.
[52] E. V. Ugbotu *et al.*, "Transfer Learning Using a CNN Fused Random Forest for SMS Spam Detection with Semantic Normalization of Text Corpus," *NIPES - J. Sci. Technol. Res.*, vol. 7, no. 2, pp. 371–382, Jun. 2025, doi: 10.37933/nipes/7.2.2025.29.
[53] T. C. Aghaunor *et al.*, "Enhanced Scorch Occurrence Prediction in Foam Production via a Fusion SMOTE-Tomek Balanced Deep Learning Scheme," *NIPES - J. Sci. Technol. Res.*, vol. 7, no. 2, pp. 330–339, May 2025, doi: 10.37933/nipes/7.2.2025.25.
[54] M. I. Akazue, A. Clive, E. Abel, O. Edith, and E. Ufiofio, "Cybershield: Harnessing Ensemble Feature Selection Technique for Robust Distributed Denial of Service Attacks Detection," *Kongzhi yu Juece/Control Decis.*, vol. 38, no. 3, 2023.
[55] O. Sinayobye, R. Musabe, A. Uwitonze, and A. Ngenzi, "A Credit Card Fraud Detection Model Using Machine Learning Methods with a Hybrid of Undersampling and Oversampling for Handling Imbalanced Datasets for High Scores," 2023, pp. 142–155. doi: 10.1007/978-3-031-34222-6_12.
[56] W. Li, S. Manickam, S. U. A. Laghari, and Y.-W. Chong, "Uncovering the Cloak: A Systematic Review of Techniques Used to Conceal Phishing Websites," *IEEE Access*, vol. 11, no. December 2024, pp. 71925–71939, 2023, doi: 10.1109/ACCESS.2023.3293063.
[57] S. S. U S, "Phishing Website Detection using Machine Learning," *INTERANTIONAL J. Sci. Res. Eng. Manag.*, vol. 08, no. 06, pp. 1–5, Jun. 2024, doi: 10.55041/IJSREM36212.
[58] D. R. I. M. Setiadi, S. Widiono, A. N. Safriandono, and S. Budi, "Phishing Website Detection Using Bidirectional Gated Recurrent Unit Model and Feature Selection," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 75–83, Jul. 2024, doi: 10.62411/faith.2024-15.

[59] P. O. Ejeh, E. Adishi, E. Okoro, and A. Jisu, "Hybrid integration of organizational honeypot to aid data integration, protection and organizational resources and dissuade insider threat," *FUPRE J. Sci. Ind. Res.*, vol. 6, no. 3, pp. 80–94, 2022.
[60] W. Li, S. Ul Arfeen Laghari, S. Manickam, and Y. W. Chong, "Exploration and Evaluation of Human-centric Cloaking Techniques in Phishing Websites," *KSII Trans. Internet Inf. Syst.*, vol. 19, no. 1, pp. 232–258, Jan. 2025, doi: 10.3837/tiis.2025.01.011.
[61] S. A. A. A. Alsaidi *et al.*, "HawkPhish-DNN cybersecurity model: adaptive hybrid optimization and deep learning for enhanced multi-objective phishing URL detection," *Int. J. Inf. Technol.*, vol. 17, no. 7, pp. 3859–3875, Sep. 2025, doi: 10.1007/s41870-025-02597-8.
[62] D. Komalasari, T. B. Kurniawan, D. A. Dewi, M. Z. Zakaria, Z. Abdullah, and A. Alanda, "Phishing Domain Detection Using Machine Learning Algorithms," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 15, no. 1, pp. 318–327, Feb. 2025, doi: 10.18517/ijaseit.15.1.12553.
[63] H. Nakano, T. Koide, and D. Chiba, "PhishParrot: LLM-Driven Adaptive Crawling to Unveil Cloaked Phishing Sites," in *Proceedings of IEEE Global Communications Conference (GLOBECOM), 2025.*, 2025.
[64] N. R. Pratama, D. R. I. M. Setiadi, I. Harkespan, and A. A. Ojugo, "Feature Fusion with Albumentation for Enhancing Monkeypox Detection Using Deep Learning Models," *J. Comput. Theor. Appl.*, vol. 2, no. 3, pp. 427–440, Feb. 2025, doi: 10.62411/jcta.12255.
[65] M. D. Okpor *et al.*, "Comparative Data Resample to Predict Subscription Services Attrition Using Tree-based Ensembles," *J. Fuzzy Syst. Control*, vol. 2, no. 2, pp. 117–128, Jul. 2024, doi: 10.59247/jfsc.v2i2.213.
[66] D. R. I. M. Setiadi, K. Nugroho, A. R. Muslikh, S. W. Iriananda, and A. A. Ojugo, "Integrating SMOTE-Tomek and Fusion Learning with XGBoost Meta-Learner for Robust Diabetes Recognition," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 1, pp. 23–38, May 2024, doi: 10.62411/faith.2024-11.
[67] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," *J. Inf. Secur. Appl.*, vol. 55, p. 102596, Dec. 2020, doi: 10.1016/j.jisa.2020.102596.
[68] F. Omoruwou, A. A. Ojugo, and S. E. Ilodigwe, "Strategic Feature Selection for Enhanced Scorch Prediction in Flexible Polyurethane Form Manufacturing," *J. Comput. Theor. Appl.*, vol. 1, no. 3, pp. 346–357, Feb. 2024, doi: 10.62411/jcta.9539.
[69] R. E. Yoro *et al.*, "Adaptive DDoS detection mode in software-defined SIP-VoIP using transfer learning with boosted meta-learner," *PLoS One*, vol. 20, no. 6, p. e0326571, Jun. 2025, doi: 10.1371/journal.pone.0326571.
[70] S. Ju, H. Lim, and J. Heo, "Machine learning approaches for crop yield prediction with MODIS and weather data," *40th Asian Conf. Remote Sensing, ACRS 2019 Prog. Remote Sens. Technol. Smart Futur.*, no. Acrs, pp. 1–4, 2020.
[71] Y. Abakarim, M. Lahby, and A. Attioui, "An Efficient Real Time Model For Credit Card Fraud Detection Based On Deep Learning," in *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, Oct. 2018, pp. 1–7. doi: 10.1145/3289402.3289530.
[72] A. Ghasemieh, A. Lloyed, P. Bahrami, P. Vajar, and R. Kashef, "A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients," *Decis. Anal. J.*, vol. 7, no. April, p. 100242, Jun. 2023, doi: 10.1016/j.dajour.2023.100242.
[73] D. Sun, M. Liu, M. Li, Z. Shi, P. Liu, and X. Wang, "DeepMIT: A Novel Malicious Insider Threat Detection Framework based on Recurrent Neural Network," in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, May 2021, pp. 335–341. doi: 10.1109/CSCWD49262.2021.9437887.
[74] A. Falconnet, C. K. Coursaris, J. Beringer, W. Van Osch, S. Sénécal, and P.-M. Léger, "Improving User Experience with Recommender Systems by Informing the Design of Recommendation Messages," *Appl. Sci.*, vol. 13, no. 4, p. 2706, Feb. 2023, doi: 10.3390/app13042706.
[75] S. K. Bajaj and S. Hansen, "Social effects of phishing on e-commerce," in *IADIS International Conference on e-Commerce 2008*, 2008, no. January 2008, pp. 215–219.
[76] F. O. Aghware, R. E. Yoro, P. O. Ejeh, C. C. Odiakaose, F. U. Emordi, and A. A. Ojugo, "DeLClustE: Protecting Users from Credit-Card Fraud Transaction via the Deep-Learning Cluster Ensemble," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 6, pp. 94–100, 2023, doi: 10.14569/IJACSA.2023.0140610.
[77] A. R. Erhovwo, O. A. Ejaita, and D. Oghorodi, "A methodology for e-banking risk assessment using fuzzy logic and Bayesian network," *Sci. Africana*, vol. 19, no. 3, pp. 101–124, Feb. 2021, doi: 10.4314/sa.v19i3.8.
[78] L. Lakshmi, M. P. Reddy, C. Santhaiah, and U. J. Reddy, "Smart Phishing Detection in Web Pages using Supervised Deep Learning Classification and Optimization Technique ADAM," *Wirel. Pers. Commun.*, vol. 118, no. 4, pp. 3549–3564, Jun. 2021, doi: 10.1007/s11277-021-08196-7.
[79] K. Omari, "Phishing Detection using Gradient Boosting Classifier," *Procedia Comput. Sci.*, vol. 230, pp. 120–127, 2023, doi: 10.1016/j.procs.2023.12.067.
[80] J. H. Setu, N. Halder, A. Islam, and M. A. Amin, "RSTHFS: A Rough Set Theory-Based Hybrid Feature Selection Method for Phishing Website Classification," *IEEE Access*, vol. 13, pp. 68820–68830, 2025, doi: 10.1109/ACCESS.2025.3561237.