Whale Optimization and AutoML for Precise Phishing Detection

Divya Singhal[1], Ankit Verma[2], Ganesh V. Radhakrishnan[3], Jyoti Parashar[4], Saroj S. Date[5] and Kamal Upreti[6,*]

[1]*Department of Computer Science, Noida Institute of Engineering & Technology, Greater Noida, India*
[2]*Department of Computer Applications, KIET Group of Institutions, Delhi-NCR, Ghaziabad, India*
[3]*Department of Economics and Finance, KIIT School of Management (KSOM), KIIT University, Bhubaneswar, India*
[4]*Department of Computer Applications, Panipat Institute of Engineering &Technology College, Panipat, Haryana, India*
[5]*Department of Artificial Intelligence and Data Science, Chh.Shahu College of Engineering, Kanchanwadi, Paithan Road, Chhatrapati Sambhajinagar (Aurangabad), MS, India*
[6]*Department of Computer Science, Christ University, Delhi NCR, Ghaziabad, India*
E-mail: divyasinghal021@gmail.com; ankit.mca4u@gmail.com; vrkris2002@gmail.com; jyoti.parashar123@gmail.com; saroj.date@gmail.com; kamalupreti1989@gmail.com
[*]*Corresponding Author*

Received 18 April 2025; Accepted 05 August 2025

Abstract

Online fraud and social engineering tactics frequently use phishing websites as platforms. Phishers often modify the source code of the web pages they exploit in their attacks to create the illusion that alterations were made to authentic websites. A solitary response is insufficient to mitigate phishing due to the many methods employed in its execution. This study examines machine learning algorithms and evaluates their efficacy when trained on

Journal of Mobile Multimedia, Vol. 21_5, 855–880.
doi: 10.13052/jmm1550-4646.2153

datasets including attributes that differentiate secure websites from phishing sites. Automated algorithms facilitate real-time fraud protection by swiftly detecting suspicious URLs, domain names, and website content. This study aims to identify the optimal method for detecting a prevalent category of cyberattacks. This would enhance the security and privacy of all internet users by facilitating the identification and blocking of malicious websites. Nonetheless, there is an urgent desire for automated models that provide rapid and precise detection. This research introduces a regression-based assessment method for phishing detection to address this demand. Our approach employs a whale optimization algorithm for feature selection. An AutoML framework subsequently utilizes the selected feature subsets as input. The model showed good accuracy in its predictions with very small errors on the test data, shown by an RMSE of 0.1079, an MSE of 0.0116, and an R^2 value of 0.9534. These results demonstrate the reliability of our feature selection and modeling methods.

Keywords: Phishing attack, optimization algorithm, whale optimization algorithm, AutoML framework, AutoML H2O, regression analysis, random forest algorithm.

1 Introduction

Phishing websites aim to collect and retrieve sensitive data from users, such as personal credentials, account details, passwords, and so on. By tricking individuals into clicking on dangerous URLs, maybe expose more personal data [1]. Because more individuals than ever search the internet using their cell phones. Mobile browsers' smaller address bar makes it more difficult to determine if a URL is legitimate or phony. Various methods are employed to initiate phone-based deception [2]. Voice phishing, or "vishing," is one instance where scammers phone possible victims. SMS-based phishing, or "smishing," is another method in which fraudsters promote phishing sites' URLs within SMS and Internet-mediated, phone-to-phone text messages. The typical fraud entails receiving a fake purchase receipt that tells her to call a support line within a designated period to challenge the charge. After a phone call, the con artist either gathers the victim's personal and financial data or convinces the victim to send gift cards or money to him [3].

Deceptive phishing attempts mimic the official websites of banks, educational institutes, online marketing, government offices, and financial institutions. Researchers have observed that a free webmail domain was the primary

platform for launching most attacks [4, 5]. The "Anti-Phishing Working Group (APWG)" received 963,994 reports of phishing attacks during the first quarter of 2024. This is the lowest quarterly total since 4Q 2021 and significantly lower than the 1,624,144 attacks documented in Q1 2023, which was the highest quarter APWG has ever witnessed [6]. While the overall count of documented phishing attempts has remained consistent, the second quarter of 2024 saw a record 877,536 phishing attacks. Representing 37.6% of all phishing attempts, social media was the most often attacked industry in 2024, as seen in Figure 1. However, there was a 70% decline in the banking sector from Q3 2023 to Q1 2024. Figure 2 represents the frequency of phishing assaults from quartile 3 in 2022 2024.

Phishing attacks, a major cybersecurity threat, come in various forms, including spear phishing, whaling, vishing, smishing, and pharming [7]. These attacks aim to steal sensitive information by luring users to fake websites or through malicious emails [8]. Anti-phishing solutions can be categorized into prevention and detection methods, with detection being more crucial[9]. Detection techniques include whitelist/blacklist approaches, content-based, URL-based, visual-similarity, and machine learning methods. Unsupervised and supervised ML methods taken together can identify known and unknown attacks [10]. There are numerous standard phishing detection products, such as Mimecast. URL analysis, Microsoft Defender, and Brand Shield offer advanced protection against these attacks [11]. These tools often use domain reputation tracking, real-time threat intelligence, behavioral analysis, and signature-based techniques. However, problems arise as many tools rely heavily on blacklists, or emails that are not in the database may bypass detection. Delayed detection allows users to interact with malicious links or websites before the threat is flagged.

In response to this, existing ML-based approaches for phishing detection often struggle with interpretability and feature relevance [12]. Table 1 presents the information of the linked study by means of a comparative survey meant to highlight some of the main conclusions or approaches covered. We recommend utilizing optimization techniques, widely recognized as one of the most effective methods for addressing these limitations. This study addresses these challenges by proposing regression analysis with a whale optimization algorithm for feature selection and AutoML for model optimization. AutoML is the act of automating the application of machine learning to problems in the real world. It seeks to make model choice, hyperparameter optimization, feature preprocessing, and validation affordable and convenient by avoiding the necessity of trial-and-error work by human beings,

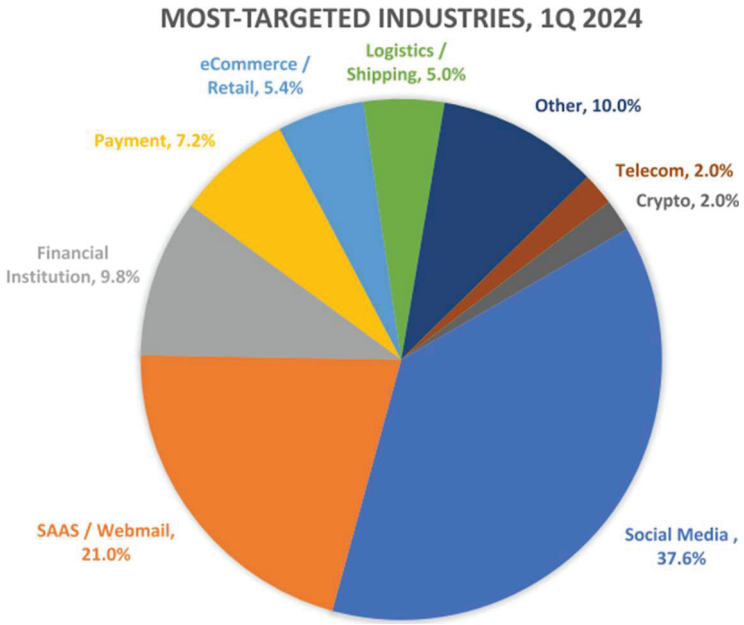

Figure 1 Distribution of Phishing Attacks based on category during Q1 2024 [6].

Figure 2 Number of Phishing attacks reported during Q3 2022–Q3 2024 [6].

particularly by non-experts. This hybrid approach ensures high performance while maintaining the interpretability of rules derived from phishing analysis. This research is motivated by the increasing complexity of assaults, which makes many outdated detection methods, including algorithms and strategies

applied by security systems, browsers, email providers, and other entities, useless. The constraints noted draw attention to the continual development of detection techniques and the ongoing threat that phishing attempts represent. One can compile the general contributions of this work as follows:

- The goal is to examine the characteristics of phishing websites and pinpoint crucial elements that aid in precise identification.
- This study employs WOA for selecting the most relevant features, reducing dimensionality, and improving model efficiency.
- To employ AutoML tools for identifying optimal models and optimizing hyperparameter settings for phishing detection.
- We evaluated the proposed approach using metrics like MSE, RMSE, and R^2, offering a deeper understanding of model performance and residual errors.

2 Proposed Workflow

In this section, we explain our proposed phishing detection system as depicted in Figure 3, which contains three phases: (a) Dataset accumulation (b) Feature Extraction (c) Model selection and evaluation.

2.1 Dataset Accumulation & Preprocessing

The dataset used for performance evaluation is "Phishing Dataset for Machine Learning: Feature Evaluation," collected from the Mendeley online repository "https://data.mendeley.com/datasets/h3cgnj8hft/1" [23], out of which the phishing webpage dataset consists of both 5000 phishing and legitimate websites. The dataset listed 48 features and a total of 10,000 websites. Prior to feature selection and model training, the dataset was preprocessed for uniformity and quality. This involved categorical feature conversion via label encoding, and min-max normalization on continuous features. We used these to normalize feature ranges and improve the convergence of the optimization and learning algorithms.

2.2 Feature Extraction Using WOA

WOA inspired by the social and predatory behaviors of humpback whales, iteratively searched the feature space to identify the optimal subset. The algorithm operated in a binary feature space, dynamically updating whale positions based on their proximity to the leader. The final output was a

Table 1 A comprehensive survey of the related work

Ref.	Main Findings	Algorithms	Strengths	Limitations
[13]	The system uses URL and domain identity mechanisms to classify phishing websites.	Ant Colony Optimization	Good for discrete feature selection	Convergence can be slow
[14]	classifies spam words in phishing emails.	Bayesian algorithm	Probabilistic & interpretable	Performance drops on imbalanced data.
[15]	The system utilizes web traffic, web content, and URLs to detect phishing and zero-day attacks.	Web Crawler-based detector	Captures diverse features	Web crawling is time-consuming & may miss real-time threats.
[16]	Perform semantic analysis of phishing emails.	NLP & semantic analysis	Understands language context	This task requires high complexity and computation.
[17]	Discusses various methods, challenges, and methodologies for phishing detection.	Hybrid/ML-based methods	Broad coverage of attacks	May lack real-time adaptability
[18]	Identify electronic phishing messages using hyperlinks' generic attributes in phishing attacks.	The Linkguard Algorithm	Effectively analyzes link structure of phishing attempts	Advanced obfuscation & URL redirection
[19, 20]	A phishing attack uses social engineering to steal sensitive information.	Content-based, heuristic-based, and fuzzy rule-based detection	Capable of detecting phishing patterns & adaptive rules	May generate false results & require manual tuning
[21]	Uses multiple classifiers on the PhishTank dataset	Naïve Bayes, decision tree to detect attacks, random forest, logistic regression, and fictitious classifier to achieve high accuracy.	High accuracy on benchmark data	Needs extensive feature engineering
[22]	Applies deep learning to large datasets.	CNN-LSTM	Excellent for feature learning	It requires large data and compute power.

binary vector representing the optimal subset of features. Initially, the dataset comprised a set of features extracted from raw data. These features were encoded as a binary vector $X = f1, f2, f3, \ldots\ fn$, where each element fi indicates whether a particular feature is selected or excluded for evaluation. The WOA was set up with a population of size 30 and up to 50 iterations. The 'a' coefficient was reduced linearly from 2 to 0, and a sigmoid transfer

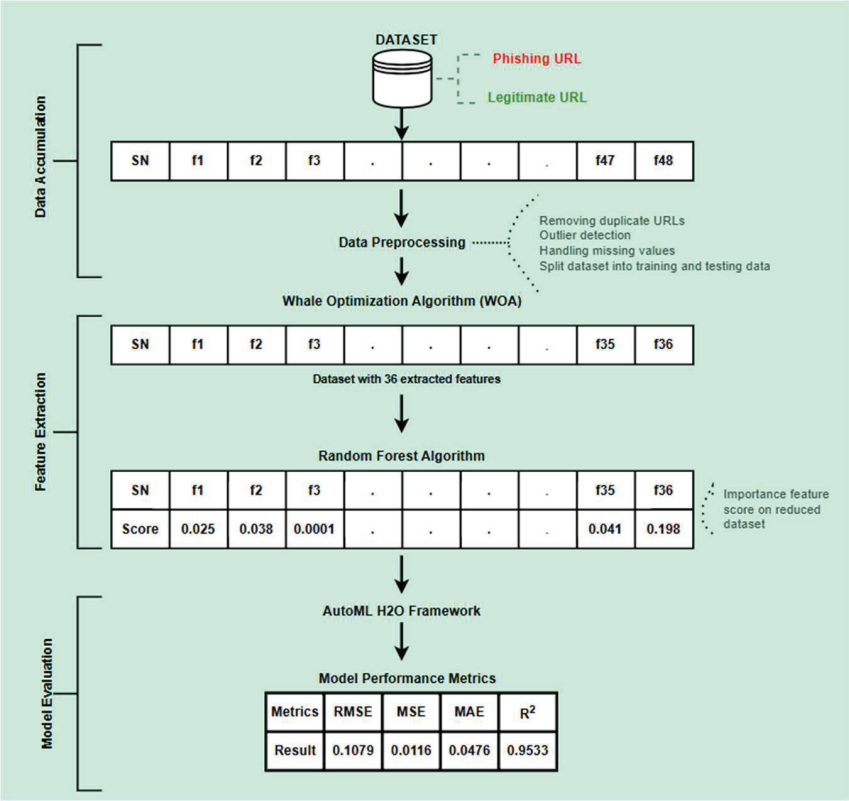

Figure 3 Methodological Flow represents three phases.

function transformed continuous position vectors into binary feature selection choices. These parameter settings were chosen empirically to achieve a trade-off between the speed of convergence and solution quality without too high a computational burden. These selected features were then used to reduce the original dataset (X_{train} and x_test), retaining only the most informative dimensions. Subsequently, the selected features and target labels were used for model training. Subsequently, Figure 4 lists 36 out of 48 features extracted to generate a particular feature vector. To assess the quality of selected features, a random forest (RF) classifier was used in conjunction with 5-fold cross-validation. Figure 5 lists the 36 features in hierarchical order based on their significance rankings. Although the presence of hostname or links in the status bar was the least important characteristic, the results revealed that the percentage of external links, domain name mismatch, and external script

SN	Feature	Type	Description
3	PathLevel	Numeric	Determining the level of the path in the URL
5	NumDash	Numeric	Total number of dashes in a URL
7	AtSymbol	Boolean	Total number of '@' symbols in the URL
9	NumUnderscore	Numeric	Number of underscores'_' used in the URL
10	NumPercent	Numeric	Total number of percent symbols present in the URL
11	NumQueryComponents	Numeric	Total number of query components
12	NumAmpersand	Numeric	Total number of '&' character
13	NumHash	Numeric	Total number of '#' character
14	NumNumericChars	Numeric	The total number of numeric characters
15	NoHttps	Boolean	Check if there is a HTTPS in the URL
17	IpAddress	Boolean	Check if the hostname of the URL uses the IP address
19	DomainInPaths	Boolean	Determines if the website link has used TLD or CCTLD
20	HttpsInHostname	Boolean	Determines if HTTPS is disorderly in the hostname of the URL
22	PathLength	Numeric	Length of all paths in each URL
23	QueryLength	Numeric	Length of query in the URL
24	DoubleSlashInPath	Boolean	Check if there is a double slash in the path
25	NumSensitiveWords	Numeric	Check if there are any sensitive words like secure, sign in, login, etc
27	PctExtHyperlinks	Float	Check the percentage of external hyperlinks in the source code
28	PctExtResourceUrls	Float	Checks the percentage of URL external resources in the source code
29	ExtFavicon	Boolean	Check if the favicon is installed from a different hostname.
30	InsecureForms	Boolean	Will see if the action in forms follows the HTTPS protocol
32	ExtFormAction	Boolean	Check if the action form contains an external URL
33	AbnormalFormAction	Boolean	Check if the action form contains an abnormal URL
34	PctNullSelfRedirectHyperlinks	Float	Check the percentage of hyperlinks that have an empty value and if it has an auto-directing value
35	FrequentDomainNameMismatch	Boolean	Checks if the URL, when accessed, shows a mismatch in the frequent domain name
36	FakeLinkInStatusBar	Boolean	Check if any fake links in the status bar lure the user towards unsafe websites.
39	SubmitInfoToEmail	Boolean	Check whether a URL requires you to submit your information to email
40	IframeOrFrame	Boolean	Check if the given URL has used iframes or frames
41	MissingTitle	Boolean	Check if there are any missing title
42	ImagesOnlyInForm	Boolean	Check if there are only images in the action form
43	SubdomainLevelRT	-1,0,1	Check if the subdomain levels are correlated.
44	UrlLengthRT	-1,0,1	Check if the URL lengths are correlated
45	PctExtResourceUrlsRT	-1,0,1	Check if the percentage of external URLs is correlated
46	AbnormalExtFormActionR	-1,0,1	Checks the relationship of different abnormal action forms in the URL
47	ExtMetaScriptLinkRT	-1,0,1	Check the correlation of meta-script links
48	PctExtNullSelfRedirectHyperlinksRT	-1,0,1	Checks the correlation of the percentage of self-directed hyperlinks

Figure 4 Number of Features Extracted by WOA.

links were the most notable aspects in terms of their contributions to threat identification.

2.3 Model Selection and Evaluation Metrics

The purpose of this phase was to autonomously examine, train, and optimize various ML models to achieve superior performance in the phishing detection

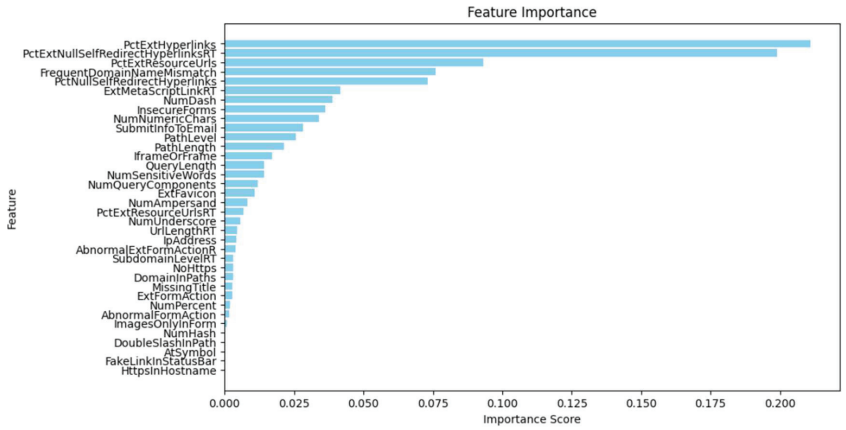

Figure 5 Relative ranking of 36 features according to their importance score.

problem. We employed AutoML with the Hydrology 2.0 (H2O) framework to develop and evaluate predictive models utilizing the condensed feature set obtained from WOA. H2O AutoML achieves results better than other frameworks using approaches like evolutionary algorithms or Bayesian optimization by combining quick random search with stacked ensembles [24]. The data structure optimized for the H2O framework is formed by integrating the diminished feature set $reduced_{X_{train}}$ With the target label y_{train}. H2O AutoML was started with the overall runtime limit of 3600 seconds and stopping metric as RMSE. A random seed value of 1234 was applied to ensure reproducibility of results. Such selection enabled efficient balance between systematic model exploration and training performance, allowing for reasonable comparison among various model architectures. Figure 6 represents the scoreboard performance metrics of multiple models trained and evaluated on the validation set. The stacked ensemble family consistently ranked top on the leader's board, achieving the best performance metrics. The stackedEnsemble_AllModels_4 model had the lowest root mean squared error (RMSE) of 0.116655, mean squared error (MSE) of 0.0136084, and mean absolute error (MAE) of 0.049628, showing it was very accurate in its predictions. The $GBM\_grid\_1\_AutoML\_1\_model\_53$ achieved an RMSE of 0.122612, making it competitive but slightly less accurate than the top stacked ensemble models (SEM). This work used four evaluation metrics (MSE, MAE, RMSE, and R^2) for reporting the results of the suggested model in order to evaluate the performance of the given regression tasks and ascertain the optimal model.

model_id	rmse	mse	mae	rmsle	mean_residual_deviance
StackedEnsemble_AllModels_4_AutoML_1_20250102_52110	0.116655	0.0136084	0.0492826	0.0844523	0.0136084
StackedEnsemble_BestOfFamily_5_AutoML_1_20250102_52110	0.117476	0.0138007	0.0504635	0.0852463	0.0138007
StackedEnsemble_BestOfFamily_6_AutoML_1_20250102_52110	0.118089	0.0139451	0.0271008	0.0829322	0.0139451
StackedEnsemble_AllModels_3_AutoML_1_20250102_52110	0.118237	0.01398	0.0501841	0.0851982	0.01398
StackedEnsemble_BestOfFamily_4_AutoML_1_20250102_52110	0.118917	0.0141412	0.0510037	0.0857453	0.0141412
GBM_grid_1_AutoML_1_20250102_52110_model_53	0.122612	0.0150337	0.0509825	0.0885424	0.0150337
StackedEnsem Ask fFamily_3_AutoML_1_20250102_52110	0.122867	0.0150964	0.051568	0.0880847	0.0150964
StackedEnsemble_AllModels_2_AutoML_1_20250102_52110	0.122882	0.0151	0.0508879	0.0880454	0.0151
StackedEnsemble_AllModels_1_AutoML_1_20250102_52110	0.123136	0.0151626	0.0495482	0.0882619	0.0151626
StackedEnsemble_BestOfFamily_2_AutoML_1_20250102_52110	0.123348	0.0152148	0.0500464	0.0884323	0.0152148
XGBoost_grid_1_AutoML_1_20250102_52110_model_14	0.123635	0.0152856	0.0533853	0.0899803	0.0152856
GBM_grid_1_AutoML_1_20250102_52110_model_26	0.125229	0.0156823	0.0531575	0.0906449	0.0156823
GBM_grid_1_AutoML_1_20250102_52110_model_4	0.125371	0.0157179	0.0588535	0.0915057	0.0157179
GBM_grid_1_AutoML_1_20250102_52110_model_45	0.125519	0.0157549	0.0576102	0.0908916	0.0157549
GBM_grid_1_AutoML_1_20250102_52110_model_34	0.126042	0.0158865	0.0481793	0.0912271	0.0158865
GBM_4_AutoML_1_20250102_52110	0.12608	0.0158961	0.0535871	0.0911706	0.0158961
XGBoost_grid_1_AutoML_1_20250102_52110_model_22	0.126348	0.0159639	0.0616614	0.0930222	0.0159639

Figure 6 Leaderboards generated by the H2O framework.

3 Proposed Framework

In this framework, we propose an optimized feature selection approach using the WOA and evaluate its performance using the RF classifier and H2O AutoML. The proposed method aims to enhance classification performance by selecting the most relevant features while reducing computational overhead. The goal of the optimization process is to select the best subset of features that maximizes classification accuracy. The objective function used in this study is defined as,

$$f(x) = -\frac{1}{k}\int_{i=1}^{k} Accuracy_i \tag{1}$$

Where 'x' is the binary feature selection vector, 'k' represents the number of cross-validation folds. '$Accuracy_i$'is the accuracy obtained on the ith fold. Given a binary vector x, the selected features are determined by:

$$selected\_features = \{j | x_j = 1,\ j \epsilon [1, d]\} \tag{2}$$

The total number of features in the dataset is represented by 'd'.

WOA mimics the social hunting behavior of humpback whales. It has two main strategies: encircling prey updates the position towards the leader, and

the bubble-net attacking mechanism. The whale 'X(t)' updates its position in the following manner:

$$X(t+1) = X^* - A \cdot (D) \tag{3}$$

Where 'X^*' is the position of the best solution found so far. 'A' is the coefficient vector calculated as $A = 2a \cdot r - a$, where 'a' linearly decreases from 2 to 0 over iterations, and 'r' is a random number in [0,1]. 'D' is the distance vector given by $D = |C \cdot X^* - X|$, where $C = 2r$ is a randomized factor.

The spiral update position is as follows:

$$X(t+1) = X^* + bl \cdot e^{cl} \cdot \cos(2\pi l) \tag{4}$$

The spiral shape is controlled by the constants 'b' and 'c', while 'l' represents a random number within the interval $[-1, 1]$. After selecting the best feature subset, we train an RF classifier as represented in Figure 7. The importance of each feature is extracted. Further to validate the efficiency of the selected features, we employ H2O AutoML, which automatically tests and selects the best-performing model from various algorithms.

4 Experimental Results & Discussions

The computer used for training and testing the model has an Nvidia, 8 GB of RAM, and an Intel Core i5 CPU operating at 1.19 GHz. Consequently, Python 3 is used to implement the suggested WOA and H2O framework to detect phishing attacks. A Google collaborator with the A100 GPU hardware accelerator is needed to execute multiple iterations.

(a) Evaluation of Feature Selection

The WOA was used to reduce the dimensions of the dataset while maintaining accuracy. The optimization algorithm was run for multiple iterations to identify the number of reduced features. As in Figure 8, the reduced feature count and corresponding accuracy were computed after setting the parameters of the WOA function. After identifying the optimal number of features for each iteration through WOA, the next step was to evaluate the performance of these reduced features. The objective of Figure 9 was to validate the efficacy of feature reduction by assessing how well the reduced dataset performed in regression tasks using an RF classifier. These two

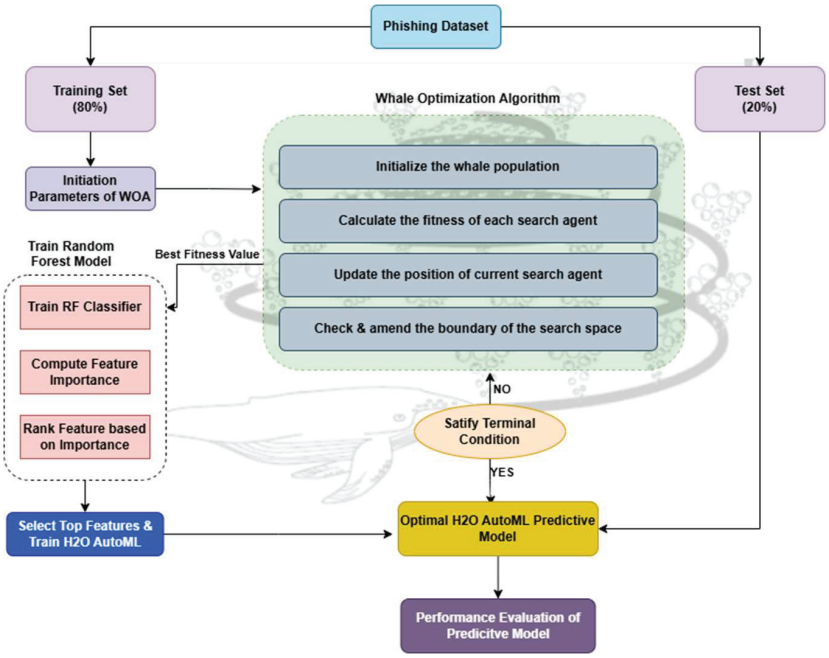

Figure 7 The proposed framework analysis of WOA-RF classifier.

measures, though similar in nature, measure different things: accuracy on reduced dataset measures the integrity of the resulting dataset structure after dimensionality reduction, and accuracy on reduced features measures how well a chosen set of features performs within a given prediction model. At iteration 70, for instance, the truncated dataset maintains a high structure-based accuracy, but the particular feature subset chosen at this point may not be the best to fit the learning pattern of the Random Forest model, leading to a decline in "Accuracy on Reduced Features". This deviation is an indication that feature selection should not only be assessed based on dataset quality but also considering model performance. Therefore, both of these metrics are essential in determining the best balanced and efficient iteration.

For each iteration, WOA reduced the original feature set to a subset containing only the most crucial features. Not only this, but we have also analyzed the scoring of each iteration by assigning weight factors for accuracy and the number of features as given in Equation (5).

$$Score = (w_1 \cdot Accuracy_{RD}) + (w_2 \cdot Accuracy_{RF}) - (w_3 \cdot Num\_Features) \tag{5}$$

Algorithm 1: Implementation of WOA in Feature Selection

Step 1: Initialize the whale population with binary values (0 or 1) representing the inclusion or exclusion of features.

Step 2: Define the maximum number of iterations, population size, and fitness function.

Step 3: Evaluate the fitness of each whale by training a classifier using the selected features and computing its performance.

Step 4: Identify the whale with the best fitness as the current global optimal solution.

Step 5: Update the positions of the whales using WOA's encircling, bubble-net, or search mechanisms based on adaptive coefficients.

Step 6: Perform encircling by adjusting the feature subset toward the best solution using a spiral or linear motion.

Step 7: Apply the bubble-net mechanism to refine the search by balancing exploration and exploitation dynamically.

Step 8: Introduce random search behavior to explore new feature combinations by updating positions randomly.

Step 9: Convert updated whale positions into a binary format using a threshold, e.g., sigmoid or step function.

Step 10: Recalculate the fitness of each whale and update the global optimum if a better solution is found.

Step 11: Return the best whale's binary vector as the optimal feature subset.

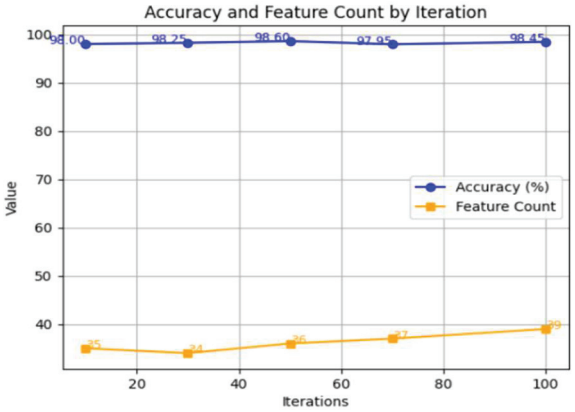

Figure 8 Accuracy on the number of reduced features at each iteration.

Where w_1, w_2, w_3 weights are defined based on the relative importance of each factor. This formula pools three essential factors accuracy on the reduced dataset, accuracy employing the Random Forest classifier, and number of chosen features, each given different weights to represent their significance, where increased accuracy enhances the score, and an increased number of

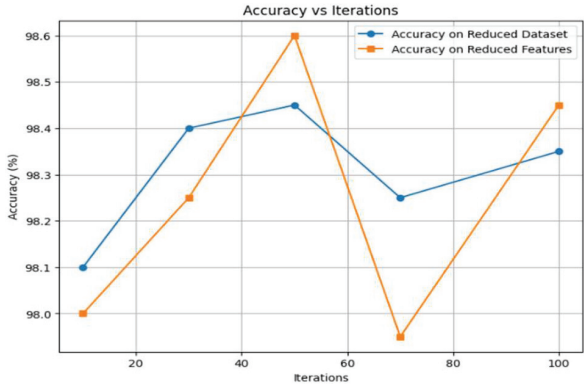

Figure 9 Accuracy on Reduced dataset vs reduced features.

	Iterations	Accuracy_RD	Accuracy_RF	Num_Features	Score
0	50	98.45	98.60	36	9816.5
1	100	98.35	98.45	39	9801.0
2	30	98.40	98.25	34	9798.5
3	70	98.25	97.95	37	9773.0
4	10	98.10	98.00	35	9770.0

Figure 10 Score value based on weights assigned.

features decreases it. The weights of $w_1 = w_2 = 50$ were chosen empirically following the reasoning that equal weight was given to both measures of accuracy through this choice, to signify their joint contribution to model reliability. A low weight of w_3=1 was given to the number of features to impose a mild penalty on excess complexity without overwhelming the effect of accuracy. As the iteration progressed, the reduced features and datasets were assessed, yielding varying accuracy scores. As seen in Figure 10, iteration 50 has the best overall score, not just because of its precision, but because it finds the best possible equilibrium between model precision and feature simplicity. The balanced compromise justifies iteration 50 as the most effective setup for AutoML usage for downstream model training. This analysis validates the hypothesis that reduced datasets were carefully selected.

After selecting the reduced dataset at iteration 50, the next step was to evaluate its performance compared to the original dataset. The reduced dataset retained a high accuracy of 98.45% identical to the original dataset's accuracy. The benchmarking results in Table 2 suggest that the idea of implementing an optimization technique on the Random Forest classifier for feature extraction has performed significantly better than other works by achieving an accuracy of 98.60% with 36 selected features.

Table 2 Performance benchmark between the proposed framework and existing work

Ref.	Classifier	Technique	No. of Features	Accuracy (%)
	Random Forest		30	94.27
[25]	FACA		30	92.40
[26]	Random Forest	HEFS	5	93.22
Proposed Method	Random Forest	WOA	36	98.60

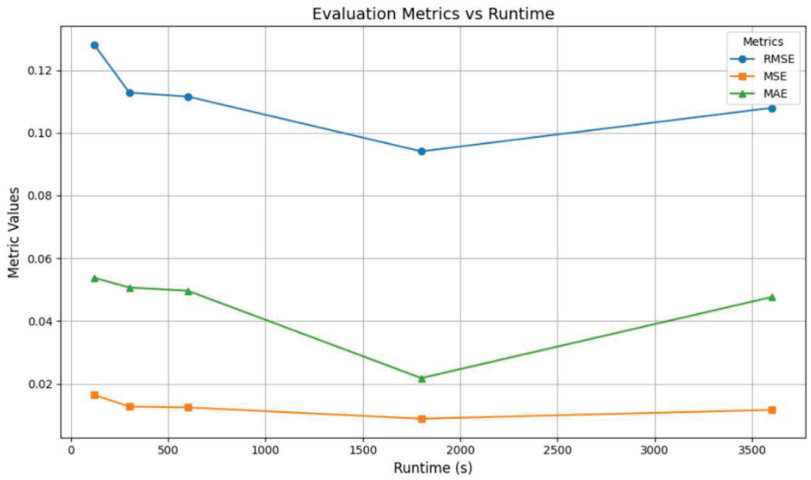

Figure 11 Evaluation metrics at different runtimes using AutoML.

(b) Model Performance

The H2O AutoML framework was employed to identify the best-performing ML model for reduced datasets. Experiments were conducted with runtime durations ranging from 120 sec to 3600 sec. Figure 11 provides a detailed evaluation of model performance at various runtime durations using three key metrics: RMSE, MSE, and MAE. The analysis reveals that longer runtimes led to improved performance across all metrics. For instance, at 120 sec the RMSE is at 0.1280, the MSE is at 0.0163, and the MAE values are at 0.0538, indicating a less precise model. As the runtime increases, the model sees a significant decrease in key metrics. At 3600 sec, the leaderboard indicated that SEM consistently outperformed other models with key metrics as shown in Table 3. The stacked ensemble model combines predictions from multiple base models, leveraging the strengths of diverse algorithms. The ensemble included 15 base models selected from a pool of 194 total models in this case. Among these, 8 GBM models, 6 XGBoost models, and 1 deep learning model were utilized. The meta-learner used a generalized linear

Table 3 Evaluation metrics at different runtime

Runtime (sec)	MSE	RMSE	MAE	RMSLE	Mean Residual Deviance	R^2	Residual Deviance	Akaike Information Criterion
120	0.0164	0.1280	0.0538	0.0905	0.0164	0.9344	32.7729	−2526.85
300	0.0127	0.1128	0.0507	0.0818	0.0127	0.9491	25.4520	−3016.47
600	0.0124	0.1116	0.0496	0.0807	0.0124	0.9502	24.8886	−3061.23
1800	0.0089	0.0941	0.0218	0.0660	0.0089	0.9522	24.1038	−3101.53
3600	0.0116	0.1079	0.0476	0.0788	0.0116	0.9534	23.2985	−3195.27

	Model Name	Coefficient Value
2	XGBoost_grid_1_AutoML_1_20250102_52110_model_14	0.254475
1	GBM_grid_1_AutoML_1_20250102_52110_model_53	0.148098
183	DeepLearning_grid_3_AutoML_1_20250102_52110_mo...	0.116113
6	GBM_grid_1_AutoML_1_20250102_52110_model_34	0.115692
4	GBM_grid_1_AutoML_1_20250102_52110_model_4	0.096220
..	...	...
74	GBM_grid_1_AutoML_1_20250102_52110_model_39	0.000000
75	XGBoost_grid_1_AutoML_1_20250102_52110_model_93	0.000000
76	XGBoost_grid_1_AutoML_1_20250102_52110_model_43	0.000000
77	XGBoost_grid_1_AutoML_1_20250102_52110_model_23	0.000000
0	Intercept	-0.018685

Figure 12 GLM meta learner coefficients to base models.

model (GLM) algorithm with a 5-fold cross-validation strategy to ensure robust performance. The GLM assigns coefficients to the base models based on their predictive performance and relevance, as shown in Figure 12.

XGBoost Model 14 with 0.254475 has the highest coefficient, indicating that it is the most influential base model in the ensemble. Models like GBM & XGBoost have zero coefficients, which means they are not contributing to the stacked ensemble's predictive accuracy. These values are indicative of good predictive performance of the model. The low RMSE and MSE show that there is little deviation between actual and predicted values, while the high R^2 value of 0.9534 shows that more than 95% of the variance is captured by the model. This validates the capability of the model to generalize well and attests to the strength of the proposed WOA-AutoML framework.

(c) Model Performance on Training and Cross-Validation Data

As in Figure 13, SEM exhibited near-perfect accuracy on training data, with an MSE of 0.0013, RMSE of 0.0365, and MAE of 0.0211. The R2 value of 0.9947 indicates that the model explains 99.5% of the variance in the

```
ModelMetricsRegressionGLM: stackedensemble
** Reported on train data. **

MSE: 0.001329665406616605
RMSE: 0.036464577422707166
MAE: 0.021099590590523708
RMSLE: 0.028560606107289745
Mean Residual Deviance: 0.001329665406616605
R^2: 0.9946812905051481
Null degrees of freedom: 7999
Residual degrees of freedom: 7984
Null deviance: 1999.981999999908
Residual deviance: 10.63732325293284
AIC: -30245.607007693172
```

Figure 13 SEM report on training data.

```
ModelMetricsRegressionGLM: stackedensemble
** Reported on cross-validation data. **

MSE: 0.013608384836739906
RMSE: 0.11665498204851735
MAE: 0.049282557528983105
RMSLE: 0.0844523007134859
Mean Residual Deviance: 0.013608384836739906
R^2: 0.9455659707467771
Null degrees of freedom: 7999
Residual degrees of freedom: 7984
Null deviance: 2000.4249364693624
Residual deviance: 108.86707869391925
AIC: -11639.536621563195
```

Figure 14 SEM report on cross-validation data.

training dataset. The high negative AIC (−30245.61) highlights the model's efficiency, balancing complexity and performance. In Figure 14, on cross-validation data, the metrics are slightly higher than the training metrics. The R2 value of 0.9456 indicates that the model explains approximately 94.5% of the variance in unseen data. The CV AIC of −11639.54 further supports the model's capability to generalize effectively.

(d) Statistical Validation of the Model Performance

A statistical significance test between the discussed WOA-based feature selection method and a baseline model that was trained on the entire feature set without reduction was performed. We compared the RMSE on 5-fold

Table 4 Paired t-test results comparing WOA – AutoML with the baseline model

Model	Mean RMSE	Standard Deviation	t-statistic	p-value
Baseline (All Features)	0.1324	0.0051	–	–
Proposed (WOA-AutoML)	0.1166	0.0042	5.41	0.0057

cross-validation through a paired t-test. As is evident from Table 4, the WOA-AutoML model generated a mean RMSE of 0.1166, while the baseline model had a greater mean RMSE of 0.1324. The paired t-test produced a t-statistic of 5.41 with a p-value of 0.0057, which is statistically significant at the 95% confidence interval ($p < 0.05$). This verifies that the gains in performance by our approach are not a result of random fluctuations and approves the efficacy of the suggested feature choice and model optimization process.

(e) Limitations & Trade-offs

WOA is a random algorithm that decreases dimensionality and increases precision, but whose performance can be different with every run because of initialization and convergence properties. AutoML H2O employs reduced model selection and tuning, which can be computationally costly and is perhaps not ideal for resource-constrained environments or real-time applications unless optimized. The data employed in this study is static, whereas phishing techniques are dynamic. Good learning on existing data comes from high R^2 values, but the generalizability of the model to newer or geographically targeted attacks needs to be established by testing on shifting datasets. The stacked ensemble offers high accuracy but higher inference time, which is a critical consideration for real-time use in applications such as browser add-ons or intrusion detection software. Whether the model will be applied in real-time is dependent on operational considerations. The chosen stacked ensemble under a 3600-second runtime yields the best accuracy but with increased model complexity. Trained models under lower runtimes give lower accuracy with fewer training and inference time. This indicates that an efficiency-focused runtime model can be selected for resource-constrained or latency-prone environments.

5 Conclusion

Phishing websites can be compromised by multiple factors such as suspicious domain names, mismatched URLs, unusual pop-ups and SSL certificates, phishing emails, and so on. It is required to identify the crucial parameters.

In this study, we have used WOA for dimensionality reduction and verified the accuracy of the reduced dataset. The number of selected features is also evaluated based on the importance score to determine the efficiency of the selected features. The AutoML H2O framework identifies the best-performing model at various runtimes. While longer runtimes improved the model's performance marginally, the results at shorter runtimes were competitive. The Stacked Ensemble Model outperformed individual base models in terms of R^2 and error metrics, such as an RMSE of 0.1166 on cross-validation. The inclusion of diverse base models, e.g., XGBoost, GBM, Deep Learning, etc., enabled the ensemble to capture complex patterns in the data. The meta-learner's coefficients revealed that XGBoost and GBM contributed the most to the ensemble's success, validating their reliability in handling structured data. The high R^2 values of 0.994 on training data and 0.945 on cross-validation data indicate that the ensemble can generalize well. The feature selection procedure can further be improved through the inclusion of hybrid metaheuristic techniques to enhance convergence and feature importance in high-dimensional data. Furthermore, real-time testing and hosting of the model on actual phishing data streams would serve to establish its real-world applicability. The present work does not account for hardware-level latency benchmarking, e.g., CPU/GPU inference time or throughput. Integration with light-weight ML frameworks or real-time scoring engines (e.g., TensorFlow Lite, ONNX) can potentially make deployment even more suitable.

Data Availability

The datasets generated or analyzed during the present study are available in the Mendeley repository, https://data.mendeley.com/datasets/h3cgnj8hft/1.

References

[1] M. Nanda, M. Saraswat, and P. K. Sharma, "Enhancing cybersecurity: A review and comparative analysis of convolutional neural network approaches for detecting URL-based phishing attacks," *e-Prime – Adv. Electr. Eng. Electron. Energy*, vol. 8, no. March, p. 100533, 2024, doi: 10.1016/j.prime.2024.100533.

[2] E. S. Shombot, G. Dusserre, R. Bestak, and N. B. Ahmed, "An application for predicting phishing attacks: A case of implementing a support

vector machine learning model," *Cyber Secur. Appl.*, vol. 2, no. January, 2024, doi: 10.1016/j.csa.2024.100036.

[3] V. Dixit and D. Kaur, "Development of Two-Factor Authentication to Mitigate Phishing Attack," no. July 2019, pp. 787–802, 2024, doi: 10.4236/jsea.2024.1711043.

[4] R. Basnet and A. H. Sung, "Rule-Based Phishing Attack Detection Rule-Based Phishing Attack Detection," no. October, 2016.

[5] S. Hossain, D. Sarma, and R. J. Chakma, "Machine learning-based phishing attack detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 9, pp. 378–388, 2020, doi: 10.14569/IJACSA.2020.0110945.

[6] P. E. Reports, P. S. Trends, B. P. Measurement, E. P. Attacks, M. Targeted, and I. Sectors, "Peter Cassidy, PHISHING ACTIVITY TRENDS REPOR, 2024," no. March, pp. 1–11, 2024.

[7] P. Kalaharsha and B. M. Mehtre, "Detecting Phishing Sites – An Overview," pp. 1–13, 2021, [Online]. Available: http://arxiv.org/abs/2103.12739.

[8] M. Dadkhah, M. D. Jazi, M. S. Mobarakeh, S. Shamshirband, X. Wang, and S. Raste, "An overview of phishing attacks and their detection techniques," *Int. J. Internet Protoc. Technol.*, vol. 9, no. 4, 2016, doi: 10.1504/IJIPT.2016.081319.

[9] S. Hawa Apandi, J. Sallim, and R. Mohd Sidek, "Types of anti-phishing solutions for phishing attack," in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/769/1/012072.

[10] G. J. W. Kathrine, P. M. Praise, A. A. Rose, and E. C. Kalaivani, "Variants of phishing attacks and their detection techniques," in *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, 2019. doi: 10.1109/ICOEI.2019.8862697.

[11] Z. Salah, H. Abu Owida, E. Abu Elsoud, E. Alhenawi, S. Abuowaida, and N. Alshdaifat, "An Effective Ensemble Approach for Preventing and Detecting Phishing Attacks in Textual Form," *Futur. Internet*, vol. 16, no. 11, pp. 1–24, 2024, doi: 10.3390/fi16110414.

[12] K. H. Chy, "Securing the web: Machine learning's role in predicting and preventing phishing attacks Securing the web: Machine learning's role in predicting and preventing phishing attacks," no. September, 2024, doi: 10.30574/ijsra.2024.13.1.1770.

[13] Jain, A. K., and Gupta, B. B. (2021). A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise*

Information Systems, *16*(4), 527–565. https://doi.org/10.1080/17517575.2021.1896786.

[14] M. Baykara and Z. Z. Gürel, "Detection of phishing attacks," *6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 – Proceeding*, vol. 2018-January, pp. 1–5, May 2018, doi: 10.1109/ISDFS.2018.8355389.

[15] "Web Phishing Detection Using Web Crawling, Cloud Infrastructure and Deep Learning Framework", *JASTT*, vol. 4, no. 01, pp. 54–71, Mar. 2023, doi: 10.38094/jastt401144.

[16] T. Peng, I. Harris, and Y. Sawa, "Detecting Phishing Attacks Using Natural Language Processing and Machine Learning," in *Proceedings – 12th IEEE International Conference on Semantic Computing, ICSC 2018*, 2018. doi: 10.1109/ICSC.2018.00056.

[17] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, Jan. 2021, doi: 10.1007/S11235-020-00733-2/TABLES/5.

[18] R. O. Akinyede and J. A. Adelakun, "Detection and Prevention of Phishing Attack Using Linkguard Algorithm," *J. Inf.*, vol. 4, no. 1, 2018, doi: 10.18488/journal.104.2018.41.10.23.

[19] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," in IEEE Access, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[20] Aljofey, A., Jiang, Q., Rasool, A. et al. An effective detection approach for phishing websites using URL and HTML features. *Sci Rep* **12**, 8842 (2022). https://doi.org/10.1038/s41598-022-10841-5.

[21] B. Espinoza, J. Simba, W. Fuertes, E. Benavides, R. Andrade, and T. Toulkeridis, "Phishing attack detection: A solution based on the typical machine learning modeling cycle," in *Proceedings – 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*, 2019. doi: 10.1109/CSCI49370.2019.00041.

[22] L. M. Abdulrahman, S. H. Ahmed, Z. N. Rashid, Y. S. Jghef, T. M. Ghazi, and U. H. Jader, "Web Phishing Detection Using Web Crawling, Cloud Infrastructure and Deep Learning Framework," *J. Appl. Sci. Technol. Trends*, vol. 4, no. 01, 2023, doi: 10.38094/jastt401144.

[23] Tan, Choon Lin (2018), "Phishing Dataset for Machine Learning: Feature Evaluation", Mendeley Data, V1, doi: 10.17632/h3cgnj8hft.1.

[24] H2O.ai, "H2O AutoML: Automatic Machine Learning," *H2O.ai*, 2024, [Online]. Available: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html.
[25] R. O. Akinyede and J. A. Adelakun, "Detection and Prevention of Phishing Attack Using Linkguard Algorithm," *J. Inf.*, vol. 4, no. 1, pp. 10–23, 2018, doi: 10.18488/journal.104.2018.41.10.23.
[26] K. L. Chiew, C. L. Tan, K. S. Wong, K. S. C. Yong, and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system," *Inf. Sci. (Ny).*, vol. 484, pp. 153–166, 2019, doi: 10.1016/j.ins.2019.01.064.

Biographies

Divya Singhal is working as an assistant professor in Noida Institute of Engineering and Technology. She has been in academics from last 8 years. She has completed her doctorate from Amity University in smart grid & information security. She has published various papers in reputed journals & conferences.

Ankit Verma is currently serving as an Associate Professor in the Department of Computer Applications at KIET Group of Institutions. He brings

with him over 18 years of experience in teaching and research. He regularly serves as a reviewer for reputed journals and international conferences and has been invited as a keynote speaker and session chair at several prestigious IEEE conferences. His research interests include artificial intelligence, IoT, fuzzy logic, and web technologies. He has strong technical expertise in programming languages and technologies. In December 2023, he successfully organized the Scopus-indexed International Conference on Recent Advancements in Computing Technologies & Engineering (RACTE-2023). He has also published books on web technologies. Dr. Ankit Verma is a seasoned faculty member with a focus on AI, IoT, fuzzy logic, and web technologies. He is deeply engaged in research, peer-reviewed publications, and academic leadership, such as organizing conferences and supervising student research.

Ganesh V. Radhakrishnan is a senior academic, interdisciplinary researcher, and policy consultant with over forty years of experience spanning academia, industry, and government. He holds a Ph.D. in Public Systems from IIM Ahmedabad and an MBA in Operations and Finance from IIM Kozhikode. His research spans economic regulation, infrastructure finance, maritime strategy, and the application of artificial intelligence in complex systems, including supply chains and digital public services.

Currently Senior Professor at KIIT University, Dr. Radhakrishnan has also served as Dean of Faculty Affairs at MIT World Peace University and Associate Dean at Jindal Global Business School. He has led the development of forward-looking academic programs in analytics, financial technology, and digital transformation. He has delivered lectures on emerging topics such as AI applications in logistics, predictive analytics, and digital governance at leading institutions across India and Europe.

Jyoti Parashar is currently working as a Professor in the Panipat Institute of Engineering & Technology college (AICTE-approved multidisciplinary institution, PIET is affiliated to Kurukshetra University) Haryana, India. With a strong academic background and expertise in computer science, she plays a pivotal role in shaping the knowledge and skills of engineering students.in the Delhi. She has done her Ph.D. in Computer Science from Maharishi Markedeshwar University, Ambala, Haryana with A++ Grade in India. She has research and teaching experience of more than 10 years. Her research interests span various domains, including artificial intelligence, data science, and emerging technologies in computing. Through her dedication to teaching and research, she actively contributes to the academic community by mentoring students, publishing research papers, and participating in conferences and workshops.

Saroj S. Date is an accomplished academician and researcher with over 18+ years of teaching experience in Computer Science & Engineering. Currently she is working as an Associate Professor in the Department of Artificial Intelligence and Data Science at CSMSS Chh. Shahu College of Engineering, Chh. Sambhajinagar. She has an extensive background in

Computer Engineering, holding a Ph.D. from Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar. She pursued Bachelor of Engg. from SGGS College of Engg. & Tech, Swami Ramanand Teerth Marathwada University, Nanded and Master of Engg. from Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar. Her expertise spans diverse subjects, including Machine Learning, Theory of Computation, Compiler Design, and Big Data Analytics. Dr. Date is proficient in programming languages such as Python, Java, C, C++. She has contributed significantly to research with publications on sentiment analysis, natural language processing, and machine learning, including Scopus-indexed journal articles, International journals/conferences and book chapters. Her main research work focuses on Sentiment Analysis, Natural Language Processing, Data Mining, Text Mining, Artificial Intelligence, Machine learning, Deep Learning, Mobile Computing, Big Data Analytics, etc.

Kamal Upreti is currently working as an Associate Professor in Department of Computer Science, CHRIST (Deemed to be University), Delhi NCR, Ghaziabad, India. He completed is B. Tech (Hons) Degree from UPTU, M. Tech (Gold Medalist), PGDM(Executive) from IMT Ghaziabad and PhD in Department of Computer Science by & Engineering. He has completed Postdoc from National Taipei University of Business, TAIWAN.

He has published 50+ Patents, 45+ Books, 32+ Magazine issues and 185+ Research papers in in various reputed Journals and international Conferences. His areas of Interest such Artificial Intelligence, Machine Learning, Data Analytics, Cyber Security, Machine Learning, Health Care, Embedded System and Cloud Computing. He has published more than 45+ authored and edited books under CRC Press, IGI Global, Oxford Press and Arihant Publication.

He is the main guest editor of more than 10 special issues of journals including Springer, Taylor and Francis, Inderscience, IGI Global, and Elsevier. He is the main guest associate editor in *Frontier Journal Convergence of Artificial Intelligence and Cognitive Systems which is SCIE and SCOPUS* having impact factor: 4.7 and cite score: 7.3. He is having enriched years' experience in corporate and teaching experience in Engineering Colleges.

He worked with HCL, NECHCL, Hindustan Times, Dehradun Institute of Technology and Delhi Institute of Advanced Studies, with more than 15+ years of enrich experience in research, Academics and Corporate.

Currently, he has completed work with Joint collaboration with GB PANT & AIIMS Delhi, under funded project of ICMR Scheme on Cardiovascular diseases prediction strokes using Machine Learning Techniques from year 2017–2020.

He got fund from DST SERB for conducting International Conference, ICSCPS-2024, 13–14 Sept 2024. Recently, he got fund from AICTE – Inter-Institutional Biomedical Innovations and Entrepreneurship Program (AICTE-IBIP) for 2024–2026. He has attended as a Session Chair Person in National, International conference and key note speaker in various platforms such as Skill based training, Corporate Trainer, Guest faculty and faculty development Programme. He awarded as best teacher, best researcher, extra academic performer and Gold Medalist in M.Tech programme.