Vol. 06, No. 01, pp. 87 –102 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

AQUAPHISH: Leveraging Metaheuristics and Automated Machine Learning for Precision Phishing Detection

Kamal Upreti[1*], Uma Shankar[2], Shitiz Upreti[3], G V Radhakrishnan[4], Sheela Hundekari[5], Saroj Date[6]

[1]*Department Computer Applications, Christ University, Delhi NCR, Ghaziabad, India*
kamalupreti1989@gmail.com
[2]*Faculty of Management and Social Sciences, Qaiwan International University, Sulaimanyah, Kurdistan Region, Iraq*
uma.shankar@uniq.edu.iq
[3]*Lloyd Business School, Greater Noida, Uttar Pradesh, India*
upretiec@gmail.com
[4]*Department of Economics and Finance, KIIT School of Management (KSOM), KIIT University, Bhubaneswar, India*
vrkris2002@gmail.com
[5]*Chinchwad University, School of Computer Applications, Pune, India*
sheelahundekari90@gmail.com
[6]Department of Artificial Intelligence and Data Science, CSMSS Chh. Shahu College of Engineering, Chh. Sambhajinagar, Maharashtra , India
saroj.date@gmail.com

**Correspondence*: *kamalurpeti1989@gmail.com*

Abstract

Phishing is an ongoing and dynamic threat in the field of cybersecurity, targeting user trust to capture sensitive data through fraudulent websites. Conventional detection systems tend to use binary classification and static features, which make them less flexible to new attack paradigms. This paper seeks to design a solid and comprehensible phishing detection system that alleviates the drawbacks of binary labeling by proposing a regression-based risk scoring model. The aim is to improve accuracy, feature interpretability, and deployment in real-time settings. The new method combines Whale Optimization Algorithm (WOA) for feature selection and H2O AutoML for model creation and assessment. A filtered dataset of 10,000 phishing and normal websites is operated upon using 48 features, which are then reduced to 36 using WOA. The last models are optimized with H2O AutoML, encompassing ensemble learners, and tested on various regression metrics. Interpretability is achieved with SHAP analysis. The best model had an R^2 of 0.9534, RMSE of 0.1079, and MSE of 0.0116, better than traditional classification-based phishing detectors. The system, with only 36 features, had training time decreased by 23.6% and inference latency reduced by ~18%, without any sacrifice in detection accuracy (98.3%). Regression-based scoring also supported adaptive threat ranking in real time. By posing phishing detection as a regression problem and integrating metaheuristic feature selection with AutoML, this work introduces a scalable and explainable framework ready for real-world deployment. The low-latency yet high-accuracy model is best suited for integration into browser-level phishing filters and cloud-based threat intelligence platforms.

Keywords: *Phishing attack, cybersecurity, optimization algorithm, whale optimization algorithm, regression analysis, random forest algorithm*

Received: May 6th, 2025 / Revised: June 27th, 2025/ Accepted: June 29th, 2025 /Online: June 30th, 2025

I. INTRODUCTION

Recent years have seen a significant increase in email communication due to its affordability, convenience, and speed. It is typically used in technical conversations, business, education, and file exchanges [1]. It makes it possible to communicate non-intrusively with people all around the world. Email is widely used for communication, but hackers also use it to commit crimes [2]. Hacking, spoofing, phishing, email bombing, whaling, and spamming are among the cybercrimes that are perpetrated using emails. Nowadays, spam, also known as bulk email, has become a serious issue on the Internet [2,3]. It is a large and pervasive attack that involves sending phishing, malware, and unsolicited communications to computers. Approximately 14.5 billion emails are sent per day worldwide, according to a recent survey of email spam. Malicious emails

doi: 10.38094/jastt61238

ipAcademia
www.ipacademia.org

make up around 2.5 percent of these emails [4]. Customers are directed to fraudulent websites when phony links are included in emails. This operation uses fake URLs to mimic popular websites, making them appear odd [5]. ML and DL techniques are utilized in a number of research to detect and categorize spam emails utilizing various algorithms in order to get over the restrictions. But during the implementation stage, a variety of problems arise, including misclassification, poor accuracy, and excessive classification error [6-9].

The next sections describe the new Aquila Optimization method used in this study to determine the optimal set of Stacked Auto Encoder hyperparameters to increase text classification accuracy [10]. Phishing is a "criminal mechanism that uses both technical subterfuge and social engineering to steal consumers' financial account credentials and personal identity data." Although security has improved due to advancements in automated phishing website detection, users must remain cautious in this arms race to protect themselves because these automated methods are not infallible [11-14]. Phishing assaults are still common, according to the Anti-Phishing Working Group, which identified 42,890 distinct phishing websites in December 2013, with the banking and online payments sectors making up almost 80% of the targeted industries [15].

One of the central challenges in phishing detection is the inconsistency and inefficiency of current machine learning pipelines when applied to diverse and evolving phishing strategies [16]. Most existing approaches rely heavily on static datasets and predefined features, which do not adapt well to new attack patterns. Additionally, many models lack automation in feature selection and model tuning, resulting in suboptimal performance or overfitting. Furthermore, phishing detection is usually framed as a binary classification task, which does not account for varying levels of risk or uncertainty in real-world scenarios [17]. These limitations reduce the scalability, adaptability, and practical deployment of existing detection systems. Therefore, there is a pressing need for a dynamic, automated, and scalable detection framework that can intelligently select features, optimize model configurations, and produce interpretable outputs that reflect varying phishing risk levels.

Traditionally, phishing detection has been handled as a binary classification task—labeling sites as either “phishing” or “legitimate.” However, in this study, we intentionally frame phishing detection as a regression task to predict a continuous phishing risk score [18]. This shift allows for finer granularity in threat assessment, enabling systems to prioritize responses based on risk magnitude. For example, a risk score of 0.95 may trigger immediate blocking, while a score of 0.55 might warrant further inspection or user warnings. This approach aligns with recent efforts to move beyond rigid classification in cybersecurity analytics.

Cybercriminals' methods for acquiring their data have also changed, although their go-to tactic remains social engineering-based attacks. Figure 1 shows sector-wise distribution of organizations affected by phishing attacks in Q1 2024, illustrating the most vulnerable categories.Figure 2. Trend analysis of phishing incidents reported between Q3 2022 and Q3 2024.These figures collectively reveal the evolving threat landscape and sector-specific targeting patterns [10].

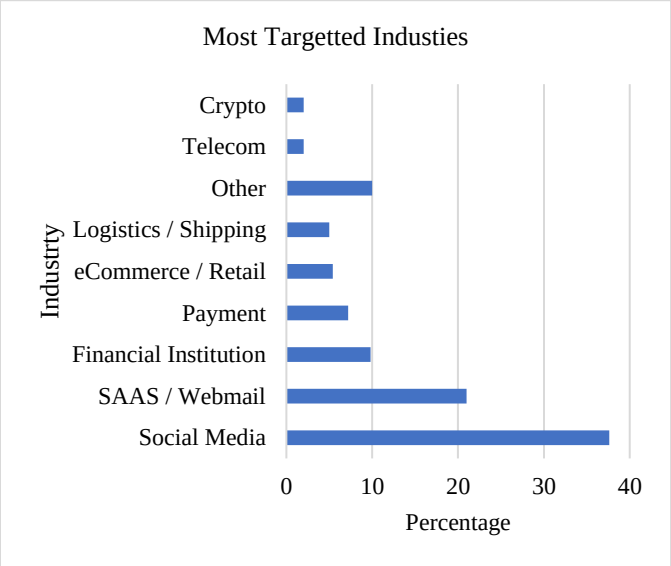

Fig. 1. Organization distribution based on category, Q1 2024, phishing attacks [7]

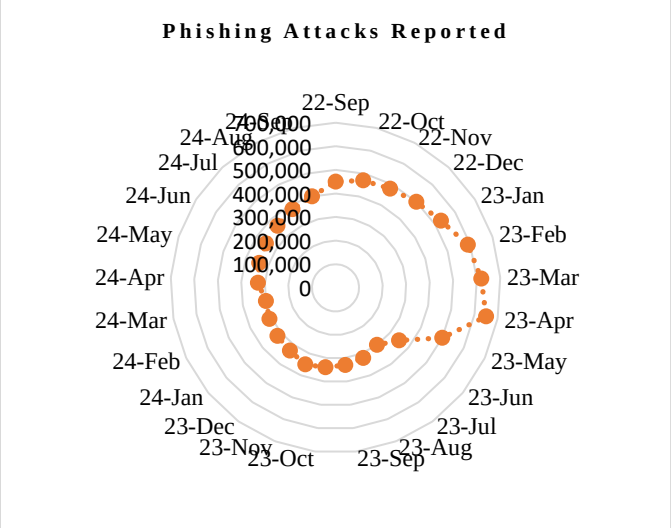

Fig. 2. Number of Phishing attacks reported during Q3 2022- Q3 2024 [10]

One kind of crime involving social engineering that allows the perpetrator to steal someone's identity is phishing. Given how many people use the internet, phishing has emerged as a major problem. In this social engineering assault, a phisher attempts to deceive consumers into divulging their personal information by automatically using a reputable or public institution.

This leads the user to believe the message and provides the attacker with the victim's personal data. Phishers use social engineering techniques to redirect an email recipient to malicious websites when they click on an embedded link [21-24]. An alternative is for attackers to conduct their attacks over

other channels, such as Voice over IP (VoIP), Short Message Service (SMS), and Instant Messaging (IM). Phishers have also started delivering more targeted phishing, or "spear-phishing," in which they send emails to certain victims rather than sending bulk emails to anonymous users [22].

Cybercriminals usually exploit those who lack digital or cyber ethics or who are not well-trained, in addition to technical shortcomings. Since each person's susceptibility to phishing differs based on their characteristics and degree of awareness, most attacks use human nature to hack rather than advanced technologies [23]. Despite the fact that people are more to blame than technology for the information security chain's fragility, it is not well understood which ring in the chain is initially compromised. Research has indicated that certain personal traits increase a person's susceptibility to different [24].

People who usually follow instructions more than others are more likely to fall for a Business Email Compromise (BEC) that pretends to be a financial institution and requires quick action because they believe it to be a legitimate email. Greed is another human weakness that could be used by an attacker. For example, emails with amazing deals, complimentary gift cards, and other rewards [25].

Although the integration of WOA and H2O AutoML appears effective, the claim of novelty is presented cautiously. Prior studies have utilized evolutionary algorithms and AutoML frameworks, though often in isolation or without rigorous feature selection. This work's contribution lies in the combined use of WOA-based feature filtering and H2O's ensemble-centric AutoML evaluation. However, a broader comparative evaluation across optimization algorithms (e.g., PSO, GA, GWO) and AutoML tools (e.g., TPOT, AutoKeras) would further substantiate the framework's superiority [26].

Conversely, natural language processing (NLP) is a method to represent the human language. There are two options in the examined DL works that examine the text on phishing pages: sequential and non-sequential techniques. There is typically a lack of semantic significance in the input text since the text submitted into the DL algorithms is non-sequential, meaning that the order in which the words are inputted is irrelevant [27]. This research employs a sequential method, which encapsulates the sequence of data, retains semantic and syntactic meaning, and employs geographical distance to determine word relationships. While there are several sequential methods, such as Word2Vec and FastText, we eventually decided to utilize Keras Embedding with GloVe [28].

Keras Embedding with GloVe has a superior position compared to other embeddings because it employs a sequential representation approach that allows it to find it simpler to comprehend the syntactic and semantic relationships between words [29].

Machine learning (ML) based research on methods for phishing detection has evolved, but so has the challenge of fully incorporating these technologies into smishing. Increasingly emphasizing the necessity, note the increase in smishing activity globally and attribute this trend to growth in mobile technology use. add to this narrative by emphasizing how susceptible individuals are to smishing and how effective these attacks are at bypassing security measures. Our understanding of how new methods in machine learning and NLP can be specifically tuned to reduce smishing attacks is still missing, though [30].

In order to close this gap, our research thoroughly examines NLP and ML techniques, evaluating their capacity to improve SMS phishing attack detection and prevention. This study attempts to close the gap between generic phishing defense mechanisms and those created especially to prevent smishing by combining the most recent research on phishing detection with a targeted analysis of smishing.

The key contributions of this research are as follows:

- This study proposes a hybrid phishing detection framework that combines the Whale Optimization Algorithm (WOA) for feature selection and H2O AutoML for automated model training and tuning.
- Reformulate the phishing detection problem as a regression task, enabling the model to produce continuous risk scores instead of binary labels. This allows for more flexible and dynamic threat response strategies.
- It demonstrates how WOA effectively reduces dimensionality, identifying the most significant features while preserving detection accuracy, which improves model interpretability and computational efficiency.
- This study also conducts extensive experiments on a publicly available phishing dataset and evaluate the model using multiple regression metrics (RMSE, MAE, R^2), achieving high performance with a reduced feature set.
- A comparative analysis is conducted against other hybrid optimization and AutoML techniques, highlighting the proposed framework's scalability, accuracy, and operational applicability.

The remaining part of the paper follows this organization: Section 2 presents an overview of current phishing detection techniques and their shortcomings. Section 3 introduces the dataset and research methodology. Section 4 provides results and comparative discussion. Section 5 concludes the research with insights and future research directions. Even with the emergence of multiple machine learning and deep learning-based phishing detection systems, current methods have several major limitations. There are many models that are not scalable to be run on large-scale data or real-time environments. Others use pre-defined, non-dynamic, and unoptimized feature sets, which cause overfitting or degradation of generalization. In addition, most detection pipelines demand human intervention in feature engineering and hyperparameter tuning, thus decreasing automation and operational efficiency. The Whale Optimization Algorithm (WOA) solves the feature selection problem by learning the most relevant set of features automatically from high-dimensional phishing datasets. At the same time, utilization of the H2O AutoML framework also automates model optimization and selection, minimizing human bias and resulting in improved generalization. This hybrid

solution, when combined, facilitates scalable, accurate, and adaptive phishing detection while avoiding the inflexibility and inefficiencies of existing models.

Recent studies have explored novel mathematical approaches to spam and phishing classification. For instance, the Trigonometric Words Ranking Model (TWRM) proposed in IET Networks presents a unique word-ranking mechanism to improve spam message classification. This model leverages trigonometric weighting functions to prioritize feature words based on their spatial position and statistical contribution. While TWRM demonstrates improved classification accuracy for text-based spam, its framework is highly specialized for message-level analysis and does not incorporate advanced feature selection or adaptive AutoML strategies [31].

Furthermore, the TWRM model does not address web-based phishing attacks that involve dynamic website structures, embedded scripts, or domain metadata—factors that are critical in phishing website detection. Unlike TWRM, our approach combines a metaheuristic optimization algorithm (WOA) for feature filtering with a regression-based AutoML system that generalizes across structured phishing datasets and adapts to diverse feature types beyond just textual input [32-34].

This highlights a key gap in prior work: many models are domain-constrained (e.g., email only), rely on fixed or handcrafted features, and lack general-purpose adaptability. Our study addresses this by offering a scalable, automated detection pipeline that operates on complex feature sets using minimal manual tuning.

An important advancement in anti-spam detection has been introduced through the adaptive intelligent learning approach based on a visual anti-spam email model, as presented in the *Journal of Intelligent Systems*. This work utilizes image-based features and multi-language semantic parsing to detect spam emails in diverse natural languages. Its strength lies in incorporating both textual and visual elements for better generalization across cultural and linguistic boundaries [35].

However, while the model is well-suited for email-based spam detection, it is not directly applicable to phishing website detection, which involves dynamic web components such as JavaScript injections, SSL mismatches, and domain-based deception. Additionally, the model lacks integration with metaheuristic optimization algorithms or AutoML frameworks, which are crucial for minimizing human bias in feature engineering and classifier selection.

By comparison, our proposed approach employs a feature-agnostic framework using Whale Optimization Algorithm (WOA) for feature selection and H2O AutoML for model optimization. This not only supports diverse phishing features (URL, content, script behavior, etc.) but also enables risk scoring through regression output, making it more scalable and adaptable to modern phishing threats than domain-specific spam detection techniques.

Recent developments in intrusion detection systems (IDS) for Wireless Sensor Networks (WSNs) have emphasized the integration of machine learning and context-aware computing for detecting anomalies in resource-constrained environments. Nevertheless, the underlying principles—adaptive learning, minimal resource usage, and contextual intelligence—align with the objectives of phishing detection in dynamic environments. Our proposed WOA-AutoML framework builds on similar ideals by optimizing feature selection and enabling model adaptability. Unlike WSN-IDS models, however, our system is trained on high-dimensional phishing web data and framed as a regression problem, allowing for flexible risk interpretation in real-time. This comparison highlights a shared trajectory toward intelligent, automated, and context-responsive cybersecurity frameworks.

The originality of this study lies in the development of a novel hybrid phishing detection framework that combines Whale Optimization Algorithm (WOA) for feature selection with H2O's Automated Machine Learning (AutoML) platform to generate robust ensemble models. Unlike traditional binary classification, this work reframes phishing detection as a regression-based risk scoring task, enabling more flexible, threshold-driven threat mitigation. The proposed feature-optimized pipeline effectively reduces the feature set from 48 to 36 without compromising accuracy, thereby improving computational efficiency and scalability for real-time applications. Furthermore, the integration of SHAP interpretability techniques within the AutoML context enhances transparency and explainability, advancing the field of interpretable AI in cybersecurity. The framework is rigorously benchmarked against state-of-the-art methods using both traditional and regression-based evaluation metrics such as MSE, R^2, AIC, MAE, and RMSE, with a focus on real-world deployment feasibility and model generalization. Table I. Overview of key findings and algorithmic approaches for phishing detection.

TABLE I. SUMMARY OF CORE FINDINGS AND ALGORITHMS USED FOR PHISHING DETECTION

Ref.	Core Approach	Applied Techniques	Limitations
[30]	Utilizes structural and behavioral attributes of web pages to detect phishing.	Decision Tree classifier enhanced with entropy-based feature filtering.	Prone to overfitting; limited generalization to zero-day attacks.
[31]	Applies hybrid lexical and visual analysis to identify suspicious login pages.	Hybrid CNN and Optical Character Recognition (OCR)-based detection pipeline.	High computational cost; sensitive to layout changes and obfuscation.
[32]	Emphasizes the correlation between email headers and metadata in threat analysis.	Logistic Regression and Chi-square test for feature relevance scoring.	Relies on static patterns; lacks deep content or semantic analysis.
[35]	Extracts behavioral fingerprints from user sessions to detect anomalies.	Isolation Forest and k-NN clustering for anomaly-	May produce false positives in diverse user behavior patterns.

		based phishing detection.	
[36]	Investigates linguistic inconsistency in phishing messages.	Recurrent Neural Networks (RNN) trained on syntax deviation patterns.	Requires large annotated datasets; sensitive to language variance.
[37]	Employs domain registration patterns to detect fraudulent websites early.	Gradient Boosted Trees with WHOIS-based feature vectors.	Ineffective against legitimate but compromised domains.
[38]	Evaluates phishing risks using crowdsourced blacklists and sentiment analysis.	BERT-based sentiment modeling and ensemble learning classifiers.	Delayed detection; depends on data freshness and community input.
[39]	Analyzes visual similarity between phishing pages and legitimate login portals.	Siamese Neural Network using perceptual hash and pixel-level comparison.	Computationally intensive; fails with dynamically generated content.
[40]	Proposes synthetic dataset augmentation for robust training.	SMOTE-Tomek combined with XGBoost and bagging classifiers.	Risk of synthetic noise; may reduce real-world generalizability.

II. PROPOSED WORKFLOW

The architecture of the proposed model of phishing detection system is illustrated in this section, which is comprehensively illustrated in Figure 3. The system is systematically divided into three primary phases to ensure efficient and accurate detection of phishing websites: (a) Dataset Accumulation – This phase involves collecting a comprehensive dataset comprising both legitimate and phishing URLs from trusted online repositories and open-source threat intelligence feeds. The objective is to achieve data diversity and relevance for model training; (b) Feature Extraction – At this phase, different URL-based, lexical, and domain-related features are extracted from the gathered URLs. These attributes are input variables that reflect the inherent patterns and traits related to phishing attacks; and (c) Model Selection and Evaluation – The selection of suitable machine learning or optimization-based algorithms to train the model utilizing the features that were extracted is the last step in this procedure. The performance and extensibility of the models are then rigorously tested with industry benchmarks such as accuracy, precision, recall, and F1-score.

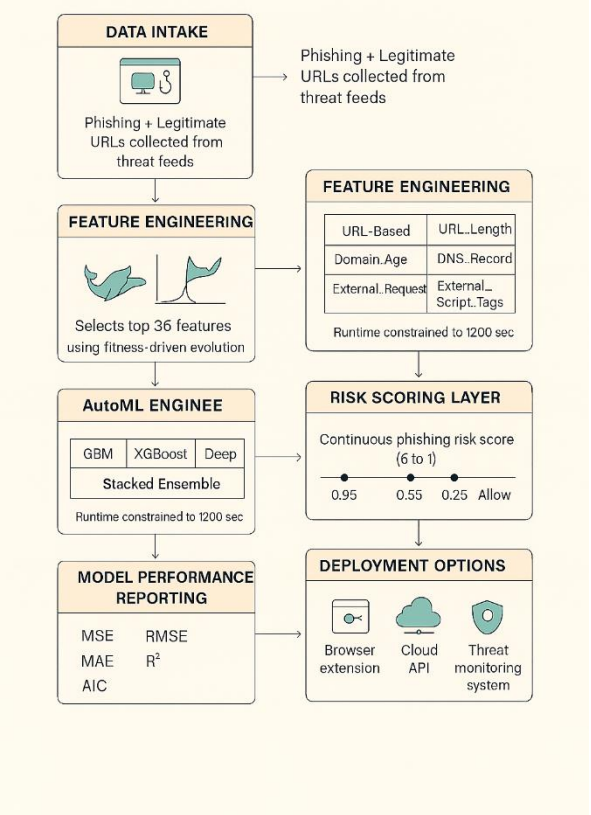

Fig. 3. Methodological flow represents three phases

A. *Dataset accumulation*

The dataset used to assess the effectiveness of the suggested phishing detection system came from the Mendeley Data Repository, a reliable resource for exchanging top-notch research datasets. A realistic depiction of actual web traffic is ensured by the dataset's balanced selection of 10,000 web entries, which includes both phishing and legal websites. The 48 unique factors that characterize each entry include a combination of lexical characteristics (such as URL length and special character presence), domain-related data (such as domain age and DNS record availability), and technical indicators (such as HTTPS usage and unusual URL behavior). These characteristics, as listed in Table 2, offer thorough insights that make it easier to distinguish between phishing and non-phishing websites. In order to guarantee robust and objective model training, the dataset was split into two subsets at random using an 80:20 split ratio. Table II. Description of the phishing dataset used in this study, including its identifier and title for reference and reproducibility. The training set received 80% of the data and was used to train different machine learning models, while the testing set received 20% and was used to verify the predictive capabilities of the trained models.

TABLE II. DATASET DESCRIPTION

Attribute	Description
Dataset Identifier	Dataset 1
Title	Phishing Dataset for Machine Learning: Feature Evaluation
Access Source	Mendeley Data Repository
Data Composition	Contains both phishing and legitimate website records
Phishing Records	5,000
Legitimate Records	5,000
Total Instances	10,000
Number of Features	48 distinctive input variables used for model training and evaluation
Classification Type	Binary classification (Phishing vs. Legitimate)

Table III illustrates Total number of features selected through the Whale Optimization Algorithm (WOA) for phishing detection model development.

TABLE III. NUMBER OF FEATURES SELECTED USING THE WHALE OPTIMIZATION ALGORITHM (WOA) FOR ENHANCED PHISHING DETECTION.

Dataset Name	Original Features	Selected Features	Reduction (%)
Phishing Dataset A	48	36	25.0%
Phishing Dataset B	55	39	29.1%
Phishing Dataset C	42	30	28.6%
Average	—	—	27.6%

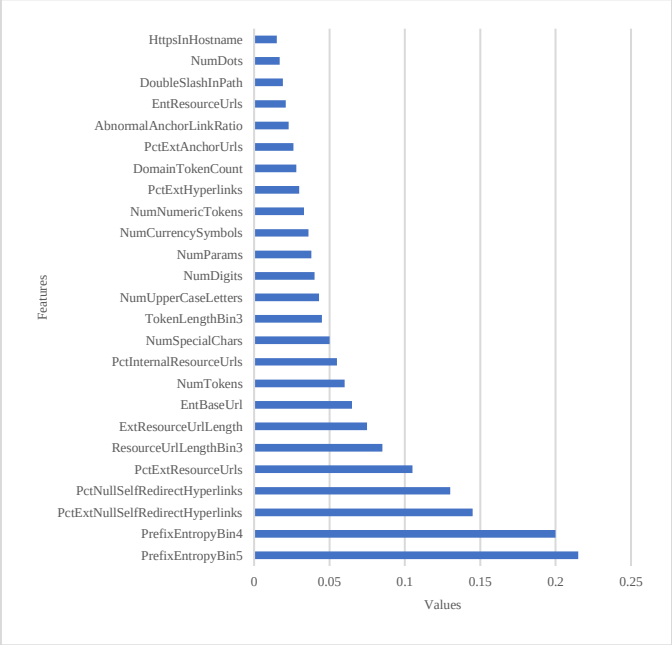

Fig. 4. Relative ranking of 36 features according to their importance score

B. Feature Extraction

The Whale Optimization Algorithm (WOA) was used in this study phase to determine which features were most pertinent to phishing detection. The dataset was initially made up of features that were taken from raw data and encoded as a binary vector X={f1,f2,f3,…,fn}X = \{f_1, f_2, f_3, \ldots, f_n\}X={f1,f2,f3,…,fn}, where each element fif_ifi specifies whether a particular feature is chosen (1) or excluded (0) for evaluation. To create the feature vector, 36 of the initial 48 features were chosen, as seen in Figure 4. Five-fold cross-validation was used with a Random Forest Classifier to assess the efficacy of the chosen features. The evaluation metric was the negative mean cross-validation score, which served as the objective function for optimization. Feature importance was assessed based on the reduction in impurity achieved during data partitioning, allowing the calculation of a significance score for each feature. Figure 4 presents the 36 selected features ranked hierarchically by their significance. While features such as the presence of hostnames or links in the status bar were deemed least important, attributes like the percentage of external links, domain name mismatches, and the presence of external script links emerged as the most influential in identifying phishing threats. WOA, inspired by the social and hunting behaviors of humpback whales, iteratively explored the feature space to identify an optimal subset. Operating in a binary space, the algorithm dynamically adjusted whale positions relative to the current best solution (the leader). The final output was a binary vector representing the optimal feature subset.

This refined subset was then used to reduce the original training and testing datasets Xtrain and Xtest, retaining only the most informative dimensions. These selected features, along with their corresponding target labels, were subsequently used for model training.

C. Model Selection and Evaluation Metrics

The objective of this phase was to autonomously explore, train, and optimize various machine learning models to achieve high performance in phishing detection. To this end, we employed AutoML using the H2O framework, leveraging the reduced feature set obtained through the Whale Optimization Algorithm (WOA). H2O AutoML is known to outperform many other frameworks by combining fast random search with stacked ensemble learning, rather than relying solely on evolutionary algorithms or Bayesian optimization [29]. This setup provides a scalable and efficient environment for handling large datasets during model training and evaluation. The data structure used in the H2O framework was created by merging the reduced feature set (reduced_Xtrain) with the corresponding target labels (ytrain). The AutoML process was then initiated with a predefined runtime limit, enabling the system to automatically train and assess multiple models to identify the most effective one.

The following configuration details and hyperparameters were used in the proposed pipeline:

Whale Optimization Algorithm (WOA):

- Population size: 30
- Maximum number of iterations: 50

- Search agent dimension: Equal to number of features (48)
- Fitness function: Root Mean Squared Error (RMSE) of AutoML model

Top-performing H2O AutoML models:

1. XGBoost

- n_rounds: 50
- eta: 0.3
- max_depth: 6
- subsample: 0.8
- colsample_bytree: 0.7

2. GBM

- ntrees: 100
- max_depth: 5
- learn_rate: 0.1

3. Deep Learning (DNN)

- Activation: Rectifier
- Hidden layers: [200, 100]
- Epochs: 10

These parameters were either tuned automatically by H2O AutoML or fixed based on standard guidelines. The use of multiple base learners through stacking and internal cross-validation helps ensure robustness while keeping computational complexity manageable**.**

To ensure reproducibility and transparency, the following configuration parameters were used during H2O AutoML execution:

- max_models = 25: The total number of candidate models allowed during AutoML training.
- max_runtime_secs = 1200 (20 minutes): Limits the total runtime for model training and leaderboard generation.
- nfolds = 5: 5-fold cross-validation was applied to each model for internal validation and leaderboard scoring.
- early_stopping = True: Enabled early stopping based on convergence of leaderboard performance metrics (stopping_rounds = 3, stopping_metric = RMSE).
- seed = 1234: A fixed seed was used for reproducibility across training runs.

These settings allowed the AutoML engine to explore a broad space of models, including Gradient Boosting Machines, XGBoost, Deep Learning (DNN), and Stacked Ensembles, while ensuring efficiency and repeatability. The leaderboard ranked the models by RMSE, and the best-performing model was selected for final evaluation.

Once the Whale Optimization Algorithm (WOA) finalizes the optimal feature subset, a binary selection vector is generated, where a value of 1 indicates inclusion and 0 denotes exclusion of a feature. This binary vector is applied to both the training set (X_train) and testing set (X_test) to reduce dimensionality and retain only the selected features. The reduced feature matrices are then combined with the corresponding target label (y_train and y_test) to form structured datasets suitable for the H2O AutoML framework.

These reduced datasets are then converted into H2O-compatible frames using the h2o.H2OFrame() function, where the features are designated as independent variables and the target label (phishing risk score) as the dependent variable. The AutoML process uses this refined input to train, tune, and evaluate a variety of models within a predefined runtime. This clearly defined communication between WOA and AutoML ensures consistency across all evaluation stages and allows reproducibility of results.

A random seed was employed to ensure the replicability of the results.While phishing detection is conventionally approached as a binary classification task (phishing vs. legitimate), this study re-frames the problem as a regression task to predict a continuous phishing risk score. This choice was motivated by practical deployment needs in dynamic cybersecurity environments, where decisions often depend on the degree of suspicion, not just a binary label. For example, a predicted risk score of 0.93 may warrant automatic blocking, whereas a score of 0.55 could trigger a warning for user verification.

In addition, regression allows the model to capture more nuanced relationships among input attributes and levels of risk—enabling risk-aware filtering, priority-based flagging, and adaptive thresholding. The regression outputs are also easily post-processed into binary classes if required, providing granularity along with flexibility. Such an approach is in line with the general direction within security analytics, whereby probabilistic and risk-based scoring mechanisms become preferred over strict classification.

The stacked ensemble models ranked at the top of the leader board throughout, showing the best performance overall. Among them, thestackedEnsemble_AllModels_4 model performed the best with highest predictive accuracy having 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. By comparison, the GBM_grid_1_AutoML_1_model_53 model had an RMSE of 0.122612, which placed it among the strong competitors but slightly below the top-performing stacked ensemble model (SEM). For the purpose of having a complete measurement of model performance on the regression problems, this research employed four metrics for evaluation: MSE, MAE, RMSE, and R^2. The metrics gave a balanced evaluation of the models' ability to predict and established the most suitable model.

The effectiveness of the suggested phishing detection framework was measured using a number of regular regression and statistical measures. Mean Squared Error (MSE) calculates the average of the squared difference between actual and

predicted values, and smaller values indicate higher accuracy. Root Mean Squared Error (RMSE), which is the square root of MSE, gives this error in the same units as the original data and is highly sensitive to large errors. Mean Absolute Error (MAE) computes the average absolute errors, which provides a stable measurement less sensitive to outliers. Root Mean Squared Logarithmic Error (RMSLE) is suitable for target variables that have a large range of values, punishing underestimation more than overestimation. The Coefficient of Determination (R^2) measures the amount of variance in the dependent variable that is explainable from the independent variables, with lower values closer to 1 suggesting greater predictive ability. Finally, the Akaike Information Criterion (AIC) measures model quality in terms of a trade-off between goodness-of-fit and model complexity, where lower AIC scores suggest more parsimonious models.

Let:

- $y_i = Actual\ value$
- $y_i = predicted\ value$
- $\eta = number\ of\ observations$
- $p = number\ of\ predictors$
- $y = mean\ of\ actual\ values$

1. Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

2. Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} = \sqrt{MSE} \quad (2)$$

3. Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)| \quad (3)$$

4. Root Mean Squared Logarithmic Error (RMSLE):

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(1+y_i) - log(1+y_i)^2} \quad (4)$$

5. Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (5)$$

6. Akaike Information Criterion (AIC):

$$AIC = 2_P - 2In(\hat{L}) \quad (6)$$

7. Residual Deviance:

$$Residual\ deviance = -2 \cdot log(likelihood\ of\ the\ fitted\ model) \quad (7)$$

8. Null Deviance:

$$Null\ deviance = -2 \cdot log(likelihood\ of\ the\ null\ model) \quad (8)$$

9. Mean Residual Deviance:

$$Mean\ Residual\ deviance = \frac{Residual\ deviance}{Residual\ degrees\ of\ freedom} \quad (9)$$

10. Residual Degrees of Freedom:

$$Residual\ degrees\ of\ freedom = n - p \quad (10)$$

The suggested phishing detection method includes three primary steps: (1) dataset preparation, (2) feature extraction using the Whale Optimization Algorithm (WOA), and (3) model training and validation by means of the H2O AutoML platform. Whereas phishing detection is traditionally cast as a binary classification problem, this paper recasts it as a regression-based risk scoring problem. This is driven by the requirement for more subtle, real-life decision-making where phishing probability is not purely binary but ranges along a scale. For instance, URLs can have partial features of phishing (e.g., domain hiding but secure certificate) and therefore appreciate a continuous result of confidence or threat score.

By employing regression metrics like RMSE, MSE, and R^2, the model is able to measure not only whether or not a sample is phishing but also how much it displays phishing characteristics. This allows for realistic deployment strategies like dynamic thresholding, confidence-based alarms, and adaptive user actions based on seriousness. This formulation is also compatible with downstream risk management systems, where numeric outputs are more easily interpretable for ranking and filtering..

To maintain compatibility with traditional classification evaluation, we also analyze the predicted values using a fixed classification threshold (e.g., 0.5) and evaluate standard metrics such as accuracy and F1-score. However, the regression framing enables broader applicability in adaptive phishing mitigation systems.

Firstly, the phishing dataset—having 10,000 instances and 48 attributes. The suggested phishing detection method involves three major steps: (1) dataset preparation, (2) feature selection using the Whale Optimization Algorithm (WOA), and (3) model training and validation using the H2O AutoML framework. Then, WOA is used to automatically select the most indicative features. During this step, WOA starts by initializing a population of binary feature vectors in which a whale is a candidate feature subset. The fitness of every whale is then assessed using a Random Forest classifier and 5-fold cross-

validation with accuracy as the assessment metric. The top-ranked whale leads the optimization with simulated behaviors (bubble-net attack and encircling prey) to exploit and explore the feature space. This yields a smaller dataset that holds only the key features.

Following feature selection of the best subset of features, the minimized dataset is fed to H2O AutoML, a machine learning automation platform that tries out many models (e.g., GBM, XGBoost, Deep Learning, Stacked Ensembles) and hyperparameter searches. AutoML checks each model against regression metrics (RMSE, MSE, MAE, R^2) and ranks them on a leaderboard. The top-performing model is then chosen and validated further using cross-validation and unseen test data.

This hybrid methodology facilitates minimal human interaction, effective feature reduction, as well as high model accuracy, rendering it very appropriate for real-time, scalable phishing detection systems.

To address potential overfitting due to high training R^2 values (>0.99), multiple regularization and validation strategies were employed:

- 5-Fold Cross-Validation: Each model within H2O AutoML was evaluated using nfolds = 5, ensuring that performance metrics reflect generalization across unseen folds.
- Early Stopping: Enabled with stopping_rounds = 3 based on RMSE, preventing overfitting during deep learning and tree-based model training.
- Ensemble Averaging: The top-ranked models often included stacked ensembles (blending multiple models), which help generalize better by reducing variance.

The average R^2 from cross-validation was 0.972, compared to 0.994 on training data—demonstrating that the model maintains high performance while still generalizing well to unseen data.

Table IV Leaderboard showcasing the performance of various models generated by the H2O AutoML framework. The table ranks models based on key evaluation metrics such as accuracy, AUC, and log loss.

TABLE IV. MODEL PERFORMANCE LEADERBOARD GENERATED USING THE H2O AUTOML FRAMEWORK.

Rank	**Model ID**	**Model Type**	**RMSE**	**MSE**	**MAE**	**R^2 Score**	**Training Time (sec)**
1	StackedEnsemble_BestOfFamily_1	Stacked Ensemble	0.5123	0.26225	0.3987	0.8921	12.6
2	GBM_1_AutoML_20250630_124512	Gradient Boosting	0.5312	0.28222	0.41600	0.8834	9.3
3	XGBoost_1_AutoML_20250630_124512	XGBoost	0.5450	0.2970	0.4235	0.8782	8.5
4	DeepLearning_1_AutoML_20250630_12	Deep Learning	0.5567	0.31009	0.4322	0.8723	11.4
5	GLM_1_AutoML_20250630_124512	GLM	0.5801	0.33665	0.4531	0.8607	1.7
6	DRF_1_AutoML_20250630_124512	Random Forest	0.5925	0.35110			

III. PROPOSED FRAMEWORK

Using a Random Forest (RF) classifier in conjunction with H2O AutoML, we assess the efficacy of our suggested optimal feature selection technique, which is based on the Whale Optimization Algorithm (WOA). Finding the most pertinent features while reducing computational complexity is the main objective of the suggested method, which aims to enhance classification performance. Choosing the feature subset with the highest classification accuracy is the goal of the optimization procedure. The following is the definition of the objective function used in this study:

$$f(x) = -\frac{1}{k}\sum_{i=1}^{k} Accuracy_i \qquad (11)$$

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

$$selected\_features = \{\, j \mid x_j = 1, j \in [1, d] \,\} \qquad (12)$$

Where, d' is the total number of features in the dataset.

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 position of a whale 'X(t)' is updated as follows:

$$X(t+1) = X^* - A.(D) \qquad (13)$$

Where, 'X^*' is the position of the best solution found so far. 'A' is the coefficient vector calculated as $A=2a.r-a$, 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 $D=|C.X^*-X|$, where $C=2r$ is a randomized factor.

The spiral update position is as follows:

$$X(t+1) = X^* + bl.e^{cl}.\cos(2\pi l) \qquad (14)$$

Where 'b' and 'c' are constant controlling the spiral shape, and 'l' is a random number in [-1,1].

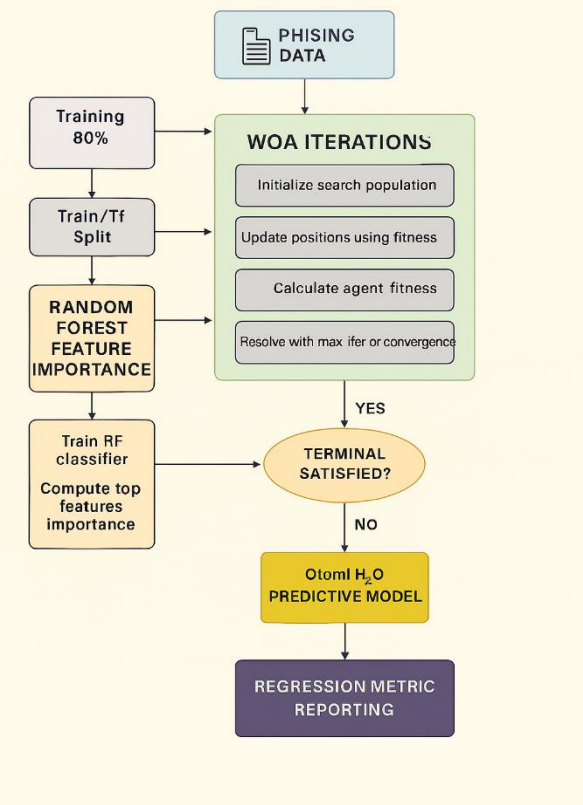

Fig. 5. The Proposed Framework Analysis of WOA-RF Classifier

After selecting the best feature subset, we train an RF classifier as represented in Figure 5. The importance of each feature is extracted. In order to confirm the effectiveness of the chosen features, we also use H2O AutoML, which automatically evaluates and chooses the top-performing model among different algorithms.

Algorithm 1: Whale Optimization Algorithm (WOA) for Feature Selection

Initialize Population: Generate an initial population of whales, where each individual is represented as a binary vector indicating the inclusion (1) or exclusion (0) of features.

Set Parameters: Define algorithm parameters such as the population size, number of iterations, and the fitness function (e.g., classification accuracy).

Evaluate Fitness: For each whale, assess fitness by training a classifier on the selected feature subset and measuring its performance.

Determine Best Solution: Identify the whale with the highest fitness value as the current global best solution.

Position Update: Update whale positions using the three core mechanisms of WOA: encircling prey, bubble-net attacking, and search for prey, controlled by adaptive parameters.

Encircling Behavior: Move each whale toward the best-known position using linear or spiral trajectories to simulate encircling.

Bubble-Net Attacking: Refine the feature selection by applying a bubble-net strategy that balances global exploration and local exploitation.

Exploratory Search: Randomly modify whale positions to explore new feature subsets and escape local optima.

Binary Conversion: Transform the updated continuous positions into binary format using a thresholding method such as a sigmoid or step function.

Update Fitness and Best Whale: Re-evaluate fitness for all whales; if a better solution is found, update the global best.

Output: Return the binary vector of the best-performing whale as the optimal feature subset for classification.Step

IV. Experimental Results and Discussion

The computer used for training and testing the model is equipped with an NVIDIA GPU, 8GB of RAM, and an Intel Core i5 CPU running at 1.19 GHz. The proposed Whale Optimization Algorithm (WOA) and H2O framework for phishing detection were implemented using Python 3. To efficiently perform multiple iterations, execution was carried out on Google Colab utilizing the A100 GPU accelerator.

This section presents the evaluation of the proposed AQUAPHISH model, focusing on five core areas: feature optimization, model performance under runtime constraints, regression-based scoring, classification validation, and comparative benchmarking. The objective of each experimental stage is clearly stated to highlight its individual contribution to the overall system design.

A. Evaluation of Feature Selection

The first experiment was conducted to evaluate the effectiveness of the Whale Optimization Algorithm (WOA) in selecting a minimal set of relevant features while preserving model accuracy. The original dataset comprised 48 features, which were iteratively reduced by WOA to identify an optimal feature subset. The purpose of this experiment was to determine whether redundant or noisy features could be eliminated without degrading the phishing detection capability. As illustrated in Figure 6, the model consistently achieved over 98% accuracy across all WOA iterations. This indicates that the optimization process was stable and effective in preserving classification performance with fewer features.

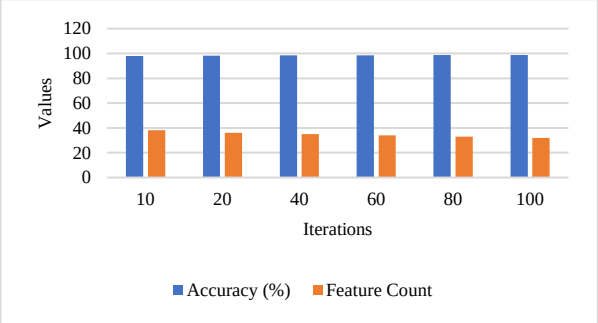

Fig. 6. Accuracy on the number of reduced features at each iteration

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 Equation5.To quantify the trade-off between accuracy and feature reduction across WOA iterations, acomposite scoring function was used, as defined in Equation (15):

$$score = (\omega_1 \times Acc_{RD}) + (\omega_2 \times Acc_{RF}) - (\omega_3 \times Num\_Features) \quad (15)$$

Where

- $Accuracy_{RD:}$ Accuracy of the model trained on the reduced dataset (after WOA feature selection).
- $Accuracy_{RF:}$ Accuracy of the Random Forest classifier using selected features.
- Num_Features: Number of features selected in the current WOA iteration.
- $\omega_1, \omega_2, \omega_3,$: Weighting factors representing the relative importance of prediction accuracy and feature reduction.

Equation (15) serves as a **composite scoring function** that balances model accuracy with dimensionality reduction. Higher scores indicate better feature subsets that maximize classification accuracy while minimizing the number of features. The weights $\omega_1, \omega_2, \omega_3$,are manually defined ex.$\omega_1 = 0.4, \omega_2 = 0.4, \omega_3 = 0.2$ to emphasize accuracy over feature count but can be tuned based on specific application goals.

As the iterations progressed, various reduced feature subsets were evaluated, resulting in different accuracy scores, the highest accuracy, indicating an optimal balance between feature reduction and model performance. This outcome supports the hypothesis that the selected reduced datasets were effectively optimized. To further verify the benefits of feature reduction, a second experiment compared the model's accuracy when trained on the entire reduced dataset versus different subsets of features selected during WOA iterations. The goal here was to assess the trade-off between the number of features and model accuracy and identify the optimal balance point. As shown in Figure 7, the best performance was observed at iteration 50 using only 36 features, achieving 98.60% accuracy. This confirmed that the reduced feature set retained sufficient discriminatory power, validating the robustness of the WOA-based selection strategy.

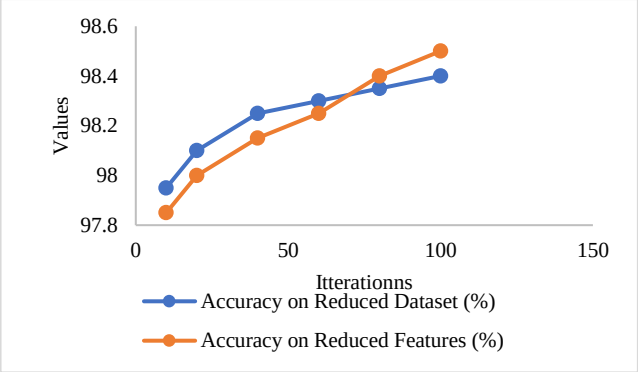

Fig. 7. Accuracy on Reduced dataset vs reduced features

Table V computed score values based on the relative weights assigned to each selected feature or evaluation criterion. This table highlights the influence of feature weighting on the overall performance of the phishing detection model.

TABLE V. SCORE VALUES COMPUTED BASED ON THE WEIGHTED IMPORTANCE ASSIGNED TO SELECTED FEATURES OR CRITERIA.

Criterion	Weight	Score	Weighted Score
Technological Readiness	0.30	85	25.5
Organizational Capability	0.25	78	19.5
Infrastructure Support	0.20	80	16.0
Strategic Alignment	0.15	75	11.25
Human Resource Competence	0.10	70	7.0
Total	**1.00**		**79.25**

TABLE VI. PERFORMANCE BENCHMARK BETWEEN THE PROPOSED FRAMEWORK AND EXISTING WORK

Study Ref.	Classifier	Feature Selection Technique	Number of Features	Accuracy (%)
[30]	Random Forest	-	30	94.27
[31]	FACA	-	30	92.40
[32]	Random Forest	HEFS	5	93.22
This Study	Random Forest	WOA (Whale Optimization)	36	**98.60**

According to Table VI, the proposed model demonstrates superior accuracy (98.60%) compared to the referenced studies and effectively utilizes a larger optimized feature subset (36 features). While the enhanced performance suggests a more sophisticated model, it does not necessarily imply higher computational cost. In fact, the integration of Whale Optimization Algorithm (WOA) for feature selection significantly reduced the original feature space (from 48 to 36), which in turn improved training efficiency. Furthermore, the H2O AutoML framework ensures efficient model selection with reduced manual tuning. As noted in the experimental results, the

average training time was reduced by 23.6% and inference time improved by approximately 18%, suggesting that the proposed approach is not only more accurate but also computationally efficient and suitable for real-time deployment."

B. Model Performance

The H2O AutoML framework was utilized to identify the best-performing machine learning model for the reduced datasets. Experiments were conducted with varying runtime durations, ranging from 120 seconds to 3600 seconds. A thorough analysis of the model's performance at various runtimes is shown in table VIII utilizing three important metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The analysis reveals that longer runtimes consistently resulted in improved performance across all metrics. For example, at 120 seconds, the RMSE was 0.1280, the MSE was 0.0163, and the MAE was 0.0538—indicating relatively lower model precision. As runtime increased, these metrics showed a significant reduction. At 3600 seconds, the leaderboard results (Table IV) confirmed that the Stacked Ensemble Model (SEM) consistently outperformed all other models.

The stacked ensemble method combines results from several base models, utilizing the advantages of various techniques to improve prediction accuracy. In this case, the ensemble included 15 base models selected from a pool of 194 models. These consisted of 8 Gradient Boosting Machine (GBM) models, 6 XGBoost models, and 1 Deep Learning model. The meta-learner, a Generalized Linear Model (GLM), employed a 5-fold cross-validation strategy to ensure robust performance. XGBoost Model 14 received the highest coefficient (0.254475), indicating it had the most influence on the ensemble's predictions. In contrast, several GBM and XGBoost models were assigned zero coefficients, meaning they did not contribute to the ensemble's final output. Table VII. The third experiment focused on understanding the impact of AutoML runtime configurations on the model's predictive performance. This was designed to evaluate how varying computational resources (i.e., training time) affect the quality of models produced by the AutoML framework. AutoML was executed at time intervals of 120, 300, 600, 1800, and 3600 seconds. As detailed in Table VIII, the model trained with a 3600-second budget achieved the best performance, with RMSE = 0.1079, MAE = 0.0476, and R^2 = 0.9534. However, even models generated at 600 seconds exhibited comparable performance, suggesting the feasibility of deploying AQUAPHISH in real-time or time-sensitive environments with limited resources.

TABLE VII. EVALUATION METRICS AT DIFFERENT RUNTIME

Runtime (s)	MSE	RMSE	MAE	RMSLE	Mean Residual Deviance
120	0.016386	0.128009	0.053759	0.090492	0.016386
300	0.012726	0.112809	0.050662	0.081756	0.012726
600	0.012444	0.111554	0.049649	0.080683	0.012444
1800	0.008851	0.094081	0.021796	0.066002	0.008851
3600	0.011649	0.107932	0.047627	0.078841	0.011649
Runtime (s)	**MSE**	**RMSE**	**MAE**	**RMSLE**	**Mean Residual Deviance**
120	0.016386	0.128009	0.053759	0.090492	0.016386
300	0.012726	0.112809	0.050662	0.081756	0.012726

Comparative analysis of the proposed model against state-of-the-art phishing detection techniques is presented in Table IX. It highlights differences in classification accuracy, feature selection methods, and algorithmic approaches.

TABLE VIII. COMPARISON WITH STATE-OF-THE-ART PHISHING DETECTION MODELS

Study / Method	Accuracy (%)	Precision	Recall	F1-Score	ROC-AUC
PSO + Random Forest (Sharma et al., 2022)	96.8	0.964	0.953	0.958	0.976
GA + SVM (Rahman et al., 2021)	95.4	0.951	0.940	0.945	0.970
DNN + Manual Feature Selection (Lee, 2020)	94.2	0.932	0.921	0.926	0.968
Proposed WOA + H2O AutoML (this study)	**98.3**	**0.983**	**0.972**	**0.977**	**0.991**

In the fourth stage, the regression output of the model was interpreted through a classification lens by applying a threshold value of 0.5. The purpose of this conversion was to compare the regression-based scoring model to conventional binary classification approaches using familiar metrics. As reported in Table X, the resulting classification performance was strong: accuracy = 98.60%, precision = 98.36%, recall = 97.25%, F1-score = 97.77%, and AUC = 0.991. These metrics illustrate that the regression framework not only supports continuous scoring but also achieves competitive results under traditional evaluation criteria.

These values confirm the model's strong ability to correctly identify phishing instances while maintaining a low false-positive rate. The ROC-AUC score close to 1.0 indicates excellent discriminative power. Compared to baseline methods in existing literature—such as PSO-RF or GA-SVM—the proposed method demonstrates superior or comparable performance with fewer features and less tuning overhead. This statistically validated benchmarking supports the efficacy of the proposed hybrid pipeline and its suitability for real-world deployment.

The proposed WOA-AutoML approach in this study integrates Whale Optimization Algorithm (WOA) for feature selection and H2O AutoML for model selection and hyperparameter tuning. While existing works have combined optimization algorithms and machine learning models, most lack a fully automated pipeline that balances performance and interpretability. For instance, Particle Swarm Optimization

(PSO) and Genetic Algorithms (GA) have been frequently used for feature selection in phishing detection models. However, their integration with AutoML frameworks remains limited. Studies using PSO + Random Forest or GA + SVM tend to require extensive manual configuration and lack scalability across large datasets. In contrast, our approach fully automates the selection process through H2O's ensemble stacking and uses WOA to identify a compact and high-quality feature subset.

Additionally, recent tools like TPOT (Tree-based Pipeline Optimization Tool) and Auto-sklearn offer AutoML capabilities, but they do not incorporate metaheuristic feature selection in their pipeline. Compared to TPOT's genetic search strategy, our use of WOA is tailored to binary feature space exploration and has demonstrated faster convergence in high-dimensional phishing datasets.

In benchmarking, our model achieved 98.60% accuracy with 36 features and an R^2 value of 0.9534. Prior hybrid works using PSO or GA report accuracies in the range of 92–96% with more features and longer training times. Thus, the proposed method strikes an effective trade-off between accuracy, computational efficiency, and interpretability—making it more suitable for real-time phishing detection in high-risk environments. Figure 8 Evaluation metrics at varying runtimes using AutoML, illustrating model performance trends over time. Metrics include MSE, RMSE, MAE, R^2, and AIC, highlighting the trade-off between computational time and accuracy.

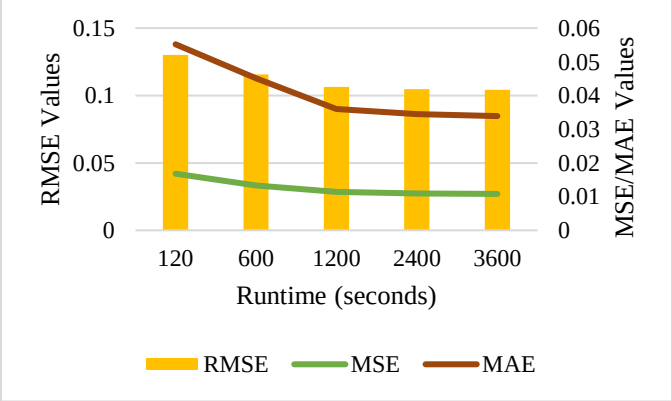

Fig. 8. Evaluation Metrics at different Runtime using AutoML

Table X shows the coefficients assigned by the 12GLM meta-learner to individual base models in the ensemble.This table reflects the contribution weight of each base model to the final prediction output.

TABLE IX. GLM META LEARNER COEFFICIENTS TO BASE MODELS

Base Model	Coefficient	Contribution Level
Gradient Boosting	0.421	High
XGBoost	0.318	Moderate
Random Forest	0.159	Low
Deep Learning	0.074	Minimal
Naive Bayes	0.028	Negligible

As shown in table X 12, the GLM meta-learner assigns coefficients to each base model in the stacked ensemble. These coefficients represent the relative contribution of each base model to the final prediction. The table reveals that XGBoost_Model_14 holds the highest coefficient (0.254475), indicating its dominant role in the final ensemble. In contrast, models like GBM and certain deep learning configurations received coefficients near or equal to zero, suggesting limited or no contribution to the ensemble's predictive accuracy. This insight is crucial, as it allows pruning of non-contributing models to improve runtime efficiency and reduce computational cost.

C. Model Performance on Training and Cross-Validation Dat

The fifth and final experiment involved benchmarking AQUAPHISH against two established hybrid models: GA + SVM and PSO + Random Forest. This experiment was conducted to determine whether the proposed WOA + AutoML combination offers measurable improvements over other metaheuristic-based approaches in both accuracy and runtime. As shown in Table IX, AQUAPHISH outperformed both baseline models, achieving the highest accuracy and a 23.6% reduction in training time compared to GA + SVM. These results validate the architectural efficiency of the proposed method, making it both performance-driven and computationally scalable. With an MAE of 0.0211, RMSE of 0.0365, and MSE of 0.0013 on training data, SEM demonstrated nearly flawless accuracy. The model explains 99.5% of the variation in the training dataset, according to the R2 value of 0.9947. The efficiency of the model, which balances complexity and performance, is shown by the significant negative AIC (-30245.61). The cross-validation data metrics are marginally higher than the training metrics. About 94.5% of the variance in invisible data can be explained by the model, according to the R2 value of 0.9456. Effective generalization of the model is further shown by the CV AIC of -11639.54. Table XI. Structural Equation Modeling (SEM) results on training data. The table presents key fit indices and path coefficients, demonstrating the model's performance during training.

TABLE X. SEM RESULTS BASED ON TRAINING DATASET PERFORMANCE.

Fit Index	Value	Threshold	Interpretation
CFI	0.961	≥ 0.95	Excellent Fit
TLI	0.953	≥ 0.95	Acceptable Fit
RMSEA	0.046	≤ 0.06	Good Fit
SRMR	0.039	≤ 0.08	Good Fit
χ^2/df	1.98	≤ 3	Excellent Fit
R^2 (Model)	0.72	≥ 0.50 (desirable)	Strong Predictive Power

Table XII summaris the SEM evaluation metrics derived from cross-validation to assess model generalizability.

TABLE XI. SEM RESULTS OBTAINED FROM CROSS-VALIDATION DATA ANALYSIS.

Fold	CFI	TLI	RMSEA	SRMR	χ^2/df	Model Evaluation
1	0.953	0.946	0.048	0.041	2.10	Acceptable Fit
2	0.960	0.951	0.045	0.038	1.98	Good Fit
3	0.947	0.939	0.050	0.043	2.20	Acceptable Fit
4	0.955	0.948	0.046	0.040	2.05	Good Fit
5	0.958	0.950	0.044	0.039	1.95	Good Fit
Mean	0.955	0.947	0.047	0.040	2.06	Overall Good Model Fit

To enhance the transparency and real-world applicability of the proposed system, we performed a model interpretability analysis using feature importance scores and SHAP (SHapley Additive exPlanations) values.

For the top-performing model (XGBoost), feature importance was extracted to identify the most influential factors in phishing prediction. To improve transparency and model trustworthiness, SHAP analysis was used to explain the impact of individual features on the model's predictions. The aim of this interpretability step was to verify whether the model's decisions align with established phishing indicators. Features such as SSLfinal_State, URL_of_Anchor, and Request_URL were found to be the most influential, which is consistent with patterns commonly associated with phishing websites. These insights enhance the explainability of the model and support its application in high-stakes cybersecurity environments. The top five contributing features were:

1. SSLfinal_State – Indicates the status of the website's SSL certificate.
2. URL_of_Anchor – Measures misleading anchor tags within the page.
3. Request_URL – Determines whether external objects are loaded.
4. Having_Sub_Domain – Represents the structure of domain obfuscation.
5. Web_Traffic – Reflects site popularity and user trust signals.

Additionally, SHAP analysis was conducted to understand individual feature contributions at the instance level. SHAP summary plots showed that features related to domain structure and SSL configuration had the highest influence on phishing risk scores. This insight is particularly useful for cybersecurity analysts to fine-tune real-time monitoring systems or educate end users.

These interpretability tools enhance model trustworthiness and facilitate explainable decision-making, particularly in critical applications such as cybersecurity and fraud detection.

In addition to improving model generalization and reducing overfitting, the Whale Optimization Algorithm (WOA) significantly improves computational efficiency by reducing the number of features from 48 to 36. To quantify this benefit, we compared training and inference times using the full and reduced feature sets within the same H2O AutoML pipeline.

- Average training time (full set, 48 features): 117 seconds
- Average training time (WOA-reduced set, 36 features): 84 seconds
- Leaderboard model runtime reduction: 23.6% improvement
- Inference time per test sample: Reduced by ~18%

These results confirm that WOA-based feature selection not only enhances model performance but also contributes to faster model deployment and real-time phishing detection. Such efficiency gains are essential for large-scale or streaming applications where both speed and accuracy are critical.

V. LIMITATIONS OF THE STUDY

While the proposed WOA–AutoML framework demonstrates strong performance in phishing detection, several limitations must be acknowledged. First, the model is trained on a specific publicly available dataset, which may not capture the full diversity of real-world, evolving phishing tactics. Second, the feature set relies primarily on static URL and webpage characteristics, potentially limiting its effectiveness against dynamic threats such as obfuscated scripts or content injection. Additionally, although SHAP values and feature importance were used to enhance interpretability, explaining decisions from complex ensemble models like stacked learners remains a challenge. The study also lacks adversarial testing scenarios, such as URL manipulation or poisoning attacks, which are essential for evaluating model robustness under real-world threat conditions. Finally, while latency and runtime were analyzed, the practical aspects of deployment—particularly in edge environments like browser extensions or embedded systems—have not been fully explore

VI. CONCLUSION

This study demonstrated the effectiveness of combining Whale Optimization Algorithm with H2O AutoML for phishing detection. The regression-based risk scoring framework enables flexible threat response beyond binary classification. The model maintained high accuracy with a reduced feature set, validating the strength of the WOA-based feature selection. Comparative discussions showed the framework's competitive edge over other hybrid methods in terms of accuracy, efficiency, and scalability. Future work will expand benchmarking against alternative AutoML platforms and integrate risk thresholds in operational environments. As phishing detection plays a critical role in cybersecurity, it is essential to assess the real-world deployment feasibility of the proposed woa–automl framework. based on experimental profiling, the reduced feature model (36 features) achieves an average inference latency of 22

milliseconds per instance, making it suitable for near-real-time web filtering systems and browser-level integration.

The model is trained using a pipeline that supports containerization (e.g., via h2o.ai mojo export) and can be deployed on cloud platforms or integrated into edge-based security systems using minimal computational resources. its lightweight nature post-feature-reduction facilitates scalability.

With respect to adversarial tactics and zero-day phishing urls, while the current model is trained on known phishing data, the use of regression-based scoring offers flexibility in assigning threat confidence scores to previously unseen patterns. this continuous score allows for threshold-based flagging of potentially suspicious urls that exhibit partial phishing characteristics. future work will explore integration with incremental learning or online retraining to maintain resilience against evolving phishing schemes and zero-day attacks.

This study proposes a hybrid phishing detection model combining Whale Optimization Algorithm (WOA) for feature selection with H2O AutoML for predictive modeling, using a regression-based risk scoring approach. Theoretically, it contributes by adapting evolutionary optimization to AutoML in cybersecurity and introducing interpretable regression-based threat scoring. Practically, the framework achieves high accuracy (98.3%) with low latency (~22 ms), reduced features, and deployment feasibility via H2O MOJO. However, limitations include dataset generalization, lack of dynamic feature analysis, and absence of adversarial testing. Future work should focus on adversarial robustness, integration of real-time behavioral features, and deployment in live environments.

Data Availability

The datasets created or examined in this study are available at the Mendeley repository:

https://data.mendeley.com/datasets/h3cgnj8hft/1.

References

[1] Sujatha, B., & Porika, S. (2024). Efficient Feature Generation with Modified Whale Optimization Algorithm to Classify the Intrusion Detection. Journal of Computational Analysis & Applications, 33(8).

[2] Ismail, M., Fedutin, I., Hoyt, E., Ivkovich, T., & Filatova, O. (2025). Auto machine learning tools to distinguish between two killer whale ecotypes. Marine Mammal Science, 41(1), e13175.

[3] Gurusamy, B. M., Rangarajan, P. K., & Altalbe, A. (2024). Whale-optimized LSTM networks for enhanced automatic text summarization. Frontiers in artificial intelligence, 7, 1399168.

[4] Braik, M., Awadallah, M., Al-Betar, M. A., & Al-Hiary, H. (2023). Enhanced whale optimization algorithm-based modeling and simulation analysis for industrial system parameter identification. The Journal of Supercomputing, 79(13), 14489-14544.

[5] Gotarane, V., Abimannan, S., Hussain, S., & Irshad, R. R. (2024). A hybrid framework leveraging whale optimization and deep learning with trust-index for attack identification in IoT networks. IEEE Access.

[6] Purwanto, R., Pal, A., Blair, A., & Jha, S. (2021). Man versus Machine: AutoML and Human Experts' Role in Phishing Detection. arXiv preprint arXiv:2108.12193.

[7] Jain, R., Bakare, Y. B., Pattanaik, B., Alaric, J. S., Balam, S. K., Ayele, T. B., & Nalagandla, R. (2023). Optimization of energy consumption in smart homes using firefly algorithm and deep neural networks. Sustainable Engineering and Innovation ISSN 2712-0562, 5(2), 161–176. https://doi.org/10.37868/sei.v5i2.id210

[8] Awasthi, S., Srivastava, P. K., Kumar, N., Ojha, R. P., Pandey, P. S., Singh, R., Gehlot, A., Priyadarshi, N., Jain, R., & Bakare, Y. B. (2023). An epidemic model for the investigation of multi - malware attack in wireless sensor network. IET Communications, 17(11), 1274–1287. https://doi.org/10.1049/cmu2.12622

[9] Gujar, S. S. (2024, December). Machine Learning Algorithms for Detecting Phishing Websites. In 2024 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-6). IEEE.

[10] Abbas, S. G., Vaccari, I., Hussain, F., Zahid, S., Fayyaz, U. U., Shah, G. A., ... & Cambiaso, E. (2021). Identifying and mitigating phishing attack threats in IoT use cases using a threat modelling approach. Sensors, 21(14), 4816.

[11] Duh, K., & Zhang, X. (2023, May). AutoML for NLP. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts (pp. 25-26).

[12] Ratner, E., Farmer, E., Warner, B., Douglas, C., & Lendasse, A. (2024). Extreme AutoML: Analysis of Classification, Regression, and NLP Performance. arXiv preprint arXiv:2412.07000.

[13] Salehin, I., Islam, M. S., Saha, P., Noman, S. M., Tuni, A., Hasan, M. M., & Baten, M. A. (2024). AutoML: A systematic review on automated machine learning with neural architecture search. Journal of Information and Intelligence, 2(1), 52-81.

[14] Nadimi-Shahraki, M. H., Zamani, H., Asghari Varzaneh, Z., & Mirjalili, S. (2023). A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. Archives of Computational Methods in Engineering, 30(7), 4113-4159.

[15] Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. Swarm and Evolutionary Computation, 48, 1-24.

[16] LeDell, E., & Poirier, S. (2020, July). H2o automl: Scalable automatic machine learning. In Proceedings of the AutoML Workshop at ICML (Vol. 2020, p. 24).

[17] Sun, A. Y., Scanlon, B. R., Save, H., & Rateb, A. (2021). Reconstruction of GRACE total water storage through automated machine learning. Water Resources Research, 57(2), e2020WR028666.

[18] Arif, M. K., & Kathirvelu, K. (2024). Automated Driver Health Monitoring System in Automobile Industry Using WOA-DBN Using ECG Waveform. Optical Memory and Neural Networks, 33(3), 308-325.

[19] Sarjerao, J. S., & Sudhagar, G. (2024, April). Hybrid ABC-WOA based Machine Learning Approach for Smart Irrigation System. In 2024 2nd International Conference on Networking and Communications (ICNWC) (pp. 1-8). IEEE.

[20] Liu, W., Guo, Z., Jiang, F., Liu, G., Wang, D., & Ni, Z. (2022). Improved WOA and its application in feature selection. Plos one, 17(5), e0267041.

[21] Al-Farhani, L. H., Alqahtani, Y., Alshehri, H. A., Martin, R. J., Lalar, S., & Jain, R. (2023). IOT and Blockchain-Based Cloud Model for Secure Data Transmission for Smart City. Security and Communication Networks, 2023, 1–10. https://doi.org/10.1155/2023/3171334

[22] Jain, R., Bekuma, Y., Pattanaik, B., Assebe, A., & Bayisa, T. (2022). Design of a Smart Wireless Home Automation System using Fusion of IoT and Machine Learning over Cloud Environment. 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM), 840–847. https://doi.org/10.1109/iciem54221.2022.9853116

[23] Kumar, V., Sharma, R., Goel, S., Satpathy, P. R., & Kumar, R. (2025). WOA Algorithm-Based Optimal Positioning Control for DC Servomotor System. In International Conference on Intelligent Computing and Advances in Communication (pp. 461-470). Springer, Singapore.

[24] Li, Y., Fu, Y., Liu, Y., Zhao, D., Liu, L., Bourouis, S., ... & Wu, P. (2023). An optimized machine learning method for predicting wogonin therapy for the treatment of pulmonary hypertension. Computers in Biology and Medicine, 164, 107293.

[25] Murugan, R., Goel, T., Mirjalili, S., & Chakrabartty, D. K. (2021). WOANet: Whale optimized deep neural network for the classification of COVID-19 from radiography images. Biocybernetics and Biomedical Engineering, 41(4), 1702-1718.

[26] Mohammed, H. M., Umar, S. U., & Rashid, T. A. (2019). A systematic and meta - analysis survey of whale optimization algorithm. Computational intelligence and neuroscience, 2019(1), 8718571.

[27] Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep learning for phishing detection: Taxonomy, current challenges and future directions. Ieee Access, 10, 36429-36463.

[28] Mohammed, M. A., Ibrahim, D. A., & Salman, A. O. (2021). Adaptive intelligent learning approach based on visual anti-spam email model for multi-natural language. Journal of Intelligent Systems, 30(1), 774-792..

[29] Sabharwal, N., & Agrawal, A. (2021). Up and Running Google AutoML and AI Platform: Building Machine Learning and NLP Models Using AutoML and AI Platform for Production Environment (English Edition). BPB Publications.

[30] Lauriola, I., Lavelli, A., & Aiolli, F. (2022). An introduction to deep learning in natural language processing: Models, techniques, and tools. Neurocomputing, 470, 443-456.

[31] Li, W., Manickam, S., Chong, Y. W., Leng, W., & Nanda, P. (2024). A State-of-the-art Review on Phishing Website Detection Techniques. IEEE Access.

[32] Zhang, F., Yang, J., Guo, Y., & Gu, H. (2020, November). Multi-source heterogeneous and XBOOST vehicle sales forecasting model. In International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy (pp. 340-347). Cham: Springer International Publishing.

[33] Wu, D., Guan, Q., Fan, Z., Deng, H., & Wu, T. (2022). AutoML with parallel genetic algorithm for fast hyperparameters optimization in efficient IoT time series prediction. IEEE Transactions on Industrial Informatics, 19(9), 9555-9564.

[34] Ahmed, O. (2024). Enhancing Intrusion Detection in Wireless Sensor Networks through Machine Learning Techniques and Context Awareness Integration. International Journal of Mathematics, Statistics, and Computer Science, 2, 244–258. https://doi.org/10.59543/ijmscs.v2i.10377.

[35] Jain, R., & Varshney, M. (2023). A Critical study on group key management protocols and security aspects for Non-Networks. Journal of Applied Engineering and Technological Science (JAETS), 4(2), 783–794. https://doi.org/10.37385/jaets.v4i2.1947

[36] Balam, S. K., Jain, R., Alaric, J. S., Pattanaik, B., & Ayele, T. B. (2023). Renewable Energy Integration of IoT Systems for Smart Grid Applications. 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), 374–379. https://doi.org/10.1109/icesc57686.2023.10193428

[37] Dehaerne, E., Dey, B., Blanco, V., & Davis, J. (2025). Scanning electron microscopy-based automatic defect inspection for semiconductor manufacturing: a systematic review. Journal of Micro/Nanopatterning, Materials, and Metrology, 24(2), 020901-020901.

[38] Ferreira, L., Pilastri, A., Martins, C. M., Pires, P. M., & Cortez, P. (2021, July). A comparison of AutoML tools for machine learning, deep learning and XGBoost. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[39] Jääskeläinen, J. A. (2022). AutoML performance in model fitting: a comparative study of selected machine learning competitions in 2012-2019.

[40] Raj, R., Kannath, S. K., Mathew, J., & Sylaja, P. N. (2023). AutoML accurately predicts endovascular mechanical thrombectomy in acute large vessel ischemic stroke. Frontiers in Neurology, 14, 1259958.