International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Enhancing Cybersecurity Through AI: A Transparent, High-Performance System for Detecting and Preventing Phishing-Based Intrusions

Abd Alkareem Mostafa Jejawi
Master's in Web Science (MWS),
Syrian Virtual University, Syria.

Dr. Sira Astour
ORCID: 0000-0003-3976-9258
Ph.D., Eng. Faculty of Information and Communication Engineering,
Arab International University (AIU), Syria

Abstract - Phishing attacks represent one of the most common and damaging forms of cybercrime currently occurring. Phishing involves complex fraudulent schemes aimed at obtaining personal data such as passwords and banking details by either directly attacking the user's computer system through the process of illegal access or tricking the victim into providing such data. With new approaches constantly emerging, current techniques such as blacklists and other static-based methods become less effective and, thus, should be supplemented with new technologies involving artificial intelligence. The objective of this research paper is to develop a system capable of detecting phishing websites with the use of machine learning techniques. The proposed methodology will include the application of experimentally-based techniques. To start with, data gathering from public resources and its preparation will be performed. In the following stage, the effectiveness of multiple models will be tested and compared to determine the best approach to detecting phishing websites. Three models will be designed including a natural language processing-based Bidirectional encoder representations from transformers (BERT) model, convolutional neural network with an attention mechanism model (Convolution Neural Network (CNN) + Attention) and the Random Forest classifier. It appears from the results obtained that the BERT model performed better compared to other models due to its high accuracy and recall rate; however, it possessed a very low level of false positives. It is noteworthy that the CNN+Attention model provides a compromise in terms of accuracy rate and effectiveness of the classifying procedure. Additionally, SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), which are considered interpretable models, have been used in order to reveal what aspects were used by the model to conclude about a certain message being a phishing one. Based on the above outcomes, it can be inferred that using deep learning algorithms along with Natural Language Processing (NLP) techniques is a successful method for detecting phishing attacks. The importance of interpretation and implementation in order to increase efficiency of intelligent systems in reality is also highlighted here.

1. INTRODUCTION:

The emergence of technology in the online world, alongside the dependence that individuals and firms have developed on using the internet to accomplish several functions, has resulted in numerous security issues. Among the biggest risks associated with cyber-attacks include phishing attacks which constitute one of the most common ways of deception in the current cyber-crime environment [1]. The attacks employ the technique of deception to convince the victim that the sender of an invitation to download certain information or supply sensitive data is genuine, be it a bank, reputable business organization, governmental institution, or even a social networking site [2] [3].

Phishing attacks are dangerous because they don't just exploit technical vulnerabilities, but rely heavily on manipulating human behavior [4]. The current cyber attacks are conducted by sending certain emails, legal sites, and reliable web addresses. It is important to highlight that during the past few years, a significant modification in the kind of cyber attacks took place – today, the threats are very versatile, and instead of being transmitted only via emails, they can also be sent through SMS, instant messages,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

social networks, advertisements, etc. Therefore, the consequences of such phishing can be catastrophic for individuals not only economically, but also in terms of data breach, privacy violation, etc.

Because of the continuously increasing threat, it is becoming more and more important to conduct extensive studies concerning the phishing phenomenon, its various types and mechanisms, and, finally, offer a solution by creating an intelligent system able to detect and mitigate the phishing process. The basic concept lies in employing the technique of artificial intelligence, which nowadays is regarded as one of the most advanced solutions in the field of IT security due to its high capabilities in data analysis and learning from past experience [5].

The objective of the suggested study is to examine the potential uses of artificial intelligence technologies in detecting attacks in different formats. Additionally, the suggested research will create a certain model allowing to identify whether information presented in the emails and SMS is authentic or fraudulent, based on some factors such as language used in the email/SMS composition and technical aspects of web links and web pages. The research aims to develop a system that can classify messages and URLs in a way that helps users to make decisions before falling victim to fraud.

2. LITERATURE REVIEWS:

In this section, we review the earlier studies conducted regarding machine learning techniques, hybrid machine learning techniques, and deep learning techniques used for detecting and mitigating phishing attacks.

It is advised to apply a three-layer architecture that utilizes hybrid machine learning, as discussed in [6], in order to recognize enticing ad phishing attacks. Such a model would be able to detect advanced phishing web pages because they are capable of bypassing present methods due to their capability to mimic real websites. As test data sets, the authors have employed two databases: the first one contains 20,000 phishing URLs extracted from PhishTank while the other includes 20,000 legitimate URLs obtained from Alexa. They conducted a number of tests by applying several algorithms in order to determine which one will help them develop their model efficiently. The findings have revealed that XGBoost is much better than others, obtaining accuracy rates of 94.1%, decision trees 91%, random forests 90.7%, and MLP 91.6%. In turn, when comparing the performance of algorithms in detecting phishing via text messages, linear SVC is superior to Naive Bayes: they get accuracy rates of 98.9% and 91.19%, respectively. However, despite its efficiency, this model is expensive due to three layers, making real-time detection impossible.

In study [7], the authors presented an effective system for detecting anti-phishing called “Phish-Jam” which can be used in mobile environment. “Phish-Jam” categorizes URLs without performing analysis on content of webpage as either Phishing or Non-Phishing. In accordance with [7], the authors utilized a hybrid approach in which three types of feature sets were considered; handcrafted features, deep learning features with LSTM and transformer embedding using BERT. It was observed that classifiers trained on handcrafted features were more accurate than others with 91.79% accuracy using the model of XGBoost. With regard to deep learning, it was achieved by LSTM and LSTM with multi-attention 96.49% and 96.71% accuracy respectively. Of all transformer features using BERT, the KNN classifier proved to be most accurate with 95.38% rate while hybrid model with handcrafted, LSTM, and transformer sets had 98.93% accuracy rate. However, considering both transformer and LSTM models increase size and time complexity making it unsuitable for mobile applications.

In study [8], the authors present a novel ML-based hybrid botnet detection approach based on three techniques, which are Correlation Analysis (CA), Mutual Information (MI), and Principal Component Analysis (PCA). Five classifiers have been employed for ensemble ML models such as Extreme Tree (ET), Random Forest (RF), XGBoost, Bagging, and Stacking (meta-classifier, i.e., Logistic Regression). To assess the performance of the proposed method, six famous datasets are used: N-BaIoT, Bot-IoT, CTU-13, ISCX, CCC, and CICIDS. Authors say Extra Trees and XGBoost ensemble models achieved top accuracy of 99.9%, while remaining models they used achieved around 90%. However, a major limitation of this approach is its domain specificity where the model is tailored for botnet detection rather than phishing, particularly email or URL-based attacks, so, its generalizability to phishing detection scenarios is limited.

Another study [9] proposes a novel model called “Boosting-based Multi-Layer Stacked Ensemble Learning Model (BMLSELM)” to detect phishing websites. The model combines hybrid feature selection with a multi-layer stacked ensemble approach to improve accuracy and reduce computational overhead. BMLSELM consist of three layers: Layer 1: Four boosting models (XGBoost, LightGBM, CatBoost, AdaBoost). Layer 2: Three models (XGBoost, CatBoost, AdaBoost). Layer 3 (Meta-Learner): XGBoost for

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

final prediction. Where predictions from lower layers passes as input to upper layers for hierarchical learning. Authors used four datasets which are UCI, Mendeley-small, Mendeley-large, Mendeley-extended, BMLSELM model achieved high accuracy up to 98.95% in phishing detection while reducing feature dimensionality by average 73%. Nonetheless, the model's multi-layered ensemble structure introduces considerable complexity, which significantly increases training time which could pose challenges for real-time or large-scale deployment scenarios where rapid model updates are necessary.

In study [10] authors proposes a HFST combining filter and wrapper methods to detect malicious URLs in IoT environments. The goal of study [10] is to reduce computational overhead while maintaining high accuracy by selecting optimal features and evaluating them using ensemble classifiers like XGBoost. The dataset consisted of balanced data with 10,000 URLs, with a total of 48 features being reduced to 26 features (46%). The 26 features were chosen using XGBoost, GB (Gradient Boosting), AdaBoost, Bagging, and k-NN (k-Nearest Neighbor). The performance of XGBoost is the highest at 98.3%. Although these results appear quite promising, the use of genetic algorithms for feature selection increases computational costs. Moreover, the smaller data set size is an additional limitation.

Authors of study [11] evaluate eight machine learning algorithms on two datasets "UCI" imbalanced dataset and "Mendeley" balanced dataset. SMOTE oversampling was applied to balance the classes in UCI. For (UCI) dataset SVM accuracy improved from 94.95% (baseline) to 96.90% (tuned), GB accuracy jumped from 94.71% to 97.08%, RF achieved stable performance before and after tuning with accuracy about 97.4 % and XGB also achieved stable performance with accuracy of 97.1%. For (Mendeley) dataset XGBoost accuracy improved from 98.15% to 98.26%, GB accuracy increased from 97.25% to 97.98%, SVM achieved the most significant improvements in accuracy from 94.48% to 96.66%, and KNN improved from 93.63% to 94.73% accuracy. Using feature selection on (UCI) dataset, RF and GB achieved highest accuracy of 97.13% and 97.18%. SVM reach to 96.02% accuracy, and KNN reach same accuracy 96.02%. For (Mendeley) dataset, GB and XGBoost achieved peak accuracy about 98.07% and 98.08%. RF model performed best 97.3% accuracy. However, despite some improvements, the gains after balancing and tuning were relatively modest for several models. In some cases, the tuning introduced additional computational complexity without yielding significant accuracy benefits which indicates that not all algorithms respond equally to such enhancements.

According to [12] the purpose of the study is to create a machine learning model for detecting "keylogger" malware that operates via websites without requiring physical access. They suggest the use of HEFS method to minimize the feature space. Authors of study [12] used three classifiers namely RF, LightGBM, and CatBoost to evaluate on a publicly available dataset. They used Canadian "Institute for Cybersecurity (CIC)" imbalanced dataset which contains 523,617 samples. Results of evaluation models as demonstrated in study [12], RF using 13 selected features achieved accuracy of 96.12% and using full features (85 features) achieved accuracy of 97.71%, this result shows a trade-off Only 1.59% accuracy drop despite 84.5% fewer features. Nevertheless, the model was specifically designed for keylogger detection rather than phishing or malicious URL analysis, so, its generalizability to phishing detection tasks remains untested.

In study [13], authors compare two ensemble learning techniques "Adaboost and MultiBoosting" with baseline classifiers like "SVM, k-NN, ANN, C4.5, and RF. Authors used a dataset from "UCI" Machine Learning Repository named "Phishing Websites Dataset". The results showed that RF achieved 97.26% accuracy. The best Ensemble Adaboost with SVM achieved 97.61% accuracy. MultiBoosting with Rotation Forest achieved 97.30% accuracy. However, the accuracy improvements offered by the ensemble techniques were relatively marginal compared to the best-performing single classifiers but these slight gains come at the cost of increased computational complexity.

Study [14] proposes a hybrid two-level framework named "DOFA-XGBoost", combining an improved metaheuristic algorithm DOFA with XGBoost to enhance phishing website detection. The study used two datasets "Mendeley" and "UCI". DOFA achieved 94.46% accuracy on Mendeley and 97.60 on UCI.

Authors of study [15] proposes a hybrid model combining a GA with a CRNN to optimize URL feature selection. Authors used different datasets include "ISCX-URL-2016", "PhishStorm", and "PhishTank". They addressed class imbalance via stratified sampling. PhishTank dataset model "GA-CRNN" achieved an accuracy of 96.85% with +4.13% over baseline CNN-LSTM, and achieved Recall 95.10% with +7.07% over baseline. For "ISCX-URL-2016" dataset "GA-CRNN" model achieved Recall improved to 95.10% compared to 88.03% for CNN-LSTM. However, this integration results in high training time and substantial memory consumption.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Study [16] authors propose another effective phishing detection model by combining feature selection techniques with ML classifiers. It used "UCI" Phishing Website dataset. Authors analyzed 6 common features shared by ET, Forward Selection, and Pearson Correlation. Highest accuracy 93.86% achieved by Random Forest and Bagging. In another scenario they evaluated 25 features selected by the Logit (LR) model. RF achieved accuracy 97.30%. Also, they evaluate the models on all features in the dataset used. RF again achieved best performance with 97.10% accuracy. When they applied PCA Random Forest achieved best performance with 95.44% accuracy. However, it is noteworthy that the Naïve Bayes (NB) classifier consistently underperformed across all feature selection scenarios which may suggest that the dataset lacks sufficient diversity or feature richness needed for probabilistic classifiers to perform effectively.

Table 2.1 shows a comparison of reviewed studies.

Table 2.1. A comparison of reviewed studies

Ref.	Year	Model / Method	Feature Types	Dataset	Classifiers	Performance	Limitations
[6]	2023	3-layer hybrid ML (URL + NLP + OCR)	URL, Text (NLP), Image (OCR)	20k phishing (PhishTank) + 20k legitimate (Alexa)	XGBoost, SVM, Naïve Bayes	XGBoost: 94.1%, SVC on NLP & OCR: 98.9%	High computational cost due to multi-layer design and OCR-dependent on image clarity
[7]	2025	"Phish-Jam" hybrid ensemble	Handcrafted URL, LSTM, BERT	PhishDump (331,552 URLs)	XGBoost, KNN, LSTM, Ensemble	Hybrid: 98.93%	Transformer and LSTM layers increase model size and latency on mobile devices
[8]	2023	Hybrid with 3 FS + 5 ML	Network traffic features	6 botnet datasets (e.g., N-BaIoT, CICIDS)	XGBoost, ET, RF, Stacking	Up to 99.9%	Targeted at botnet, not email or URL-based phishing, so limited generalizability
[9]	2023	BMLSELM multi-layer stack	66–86% top features via boosting	UCI, Mendeley (4 versions)	XGBoost, CatBoost, AdaBoost	Up to 98.95%	Complex architecture increasing the training time
[10]	2024	HFST (Hybrid FS with MI + GA + FS)	Lexical, content, advanced lexical	10,000 balanced URLs	XGBoost, GB, Bagging, k-NN	XGBoost: 98.3%	GA and wrapper FS add computational load and the dataset is small which limit the generalizability
[11]	2023	Tuned ML (balancing + FS + grid search)	URL + content	UCI, Mendeley	SVM, GB, RF, XGBoost	RF: 97.4%, SVM improved by 2%	Slight gains after balancing, so, tuning doesn't help all models and increase computational cost
[12]	2023	HEFS for keylogger detection	85 → 13 selected features	CIC keylogger dataset	RF, CatBoost, LightGBM	RF: 96.1%, CatBoost improved by 15.6%	Designed for keyloggers, not phishing; generalizability not validated
[13]	2020	AdaBoost vs. MultiBoost	30 standard phishing features	UCI Phishing Websites Dataset	SVM, RF, k-NN, AdaBoost	Adaboost+SVM: 97.61%	Minor accuracy gains with high cost
[14]	2023	DOFA-XGBoost (2-level FS & tuning)	Full + Selected phishing features	UCI, Mendeley	XGBoost	Up to 97.6%	High parameter tuning complexity
[15]	2022	GA-CRNN (Genetic + CNN + LSTM)	90k URL features (char/word level)	PhishTank, ISCX	CRNN, CNN-LSTM	Up to 96.85%, recall improved +7%	High training time and memory due to GA and DL layers

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

[16]	2023	Multiple FS + ML comparison	PCA, LR, ET, Pearson	UCI Phishing Dataset	RF, DT, SVM, KNN	RF: 97.48% (PCA), NB lowest	NB consistently underperformed, so, dataset may lack diversity

3. METHODOLOGY:

Given the above dynamic aspect of phishing attacks, it is necessary for any system developed in order to counteract these attacks to not only be highly accurate but resilient and adaptable to any form of attack vector [17]. With the increasing sophistication in phishing attacks, traditional approaches like blacklists and signature filters can no longer cope with this problem [18]. In this section, we will discuss, in detail, the process of developing an intelligent system that could aid in phishing attack detection and prevention. In our proposed solution, we will utilize AI techniques, including deep learning, NLP, and model explainability, to construct a practical and transparent detection framework.

The methodology employed for this research work involves the use of a pipeline methodology approach. It includes data gathering and pre-processing, exploratory data analysis, modeling, training, testing, and deployment phases. The crux of the architecture involves the utilization of the comparison between three distinct types of classifiers, where Fine-Tuned BERT will be used to factor in the context, CNN with attention to capture features in the input sequence, and the final one being a random forest benchmark classifier.

3.1 Overview of the Proposed Methodology:

The objective of this methodology is to design an intelligent system that able to identify benign URLs from phishing URLs with high performance.

The complete methodology is composed of seven interconnected phases:

1. Data Collection: Acquiring data that are labeled to describe URL characteristics.
2. Data Pre-processing: Performing pre-processing procedures on acquired data to use for machine learning techniques.
3. Model Building: Constructing three classifiers: Random Forest, CNN with attention, and BERT.
4. Model Training and Evaluation: Training and evaluating the models according to Accuracy, Recall, and False Positive Rate.
5. Model Interpretation: Utilizing SHAP and LIME techniques to explain the outcome of the model.

3.2 Data Collection:

It comprised of gathering enough data that could later on be employed in the process of supervised learning. We used a publicly available dataset from Kaggle. We selected this dataset due to the diversity of data provided. The number of samples of data used in the final dataset was 10,000, equally divided between phishing URLs and legitimate URLs. Each instance in the dataset is represented by a multi-dimensional feature vector, encapsulating a wide array of URL characteristics. These features are categorized as follows:

1. URL Structure Features: URL length, domain length, number of dots (.), number of dashes (-), usage of @ signs, number of slashes (/) and other symbols.
2. Technical Features: Use of an IP address for the website instead of a domain name, use of any unusual ports and use of HTTPS encryption.
3. Domain Features: Number of subdomains, dash symbol usage in the domain name, domain's age (days since registration) and whether it is shortened via URL shortening tools.
4. Page Content & Behavior Features: Number of internal and external links on the website, presence of any login form and e-mail submission form, presence of any iframes and popups, use of favicon from an external resource and external media percentage.
5. Reputation & Statistics Features: Web traffic rank, Google index status, page rank, and the number of times the URL has been reported.

The mathematical representation of a single data instance can be defined as a feature vector $x_i \in \mathbb{R}^M$ where M is the total number of extracted features. Each instance is associated with a binary label $y_i \in \{0,1\}$ where 0 means that the URL is legitimate and 1 donates to phishing URL. We define the full dataset as $D = \{(x_i, y_i)\}$ Where $i \in \{1, 2, 3, \dots, N\}$ and $N = 10000$ the samples number. Figure 1 illustrates the data collection workflow:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

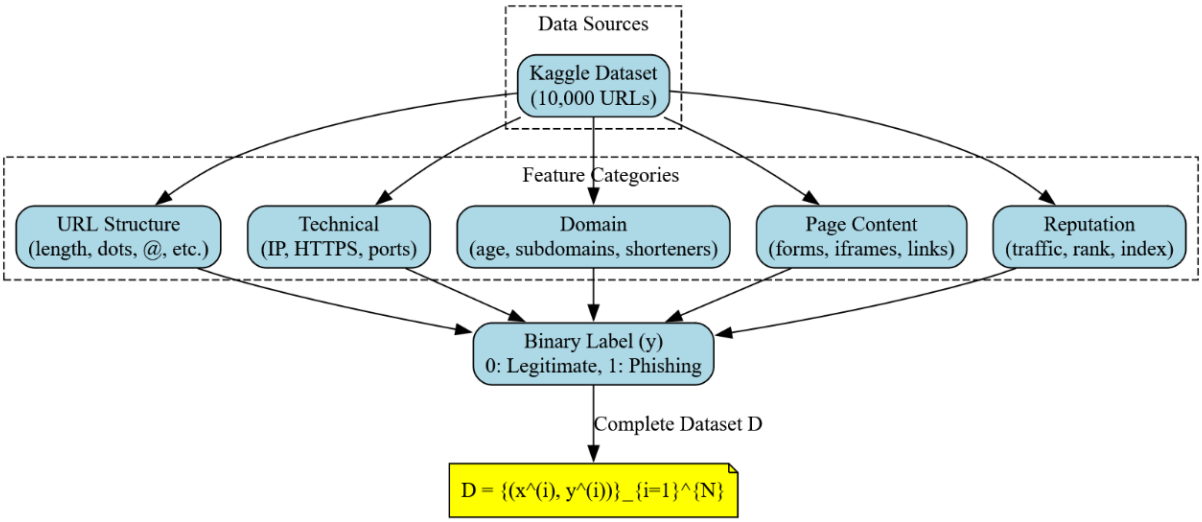

Figure 1. The data collection workflow

3.3 Data Preprocessing:

Raw data is normally noisy, making it hard for the model to perform well with such data. Data pre-processing involves converting raw data to workable data. This process requires you to undertake the following steps:

1. Data Cleaning: The instances, which contains missing URL string or label, were removed.
2. Handling Missing Values: For numerical features with missing values, the median value of that feature across the dataset was used for imputation.
3. Standardization: Standardization is the process of scaling data so that all values fall within a specified range, with the mean being 0 and the standard deviation being 1 [19]. This technique is widely used in data preprocessing to ensure that data points across different variables can be compared on the same scale to prevent variables with larger numerical ranges from dominating variables with smaller ranges. The formula for standardization is as follows:

$$x_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j} \qquad (3.20)$$

Where:
$x_{i,j}$: The feature j on the sample i.
μ_j: The mean of the feature j across all training samples.
σ_j: The standard deviation of values in feature j.

4. Text Preparation for NLP Models: For the BERT model, the raw URL strings were tokenized using the BERT tokenizer [20]. This process converts the URL into a sequence of tokens, with special [CLS] and [SEP] tokens to mark the sequence start and end [21]. The tokenization function is defined as $Tokenize(u) = [[CLS], t_1, t_2, \dots, t_k, [SEP]]$.
5. Data Splitting: After cleaning and preprocessing, the dataset was split into three subsets Training Set (70%), Validation Set (15%), and Test Set (15%).

Figure 2 shows the preprocessing pipeline.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

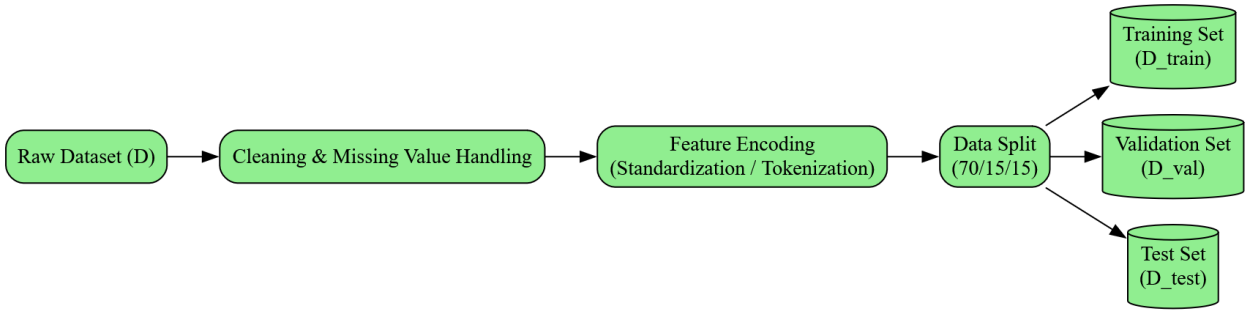

Figure 2. The preprocessing pipeline

3.4 Model Building:

We used three distinct models to provide a comparative analysis of performance and efficiency.

3.4.1 Random Forest:

Random forests may be defined as the extension of the decision tree concept that is focused on the development of independent trees and the determination of the final outcome via the majority vote system.

The RF algorithm is one of the most powerful algorithms in the context of machine learning that may be utilized in the context of tree learning in the domain of machine learning. While developing the decision tree via the RF algorithm, the primary focus is on the development of decision trees in such a way that the features are selected via the process of random selection at each section of the decision tree developed via the process of random selection [22]. The primary objective of the process of randomization is to minimize the possibility of over-allocation in the context of decision tree development. While developing the decision tree via the RF algorithm, the results are summed up via the voting process in the context of classification and average values in the context of regression [22]. Random tree forests are known for their ability to handle complex datasets, minimize over-allocation, and provide reliable predictions in diverse environments, and are therefore widely used in classification and regression tasks. Figure 3 illustrates the workings of a random forest algorithm.

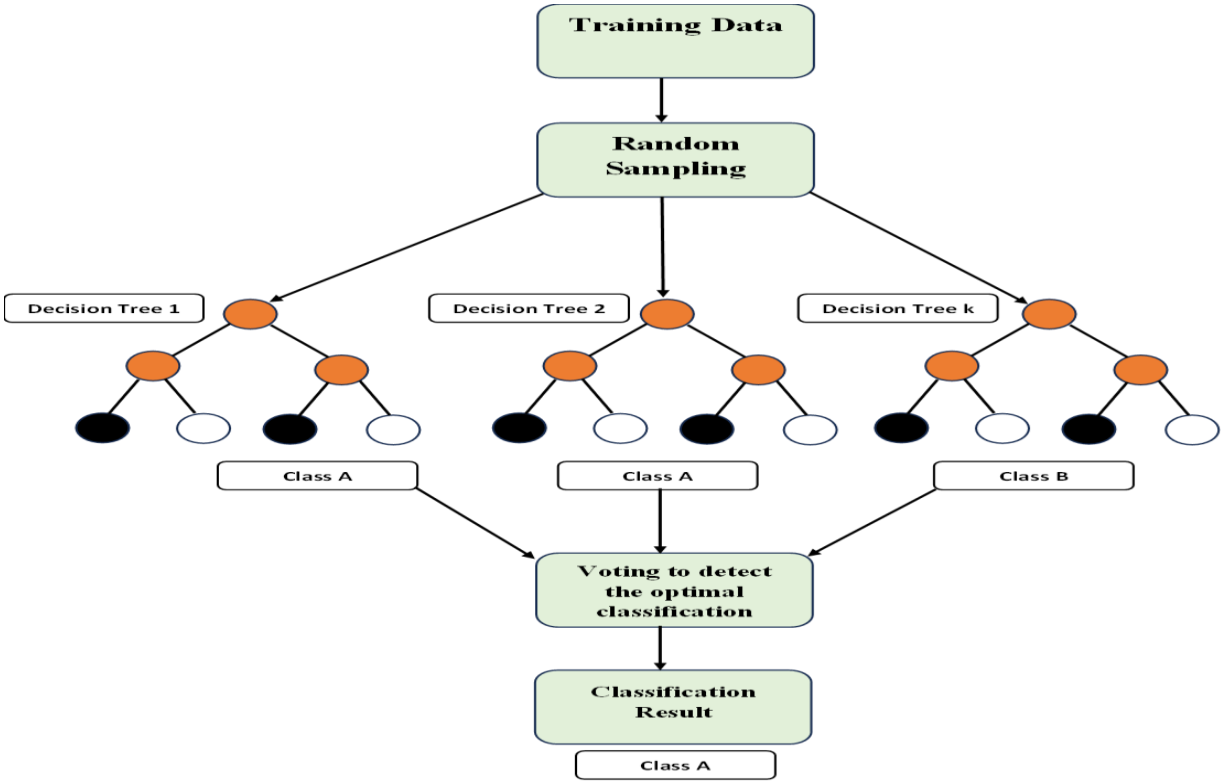

Figure 3. The workings of a random forest algorithm

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

3.4.2 CNN Model with Attention Mechanism:

This model is designed to capture local patterns and focus on the most important parts of a URL string. This architecture processes the tokenized URL sequence.

1. Embedding layer: This layer is responsible for the creation of a mapping from the input tokens into a dense vector of numeric features. The consequence of this is that there is a compressed representation of the text with less dimensionality [23]. One could compare the purpose of the layer to a look-up table in which each word has an associated vector representation. Word embeddings have been used extensively in NLP tasks because they consider both the context and semantics involved in relations between various words.
2. Convolution Layer: Conv1D layer, when used in deep learning, operates based on the mechanism of performing convolutions in only one direction. When using a Conv1D layer, the kernel moves across the data, and every element in the kernel is multiplied by the corresponding element in the receptive field [24]. After multiplying these elements together, the results are summed up to give just one result. This procedure goes on until all elements in the dataset are convoluted to form a new set of features. There are adjustable parameters in the kernel that are responsible for training. Figure 4 illustrates an example of a Conv1D operation.

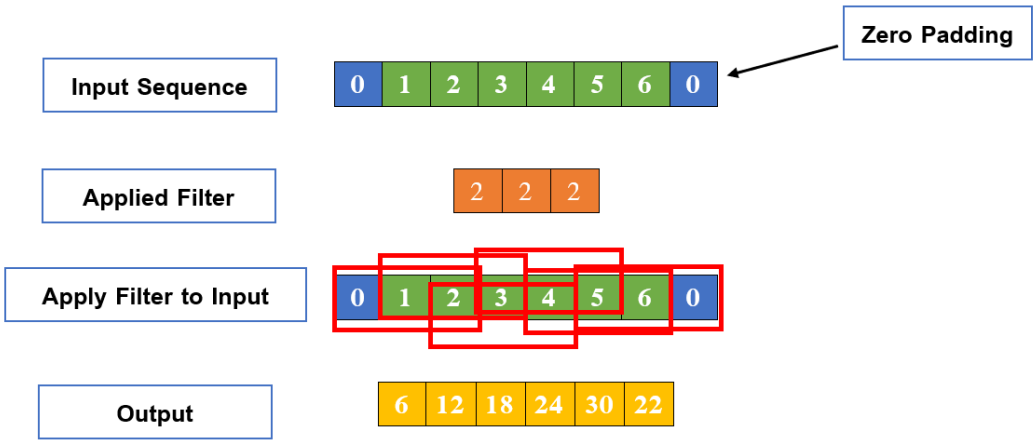

Figure 4. An example of a Conv1D operation

3. Attention Layer: It ensures that the deep learning models concentrate on the important features. This not only helps in improving the accuracy of the predictions but also ensures that the cost of computation is reduced. As such, the important features are made to stand out through the attention mechanisms by identifying them as the most critical elements of the particular problem. The main components of the attention-based model include: (1) Encoder; (2) Attention Mechanism; and (3) Decoder [25]. Figure 5 presents the attention structure.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

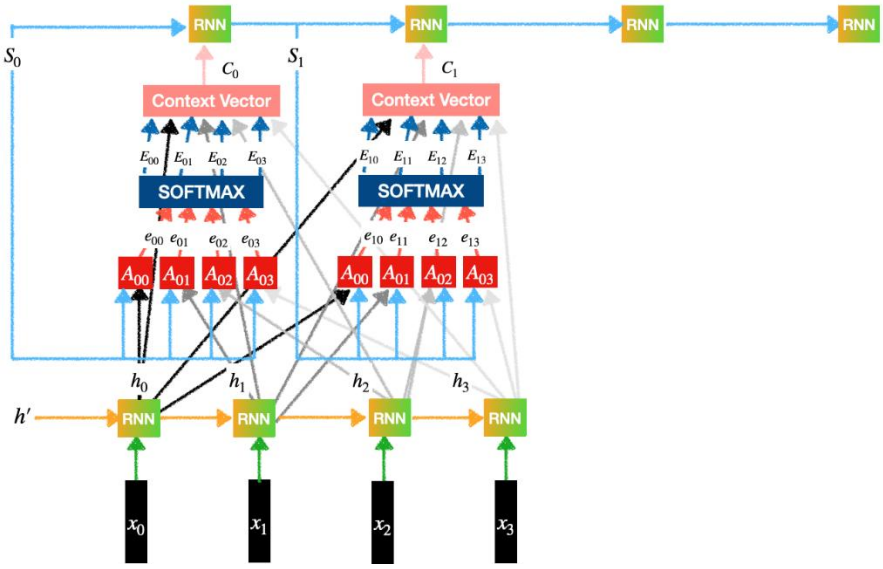

Figure 5. The attention structure

4. Dense layer: Here, the connections are established between all neurons and all the other neurons of the previous layer, hence the name "fully connected." All neurons get input from all activations in the previous layer. The connection has a value of weight associated with it, and the value is optimized during the training process. All the inputs are then summed up based on their weights.
5. Classification Layer: The context vector is passed through a dense layer with a sigmoid activation function to produce a probability score for each category.
6.

3.4.2.1 Model optimization and efficiency improvement techniques:

Because deep learning models may require high computing resources and longer execution times compared to traditional models [26], the proposed system incorporates a set of optimization techniques aimed at reducing computational complexity and improving model efficiency without significantly impacting detection accuracy. This is particularly important when running the model in real-world environments where response time and resource consumption are considered critical issues.

One of the optimization techniques adopted when improving the efficiency of the CNN model is Depthwise Separable Convolutions [27]. It plays an important role in reducing computational costs compared to regular convolutions and thus makes the model training and processing faster with a smaller network size without losing the ability to extract features [27]. We also used the knowledge distillation which refers to the transfer of knowledge from a big model known as the teacher model to a small model called the student model [28].

Further, pruning is one method that we used to remove superfluous parameters or connections in the model, hence reducing the size of the model [29]. Moreover, quantization takes place whereby the precision of the weights of the model is reduced from higher to lower precision; hence there is faster performance while reducing the size of the model [30].

3.4.2.2 Model interpretation and decision explanation:

Interpretability in cyber security solutions is of paramount importance because it plays a significant role in the decision-making process for intelligent models. This point becomes extremely relevant when dealing with the deployment of deep learning algorithms that can be referred to as a "black box." In this case, it becomes challenging to figure out the logic behind the classification of each individual link as either phishing or legitimate. Undoubtedly, this matter is highly relevant to identifying phishing attacks because everyone involved needs to comprehend the essence of this issue.

Thus, the solution suggested in the proposal comprises a certain step that is dedicated to interpreting the results obtained during the process of modeling by using tools such as SHAP and LIME [31]. As was noted before, the models under consideration evaluate the influence of certain features on the work of the classifier used in the system. More specifically, they measure the contribution of

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

those features that make the probability of maliciousness of the connection either lower or higher with the aim of identifying the most influential factors among all others. Thus, for example, the usage of IP addresses instead of domain names, the presence of certain suspicious words in the connection, an abundance of special characters in the connection, and usage of shortened links may be viewed as features that can affect the outcome. Interpretation of model outputs makes it possible to turn complicated mathematical calculations into understandable results perceived by people.

Hence, the incorporation of SHAP and LIME into this system will help enhance the robustness and practicality of the system, particularly in those contexts where explanations are necessary, such as government agencies, banks, and educational institutions, where the interpretation of signals is critical in the security process.

3.4.3 Bidirectional Encoder Representations from Transformers (BERT) Model:

3.4.4

Along with the CNN algorithm using the attention mechanism mentioned above, another algorithm may also be applied to the proposed system through natural language processing technologies using BERT. In this way, it will be possible to improve the efficiency of the system in the situation where it is impossible to identify phishing schemes solely based on their numeric features. It should be emphasized that in most situations, phishing cannot be considered solely as manipulation with hyperlinks, as fraudsters use the linguistic capabilities of both the hyperlink itself and the text accompanying it – whether it is a letter or a text of a website.

The deployment of BERT in the current study hinges on the concept of fine-tuning. In this regard, during the initial stage, BERT will be loaded with its massive language data that is trained to it. Afterwards, BERT will be partially retrained with phishing data. It is worth mentioning here that the main aim of this step is to allow BERT to leverage its language knowledge and further fine-tune it to comprehend malicious and benign URLs. Prior to this, it is necessary to preprocess the data. The texts or URLs will be converted into BERT-friendly inputs using the BERT Tokenizer. Ultimately, the tokenized data will serve as inputs for BERT. In addition, a classification layer will be added to the output layer to produce binary classifications.

For training, a loss function suitable for binary classification problems (Cross-Entropy Loss [32]) is used along with an optimizer AdamW to ensure the stability of the training process and achieve good weight convergence [33].

Accordingly, the BERT model was incorporated into the proposed system not only to achieve the highest possible accuracy but also to provide a clear scientific comparison between a model based on feature analysis and pattern extraction via CNN-Attention and a model based on contextual text understanding via BERT. This comparison reveals the key differences between the two models in terms of performance and resources. BERT excels in accuracy, recall, and reducing false alarms in many cases, while CNN-Attention is more efficient in terms of execution speed and resource requirements, making it more suitable for near-real-time systems or environments with limited capabilities.

The BERT model was employed for its superior contextual understanding of text. It was fine-tuned for the binary classification task. The input for BERT is a sequence of tokens derived from the raw URL, formatted as [CLS] url_tokens [SEP]. The final hidden state corresponding to the [CLS] token $h_{[CLS]} \in \mathbb{R}^{768}$, serves as an aggregate representation of the entire URL. For fine-tuning, The pre-trained BERT model was augmented with a single classification layer and the probability of the URL being phishing is:

$$\hat{p} = \sigma(W_{bert} \cdot h_{[CLS]} + b_{bert})$$

Where:
W_{bert}: The weight matrix.
$h_{[CLS]}$: The final hidden state corresponding to the [CLS] layer.
b_{bert}: The bias.

During fine-tuning, the loss (Binary Cross-Entropy) is backpropagated through the entire network to update both the new classification layer and the pre-trained BERT weights. Figure 6 shows the fine-tuning process.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

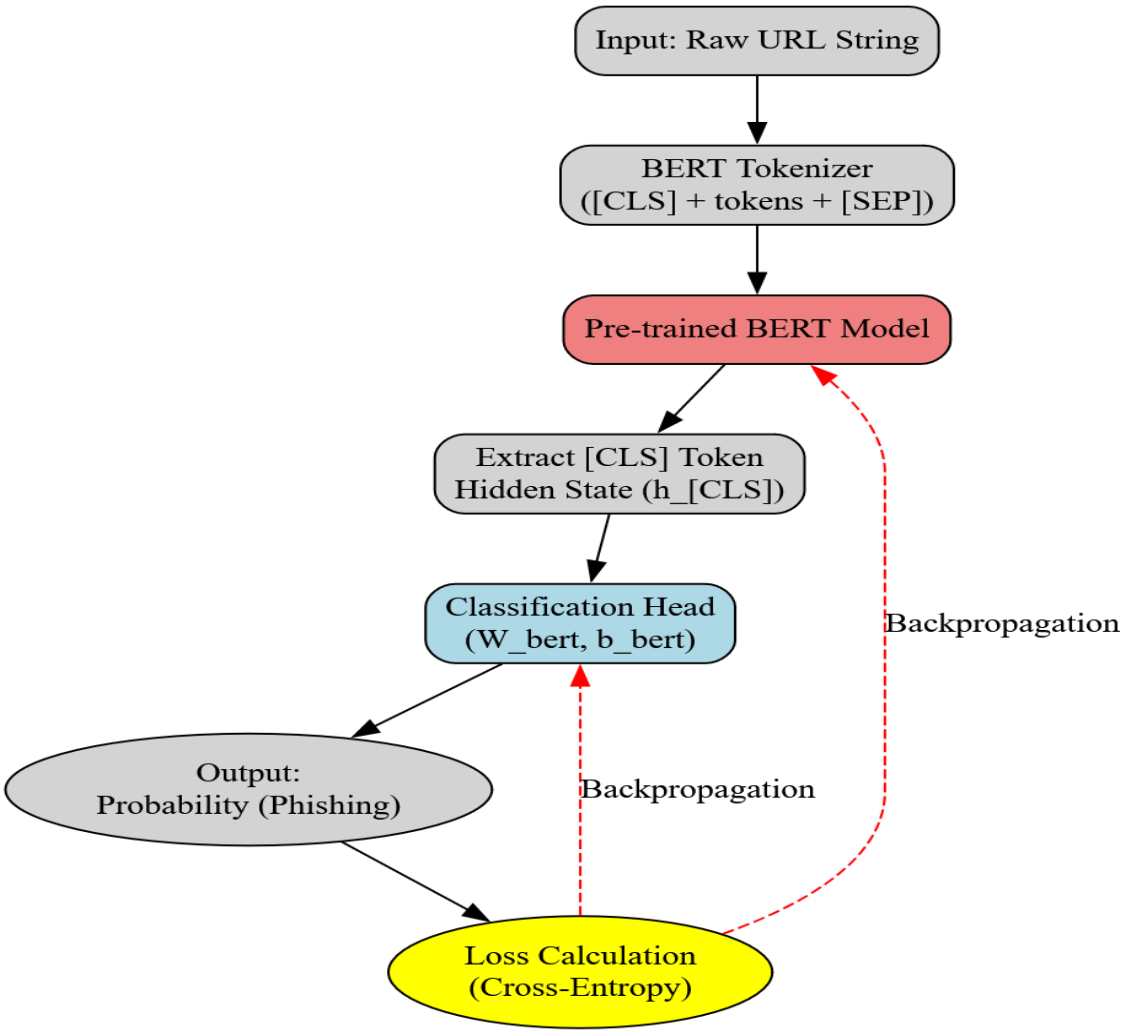

Figure 6. The fine-tuning process

3.5 Evaluation Metrices:

The confusion matrix (CM) is a basic tool used to assess the accuracy of a classification [34]. Figure 7 shows the confusion matrix in general.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Figure 7. The confusion matrix in general

It helps to calculate accuracy, recall, and False Positive Rate (FPR) .

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

4. RESULTS AND DISCUSSIONS:

Experiments conducted using a dataset containing 10,000 links, including 5,000 phishing links and 5,000 legitimate links, yielded promising results confirming the effectiveness of the proposed system in detecting and phishing URLs. Table 2 shows the results of the evaluation of the three models.

Table 2. The results of the evaluation of the three models

Model	Accuracy	Recall	FPR	Prediction Time	Training Time	Required Resources
BERT	98.7%	97.9%	1.4%	15 ms	12 minutes	High (GPU)
CNN + Attention	96.5%	94.8%	2.9%	120 ms	5 minutes	Medium
Random Forest	91.3%	88.6%	5.7%	5 ms	1 minute	Low

The performance of three different classification models was compared in the experiments: BERT, an advanced natural language processing model; CNN + Attention, a deep model for efficient feature extraction; and Random Forest, a traditional reference model. It can be observed from the findings made after the evaluation period that the BERT algorithm was more efficient, recording an accuracy rate of 98.7%, a recall rate of 97.9%, and a false positive rate of 1.4%. This is due to its ability to identify phishing links very effectively.

However, CNN+Attention produced a high performance as well, with accuracy of 96.5% and recall of 94.8%. This depicts its effectiveness in identifying the distinct patterns from the connections while having lower computational complexity as compared to BERT. On the other hand, the accuracy score obtained for Random Forest model was 91.3%, while that for recall was 88.6%.

It can thus be concluded from these results that deep learning models, specifically the BERT model, are better at understanding the link structure and extracting context and semantic cues for distinguishing phishing links from legitimate ones than traditional models, which may rely on superficial features or patterns.

In contrast, while considering the efficiency aspect concerning training time and computing requirements, an evaluation was made to determine the practicality of the models. It was observed that the BERT algorithm that was highly accurate involved substantial computational efforts as well as required more time for training compared to other models. On the other hand, the Random Forest algorithm is said to be computationally efficient due to its fast performance and relatively lower resource needs. This analysis underscores the importance of balancing high accuracy with efficiency and ease of operation, especially when considering system deployment in real-world environments.

4.1 The results explanation:

XAI methods such as SHAP and LIME have been applied to understand how some URLs are labeled as phishing or not. The main contributing factors to this classification include:

- Using IP addresses rather than domain names.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- Use of words such as login, verify, and secure.
- The amount of strange dots and symbols within the URL.
- Shortening URLs through platforms like bit.ly.

These features played an important role in explaining the decision-making process in this specific model and increasing its transparency, thus allowing users to get the required data concerning the process in a convenient manner.

The efficiency of the proposed methodology in identifying the harmful URLs without any considerable possibilities of raising false alarms was rather high. Moreover, the application of interpretation techniques increased the credibility of the model's functioning.

Results showed a highly efficient capability of recognizing phishing URLs, exceeding those of current models' accuracies in identifying such URLs, thus enhancing the system's performance in real-world applications.

4.2 A compared with previous studies:

Table 3 shows a comparison of our results with previous studies.

Table 3. A comparison with previous studies

Ref.	Year	Model / Method	Classifiers	Performance	Limitations
[6]	2023	3-layer hybrid ML (URL + NLP + OCR)	XGBoost, SVM, Naïve Bayes	XGBoost: 94.1%, SVC on NLP & OCR: 98.9%	High computational cost due to multi-layer design and OCR-dependent on image clarity
[7]	2025	"Phish-Jam" hybrid ensemble	XGBoost, KNN, LSTM, Ensemble	Hybrid: 98.93%	Transformer and LSTM layers increase model size and latency on mobile devices
[8]	2023	Hybrid with 3 FS + 5 ML	XGBoost, ET, RF, Stacking	Up to 99.9%	Targeted at botnet, not email or URL-based phishing, so limited generalizability
[9]	2023	BMLSELM multi-layer stack	XGBoost, CatBoost, AdaBoost	Up to 98.95%	Complex architecture increasing the training time
[10]	2024	HFST (Hybrid FS with MI + GA + FS)	XGBoost, GB, Bagging, k-NN	XGBoost: 98.3%	GA and wrapper FS add computational load and the dataset is small which limit the generalizability
[11]	2023	Tuned ML (balancing + FS + grid search)	SVM, GB, RF, XGBoost	RF: 97.4%, SVM improved by 2%	Slight gains after balancing, so, tuning doesn't help all models and increase computational cost
[12]	2023	HEFS for keylogger detection	RF, CatBoost, LightGBM	RF: 96.1%, CatBoost improved by 15.6%	Designed for keyloggers, not phishing; generalizability not validated
[13]	2020	AdaBoost vs. MultiBoost	SVM, RF, k-NN, AdaBoost	Adaboost+SVM: 97.61%	Minor accuracy gains with high cost

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

[14]	2023	DOFA-XGBoost (2-level FS & tuning)	XGBoost	Up to 97.6%	High parameter tuning complexity
[15]	2022	GA-CRNN (Genetic + CNN + LSTM)	CRNN, CNN-LSTM	Up to 96.85%, recall improved +7%	High training time and memory due to GA and DL layers
[16]	2023	Multiple FS + ML comparison	RF, DT, SVM, KNN	RF: 97.48% (PCA), NB lowest	NB consistently underperformed, so, dataset may lack diversity
Proposed	2026	Using BERT and CNN + Attention + Interpretation Tools (SHAP, LIME)	BERT and CNN + Attention	Accuracy 98.7%, Recall 97.9%, FPR 1.4%	Higher demand for GPU resources, relatively higher training time

5. CONCLUSIONS AND FUTURE WORKS:

This paper addresses the issue of phishing attacks, which are among widespread and harmful kinds of cybercrimes due to the use of highly-developed fraudulent tactics aimed at extracting valuable information from victims. As per the methodology presented in this study, there was a need for an intelligent approach for detecting phishing through link analysis using a practical dataset. It is a process where a few pre-processing techniques and exploratory data analysis are applied before building models to test their efficiency in terms of the different approaches that have been put forward. Some of the techniques used include advanced algorithms like BERT and CNN with attention, while the classical Random Forest algorithm acts as the control model. From the results of the experiments carried out, one can see the effectiveness of deep learning approaches in terms of accuracy, sensitivity, and reducing the frequency of false alarms. In terms of the results obtained, the most effective was the BERT model, and the CNN+Attention model is an optimal compromise between accuracy and speed. Also, what distinguishes the current research from others is the aspect of interpretability of the suggested model. In order to get some insights into the working process of the system, SHAP and LIME methods were used. There are still some promising future trends which can make this research further progress. Precision and generality of the proposed model can be improved by training it on more datasets, containing links from different websites and providing more actual information about the phishing attacks. While this research mostly considered link analysis, the logical follow-up would be developing the algorithms of predicting based on content analysis. For instance, one can add sender's IP address, context of the messages, webpage structure, and links' syntax analysis. One can think about creating a reduced version of the model to make it suitable for use on mobile devices and on websites. Some ways of achieving this result are provided below: Knowledge Distillation, Model Pruning and Quantization, and executing the algorithm via ONNX Runtime and TensorRT. The current model already enables some level of interpretation with the use of SHAP and LIME. However, a simpler and more intuitive visualization of these instruments could improve users' experience. Furthermore, practical recommendations such as "reason for risk" and "suggested action" could be provided instead of simply displaying the characteristics.

REFERENCES:

[1] P.-N. Nina, P.-N. Nina, K. B. Géza, S. Malatyinszki, and M. Szilárd, "The Human Factor of Information Security: Phishing in Cybercrime," 2024.
[2] F. Carroll, J. A. Adejobi, and R. Montasari, "How good are we at detecting a phishing attack? Investigating the evolving phishing attack email and why it continues to successfully deceive society," *SN Computer science,* vol. 3, no. 2, p. 170, 2022.
[3] A. K. Jain and B. B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," *Enterprise Information Systems,* vol. 16, no. 4, pp. 527-565, 2022.
[4] M. S. Kheruddin, M. A. E. M. Zuber, and M. M. M. Radzai, "Phishing attacks: Unraveling tactics, threats, and defenses in the cybersecurity landscape," *Authorea Preprints,* 2024.
[5] S. R. Bauskar, C. R. Madhavaram, E. P. Galla, J. R. Sunkara, and H. K. Gollangi, "AI-driven phishing email detection: Leveraging big data analytics for enhanced cybersecurity," *Library Progress International,* vol. 44, no. 3, pp. 7211-7224, 2024.
[6] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A hybrid approach for alluring ads phishing attack detection using machine learning," *Sensors,* vol. 23, no. 19, p. 8070, 2023.
[7] R. S. Rao, C. Kondaiah, A. R. Pais, and B. Lee, "A hybrid super learner ensemble for phishing detection on mobile devices," *Scientific Reports,* vol. 15, no. 1, p. 16839, 2025.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

[8] M. A. Hossain and M. S. Islam, "A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection," *Scientific Reports,* vol. 13, no. 1, p. 21207, 2023.
[9] L. R. Kalabarige, R. S. Rao, A. R. Pais, and L. A. Gabralla, "A boosting-based hybrid feature selection and multi-layer stacked ensemble learning model to detect phishing websites," *IEEE Access,* vol. 11, pp. 71180-71193, 2023.
[10] S. Mohanty, A. A. Acharya, T. Gaber, N. Panda, E. Eldesouky, and I. A. Hameed, "An efficient hybrid feature selection technique toward prediction of suspicious urls in iot environment," *IEEE access,* vol. 12, pp. 50578-50594, 2024.
[11] S. R. Abdul Samad *et al.*, "Analysis of the performance impact of fine-tuned machine learning model for phishing URL detection," *Electronics,* vol. 12, no. 7, p. 1642, 2023.
[12] M. S. Alsubaie, S. H. Atawneh, and M. S. Abual-Rub, "Building Machine Learning Model with Hybrid Feature Selection Technique for Keylogger Detection," *International Journal of Advances in Soft Computing & Its Applications,* vol. 15, no. 2, 2023.
[13] A. Subasi and E. Kremic, "Comparison of adaboost with multiboosting for phishing website detection," *Procedia Computer Science,* vol. 168, pp. 272-278, 2020.
[14] L. Jovanovic *et al.*, "Improving phishing website detection using a hybrid two-level framework for feature selection and xgboost tuning," *Journal of Web Engineering,* vol. 22, no. 3, pp. 543-574, 2023.
[15] S.-J. Bu and H.-J. Kim, "Optimized URL feature selection based on genetic-algorithm-embedded deep learning for phishing website detection," *Electronics,* vol. 11, no. 7, p. 1090, 2022.
[16] R. Pal, M. K. Pandey, S. Pal, and D. C. Yadav, "Phishing Detection: A Hybrid Model with Feature Selection and Machine Learning Techniques," *IJERR,* vol. 36, pp. 99-108, 2023.
[17] A. S. K. Joseph and S. Srinivasan, "Anti-phishing adaptive AI systems: Efficiently countering social engineering attacks by real-time analysis of email content," in *2025 International Conference on Computational Innovations and Engineering Sustainability (ICCIES)*, 2025: IEEE, pp. 1-6.
[18] O. Kadyrov and A. Batyrgaliyev, "Multi-level Spam Protection Architecture: Integrating Rules, Blacklists, and Machine Learning Techniques," *Computing & Engineering,* vol. 3, no. 2, pp. 11-22, 2025.
[19] O. Ismael, "The Standard Deviation Score: a novel similarity metric for data analysis," *Journal of Big Data,* vol. 12, no. 1, p. 58, 2025.
[20] F. Souza, R. F. Nogueira, and R. A. Lotufo, "BERT models for Brazilian Portuguese: Pretraining, evaluation and tokenization analysis," *Applied Soft Computing,* vol. 149, p. 110901, 2023.
[21] M. A. Khan, A. Khalid, M. S. Ibad, D. Khan, and F. Malik, "From Sequences to Transformers: LSTM and BERT Powering Next-Gen Sentiment Analysis," *International Journal of Information Systems and Computer Technologies,* vol. 5, no. 1, pp. 80-94, 2025.
[22] T. Prajwala, "A comparative study on decision tree and random forest using R tool," *International journal of advanced research in computer and communication engineering,* vol. 4, no. 1, pp. 196-199, 2015.
[23] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A survey of text representation and embedding techniques in nlp," *IEEe Access,* vol. 11, pp. 36120-36146, 2023.
[24] F. Xiao, J. Zhang, G. Yang, R. Liu, X. Cheng, and S. Chen, "ZoomingCOD: Zooming-like behavior for Camouflage Object Detection based on Cross-axis Perception and Adaptive Receptive Field," *IEEE Transactions on Multimedia,* 2026.
[25] B. Guo, Q. Zhang, Q. Peng, J. Zhuang, F. Wu, and Q. Zhang, "Tool health monitoring and prediction via attention-based encoder-decoder with a multi-step mechanism," *The International Journal of Advanced Manufacturing Technology,* vol. 122, no. 2, pp. 685-695, 2022.
[26] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," *ACM Computing Surveys,* vol. 55, no. 12, pp. 1-37, 2023.
[27] J.-G. Jang, C. Quan, H. D. Lee, and U. Kang, "Falcon: lightweight and accurate convolution based on depthwise separable convolution," *Knowledge & Information Systems,* vol. 65, no. 5, 2023.
[28] N. Passalis, M. Tzelepi, and A. Tefas, "Knowledge distillation," in *Deep Learning for Robot Perception and Cognition*: Elsevier, 2022, pp. 165-186.
[29] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," *Ieee Access,* vol. 10, pp. 63280-63300, 2022.
[30] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, "A comprehensive survey on model quantization for deep neural networks in image classification," *ACM Transactions on Intelligent Systems and Technology,* vol. 14, no. 6, pp. 1-50, 2023.
[31] M. Arunika, S. Saranya, S. Charulekha, S. Kabilarajan, and G. Kesavan, "A survey on explainable AI using machine learning algorithms shap and lime," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024: IEEE, pp. 1-6.
[32] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *International conference on Machine learning*, 2023: pmlr, pp. 23803-23828.
[33] P. Zhou, X. Xie, Z. Lin, and S. Yan, "Towards understanding convergence and generalization of AdamW," *IEEE transactions on pattern analysis and machine intelligence,* vol. 46, no. 9, pp. 6486-6493, 2024.
[34] S. Sathyanarayanan and B. R. Tantri, "Confusion matrix-based performance evaluation metrics," *African Journal of Biomedical Research,* vol. 27, no. 4S, pp. 4023-4031, 2024.