Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.056476

ARTICLE

# Cuckoo Search-Optimized Deep CNN for Enhanced Cyber Security in IoT Networks

**Brij B. Gupta[1,2,3,4,*], Akshat Gaurav[5], Varsha Arya[6,7], Razaz Waheeb Attar[8], Shavi Bansal[9], Ahmed Alhomoud[10] and Kwok Tai Chui[11]**

[1]Department of Computer Science and Information Engineering, Asia University, Taichung, 413, Taiwan

[2]Symbiosis Centre for Information Technology (SCIT), Symbiosis International University, Pune, 411057, Maharashtra, India

[3]Center for Interdisciplinary Research, University of Petroleum and Energy Studies (UPES), Dehradun, 248007, India

[4]University Centre for Research and Development (UCRD), Chandigarh University, Chandigarh, 140413, India

[5]Computer Engineering, Ronin Institute, Montclair, NJ 07043, USA

[6]Department of Business Administration, Asia University, Taichung, 413, Taiwan

[7]Department of Electrical and Computer Engineering, Lebanese American University, Beirut, 1102, Lebanon

[8]Management Department, College of Business Administration, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

[9]Department of Research and Innovation, Insights2Techinfo, Jaipur, 302001, India

[10]Department of Computer Science, Faculty of Science, Northern Border University, Arar, 91431, Saudi Arabia

[11]Department of Electronic Engineering and Computer Science, Hong Kong Metropolitan University (HKMU), Hong Kong, 518031, China

*Corresponding Author: Brij B. Gupta. Email: bbgupta@asia.edu.tw

Received: 23 July 2024 Accepted: 23 October 2024 Published: 19 December 2024

## ABSTRACT

Phishing attacks seriously threaten information privacy and security within the Internet of Things (IoT) ecosystem. Numerous phishing attack detection solutions have been developed for IoT; however, many of these are either not optimally efficient or lack the lightweight characteristics needed for practical application. This paper proposes and optimizes a lightweight deep-learning model for phishing attack detection. Our model employs a two-fold optimization approach: first, it utilizes the analysis of the variance (ANOVA) F-test to select the optimal features for phishing detection, and second, it applies the Cuckoo Search algorithm to tune the hyperparameters (learning rate and dropout rate) of the deep learning model. Additionally, our model is trained in only five epochs, making it more lightweight than other deep learning (DL) and machine learning (ML) models. The proposed model achieved a phishing detection accuracy of 91%, with a precision of 92% for the 'normal' class and 91% for the 'attack' class. Moreover, the model's recall and F1-score are 91% for both classes. We also compared our approach with traditional DL/ML models and past literature, demonstrating that our model is more accurate. This study enhances the security of sensitive information and IoT devices by offering a novel and effective approach to phishing detection.

## KEYWORDS

Deep learning; phishing; Cuckoo Search; cable news network (CNN); IoT; ANOVA F-test

## 1 Introduction

The Internet of Things (IoT) environment is currently experiencing several developments. One notable development is the expansion of the scale and diversity of IoT data, which has led to improvements in the industry ecosystem and the emergence of innovations in different areas [1,2]. Technological advancements have driven application products towards intelligence, convenience, low power consumption, extensive connectivity, and comprehensive coverage [1,3,4]. Additionally, there has been a focus on improving IoT intrusion detection systems (IDS) using artificial intelligence (AI) techniques, particularly machine learning (ML) algorithms [5,6]. Researchers have reviewed and compared various ML algorithms and techniques using different datasets to enhance the efficiency of IoT IDSs [5,7,8]. However, there is still a need for recent datasets collected from the IoT environment and further investigation into the most effective ML models for building efficient IoT IDSs [5,9,10]. Another critical aspect of the IoT environment is the threat of malware. Studies have characterized 64 IoT malware families, analyzing their behaviors, target architecture, target devices, delivery methods, attack vectors, persistence techniques, and evolution [11,12]. Finally, the emergence of the Narrowband of the Internet of Things (NB-IoT) has significantly contributed to the evolution of IoT towards intelligence, convenience, low power consumption, extensive connectivity, and comprehensive coverage [1,13,14]. Overall, these developments in the IoT environment highlight the continuous growth and advancements in various aspects of IoT technology

Phishing attacks can have a significant impact on the IoT environment. Phishing is a cyber attack where attackers deceive users into revealing sensitive information or performing malicious actions by posing as a trustworthy entity. In the context of the IoT, phishing attacks can target both the users of IoT devices and the devices themselves. One way phishing attacks affect the IoT environment is by compromising the security and privacy of IoT devices and networks. Attackers can use phishing techniques to trick users into providing their login credentials or other sensitive information, which can then be used to gain unauthorized access to IoT devices or networks [15]. Once attackers gain access, they can manipulate or control the devices, potentially causing disruptions or unauthorized actions. Phishing attacks can also lead to the theft of sensitive data from IoT devices. By tricking users into providing their personal or financial information, attackers can access valuable data stored on IoT devices, such as health records, financial information, or personal preferences. This can seriously affect individuals and organizations, as the compromised data can be used for identity theft, fraud, or other malicious purposes. Furthermore, phishing attacks can exploit vulnerabilities in the IoT ecosystem. The IoT environment consists of a complex network of interconnected devices, sensors, and platforms, creating opportunities for attackers to exploit security weaknesses. Phishing attacks can be used to gain initial access to the IoT network, allowing attackers to launch further attacks or spread malware [16].

### *1.1 Contribution*

Our proposed method focuses on the research gap by combining hyperparameter optimization with a unique feature selection technique within a unified framework, primarily meant to be light-weight and computationally effective. We guarantee that the model chooses the most relevant features and simultaneously optimizes the hyperparameters using CSA, minimizing training time (five epochs) and computing overhead.

### *1.2 Organization*

The rest of the paper is organized as follows: The details of the previous works and their limitations are presented in Section 2. The limitations of the previous works motivated us to build our proposed

model, the architecture of which is presented in Section 3. This section details the data preprocessing stage, feature selecting stage, and information about the Cuckoo algorithm. This section helps the reader to get in-depth knowledge about our proposed model. After this, the details of our system model are presented in Section 4. After this, the details of the dataset are presented. After this, the performance of feature selection and hyper-parameters are presented. After this, the performance of the model is presented. Also, the comparative analysis with standard ML/DL models is presented in this section, and compression of the previous work is also presented in this section. This section gives a detailed analysis of the limitations of the past results and their comparison related to our proposed model. Finally, Section 5 concludes the paper.

## 2 Related Work

In the field of phishing attack detection, a variety of methodologies have been explored, with significant advancements being made through the application of ML and DL techniques with optimizes.

For instance, Saxena et al. [17] introduce the use of the Cuckoo Search algorithm for feature selection in phishing website detection, demonstrating improvements in classification accuracy and reduction of errors using random forest and BF-tree classifiers. The paper primarily relies on data mining techniques and classifiers evaluated using a specific dataset from NASA, which may limit the generalizability of the findings to other datasets or real-world scenarios. Additionally, while the Cuckoo Search algorithm is used for feature selection, the study may not fully explore the impact of different hyperparameter settings on classifier performance. Building on the theme of feature optimization, Kumaresan et al. [18] present a novel spam classification framework that combines text and image features with a hybrid kernel-based SVM, achieving significant improvements in accuracy over existing methods, demonstrating the effectiveness of the S-Cuckoo search for feature optimization and hybrid kernel integration in SVM classifiers. However, the reliance on a hybrid kernel-based SVM (HKSVM) and S-Cuckoo search for feature selection might result in increased computational complexity and slower processing times, potentially limiting its scalability for real-time spam detection across large datasets.

Brindha et al. [19] present an Intelligent Cuckoo Search Optimization Algorithm (ICSOA) combined with a Gated Recurrent Unit (GRU) model for improved phishing email detection and classification, thereby furthering the topic of optimization algorithms. This method dramatically outperforms current methods by maximizing feature selection and hyperparameters, attaining exceptional accuracy. Using ongoing research on meta-heuristic algorithms, Sabahno et al. [20] present an enhanced spotted hyena optimization (ISHO) method that improves feature selection for phishing website detection. This work shows better performance and accuracy than several other meta-heuristic methods and conventional classifiers. Though efficient, the study's emphasis on a single feature selection and classification combination (ISHO algorithm with SVM) could not transfer well across many datasets or more complicated, real-world settings. It could also neglect the flexibility needed to change phishing strategies in Internet of Things settings. By using bio-inspired meta-heuristic algorithms for feature selection, Al-Sawwa et al. [21] present a complete method to improve phishing detection, thus expanding the spectrum of meta-heuristic techniques. The work shows that the Spark-based MVO method performs better than others in reaching high detection rates across many phishing datasets. The work mainly addresses feature selection using bio-inspired meta-heuristic algorithms, which may not consider the real-time adaptation needed for changing phishing strategies.

Furthermore, even if the Apache Spark-based decision tree technique is scalable, its performance may be restricted in more complicated or dynamic surroundings outside the evaluated data.

Ali et al. [22] propose a PSO-based feature weighting method to improve phishing website detection accuracy. This work lowers the required feature count for successful detection and significantly increases classification accuracy. Though efficient, using too much particle swarm optimization (PSO) for feature weighting might cause processing cost and complexity, hence perhaps restricting real-time applicability. Furthermore, the method could still rely on the first choice of characteristics, which might not entirely reflect the changing character of phishing strategies. The effectiveness of nature-inspired algorithms is further supported by Anupam et al. [23], who demonstrate that nature-inspired optimization algorithms, particularly the Grey Wolf Optimiser, can significantly improve the performance of SVM classifiers in detecting phishing websites. While this approach outperforms traditional grid-search optimized models, the paper focuses on optimizing a Support Vector Machine (SVM) classifier using nature-inspired algorithms, but it may not address the adaptability required for real-time phishing detection in a rapidly evolving threat landscape. Additionally, while multiple optimization algorithms are compared, the study does not explore the potential benefits of hybrid approaches that could further enhance performance. Finally, the relevance of Cuckoo Search in phishing detection is highlighted by Niu et al. [24], who introduces a Cuckoo Search-enhanced SVM (CS-SVM) model for phishing email detection. This approach demonstrates significant improvements in classification accuracy, yet it may not address scalability issues when applied to larger, more diverse datasets. Additionally, the reliance on a specific dataset with a limited number of phishing emails could restrict the generalizability of the findings.

Kumar et al. [25] present a novel application of the Swarm Intelligence Binary Bat Algorithm in designing a neural network for phishing website detection. While the paper introduces an innovative approach, it may face scalability challenges when applied to larger datasets or more complex phishing scenarios. Additionally, the model's reliance on a single optimizer (Adam) might limit its adaptability to different datasets or evolving phishing tactics.

***Research Gap***

Despite significant progress in phishing detection using many machine learning and deep learning approaches, the simultaneous optimization of feature selection and hyperparameter tweaking still shows a clear gap. Though they may coherently combine both, current methods usually concentrate on either feature selection or hyperparameter optimization. Many of these techniques either depend on sophisticated models or need time and computing resources to reach high accuracy, hence they are less appropriate for real-time or resource-limited settings. This absence of a simplified, effective method combining hyperparameter optimization with feature selection offers an opportunity to develop lightweight, quick, accurate phishing detection systems.

## 3 Proposed Approach

This section presents the details of the proposed approach (Fig. 1). Our prospered approach is divided into four phases: the first phase is for the data processing, the second phase is for selecting optimal features, and the third phase is for the optimal hyper-parameters selected. Finally, in the fourth phase, the deep learning model is used to predict results.

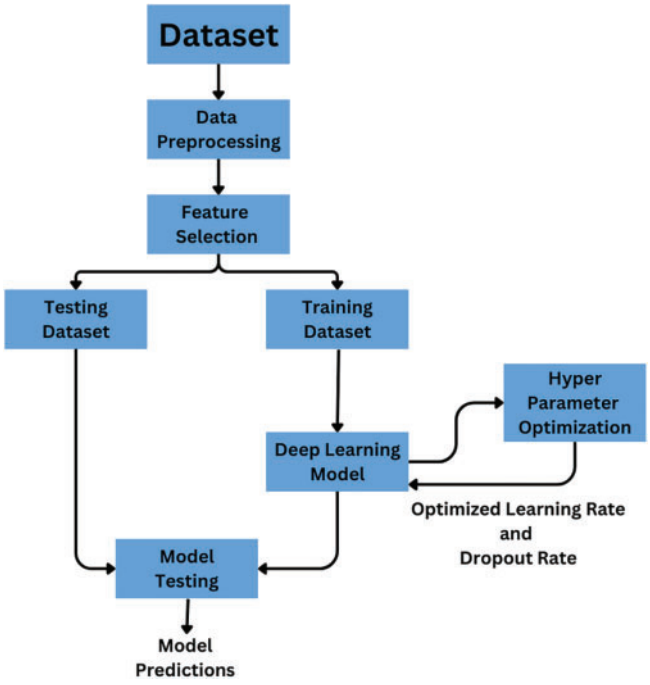


**Figure 1:** Proposed approach

### *3.1 Data Prepossessing*

This is the first phase of our proposed approach. We collect the data from the Kaggle and perform the data preprocessing.

#### *3.1.1 Data Collection*

Let $\mathscr{D} = \{(x_i, y_i)\}_{i=1}^{n}$ be the dataset, where $x_i \in \mathbb{R}^m$ represents the feature vector of the $i$-th sample, and $y_i \in \{0, 1\}$ represents the corresponding label (e.g., phishing or non-phishing).

$$\mathscr{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{1}$$

where $n$ is the total number of samples, and $m$ is the number of features.

#### *3.1.2 Data Normalization*

We normalize each feature $x_{ij}$ to ensure all features have the same scale. This can be done using min-max normalization:

$$x''_{ij} = \frac{x'_{ij} - \min(x'_j)}{\max(x'_j) - \min(x'_j)} \tag{2}$$

where $\min(x'_j)$ and $\max(x'_j)$ are the minimum and maximum values of the $j$-th feature across all samples.

The normalized dataset is represented as:

$$\mathscr{D}'' = \{(x''_1, y_1), (x''_2, y_2), \ldots, (x''_n, y_n)\} \tag{3}$$

### 3.1.3 SMOTE for Data Balancing

We apply the Synthetic Minority Over-sampling Technique (SMOTE) to balance the class distribution. Let $\mathscr{D}_0$ and $\mathscr{D}_1$ represent the sets of samples for the majority and minority classes, respectively, with $|\mathscr{D}_0| > |\mathscr{D}_1|$.

SMOTE generates synthetic samples $\tilde{x}$ for the minority class:

$$\tilde{x} = x_i'' + \lambda \cdot (x_j'' - x_i'') \tag{4}$$

where $x_i'', x_j'' \in \mathscr{D}_1$ are two randomly chosen minority class samples, and $\lambda \in [0, 1]$ is a random number.

The final balanced dataset $\mathscr{D}'''$ includes the original majority class samples and both the original and synthetic minority class samples:

$$\mathscr{D}''' = \mathscr{D}_0 \cup \mathscr{D}_1 \cup \{\tilde{x}_k\}_{k=1}^{s} \tag{5}$$

where $s$ is the number of synthetic samples generated.

## 3.2 Feature Selection Using ANOVA F-test

In this phase, we describe the feature selection process using the ANOVA F-test, which helps identify the most relevant features for classification.

### 3.2.1 ANOVA F-test

The ANOVA F-test is used to determine the statistical significance of each feature concerning the target variable. For each feature $x_j$, the F-score is computed as:

$$F_j = \frac{\text{Between-group variability}}{\text{Within-group variability}} = \frac{\frac{1}{K-1}\sum_{k=1}^{K} n_k(\bar{x}_{k\cdot} - \bar{x}_{\cdot\cdot})^2}{\frac{1}{N-K}\sum_{k=1}^{K}\sum_{i=1}^{n_k}(x_{ik} - \bar{x}_{k\cdot})^2} \tag{6}$$

where:

- $K$ is the number of groups (classes).
- $n_k$ is the number of samples in the $k$-th group.
- $\bar{x}_{k\cdot}$ is the mean of the $j$-th feature for the $k$-th group.
- $\bar{x}_{\cdot\cdot}$ is the overall mean of the $j$-th feature across all groups.
- $x_{ik}$ is the value of the $j$-th feature for the $i$-th sample in the $k$-th group.

### 3.2.2 Feature Ranking and Selection

Once the F-scores $F_j$ are computed for all features $j$, the features are ranked based on their F-scores. The top $p$ features with the highest F-scores are selected for further processing, where $p$ is the number of selected features:

$$\text{Selected features} = \{x_{j_1}, x_{j_2}, \ldots, x_{j_p}\} \quad \text{where} \quad F_{j_1} \geq F_{j_2} \geq \cdots \geq F_{j_p} \tag{7}$$

The resulting feature set $\mathscr{F}$ for model training is:

$$\mathscr{F} = \{x_{j_1}, x_{j_2}, \ldots, x_{j_p}\} \tag{8}$$

This selected feature set $\mathscr{F}$ is then used for the subsequent model training and classification tasks.

### *3.3 Feature Selection Using Cuckoo Search Algorithm for CNN Model*

In this phase, we describe the feature selection process using the Cuckoo Search algorithm (CSA) [26], which is employed to optimize the feature set for the CNN model.

#### *3.3.1 Cuckoo Search Algorithm*

The CSA is a meta-heuristic optimization algorithm inspired by the brood parasitism of some cuckoo species. It is used to select an optimal subset of features that maximizes the performance of the CNN model. The CS algorithm operates as follows:

1. **Initialization:** Initialize a population of $N$ host nests, where each nest represents a candidate solution (i.e., a subset of features). Let $\mathbf{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ represent the feature subset of the $i$-th nest.
2. **Fitness Evaluation:** Evaluate the fitness $f(\mathbf{x}_i)$ of each nest, where the fitness is defined as the performance (e.g., accuracy) of the CNN model trained on the selected features $\mathbf{x}_i$.
3. **Generation of New Solutions:** Generate new solutions (feature subsets) by Lévy flights:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha \cdot \text{Lévy}(\lambda) \tag{9}$$

   where $\alpha > 0$ is the step size, $t$ is the current iteration, and Lévy($\lambda$) is a random step drawn from a Lévy distribution.
4. **Discovery and Replacement:** Replace a fraction $p_a$ of the worst nests with new solutions generated randomly:

$$\mathbf{x}_i^{(t+1)} \leftarrow \text{Random Initialization} \quad \text{if} \quad f(\mathbf{x}_i^{(t)}) \text{ is among the worst solutions} \tag{10}$$

5. **Selection of Best Solution:** Identify the best solution $\mathbf{x}^*$ that corresponds to the highest fitness $f(\mathbf{x}^*)$.
6. **Iteration:** Repeat Steps 2–5 until a stopping criterion is met (e.g., a maximum number of iterations or convergence is achieved).

#### *3.3.2 Optimal Feature Selection*

The optimal feature subset $\mathscr{F}^*$ is defined as the feature subset corresponding to the best solution $\mathbf{x}^*$ found by the Cuckoo Search algorithm:

$$\mathscr{F}^* = \mathbf{x}^* = \{x_{j_1}, x_{j_2}, \ldots, x_{j_k}\} \tag{11}$$

where $k$ is the number of features in the optimal subset. This feature subset $\mathscr{F}^*$ is then used to train the CNN model, ensuring improved classification performance.

#### *3.3.3 CNN Model Training*

Once the optimal feature subset $\mathscr{F}^*$ is selected, the CNN model is trained using these features. The CNN model's architecture and parameters are optimized based on the chosen features, leading to enhanced accuracy and efficiency in phishing detection.

## 4 Results and Discussion

The previous section gives information about the framework of our proposed approach, and in this section, we present the details of our system specifications and simulation results.

### *4.1 Simulation Parameters*

In this study, we developed and tested models on a Windows 11 PC running Intel Core i5 CPU. We used the key libraries such as Pandas version 1.5.3 and NumPy [27] version 1.24.3 for data processing and manipulation, and PyTorch [28] version 2.2.1 for building deep learning models. Using scikit-learn [29] and mealpy library [30], we ran the suggested model using NVIDIA GeForce RTX 3050 GPU as well.

### *4.2 Dataset Representation*

We utilized the Kaggle [31] dataset to test our proposed models. The dataset comprising around 11,000 website examples, the collection is defined by thirty different criteria reflecting the features of the website along with a class label labeling the website as either phishing (‘1’) or non-phishing (‘−1’). From basic URL metrics to complex security properties like “UsingIP,” “HTTPS,” “AnchorURL,” and “RequestURL,” these parameters capture a wide range of information. Analyzing the complicated behaviors and security stances of websites depends on these characteristics.

### *4.3 Data Prepossessing*

The dataset is not balanced; this dis-balance may affect the output results. Hence, we performed the data resampling with the help of SMOTE [32]. Fig. 2 presents the distribution of the class labels after SMOTE operation. From Fig. 2, it is clear that the dataset is balanced and not affect the final results.

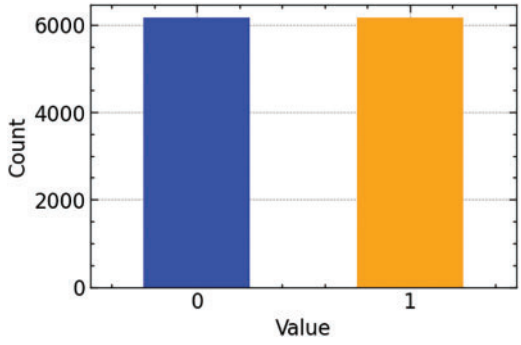


**Figure 2:** Class labels after SMOTE operation

### *4.4 Feature Selection*

After data reprocessing, we used the ANOVA F-test to select important features. As we know that not all the features aaffect the final predictions, we select the ten most important features. In the ANOVA F-test, for each feature, the F-score is calculated, and the best features are specified using the F-scores. As represented in Fig. 3, some features, like ’HTTPS’, ’AnchorURL’, and ’PrefixSuffix-’ has high F-1 scores. Through this process, we sleeted the ten features that have the highest F-1 score.

### *4.5 Performance of Cuckoo Search Algorithm (CSA)*

After selecting the ten most essential characteristics using the ANOVA F-test, we concentrated on improving the hyperparameters (learning rate and dropout rate) of our phishing detection model using the Cuckoo Search method. Aiming to create an appropriate balance between exploration and

exploitation throughout the optimization process, the setup for the Cuckoo Search included setting epochs to 10, a population size of 50, and an abandonment probability (pa) of 0.3.

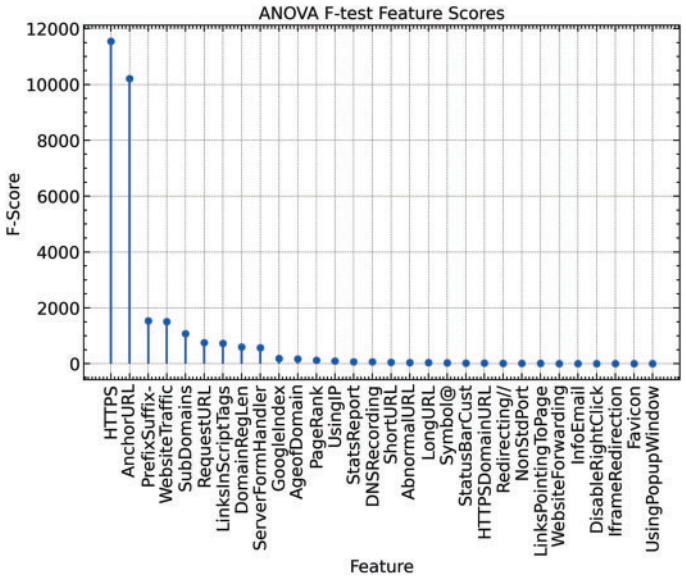


**Figure 3:** Features plot according to ANOVA

As shown in Fig. 4, the model's runtime dynamics displayed fluctuations in computing time throughout many iterations, notably declining towards the latter iterations. As the model approaches convergence, this trend shows an increase in computing efficiency because it optimizes both the learning and dropout rates efficiently.

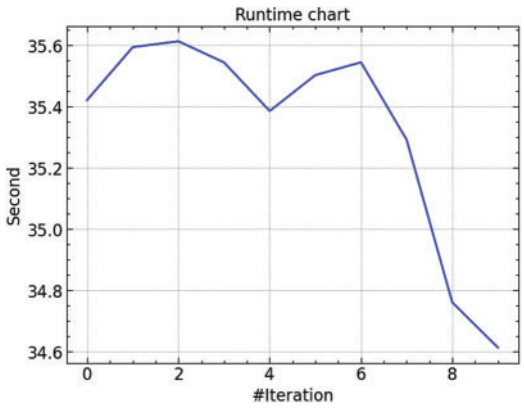


**Figure 4:** Time variation of CSA

### *4.6 Model Performance*

After hyperparameter tuning using the Cuckoo Search method and choosing the 10 most essential characteristics found by ANOVA F-test, our phishing detection model experienced extensive training and testing. The classification Report and confusion Matrix help to clarify the model's performance

criteria by offering a comprehensive perspective on its forecast accuracy. To calculate the classification report and confusion matrix, we used the following matrix:

- Accuracy is defined as the total number of correct predictions divided by the total number of predictions made:

$$Accuracy = \frac{True\ Positives\ +\ True\ Negatives}{Total\ Predictions} \tag{12}$$

- Precision for each class is the number of correct positive predictions divided by the total number of positive predictions made:

$$Precision = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives} \tag{13}$$

- Recall (or Sensitivity) for each class is the number of correct positive predictions divided by the total number of actual positives:

$$Recall = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives} \tag{14}$$

- F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics:

$$F1 - score = 2 \times \frac{Precision\ +\ Recall}{Precision\ \times\ Recall} \tag{15}$$

We used a classification report, as presented in Fig. 5, to show the performance of the proposed model with the help of precision, recall, and F1-score. Every statistic came out to be around 0.91, which suggests the model's balanced capacity to detect both classes without any notable inclination toward one. Calculated as the ratio of correctly predicted cases to the total count of occurrences, the model's general accuracy similarly came at 0.91. This homogeneity in F1-score, recall, and accuracy emphasizes the success of the hyperparameter tuning and feature selection procedures.

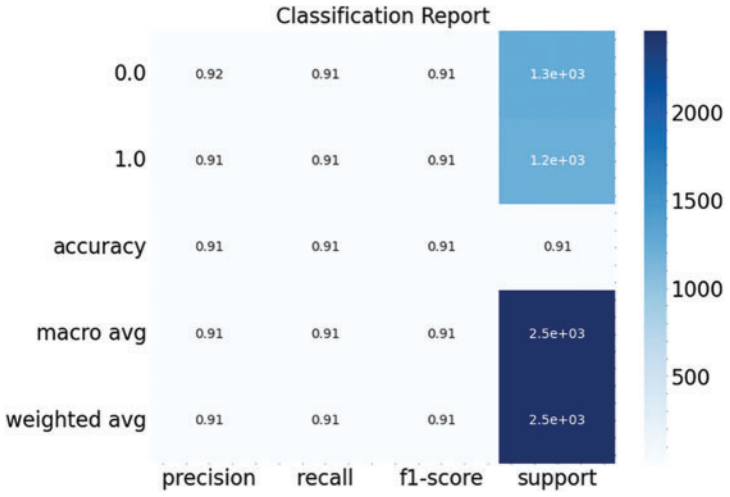


**Figure 5:** Classification report

Presented in Fig. 6, the Confusion Matrix shows the performance of the model with 1138 true positives, 1107 true negatives (attack correctly identified as attack), 114 false negatives and 104 false positives.

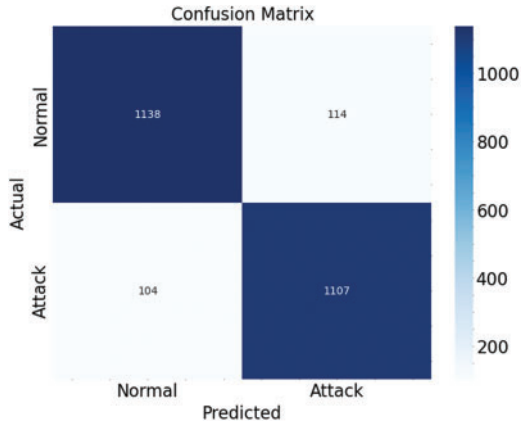


**Figure 6:** Confusion matrix

### *4.7 Comparative Analysis*

Previous sections present the details about the construction of our proposed model and the performance of the ANOVA F-test and Cuckoo Search algorithm. However, this section presents the comparative analysis of our proposed model with traditional DL/ML and past literature models.

#### *4.7.1 Quantitative Analysis*

Using the loss measure as the foundation for assessment, we evaluated our proposed model against standard machine learning and deep learning models, as represented in Fig. 7.

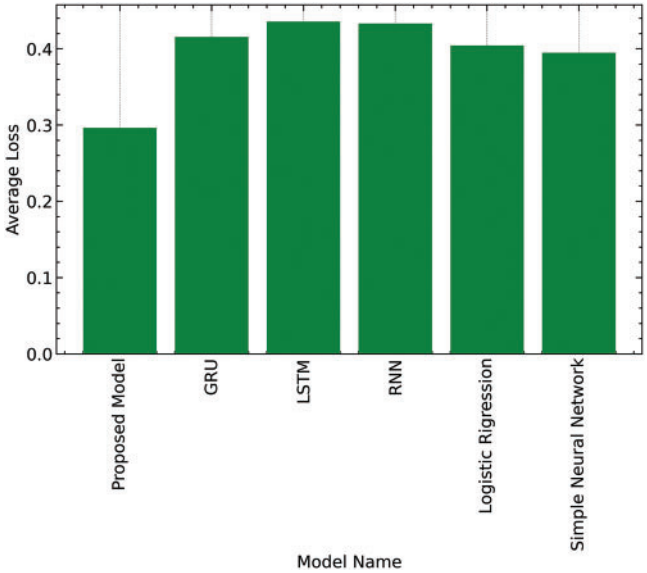


**Figure 7:** Comparative analysis

Consistently showing the lowest loss values throughout all epochs, the proposed model has better capacity to reduce the error between expected and actual results. This constant performance indicates that the model's design is very successful for the particular job of phishing detection, therefore providing a solid solution with high prediction accuracy.

By contrast, while usually suitable for sequence prediction tasks, the GRU and LSTM models showed more loss values in this regard. This result suggests that, despite their sophisticated features, these models might not be as appropriate for phishing detection as our proposed model, which seems to reflect better the complexity engaged in this particular application.

While the model learns rapidly, the Simple Neural Network showed a notable early decrease in loss, which subsequently plateaued, suggesting that it achieves its performance limit faster than the more complex designs. This behavior emphasizes the difficulty of phishing detection, which perhaps calls for the more subtle approach our suggested model offers.

Nevertheless, logistic regression and RNN had the most significant loss values among the models examined. This result underlines the superiority of the suggested model even more in this field as it implies that these models are less able to properly handle the complex patterns required for precisely separating phishing from non-phishing websites.

With its capacity to attain lower error rates and greater prediction accuracy than other conventional models, our suggested model clearly shows its usefulness in overall phishing detection. Particularly about phishing detection, the loss comparison, as shown in the figure, validates the correctness of the design decisions underpinning our model. It emphasizes their potential as a robust solution for real-world cybersecurity problems.

#### *4.7.2 Qualitative Analysis*

In this section, we presented the qualitative comparison of past work with our proposed work, the details are presented in Table 1.

**Table 1:** Qualitative analysis with past research

| Model | Technique | Feature selection | Hyper-parameter tuning | Complexity |
|---|---|---|---|---|
| Saxena et al. [17] | Random forest and BF-tree | CSA | × | High |
| Kumaresan et al. [18] | HKSVM | S-CSA | × | High |
| Brindha et al. [19] | GRU | CSA | CS | High |
| Sabahno et al. [20] | SVM | ISHO | × | High |
| Al-Sawwa et al. [21] | Apache spark-based decision tree | PSO, FFA, MVO, MFO, BAT optimization algorithm | × | High |
| Ali et al. [22] | ML | Particle swarm optimization | × | High |
| Anupam et al. [23] | SVM | × | Grey wolf optimiser | High |
| Niu et al. [24] | SVM | × | CSA | High |

(Continued)

**Table 1 (continued)**

| Model | Technique | Feature selection | Hyper-parameter tuning | Complexity |
|---|---|---|---|---|
| Kumar et al. [25] | DL | × | Adam optimizer | High |
| **Proposed model** | **CNN** | **ANOVA** | **CSA** | **Low** |

The complexity in the paper by Saxena et al. [17] arises from integrating the Cuckoo Search algorithm for feature selection with multiple classifiers (random forest and BF-tree) and evaluating their performance using a range of error metrics. The use of these different techniques and metrics requires careful tuning and evaluation, adding to the overall computational and analytical complexity.

Similarly, the complexity of the approach by Kumaresan et al. [18] stems from the integration of multiple elements: text and image-based feature extraction, hybrid feature selection using S-Cuckoo search, and the use of a hybrid kernel in SVM that blends three different kernel functions. Each of these components requires careful tuning and coordination, making the overall system more intricate and resource-intensive.

The integration of the intelligent Cuckoo Search (CS) optimization algorithm with a Gated Recurrent Unit (GRU) model, along with multi-stage pre-processing, adds significant complexity in the proposed approach by Brindha et al. [19]. This complexity arises from the need to balance feature extraction, model training, and hyperparameter tuning to achieve optimal performance. Likewise, the integration of an improved spotted hyena optimization (ISHO) algorithm for feature selection, combined with the use of support vector machines (SVM) for classification, adds complexity to the model proposed by Sabahno et al. [20]. This complexity arises from the need to finely tune the meta-heuristic algorithm parameters and ensure they effectively enhance classification accuracy.

Moreover, the complexity of the proposed approach by Al-Sawwa et al. [21] arises from the integration of five different bio-inspired meta-heuristic algorithms (PSO, FFA, MVO, MFO, BAT) for feature selection. Each algorithm requires careful tuning and evaluation, and implementing these within an Apache Spark framework adds another layer of complexity due to the need for distributed computing and optimization. Similarly, the model proposed by Ali et al. [22] is complex due to the use of PSO for feature weighting, which involves iteratively adjusting the weights of features to optimize classification performance. This process is computationally intensive and requires careful tuning of the PSO parameters, especially when applied to large datasets or in real-time detection scenarios. The complexity of the proposed approach by Anupam et al. [23] also arises from the use of four different nature-inspired optimization algorithms (Bat Algorithm, Firefly Algorithm, Grey Wolf Optimiser, and Whale Optimization Algorithm) to find the optimal hyperplane in SVM. Each of these algorithms has unique parameters that require careful tuning, adding to the computational complexity and making the approach more resource-intensive. Furthermore, the paper by Niu et al. [24] is complex due to integrating the Cuckoo Search algorithm with the SVM classifier for optimizing the Radial Basis Function (RBF) kernel parameters. This hybrid approach necessitates careful tuning and validation, which can be computationally intensive and complex to implement effectively. Lastly, the complexity of the approach by Kumar et al. [25] arises from integrating the Swarm Intelligence Binary Bat Algorithm with a deep learning neural network for phishing detection. This involves the design and training of

the neural network and the optimization process, which requires careful tuning of the algorithm's parameters to achieve the desired performance.

In contrast, our proposed approach minimizes complexity while maintaining high accuracy. Focusing on "specific technique/methodology" reduces the need for extensive hyperparameter tuning and computational overhead, making our solution more scalable and adaptable to real-world scenarios without sacrificing performance.

## 5 Conclusion

This work presents a model for phishing attack detection in an IoT environment. We used the ANOVA F-test to select the best features and the CSA to optimize key hyperparameters (learning rate, dropout) in the proposed deep CNN model. In addition, our proposed models get trained in five epochs, making it lightweight compared to the recent proposed works. Our deep CNN model detects phishing attacks with an accuracy of 91%. Furthermore, we compare the proposed model with standard ML, DL, and past research to present the effectiveness of our proposed model. However, still there is scope of improvement in our proposed model, in this context, in future, we will focus on testing the model on real-time environment.

**Acknowledgement:** The authors would like to thank Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R 343), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia for funding this research work through the Project number "NBU-FFR-2024-1092-09".

**Funding Statement:** This study was supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R 343), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia for funding this research work through the Project number "NBU-FFR-2024-1092-09".

**Author Contributions:** Final manuscript revision, funding, supervision: Brij B. Gupta, Kwok Tai Chui; study conception and design, analysis, and interpretation of results, methodology development: Akshat Gaurav, Varsha Arya; data collection, draft manuscript preparation, figure and tables: Shavi Bansal, Ahmed Alhomoud, Razaz Waheeb Attar. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** All data generated or analyzed during this study are included in this published article.

**Ethics Approval:** This article does not contain any studies with human participants or animals performed by any authors.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

[1] G. Jia, Y. Zhu, Y. Li, Z. Zhu, and L. Zhou, "Analysis of the effect of the reliability of the NB-IoT network on the intelligent system," *IEEE Access*, vol. 7, pp. 112809–112820, 2019. doi: 10.1109/ACCESS.2019.2932870.

[2] A. Almomani *et al.*, "Phishing website detection with semantic features based on machine learning classifiers: A comparative study," *Int. J. Semant. Web Inform. Syst. (IJSWIS)*, vol. 18, no. 1, pp. 1–24, 2022. doi: 10.4018/IJSWIS.

[3] A. K. Jain and B. B. Gupta, "Phishing detection: Analysis of visual similarity based approaches," *Secur. Commun. Netw.*, vol. 2017, 2017. doi: 10.1155/2017/5421046.

[4] D. Li, Q. Chen, and L. Wang, "Phishing attacks: Detection and prevention techniques," *J. Ind. Eng. Appl. Sci.*, vol. 2, no. 4, pp. 48–53, 2024.

[5] A. Abdulla and N. Jameel, "A review on iot intrusion detection systems using supervised machine learning: Techniques, datasets, and algorithms," *UHD J. Sci. Technol.*, vol. 7, pp. 53–65, 2023. doi: 10.21928/uhdjst.v7n1y2023.pp53-65.

[6] B. B. Gupta, N. A. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: Taxonomy of methods, current issues and future directions," *Telecommun. Syst.*, vol. 67, pp. 247–267, 2018. doi: 10.1007/s11235-017-0334-z.

[7] S. Li, D. Qin, X. Wu, J. Li, B. Li and W. Han, "False alert detection based on deep learning and machine learning," *Int. J. Semant. Web Inform. Syst.*, vol. 18, no. 1, pp. 1–21, 2022. doi: 10.4018/IJSWIS.

[8] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, pp. 2015–2028, 2019. doi: 10.1007/s12652-018-0798-z.

[9] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," *Telecommun. Syst.*, vol. 68, pp. 687–700, 2018. doi: 10.1007/s11235-017-0414-0.

[10] B. Biswas, A. Mukhopadhyay, A. Kumar, and D. Delen, "A hybrid framework using explainable AI (XAI) in cyber-risk management for defence and recovery against phishing attacks," *Decis. Support Syst.*, vol. 177, no. 12, 2024, Art. no. 114102. doi: 10.1016/j.dss.2023.114102.

[11] I. Gulatas, A. Zaim, and M. Aydin, "Malware threat on edge/fog computing environments from internet of things devices perspective," *IEEE Access*, vol. 11, pp. 33584–33606, 2023. doi: 10.1109/ACCESS.2023.3262614.

[12] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Comput. Commun.*, vol. 175, pp. 47–57, 2021. doi: 10.1016/j.comcom.2021.04.023.

[13] A. K. Jain and B. Gupta, "PHISH-SAFE: URL features-based phishing detection system using machine learning," in *Cyber Security*, Singapore: Springer, 2018, pp. 467–474.

[14] J. Aljabri *et al.*, "Hybrid stacked autoencoder with dwarf mongoose optimization for Phishing attack detection in internet of things environment," *Alexandria Eng. J.*, vol. 106, no. 3, pp. 164–171, 2024. doi: 10.1016/j.aej.2024.06.070.

[15] J. Song and D. Park, "Preemptive cyber response strategy and iot forensic evidence," *Int. J. Adv. Sci. Technol.*, vol. 117, pp. 129–138, 2018. doi: 10.14257/ijast.2018.117.11.

[16] M. Rytel, A. Felkner, and M. Janiszewski, "Towards a safer internet of things-a survey of iot vulnerability data sources," *Sensors*, vol. 20, 2020, Art. no. 5969. doi: 10.3390/s20215969.

[17] A. Saxena, N. Sharma, P. Agarwal, and R. Barotia, "Phishing website prediction by using cuckoo search as a feature selection and random forest and BF-tree classifier as a classification method," in *Rising Threats in Expert Applications and Solutions*, Singapore: Springer, 2020, pp. 765–776.

[18] T. Kumaresan, S. Saravanakumar, and R. Balamurugan, "Visual and textual features based email spam classification using S-Cuckoo search and hybrid kernel support vector machine," *Cluster Comput.*, vol. 22, pp. 33–46, 2019. doi: 10.1007/s10586-017-1615-8.

[19] R. Brindha, S. Nandagopal, H. Azath, V. Sathana, G. P. Joshi and S. W. Kim, "Intelligent deep learning based cybersecurity phishing email detection and classification," *Comput. Mater. Contin.*, vol. 74, no. 3, 2023. doi: 10.32604/cmc.2023.030784.

[20] M. Sabahno and F. Safara, "ISHO: Improved spotted hyena optimization algorithm for phishing website detection," *Multimed. Tools Appl.*, vol. 81, pp. 34677–34696, 2022. doi: 10.1007/s11042-021-10678-6.

[21] J. Al-Sawwa, M. Almseidin, M. Alkasassbeh, K. Alemerien, and R. Younisse, "Spark-based multi-verse optimizer as wrapper features selection algorithm for phishing attack challenge," *Cluster Comput.*, pp. 1–16, 2024. doi: 10.1007/s10586-024-04272-2.
[22] W. Ali and S. Malebary, "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection," *IEEE Access*, 2020. doi: 10.1109/ACCESS.2020.3003569.
[23] S. Anupam and A. K. Kar, "Phishing website detection using support vector machines and nature-inspired optimization algorithms," *Telecommun. Syst.*, 2021. doi: 10.1007/s11235-020-00739-w.
[24] W. Niu, X. Zhang, G. Yang, Z. Ma, and Z. Zhuo, "Phishing emails detection using CS-SVM," in *2017 IEEE Int. Symp. Parallel Distrib. Process. Appl. 2017 IEEE Int. Conf. Ubiquitous Comput. Commun. (ISPA/IUCC)*, 2017.
[25] P. P. Kumar, T. Jaya, and V. Rajendran, "SI-BBA–A novel phishing website detection based on Swarm intelligence with deep learning," *Mater. Today: Proc.*, vol. 80, pp. 3129–3139, 2023. doi: 10.1016/j.matpr.2021.07.178.
[26] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *2009 World Congr. Nature Biol. Inspired Comput. (NaBIC)*, 2009, pp. 210–214.
[27] T. E. Oliphant *et al.*, *Guide to Numpy*. USA: Trelgol Publishing, 2006, vol. 1.
[28] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan, "PyTorch," in *Programming TensorFlow: Solut. Edge Comput. Appl.*, 2021, pp. 87–104.
[29] O. Kramer and O. Kramer, "Scikit-learn," in *Mach. Learn. Evol. Strategies*, 2016, pp. 45–53.
[30] N. Van Thieu and S. Mirjalili, "MEALPY: An open-source library for latest meta-heuristic algorithms in Python," *J. Syst. Archit.*, 2023. doi: 10.1016/j.sysarc.2023.102871.
[31] E. Chand, "Phishing website Detector," 2023. Accessed: Jan. 30, 2024. [Online]. Available: https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector
[32] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinform.*, vol. 14, pp. 1–16, 2013. doi: 10.1186/1471-2105-14-106.