International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (Feb-24)

http://dx.doi.org/10.12785/ijcds/150144

An Efficient Spam and Phishing Email Filtering Approach using Deep Learning and Bio-inspired Particle Swarm Optimization

Santosh Kumar Birthriya[1], Priyanka Ahlawat[2] and Ankit Kumar Jain[1,3]

[1,2,3]*Department of Computer Engineering, National Institute of Technology, Kurukshetra, India*

Dates: Received 7 May. 2023, Revised 10 Jan. 2024, Accepted 21 Jan. 2024, Published 1 Feb. 2024

Abstract: The exponential rise in spam and phishing emails presents a critical challenge to the privacy, security, and efficiency of users. This research introduces a deep learning model with enhanced performance over existing top-tier studies. The model's strength lies in its ability to precisely classify emails into three distinct categories: legitimate (ham), unsolicited (spam), and malicious (phishing). This research employs two sophisticated feature selection techniques to enhance classification accuracy: Principal Component Analysis (PCA) and Particle Swarm Optimization (PSO). These techniques are instrumental in identifying and extracting the most informative features from the data, which are critical for the task of categorizing emails effectively. Rigorous testing has elevated the PSO-enhanced model to a position of excellence, with an accuracy rate of 99.60%. This high degree of accuracy is a testament to the strength of deep learning in the arena of email filtering. The research confirms the value of feature selection in augmenting deep learning models, laying the groundwork for innovative defenses against email threats. The study's insights offer optimistic prospects for the advancement of more resilient email systems. Utilizing the substantial computational prowess of deep learning and the precision of feature selection techniques like PCA and PSO, the research charts a course for significantly reducing spam and phishing email incidents. As such, this research marks a significant stride in digital security, equipping stakeholders with a powerful asset in the ongoing effort to safeguard against cyber threats.

Keywords: Spam email, Phishing email, Artificial neural network, Particle swarm Optimization

1. Introduction

Email spam and phishing are common issues in electronic communication that continue to cause problems for Internet users and companies [1]. These unwarranted and potentially harmful communications can have several negative consequences, such as decreased productivity, loss of sensitive information, and financial misconduct [2]. Spam, or unsolicited bulk email, is sent to multiple recipients without permission. These messages often contain advertisements, promotions, or hoaxes, which can congest inboxes, posing challenges for users to manage their emails effectively. Despite significant improvements in spam filters, fraudsters persistently devise new strategies to bypass them, presenting an enduring challenge. These strategies encompass obfuscation, employing image-based spam, and utilizing botnets to disseminate vast quantities of spam messages from diverse sources, rendering them difficult to intercept. [3]. Phishing emails, conversely, are messages sent to specific people and are meant to trick them into giving private information, like login passwords or banking details, clicking on harmful links, or downloading malware. These emails often pretend to be from real organizations, like banks or social media platforms, and use "social engineering" techniques to trick users into doing what the hackers want [4]. Phishing attacks can lead to identity theft, loss of money, and systems that aren't secure [5]. Also, spear-phishing attacks are a more advanced type of phishing that targets specific people or organizations and often uses personal information to make the attack seem more real [6], [7]. Given the ever-evolving nature of spam and phishing emails, there is a continuous need for more sophisticated and accurate detection techniques. In recent years, deep learning and bio-inspired particle swarm optimization approaches have emerged as promising solutions to improve email filtering and protect users from these threats [8], [9].

Figure 1 shows that phishing attacks in 2022 reached a record high. The Anti-Phishing Working Group (APWG) reported over 4.7 million such attacks. This represents a significant increase, exceeding 150% annually since 2019. October 2022 experienced the highest number of unique phishing emails in a single month, with APWG noting 101,104 incidents. In the final three months of 2022, the number of attacks rose slightly to 1,350,037, up from 1,270,883 in the preceding three months [10].

In Figure 2, the United States was identified as the country

E-mail address: cs.santosh1b@gmail.com, ankitjain@nitkkr.ac.in, priyankaahlawat@nitkkr.ac.in

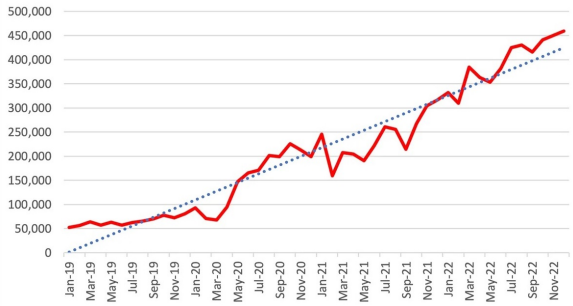

Figure 1. Phishing Attacks Recorded Between 2019 to Q4 2022[10].

sending the highest number of spam emails in a single day worldwide, with an estimated volume of approximately eight billion on January 16th, 2023. This figure positioned it as the top offender in terms of spam email volume. The second and third-ranked countries were Czechia and the Netherlands, respectively, with 7.7 billion and 7.6 billion spam emails sent on the same day [11]. It is important to note that these figures may vary depending on the source and data collection methodology. Nonetheless, they underscore the persistent issue of spam emails globally and the imperative for effective measures to combat this form of unwanted electronic communication [11].

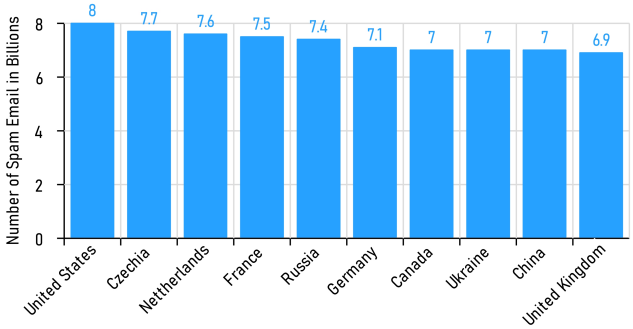

Figure 2. Global Daily Spam Email Count by Country (January 16th, 2023) [11].

Email communication is crucial for personal and professional interactions, providing efficient and convenient information exchange. Deep learning algorithms have shown great potential in filtering spam and phishing emails, ensuring email security when trained with high-quality, large-scale data.

Addressing the challenge of high-dimensional email data, bio-inspired Particle Swarm Optimization (PSO) algorithms can optimize deep learning model parameters, improving accuracy and efficiency in email filtering systems. This research explores the benefits of combining deep learning algorithms with PSO optimization techniques for efficient email filtering, ultimately contributing to a more secure and reliable email communication system.

Contribution: The suggested deep learning and particle swarm optimization-based email filtering method makes the following contributions to the field:

- Increased accuracy: This method uses deep learning models' capacity to recognize complex patterns, and particle swarm optimization improves the identification of anomalies, leading to improved accuracy compared to conventional techniques.
- Fast processing: The use of deep learning models, known for their efficiency in processing large datasets, combined with the parallel computing abilities of PSO, greatly speeds up the email filtering process. This efficiency is significant given the vast volume of emails that need to be processed in real-time.
- Tri-categorical classification: Our approach goes beyond the conventional binary classification of emails into spam and non-spam. It introduces a tri-categorical classification system that sorts emails into ham (non-spam), spam, and phishing categories. This is particularly relevant today, where phishing attacks are becoming more sophisticated and must be distinctly identified from regular spam.
- Scalability and applicability: Recognizing the exponential growth in spam and phishing emails, your method's ability to scale and handle large volumes of data while maintaining high accuracy is a significant advancement. This scalability ensures that the system remains effective even as the importance of emails continues to grow.
- Emphasis on optimization: The application of PSO, an optimization technique inspired by natural processes, highlights the importance of integrating traditional optimization strategies with modern deep learning methods. This blend showcases the potential of hybrid approaches in enhancing the performance and efficiency of email filtering systems.

The paper is divided into the following sections. Section 2 reviews earlier research on email filtering. Section 3 introduces a new method for identifying ham, spam, and phishing emails using deep learning and particle swarm optimization. Section 4 presents the experimental results. Section 5 concludes the paper and discusses possible future research in this field.

2. Related Work

The widespread use of email communication has led to an ever-increasing number of spam and phishing emails. These unsolicited messages cause irritation, waste time, and pose significant security risks if not adequately addressed. This section provides a review of related work in the areas of deep learning and bio-inspired optimization techniques for spam and phishing email filtering.

Zhang et al. (2014) [12], the experimental dataset, comprising 6,000 emails from 2012, underwent a Kolmogorov–Smirnov test, revealing significant results in capital-run-length features. An alpha value of 7 proved most effective in your model. Among several meta-heuristic algorithms, the MBPSO outperformed others like GA, RSA, PSO, and BPSO in classification efficiency. The decision tree enhanced by MBPSO feature selection exhibited high sensitivity (91.02%), specificity (97.51%), and accuracy (94.27%). Compared to traditional methods like SFS and SBS, MBPSO showed superior performance. Furthermore, your study indicated that wrapper methods surpass filter methods in classification indices, effectively reducing false positives without affecting sensitivity or accuracy.

Idris et al. (2014) [13], the developed model combines the Negative Selection Algorithm (NSA) with Particle Swarm Optimization (PSO) to enhance random detector generation. This model employs stochastic distribution for data modeling and introduces the Local Outlier Factor as a fitness function. This function identifies candidate detectors' local optimum (Pbest), ensuring they effectively distinguish between non-spam and spam. Distance measurement further improves detector distinctiveness. The comparative analysis demonstrates this hybrid NSA-PSO model's superior detection rate (91.22%) over the standard NSA (68.86%), particularly evident in a test with 2000 detectors and a threshold of 0.4.

Smadi et al. (2015) [14] proposed an intelligent method for phishing email detection that included a preprocessing step to extract information from different email segments. The J48 algorithm was used to classify the 23 features drawn from existing literature, with ten-fold cross-validation employed for training, testing, and validation. Their primary goal was to enhance email classification metrics by optimizing preprocessing and determining the best approach. The random forest method achieved the highest accuracy of 98.87% for a legitimate dataset.

Agarwal et al. (2018) [15] combined Particle Swarm Optimization (PSO) with the Naive Bayes (NB) classifier for email filtering. Their findings indicated that this composite approach surpassed the standalone NB in metrics such as accuracy, F1-score, recall, and precision.

In 2019, Taloba et al. [16] explored the synergy between Genetic Algorithm (GA) optimization and Decision Tree (DT) classifiers to address overfitting in high-dimensional feature spaces. They applied Principal Component Analysis (PCA) for feature extraction and aimed to pinpoint the optimal parameter settings for the DT. Their J-48 DT algorithm was combined with a fitness function to enhance accuracy. On the Enron spam dataset, their GA-DT model outperformed other classifiers without the use of PCA.

Dedeturk et al. (2020) [17] assessed various classification techniques, including Gaussian Naive Bayes, Linear Support Vector Machine, Radial Basis Function SVM, Multinomial Naive Bayes, Logistic Regression via gradient descent, and Logistic Regression integrated with Artificial Bee Colony optimization. They introduced a model with 1000 features, achieving an accuracy of 98.7%. This underscored the significance of feature selection in machine learning classification.

Talaei et al. (2020) [18] introduces a novel approach to enhance spam detection using artificial neural networks (ANNs) by incorporating a feature selection method based on the sine-cosine algorithm (SCA). Traditional ANN methods for spam detection often face errors due to the inclusion of all features in the training phase. The SCA is utilized to refine the feature vectors, selecting the most effective features to train the ANN. When applied to the Spambase dataset using MATLAB, this method achieved impressive results: 98.64% precision, 97.92% accuracy, and 98.36% sensitivity. This performance surpasses that of other classifiers like multilayer perceptron (MLP) neural networks, Bayesian networks, decision trees, and random forests in spam detection tasks. Specifically, the incorporation of SCA led to a reduction in feature selection error by about 2.18% in the MLP neural network, according to our testing outcomes.

Rodrigues et al. (2021)[19] investigated the use of transfer learning to detect spam and phishing emails. They employed the pre-trained deep learning model, BERT (Bidirectional Encoder Representations from Transformers), and fine-tuned it to classify emails based on their textual content. Depending on the input data, the model used a variable number of features and achieved an impressive accuracy of 99.0% Mughaid et al. (2022) [20] developed a machine-learning algorithm to distinguish between phishing and legitimate emails. The dataset was segmented into training and testing sets, and the model's performance was assessed on three separate datasets with different numbers of features. The results showed that the best accuracy and performance were obtained using the dataset with the most features. For clarity, the first dataset had 22 features, the second contained 50 features, and the third was solely based on textual features.

Alshingiti et al. (2023) [21] introduced three methods to improve phishing detection. Among them, the CNN (Convolutional Neural Network) technique achieved the top accuracy at 99.2%. In comparison, the LSTM (Long Short-Term Memory) and the combined LSTM-CNN models delivered accuracies of 96.8% and 97.6%, respectively. Given its superiority in text classification tasks, efficiency, and computational speed, CNN emerged as the most favorable choice for phishing detection.

Table I compares various spam and phishing email detection methods from different studies, detailing their methodologies, advantages, and disadvantages. The approaches focus on enhancing accuracy, with some reaching up to 99.2%, while also addressing challenges such as feature selection and model complexity.This study proposed an Artificial Neural Network (ANN) with PSO (21 Features) for classification tasks. The proposed approach achieved an accuracy of 99.60%. ANNs are a popular method for classification tasks that mimic the structure and function of the human brain.

TABLE I. Comparison of Various Methods for Detecting Spam and Phishing Emails in Related Works

Author	Methodology	Advantages	Disadvantages
Zhang et al. (2014) [12]	Decision tree enhanced by MBPSO	The approach boasts high accuracy and effective feature selection	It may be computationally intensive and potentially less generalizable to different datasets.
Idris et al. (2014) [13]	Hybrid NSA-PSO model	NSA-PSO hybrid model offers a significantly higher detection rate and improved accuracy (99.2%) in distinguishing spam.	It may require more computational resources and complexity compared to the standard NSA.
Smadi et al. (2015) [14]	J48 Decision Tree Classifier	High accuracy (98.87%), Preprocessing step to extract features	Dependent on feature selection, Requires manual feature extraction
Agarwal et al. (2018) [15]	Particle Swarm Optimization + Naive Bayes	Improved accuracy, recall, F1-score, and precision	Requires Correlation Feature Selection (CFS) for selecting most relevant features
Taloba et al. (2019) [16]	Genetic Algorithm + Decision Tree + Principal Component Analysis	Better accuracy, Addresses overfitting issue, Optimal DT parameter settings	Requires multiple preprocessing steps, Complexity of combining multiple techniques
Dedeturk et al. (2020) [17]	Multiple Classifiers (Gaussian NB, Linear SVM, RBFSVM, MultiNomial NB, LR by gradient descent, Artificial Bee Colony with LR)	1000-Feature Model, High accuracy (98.7%)	Complexity of testing multiple classifiers, Feature selection critical
Talaei et al. (2020) [18]	ANN	The proposed method enhances ANN spam detection accuracy by selectively training on optimal features using the sine-cosine algorithm (SCA), resulting in reduced error rates.	The approach may require additional computational resources and complexity for implementing and optimizing the SCA feature selection process.
Rodrigues et al. (2021) [19]	BERT Transfer Learning for Text Classification	High accuracy (99.0%), Dynamic number of features, Leveraging pre-trained model	Requires fine-tuning, Dependent on quality of pre-trained model
Mughaid et al. (2022) [20]	Machine Learning-based Detection Algorithm	High accuracy with more features, Testing with different datasets	Performance varies with different feature sets
Alshingiti et al. (2023) [21]	CNN, LSTM, LSTM-CNN Hybrid	Best accuracy with CNN (99.2%), Effective processing of sequential data	Complexity of implementing multiple techniques, Comparison of speed and performance

3. Proposed Methodology

Our research is focused on developing a neural network classifier capable of accurately categorizing emails as ham, spam, or phishing. A priority is placed on optimizing hyperparameters to ensure the classifier operates in real-time and achieves high validation accuracy on new data. The proposed framework, illustrated in Figure 3, comprises preprocessing, feature extraction, and feature selection stages, which prepare the input data for training the neural network. During the learning process, features of the input data are fed into the network through the input layer, processed across the hidden layers. Finally, the output layer classifies the type of email. Subsequent sections provide in-depth explanations of each stage.

A. Datasets

To execute the proposed method, we used three different and essential datasets: the UCI Machine Learning Repository [22], the CSDMC2010 Spam Corpus [23] and the SpamAssassin Public Corpus [24]. The UCI and CSDMC databases offer spam and ham emails, while the SpamAssassin dataset adds phishing emails to our collection, as shown in Figure 4. By adequately preparing these emails, we can extract essential aspects often present in spam and phishing emails, such as JavaScripts, HTML tags, and appealing URLs targeted to interest visitors. This comprehensive approach ensures a robust and compelling analysis for improving email security and user experience.

B. Data Preprocessing

Data preprocessing is a fundamental step in developing effective email filtering systems, especially in the context of increasing cyber threats. Preprocessing primarily aims to transform raw email text into a standardized, analyzable format. Here's a detailed look at the key preprocessing techniques:

- Parsing email content: This involves breaking down the email's components, such as the body, subject, sender, and recipient. Parsing simplifies the text by structuring and separating different parts, making it more accessible for analysis.
- Tokenization: In this step, the email text is divided into smaller units called tokens, words, phrases, or sentences. Tokenization helps represent the text in a way that's easier to process and analyze.
- Stemming: This technique reduces words to their root form by removing suffixes. For example, "running" becomes "run." Stemming helps in decreasing

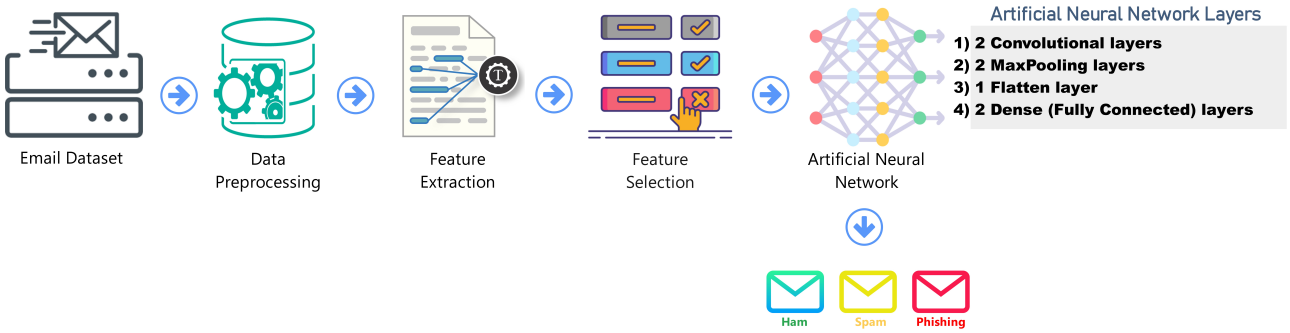

Figure 3. Architecture of Proposed Methodology.

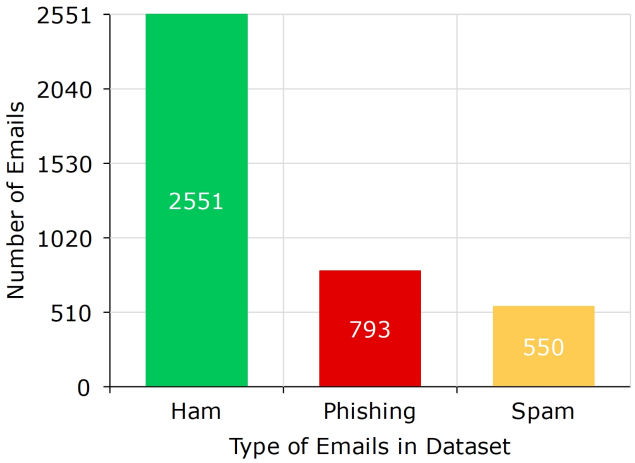

Figure 4. Type of Emails in Datasets.

the text's dimensionality, simplifying the subsequent analysis.

- Lemmatization: Similar to stemming, lemmatization converts words to their base or dictionary form (lemma). For instance, "went" is changed to "go." This technique also aids in reducing text dimensionality and enhances consistency in the text.
- Case folding: This process involves converting all characters in the text to a uniform case (either uppercase or lowercase). It's essential for maintaining consistency and reducing text complexity.
- Error correction: Spelling and typographical errors are addressed through similarity scoring, which compares the intended words with the actual spelling. Words a

These preprocessing steps are crucial for transforming raw email data into a more consistent and manageable format for analysis. By simplifying the text and reducing its dimensionality, these techniques significantly enhance the performance and effectiveness of email filtering systems.

C. Feature Extraction

Feature extraction is crucial in machine learning to highlight important information from raw data, enhancing model efficiency and accuracy. In an email ham, spam, and phishing detection study, 40 features were divided into body-based and subject-line-based groups, including boolean, numerical attributes, and keyword patterns. Properly extracting these features ensures the model concentrates on relevant data, improving accuracy and outcomes.

Feature Extraction using PCA: In Algorithm 1, PCA efficiently processes high-dimensional email data to differentiate between ham, spam, and phishing categories. It reduces dimensions and aids in feature extraction for email classification. PCA identifies the main components that capture the most significant variance by analyzing the entire dataset. These components, derived from the email data's covariance matrix, become the new features. Though they might not mirror the original email attributes, they contain the core information, streamlining the distinction between ham, spam, and phishing emails [25].

Feature Extraction using PSO: Algorithm 2, PSO simulates the collective behavior of swarming entities to solve optimization problems. In email categorization, each particle, denoted as $\mathbf{x}_i$, stands for a potential solution or a feature subset in the email data. These subsets could represent patterns or markers typical of spam, or phishing emails. By adjusting their trajectories based on individual memory, $\mathbf{p}best_i$, and the collective best-known position, $\mathbf{g}best$, the particles zero in on the most pertinent features for classifying emails. Parameters c_1 and c_2 are vital for harmonizing personal and collective learnings, enabling comprehensive email data exploration and exploitation [9], [26].

The dataset initially contained 40 features. After applying PCA feature selection, the dataset was reduced to 35 available features. Furthermore, the dataset was further reduced to 21 available features after using PSO.

D. Constructing Deep Neural Networks

In constructing deep neural networks (Artificial Neural Networks - ANN) for categorizing emails into three categories: ham (legitimate), spam (unsolicited), or phishing (deceptive), this method implements a deep learning model trained on features extracted using Particle Swarm Opti-

Algorithm 1 Email Feature Extraction using PCA

Require: X: an $n \times p$ matrix containing n email samples and p feature vectors
Require: k: number of principal components
Require: Classifier: A pre-defined classification model (e.g., SVM, Naive Bayes, etc.)
Ensure: Y: an $n \times k$ matrix containing the transformed data with k principal components
Ensure: ClassificationLabels: an array of size n containing labels (ham, spam, phishing) for each email
Feature Extraction from Emails:
Convert each email into a feature vector (e.g., using TF-IDF, word embeddings, etc.)
Store the feature vectors in X
Standardize the data:
Compute the mean of each feature vector and subtract it from the corresponding feature values in X
Compute the covariance matrix:
Compute the transpose of X
Multiply X by its transpose
Divide the result by $n - 1$
Compute the eigenvectors and eigenvalues:
Compute the eigenvalues and eigenvectors of the covariance matrix
Sort the eigenvalues in decreasing order
Select the k eigenvectors corresponding to the k largest eigenvalues
Transform the data:
Multiply the transpose of the selected eigenvectors with the transpose of X to obtain the transformed data Y
Classify Emails:
Train the Classifier using the transformed data Y and known labels (ham, spam, phishing)
Predict the labels of the email samples in Y using the trained Classifier
Store the predicted labels in ClassificationLabels
Output:
Return the transformed data Y, the eigenvectors, and the ClassificationLabels

Algorithm 2 Email Feature Extraction using PSO

Require: Email feature extraction function $f(\mathbf{x})$
Require: Number of particles P
Require: Maximum iterations M
Require: Inertia factor w
Require: Individual acceleration constant k_1
Require: Group acceleration constant k_2
Require: Search region limits $\mathbf{x}min$, $\mathbf{x}max$
Ensure: Best global classifier parameters $\mathbf{g}_{finest}$
Extract email features for classification and initialize random positions $\mathbf{x}_i$ and velocities $\mathbf{v}_i$ for particles within search region limits
for $m = 1$ to M **do**
  for $i = 1$ to P **do**
    Compute classification accuracy for particle i: $f(\mathbf{x}_i)$
    if $f(\mathbf{x}_i) > f(\mathbf{p}_{top_i})$ **then**
      Refresh personal top position: $\mathbf{p}_{top_i} = \mathbf{x}_i$
    end if
    if $f(\mathbf{x}_i) > f(\mathbf{g}_{finest})$ **then**
      Refresh finest global position: $\mathbf{g}_{finest} = \mathbf{x}_i$
    end if
  end for
  for $i = 1$ to P **do**
    Modify particle i velocity:

$$\mathbf{v}_i = w \cdot \mathbf{v}_i + k_1 \cdot r_1 \cdot (\mathbf{p}_{top_i} - \mathbf{x}_i) + k_2 \cdot r_2 \cdot (\mathbf{g}_{finest} - \mathbf{x}_i)$$

    Update particle i classifier parameters:

$$\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i$$

    Apply boundary constraints to $\mathbf{x}_i$ if needed
  end for
end for
Output the best classifier parameters $\mathbf{g}_{finest}$

mization (PSO). The model comprises several layers, each serving a specific purpose. Here is an explanation of each layer in the ANN architecture:

1) Convolution2D Layer: The convolution operation, symbolically represented as $*$, slides a filter over an input feature map to extract features useful for distinguishing between different email types. The convolution operation for feature extraction is given by:

$$(I * F)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) \cdot F(x - i, y - j) \quad (1)$$

where:
- I is the input feature map, representing the encoded email.
- F is the filter or kernel used for convolution.
- x and y are spatial coordinates on the feature map.

2) Activation Function: ReLU: After convolution, the resultant feature map values are passed through an activation function to introduce non-linearity. The activation function is:

$$ReLU(x) = \max(0, x) \quad (2)$$

where:
- x is the input to the function.

This activation function helps the model learn complex patterns, which can be crucial in email classification where emails might contain intricate wordings to deceive the receiver.

3) MaxPooling2D Layer (Downsampling): Max pooling operation reduces the spatial dimensions, making

the model faster and more invariant to small translations. MaxPooling operation is represented as:

$$MaxPooling(R) = \max_{x,y \in R} R(x, y) \quad (3)$$

where:

- R represents a region of the input feature map.
- x and y are spatial coordinates within region R.

For email classification, this step can help in retaining only the most essential features, which can be indicative of an email being ham, spam, or phishing.

4) Flatten Layer: The Flatten layer reshapes a 2D feature map to a 1D vector for the subsequent dense layers.
5) Dense Layer (Fully Connected Layer): Here, the input from the Flatten layer is transformed using weights and biases:

$$y = Wx + b \quad (4)$$

where:

- y is the output vector.
- W represents the weights matrix.
- x is the input vector.
- b is the bias vector.

In the context of email classification, this layer helps in making decisions based on the extracted features, determining whether the email is ham, spam, or phishing.

6) Softmax Activation (Classification): For classification into the three classes (ham, spam, phishing), the softmax function is:

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (5)$$

where:

- z_i is the input vector's ith element.
- K represents the number of classes, in this case, 3 (classes: ham, spam, and phishing).

E. Training Deep Neural Networks

This research proposes a deep learning model combined with PSO for identifying malicious activities using supervised learning. The neural network node weights are iteratively adjusted in each cycle to minimize the error. The 'Adam' optimizer was employed for efficient weight optimization during training, with a learning rate of 0.001. The training procedure incorporated batch learning with a batch size of 16 and a predetermined epoch count of 180.

F. Categorical Cross-entropy

The loss function measures the disparity between the true labels and the predictions in multi-classification tasks:

$$L(y, \hat{y}) = -\sum_{i}^{c} T_i \log(\hat{y}_i) \quad (6)$$

Where:

- $L(y, \hat{y})$: Value of the loss function.
- T_i: True label for the i^{th} class.
- $\hat{y}_i$: Model's predicted probability for the i^{th} class.

The aim is to minimize this loss, typically using gradient descent methods.

G. Adam Optimization

Adam, an advanced optimizer for deep learning, improves upon SGD by integrating features from AdaGrad and RMSProp. It adjusts the learning rate based on the gradients' second moments. The weight update is given by:

$$Correction = \alpha \frac{\delta i}{\delta \theta i} \quad (7)$$

Incorporating momentum:

$$Correction = \gamma \times PreviousCorrection + \alpha \frac{\delta i}{\delta \theta i} \quad (8)$$

Where:

- α: Learning rate.
- δi: Variation in the cost function.
- $\delta \theta i$: Variation in the weight parameter.
- γ: Coefficient of momentum, denoting prior correction's influence on the current one.

The value of i is then updated as shown in Equation 8. The suggested value of γ should be gradually increased from 0.5 to 0.9.

In conclusion, each layer ensures the transformation of raw email data into meaningful patterns, which are then classified as ham, spam, or phishing.

4. Experiment Results

Hardware: Experiments were conducted on an ASUS ROG Strix G17 with specifications including a 5.2 GHz AMD Ryzen 9 7845HX CPU, 16GB DDR5 RAM, and a 6GB RTX 4050 GPU.

Software and libraries: Python 3.11.5 is the programming language used. Python is popular in data science and machine learning due to its simplicity and the vast ecosystem of libraries available.

An essential evaluation tool for classification models is the confusion matrix, which provides metrics like True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). These are defined as:

- True Positives (TP): Both the actual and predicted classifications are true.
- True Negatives (TN): Both the actual and predicted classifications are false.
- False Positives (FP):The prediction is true, whereas the actual class is false.

- False Negatives (FN): The prediction is false, but the actual class is true.

From these values, essential performance metrics are derived:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

The performance of various classification methods, Deep Learning with all features, PCA processed features, and PSO processed features, is demonstrated in a table II using metrics such as precision, recall, F-score, and accuracy.

A. *Experiment 1 - Deep learning algorithm with all features*

Deep Learning algorithm implemented for a classification task demonstrates strong performance across several key metrics, as shown in Figure 5. The model has an impressive accuracy of 91.68% , indicating its ability to correctly classify most of the test data. With a precision of 88.43%, the model accurately identifies positive class instances. Furthermore, the model attains a notable recall of 80.23%, effectively recognizing the positive class in the dataset. The model's F-score, a balanced measure of precision and recall, stands at 82.25%, underscoring the model's overall efficacy in classifying both positive and negative classes.

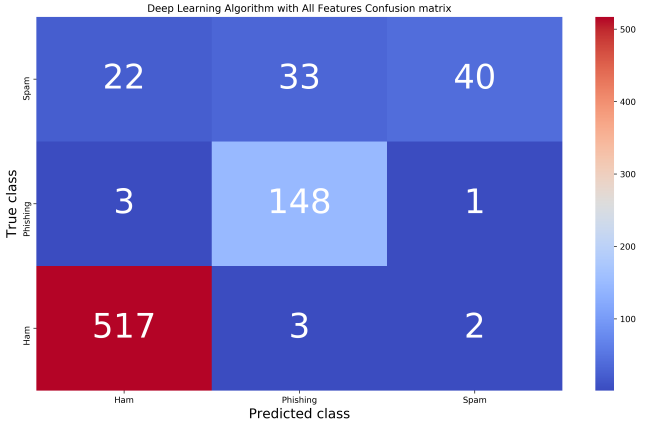

Figure 5. Deep Learning Algorithm with All Features Confusion matrix.

In Figure 6 demonstrates that the initial training accuracy of the DL with All Features model is 68.44%. As training progresses, this accuracy rises, peaking at 93.94% in epoch 18. While generally lower, the validation accuracy mirrors this trend and reaches its highest at 94.15% in epochs 18 and 19. However, there are moments when the validation accuracy dips despite a rise in training accuracy, such as in epochs 4 and 12. This showcases the model's learning ability and generalization on unseen data.

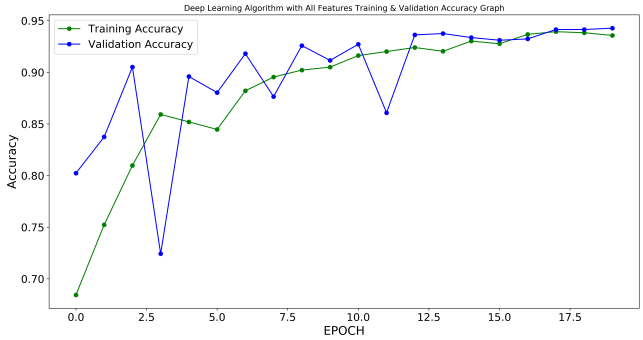

Figure 6. Training and Validation Accuracy with All Features.

In Figure 7, the performance of the deep learning (DL) with All Features model is analyzed across various epochs. Initially, between epochs 1 and 3, the model shows considerable improvement, with the training loss decreasing from 18.2 to 2.1 and the validation loss dropping from 3.3 to 0.4, indicating effective learning and generalization. However, at epoch 4, there is a noticeable increase in validation loss, suggesting a potential overfitting issue. Fortunately, the model stabilizes and improves in subsequent epochs, overcoming this overfitting tendency. By epoch 9, the validation loss decreases significantly, indicating a regained efficiency in generalization. From epochs 10 to 14, the model continues to improve steadily, with minor fluctuations in validation loss. In the final stages, from epochs 15 to 18, both training and validation losses remain broadly consistent, suggesting that the model has achieved a stable and optimal learning state, having extracted maximal insights from the training data.

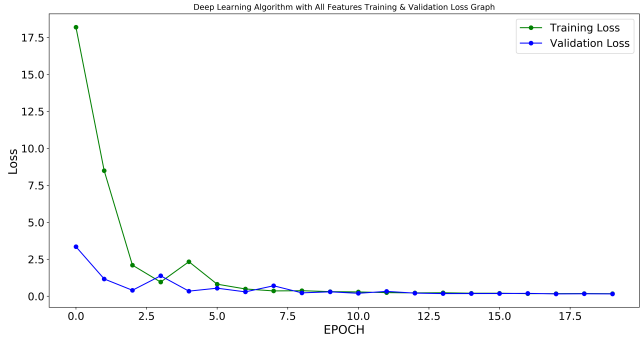

Figure 7. Training and Validation Loss with All Features.

B. *Experiment 2 - Deep learning algorithm with PCA features*

The Deep Learning model, integrated with PCA for feature selection, operates efficiently with a reduced feature set of 35. Using categorical_crossentropy as the loss function, Adam optimizer, and evaluating through accuracy metrics. The depicted confusion matrix in Figure 8 reveals

TABLE II. Performance Comparison of Deep Learning Algorithms: Test Results

Algorithm Name	Features	Accuracy	Precision	Recall	F-Score
Deep Learning (All Features)	40	91.677503	88.426217	80.234406	82.248226
Deep Learning (PCA Features)	35	94.278283	91.770529	90.659603	91.125554
Deep Learning (PSO Features)	21	99.609882	99.373220	99.511622	99.442079

the DL with PCA Features model exceptional classification performance, showcasing an accuracy of 94.28%, precision of 91.77%, recall of 90.65%, and an FSCORE of 91.12%.

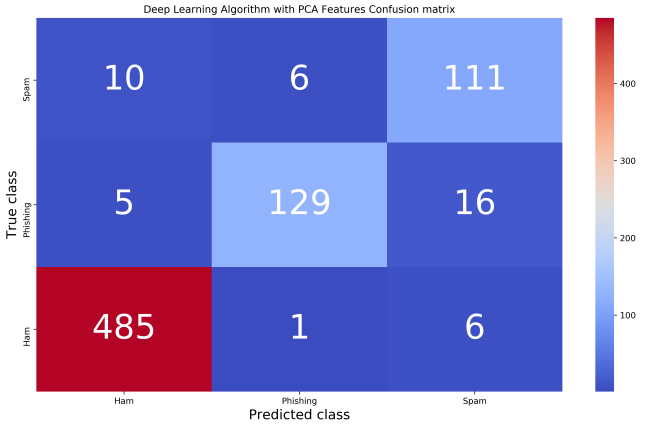

Figure 8. Deep Learning Algorithm with PCA Features Confusion matrix.

Figure 9 illustrates the performance of the DL-PCA model on the training and validation accuracy. Initially, it reports an accuracy of 93.53%. Over 20 epochs, there's a notable improvement increases to 94.20%. This pattern indicates effective learning and enhanced ability in classifying training data as the model progresses through the epochs.
Conversely, Figure 10 depicts the model's performance on the validation set. It starts with a loss of 0.1188 and an accuracy of 94.20% in the first epoch. By the final epoch, there's a slight improvement, with the loss decreasing to 0.1150 and accuracy marginally rising to 94.27%. These results demonstrate the DL-PCA model's robustness and high accuracy, consistently achieving rates between 93% and 94% on both training and validation datasets, indicating its reliability and effectiveness in classification tasks.

C. *Experiment 3 - Deep learning algorithm with PSO features*

The deep learning model applied Particle Swarm Optimization (PSO) for feature selection. PSO is a metaheuristic optimization approach to find the optimal parameter combination for a specific problem. In this instance, PSO was utilized to identify the most relevant features within the dataset, resulting in the recognition of 21 significant features. As displayed in Figure 11, the confusion matrix demonstrates excellent performance across key metrics such as accuracy, precision, recall, and F-score. The model achieved an impressive accuracy of 99.60% , indicating its ability to classify most input data points correctly.

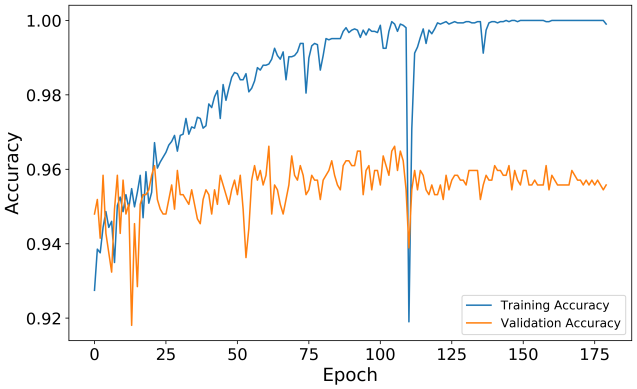

Figure 9. Training and Validation Accuracy for DL-PCA.

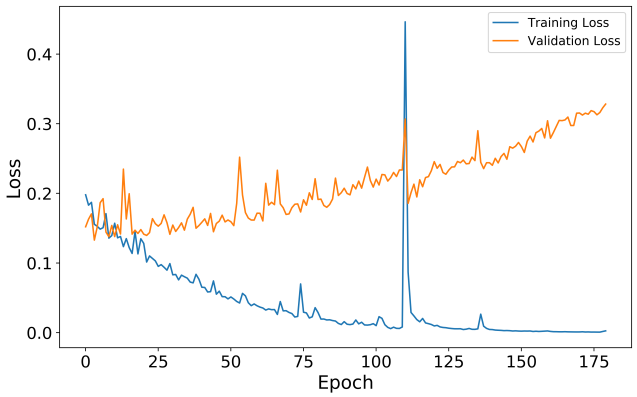

Figure 10. Training and Validation Loss for DL-PCA.

Additionally, a precision score of 99.35% highlights the model's remarkable proficiency in accurately identifying true positive instances among all predicted positives.
Figure 12 demonstrates the progression of the DL-PSO model's accuracy over the course of training. Initially, the model starts with low accuracy, signaling poor performance in the early training stages. However, as training progresses through the epochs, a clear and consistent improvement in accuracy is observed. This gradual enhancement is particularly notable around epoch 61, where the model achieves a significant accuracy level of 96.69%. The trend of increasing accuracy continues, and by the time the model reaches epoch 180, it attains an impressive accuracy of 99.83%. This upward trajectory is not only limited to training accuracy but is also mirrored in the validation accuracy. The consistent improvement in validation accuracy indicates that the model is effectively generalizing its learning to

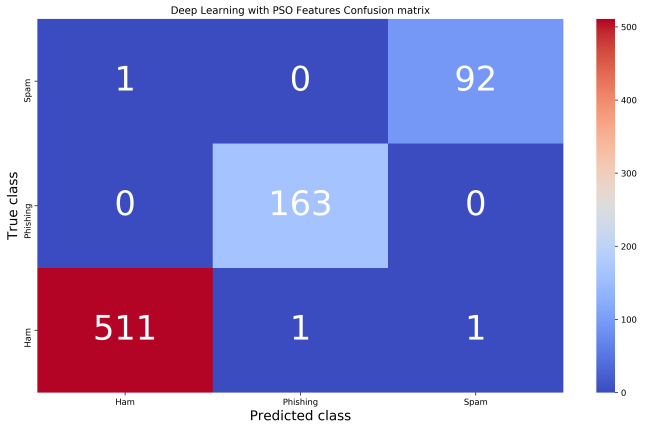

Figure 11. Deep Learning Algorithm with PSO Features Confusion matrix.

new, unseen data. Overall, the DL-PSO model exhibits a remarkable advancement in performance as it undergoes more epochs, demonstrating its learning and adaptation capabilities.

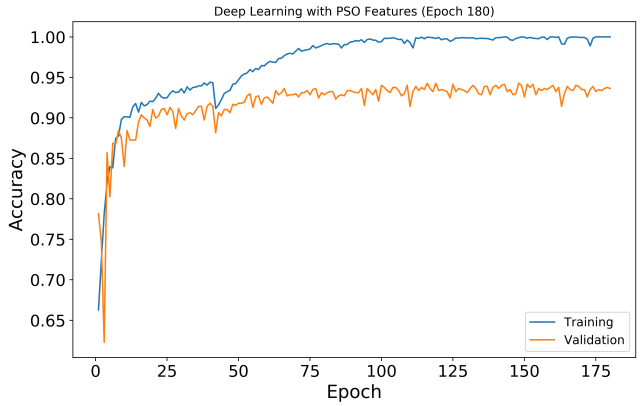

Figure 12. Training and Validation Accuracy for DL-PSO.

Based on Figure 13, the DL-PSO model initially started with high training and validation losses in the first epoch, indicating it was performing poorly. We can see that the validation loss decreases rapidly in the initial epochs, indicating that the model is learning and improving. However, after epoch 20, the validation loss seems to stabilize, and the model is no longer improving as much as it did in the initial epochs. There is a slight increase in the validation loss in epochs 2, 7, 42, and 55. The model performs well, with a final validation loss of 0.229.

Figure 14 shows that overall, these results suggest that the Deep Learning model with PSO Features is a powerful and accurate tool for classification tasks, and it can be a good choice for various real-world applications.

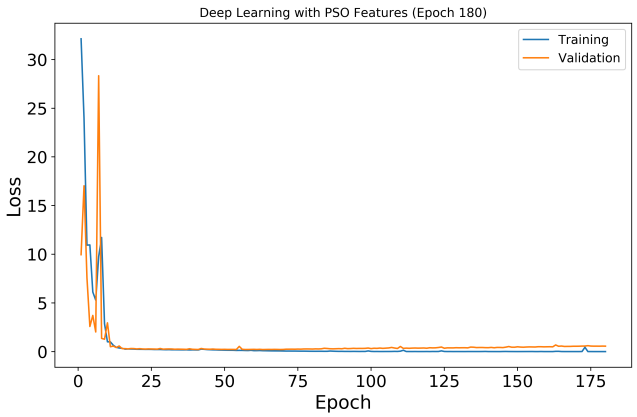

Figure 13. Training and Validation Loss for DL-PSO.

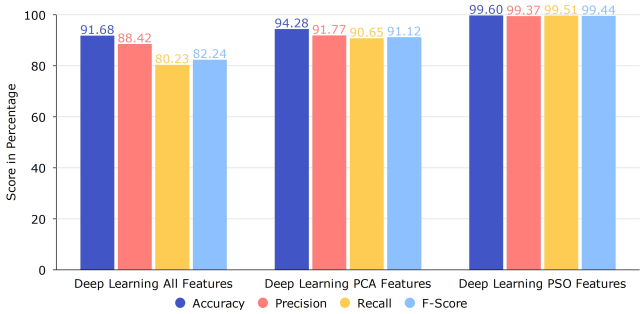

Figure 14. Deep Learning All Algorithm Performance.

D. Comparative Analysis of Our Approach with Existing Methods

Table III provides a comparative analysis of various spam and phishing detection methods, showcasing authors' different methods and their respective performances on the given datasets. In this analysis, the study by Agarwal et al. (2018) applied the Ling-Spam corpus with a Particle Swarm Optimization (PSO) integrated with Naive Bayes (NB) classifier, achieving an accuracy of 95.50%. Similarly, Taloba (2019) utilized a Genetic Algorithm (GA) combined with Decision Trees (DT) on the Enron spam dataset, achieving comparable accuracy levels. Differing from these, Mughaid's (2022) research employed neural networks on a custom dataset, resulting in varying accuracy for phishing and spam detection. More recently, Alshingiti's (2023) study took advantage of deep learning classifiers like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, showing especially high performance with CNN, achieving an accuracy of 99.2%. However, our method, which utilizes Deep Learning (DL) with PSO for feature selection, surpasses these approaches, reaching a remarkable accuracy of 99.60%, along with high precision and recall. This underscores the effectiveness of combining DL with PSO features in accurately detecting spam and phishing emails.

TABLE III. Comparative Analysis of Our Method with Existing spam and Phishing Detection Methods

Author Name	Dataset Used	Classifier	Accuracy(%)	Precision(%)	Recall(%)
Agarwal et al.(2018) [15]	Ling-Spam corpus	PSO with NB	95.50	96.42	94.50
Taloba et al.(2019) [16]	Enron spam dataset	GA with DT	95.50	95.50	97.20
Mughaid et al.(2022) [20]	Custom (Phishing+Ham)	Neural network	80.66	88.89	69.95
	Custom (Spam+Ham)	Neural network	97.7	96.4	89.3
Alshingiti et al.(2023) [21]	Custom	CNN	99.2	99	99.2
		LSTM	96.8	95.9	97.5
		LSTM-CNN	97.6	96.9	98.2
Our Method	Custom (Phishing+Spam+Ham)	DL-All Features	91.68	88.42	80.23
		DL-PCA Features	94.27%	91.77	90.65
		DL-PSO Features	**99.60**	**99.37**	**99.51**

5. Conclusion and Future Scope

In our study, we explored a novel approach to improve email filtering by combining deep learning, a powerful type of artificial intelligence, with a nature-inspired optimization technique known as particle swarm optimization (PSO). This combination aimed to identify and filter out spam and phishing emails more effectively. We used two methods, PCA (Principal Component Analysis) and PSO, to determine the most essential features for training our deep learning model. Our results showed that the PSO method was exceptionally effective, achieving a high accuracy of 99.60% , significantly better than traditional methods and PCA. This suggests that our advanced system could be highly effective if integrated into real-time email filtering systems, offering better protection against phishing and other email-based threats. The PSO algorithm, while effective, can be computationally intensive, potentially limiting its applicability in environments with restricted computing resources. The model's performance in differentiating between sophisticated phishing attempts and legitimate emails remains an area that could be further explored and refined. Looking ahead, there's potential for further research in this area, such as experimenting with different deep-learning models to improve efficiency and fine-tuning the PSO algorithm for even greater accuracy. Another promising avenue could be combining various algorithms and techniques to develop a more robust and effective email filtering solution, which could significantly enhance cybersecurity measures.

References

[1] K. C. Hwa, S. Manickam, and M. A. Al-Shareeda, "Review of peer-to-peer botnets and detection mechanisms," *arXiv preprint arXiv:2207.12937*, 2022.

[2] M. Thakral, R. R. Singh, and B. V. Kalghatgi, "Cybersecurity and ethics for iot system: A massive analysis," in *Internet of Things: Security and Privacy in Cyberspace*. Springer, 2022, pp. 209–233.

[3] A. Laha, M. T. Yasar, and Y. Cheng, "Substop: An analysis on subscription email bombing attack and machine learning based mitigation," *High-Confidence Computing*, vol. 2, no. 4, p. 100086, 2022.

[4] M. F. Alghenaim, N. A. A. Bakar, and F. A. Rahim, "Awareness of phishing attacks in the public sector: Review types and technical approaches," in *Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2022 Volume 1*. Springer, 2023, pp. 616–629.

[5] M. M. Ali and N. F. Mohd Zaharon, "Phishing—a cyber fraud: The types, implications and governance," *International Journal of Educational Reform*, p. 10567879221082966, 2022.

[6] T. Xu, K. Singh, and P. Rajivan, "Personalized persuasion: Quantifying susceptibility to information exploitation in spear-phishing attacks," *Applied Ergonomics*, vol. 108, p. 103908, 2023.

[7] S. Kumar Birthriya and A. K. Jain, "A comprehensive survey of phishing email detection and protection techniques," *Information Security Journal: A Global Perspective*, vol. 31, no. 4, pp. 411–440, 2022.

[8] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: New collection and results," *Proceedings of the 11th ACM Symposium on Document Engineering*, pp. 259–262, 2011.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[10] APWG, "Phishing Activity Trends Report: Q4 2022," https://docs.apwg.org/reports/apwg_trends_report_q4_2022.pdf, 2022.

[11] Statista, "Daily number of spam emails sent worldwide as of january 16th 2023, by country," January 2023. [Online]. Available: https://www.statista.com/statistics/1270488/spam-emails-sent-daily-by-country/

[12] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary pso with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.

[13] I. Idris and A. Selamat, "Improved email spam detection model with negative selection algorithm and particle swarm optimization," *Applied Soft Computing*, vol. 22, pp. 11–27, 2014.

[14] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "Detection of phishing emails using data mining algorithms," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. IEEE, 2015, pp. 1–8.

[15] K. Agarwal and T. Kumar, "Email spam detection using integrated approach of naïve bayes and particle swarm optimization," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2018, pp. 685–690.

[16] A. I. Taloba and S. S. I. Ismail, "An intelligent hybrid technique of decision tree and genetic algorithm for e-mail spam detection," in *2019 9th International Conference on Intelligent Computing and Information Systems (ICICIS)*. IEEE, 2019, pp. 99–104.

[17] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Applied Soft Computing*, vol. 91, p. 106229, 2020.

[18] R. Talaei Pashiri, Y. Rostami, and M. Mahrami, "Spam detection through feature selection using artificial neural network and sine–cosine algorithm," *Mathematical Sciences*, vol. 14, pp. 193–199, 2020.

[19] R. Rodrigues, M. Costa, and T. Silva, "Transfer learning for spam and phishing email detection using bert," *Journal of Machine Learning and Applications*, vol. 4, no. 1, pp. 25–38, 2021.

[20] A. Mughaid, S. AlZu'bi, A. Hnaif, S. Taamneh, A. Alnajjar, and E. A. Elsoud, "An intelligent cyber security phishing detection system using deep learning techniques," *Cluster Computing*, vol. 25, no. 6, pp. 3819–3828, 2022.

[21] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A deep learning-based phishing detection system using cnn, lstm, and lstm-cnn," *Electronics*, vol. 12, no. 1, p. 232, 2023.

[22] D. Dua and C. Graff, "UCI machine learning repository," http://archive.ics.uci.edu/ml/index.php, 2017.

[23] J. Guillaume, "Csdmc2010 spam corpus," https://plg.uwaterloo.ca/~gvcormac/treccorpus07/, 2010. [Online]. Available: https://plg.uwaterloo.ca/~gvcormac/treccorpus07/

[24] "Spamassassin public corpus," http://spamassassin.apache.org/old/publiccorpus/, n.d. [Online]. Available: http://spamassassin.apache.org/old/publiccorpus/

[25] T. Kurita, "Principal component analysis (pca)," *Computer Vision: A Reference Guide*, pp. 1–4, 2019.

[26] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1998, pp. 69–73.

Santosh Kumar Birthriya is a Ph.D. scholar at the National Institute of Technology, Kurukshetra, India. He received his M.Tech degree in Computer Engineering from the National Institute of Technology, Kurukshetra, India. His research interests include Cyber Security, Machine Learning, Deep Learning, and Information Security.

Dr. Priyanka Ahlawat received PhD in Computer Science and Engineering from National Institute of Technology, Kurukshetra, India. She is Assistant Professor in Computer Engineering Department NIT Kurukshetra Haryana, India. Her interest areas are Key management, Wireless Sensor Networks Security.

Dr. Ankit kumar Jain is presently working as Assistant Professor in National Institute of Technology, Kurukshetra, since September 2013. He received Master of technology from Indian Institute of Information Technology Allahabad (IIIT) India . Dr. Jain received PhD degree from National Institute of Technology, Kurukshetra in the area of Information and Cyber Security. He has more than 60 research papers in International journals and conferences of high repute including Elsevier, Springer, Taylor & Francis, Inderscience, IEEE, etc. His general research interest is in the area of Information and Cyber security, Phishing Website Detection, Web security, Mobile Security, IoT Security, Online Social Networks and Machine Learning.