IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 41, No. 3, June, 2024, pp. 1283-1292
Journal homepage: http://iieta.org/journals/ts

Enhanced Detection of Text and Image Spam Using Cost-Sensitive Deep Learning

Deepika Mallampati[1,2*], Nagaratna P. Hegde[3]

[1] Department of CSE, Osmania University, Hyderabad 500001, India
[2] Neil Gogte Institute of Technology, Hyderabad 500001, India
[3] Department of CSE, Vasavi College of Engineering, Hyderabad 500001, India

Corresponding Author Email: mokshhyd@gmail.com

https://doi.org/10.18280/ts.410317

Received: 2 October 2023
Revised: 23 March 2024
Accepted: 10 April 2024
Available online: 26 June 2024

Keywords:
convolutional neural network (CNN), class imbalanced, cost-sensitive (CS) strategy, loss functions

ABSTRACT

In the realm of unwanted digital content, image spam presents a distinct challenge, characterized by its evasion of traditional text-based filters. This study introduces an advanced approach for the classification of image spam through the deployment of hybrid, cost-sensitive machine learning techniques. Images laden with spam (unwanted content) and benign images (ham) are distinguished by employing a combination of textual and visual data, which enriches the interpretative depth of the analysis. By integrating multi-modal features, resilience against fluctuations in input data and noise is significantly improved. The synthesis of textual context and visual elements enables robust generalization across similar instances while compensating for variations in verbal descriptions, thus maintaining consistent model performance in diverse conditions. A novel methodology is presented wherein cost-sensitive (CS) learning is applied to optimize both feature representations and classifier parameters concurrently, using a deep convolutional neural network (CNN) integrated with a support vector machine (SVM) model. This cost-effective strategy is designed to address class imbalances and refine intermediate feature representations, facilitating rapid adaptation to class-dependent costs. The proposed CSCNN-SVM model is evaluated using the ISH dataset, demonstrating superior performance with an accuracy rate of 98.05%, an AUC of 99.01%, and a computational testing duration of one to two seconds. Furthermore, a variety of machine learning techniques including Logistic Regression, Random Forest, Decision Trees, K Nearest Neighbors, Gaussian Naive Bayes, AdaBoost, and Linear SVM are employed. Utilizing the Spam Hunter Dataset, which consists of real spam emails, these algorithms have proven effective in identifying both text and image spam, achieving comparable levels of accuracy. This innovative, hybrid model not only enhances the detection capabilities of spam classifiers but also contributes significantly to the broader field of digital content management.

1. INTRODUCTION

The internet has become an integral part of human existence, with over 4.5 billion people using it for daily activities. Most internet users consider email a reliable communication tool [1], and its effectiveness has grown over time. However, as email usage increases, so does the number of spam attacks. Spammers can transmit spam from anywhere with internet access, as long as they have the receiver's data and are not solicited. Spam emails are sent without the receiver's data [2]. Phishing emails, often containing fake content or links to malicious websites [3], are widely used to collect users' sensitive information for their own financial gain without their consent. Despite advancements in spam filtering software, there is no reliable mechanism to distinguish genuine and harmful emails due to the ever-changing nature of spam content, and despite anti-spam tools, naive end-users continue to fall prey to this harmful trap [4]. Spam has been sent for three or four decades, and, despite this, it continues to be sent. The Personalized Email Prioritization (PEP) issue is a key impediment, with the goal of assisting users in rating the relevancy of email communications [5, 6]. This is yet another issue with email classification; one potential solution would be to assign low priority to suspicious emails [7-9]. Spam emails are those that were sent without the receiver's data and were not solicited. Machine learning is a powerful tool for detecting image spam shown in Figure 1, but it's crucial to recognize its limitations. Machine learning models can be vulnerable to biased training data, which can make it difficult to detect new spam types [10]. Adversaries can exploit this vulnerability by creating intentionally obfuscated or manipulated images, which can bypass the model's detection mechanisms, particularly for image spam relying on visual elements like logos, watermarks, or text. Intelligent detection is crucial for identifying changes in spam email content and individual user priorities [11, 12]. Deep learning is a powerful technique that has shown great potential in various domains, including image processing and pattern recognition. It involves training deep neural networks with multiple layers to learn representations and features from data automatically. With image spam

detection, deep learning can be utilized to extract relevant features from images that distinguish between spam and non-spam. By leveraging multiple layers of neurons, deep learning models can learn complex patterns and relationships in the data, enhancing the accuracy of the classifier. Supervised learning approaches can be used to train deep learning models for image spam detection using a labeled dataset where each image is annotated as spam or non-spam. CNNs [13] are a common approach to deep learning for image classification tasks. They are particularly effective for image analysis because of their ability to capture spatial hierarchies and local patterns. Unsupervised feature learning algorithms, such as autoencoders or generative adversarial networks (GANs), can be employed to learn meaningful representations from the image data, which can then be used for spam detection. In conclusion, deep learning is a powerful tool for image spam identification because of its ability to automatically learn complex features and patterns from image data, leading to more accurate and robust classifiers. CNNs are built from repeated bits of information organized in this way and have been successfully employed in image spam detection [14].

Figure 1. Examples of spam images

The data augmentation [15] approach generates new training dataset samples by making random changes to existing datasets. This has a number of consequences, including faster learning process convergence and less overfitting [16]. Starting from scratch with a deep neural network demands a significant amount of data and computing resources. Transfer learning has become an essential tool for building efficient text and image spam classification models. It enables faster training, higher levels of efficiency, and greater adaptation to emerging spam threats by using pre-trained models and fine-tuning them for particular tasks [17]. To do this, most of the layers' parameters are inaccessible, and the learning process only adjusts a subset of the layer's parameters. It is also useful for dealing with data scarcity, assuming that the pre-trained model was trained with a sufficient amount of data. They differentiated between images classified as spam and those labeled ham. The Dredge dataset [18], the Image Spam Hunter (ISH) dataset [19], and the enhanced dataset [20] are some of the common datasets used in conjunction with a variety of deep learning models and their respective applications.

The objective and motivation behind the research are mentioned below:

- Our paper introduces a novel approach that integrates both text and image data to enhance the accuracy of spam image classification.
- We propose an innovative technique that automatically calculates class-dependent costs based on data statistics, such as distribution and reparability measurements.
- Unlike existing methods, our approach applies class-dependent costs exclusively during the training phase. This allows for predictions without the need to alter the trained network post-learning, streamlining the inference process while maintaining the integrity of the trained CNN parameters.

In the following section, we discussed the related work in our field. The third section explains the proposed work. Sections 4 and 5 covered the results and conclusion.

2. RELATED WORK

Several research works have explored the integration of text and image information with cost-sensitive learning for spam classification.

The authors [21] optimized a model for feature extraction and classification tasks, meeting both parties' needs, and tested it against various "improved" and "challenge" databases. The datasets were designed to enhance the classification process, and the proposed model significantly improved task accuracy compared to other methods used. Image spam, a type of spam containing text-encoded images, is classified using machine learning methods based on a comprehensive collection of image attributes. CNNs are widely used in image classification and feature extraction due to their superior results.

The authors [22] used datasets like Personal Collection, Dredze, and Spam Archives for image spam classification. Low-level metadata and image metadata are two common feature sets. SVM is the most widely used supervised machine learning method. Naive Bayes and K-Nearest Neighbor are two algorithms used. The study identifies and discusses metrics for evaluating the effectiveness of current image spam classifiers and examines the results of recent algorithms.

The authors [23] proposed a model for spam email classification that includes dataset collection, feature extraction, feature selection, and detection. They use a standard email dataset, combining text and image data, and extract text features using TF-IDF for evaluation. The study used a gray-level co-occurrence matrix (GLCM) to capture visual attributes and the Fitness-Oriented Levy Improvement-based Dragonfly Algorithm (FLI-DA) for feature selection. After selecting the best features, a hybrid learning strategy combines RNN and CNN for detection, leveraging RNN's sequential data processing skills and CNN's spatial image capture capabilities.

The authors [24] proposed deep learning techniques to detect image-based spam on Twitter, including Arabic text. This is a common issue on Online Social Networks (OSNs) like Facebook and Twitter. The "Efficient and Accurate Scene Text Detector" and "Convolutional Recurrent Neural Network' are employed for text recognition. The text is classified as spam or non-spam using blocklists and allow lists, a useful method for combating spam in OSNs, which is flexible and resilient to explicit classification problems.

The authors [25] explored the use of deep CNNs and transfer learning-based pre-trained CNN models for image spam classification. The rapid growth of the internet has led to numerous cyberattacks, with spam emails accounting for 55% of all emails. To address this, two deep CNNs models and pre-trained ImageNet architectures like VGG19 and Xception were trained using three datasets. The study also explored the impact of a cost-sensitive learning strategy to address data imbalances. The proposed models achieved impressive results, with accuracies reaching up to 99% and a false positive rate of zero in the best-case scenario.

The authors [26] studied and analyzed four deep-learning algorithms for detecting image spam. They trained these networks to distinguish specific visual properties and tested their performance on a robust dataset. They also built two other CNN architectures and provided experimental data to distinguish image spam. The study focused on images containing spam, which is unwanted bulk content, and image spam, which refers to unwanted content embedded within images. The legitimacy of email-based communication systems may be uncertain because of image spam. The field of spam classification is growing, with researchers exploring new deep learning architectures, cost-sensitive learning methods, and multimodal fusion techniques to address spam strategies.

3. MATERIAL AND METHODS

The method employs text and image spam filtering techniques to enhance accuracy. The model architecture shown in Figure 2 includes data preprocessing, feature selection, and classification models. Data normalization and standardization are performed in stage 1, followed by feature extraction selection in stage 2. In stage 3, various classification models are trained and tested for text and image features. Stage 4 evaluates performance metrics, while stage 5 demonstrates spam or ham detection.

3.1 Datasets description

In this paper, we use two datasets, namely the Spambase and ISH datasets.

Spambase dataset [27]: The dataset contains 4,601 emails, which are categorized as spam or not. The dataset comprises 57 numerical features that indicate email features, including word and character rate, capital letter sequence length, and symbol occurrence. The Spambase dataset is used for spam filtering and machine learning classification. The dataset is difficult to deal with since no one feature separates spam from legitimate emails.

ISH: In this paper, we used ISH [28], which was the first dataset made available to the public. As seen in Figure 3, it contains two unique sets of genuine images preserved in the JPEG file format. Figure 3a shows some images from 810 ham images. Similarly, the 928 spam images were collected from real spam e-mails (see Figure 3b). It should be noted that it corrupted eight spam images during the extraction process from the spam image collection and is thus not included in the sample set. Table 1 lists all the details of an ISH dataset, which contains 1730 images.

Table 1. Summary of ISH dataset

Dataset	Number of Spam	Number of Ham	Total
ISH	920/8	810	1730/8

3.2 Data preprocessing

Preprocessing organizes and modifies raw data before training and evaluating classifiers. A data mining approach organizes raw data into usable formats. Preprocessing is the initial step in building a machine-learning model. This stage takes real-world data, which often contains errors, inconsistencies, and inaccuracies, and converts it into accurate, precise, and usable input variables and patterns.

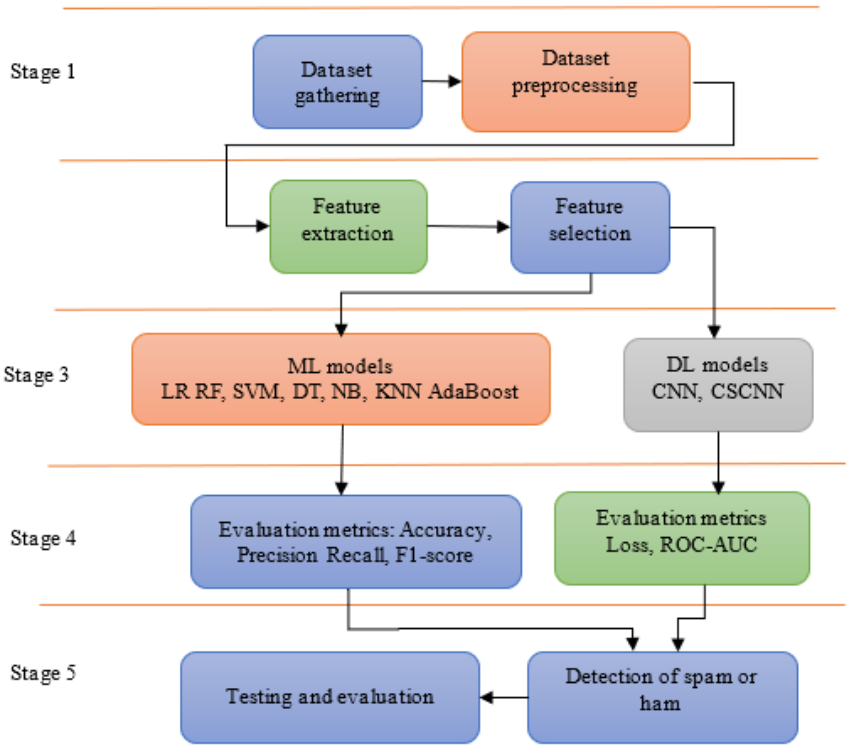

Figure 2. Architecture of proposed model

(a) Ham images from ISH dataset

(b) Spam images from ISH dataset

Figure 3. Some examples of images from ISH dataset

3.3 Data extraction

Feature extraction simplifies large raw datasets, which depend on the initial dataset. At this stage of the process, we can extract data from the dataset, such as variables, attributes, or classes. One of the most critical processes in training the model is feature extraction [29]. This contributes to more reliable and accurate outcomes. We refer to the strategy of selecting a few significant variables that accurately define the data throughout the feature extraction process as feature selection. The method employed is feature selection from among the various attributes. Following that, we construct the model by combining the traits or variables that were chosen. If we follow the feature selection procedure correctly, it will minimize the time spent on the model.

Monarch Butterfly Optimization: After discussing the concept of a self-adaptive strategy, an enhanced MBO algorithm called monarch butterfly optimization with a self-adaptive population (SPMBO) is presented [30]. New individuals in SPMBO are only accepted by those who are superior to previous generations.

3.4 Proposed model

The primary goal of the proposed model is to build a deep learning-based model and evaluate and compare it with several machine learning algorithms from more traditional approaches that were tested.

Gaussian Naive Bayes: Supervised machine learning has been used to detect spam. The ability to discriminate between various things based on defined features is a key component of its success. We calculated the probability of a word or event reoccurring in the future using this method [31]. For instance, if an e-mail has a word that is only present in spam e-mails and not in ham e-mails, the algorithm will almost certainly classify it as spam.

$$(c/x) = \left(P(x/c)P(c)/(P(x))\right) \tag{1}$$

$$P(x) = \sum_{y} P(x/cP(c)) \tag{2}$$

In this case, x represents function vectors, and c denotes a class variable.

SVM: Another tool for supervised machine learning is the SVM. It will only function with pre-classified datasets. When training SVMs, it is most often used for classification and regression models. The data classification is more reliable than another model when there is a limited amount of labeled data. The SVM serves better for the classification. To distinguish between positive and negative values in the dataset (also known as spam and ham), a hyperplane is used. Then, determine which values are close to the decision boundary. Figure 4 is an illustration of the SVM.

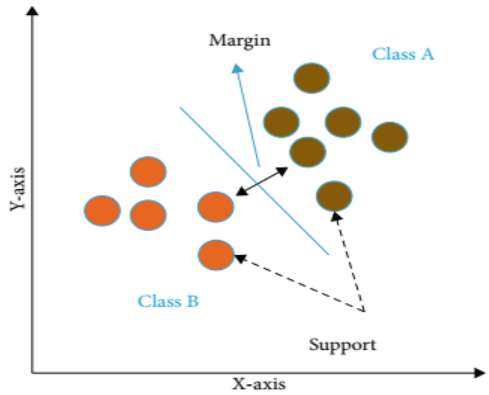

Figure 4. SVM [32]

KNN: This serves as a classification method that represents items as points in a space and calculates the distances between them. In the learning phase, the algorithm assigns training data points to clusters based on their proximity to the center. The algorithm takes a parameter, k, which represents the number of nearest neighbors to consider, and its value can be optimized. The choice of k has an impact on the accuracy of the classification. The k nearest neighbors are the instances in the training sample that are closest to the object being classified. The object is assigned to the class that shares the most attributes with its k nearest neighbors. However, the KNN algorithm can be unstable when dealing with outliers and does not perform well with many features.

Logistic Regression: This is a suitable method for modeling and explaining the relationship between a binary response variable and instructive modules [33]. It is used to analyze data and predict the probability of assigning a specific class, where the values range from 0 to 1. Logistic regression provides a way to model the likelihood of an event occurring based on the given inputs.

Decision tree: This is a hierarchically structured structure that divides the feature space into subspaces. Then, for each object encompassed within this subspace, predict the result, in which algorithm overfitting is a typical issue.

Random forest: This prediction method relies on the construction of trees. By combining multiple trees into a forest, the predictive power of each tree can be enhanced. In the training phase, we construct multiple decision trees based on the programmer's specifications [34]. These trees are then utilized to predict the class. The prediction is made by examining the class votes from each tree, and the output is determined by selecting the class with the highest number of votes.

Adaptive Boosting: AdaBoost is a machine learning ensemble method used for classification tasks. It involves

iteratively training weak learners, such as decision trees, into a single strong learner with improved accuracy. The process involves assigning equal weights to all data points, evaluating the learner's performance, and increasing or decreasing the weights of misclassified data points.

Cost-insensitive models: Cost-insensitive models refer to machine learning models that do not explicitly consider the cost associated with misclassifying instances during the training process. In many classification problems, especially in scenarios where the cost of false positives and false negatives is uneven, it becomes important to account for these costs to optimize the model's performance.

In a cost-insensitive model, the misclassifications are treated equally during the training process, and the model is optimized based on a standard loss function without considering the specific costs associated with different types of errors. However, there are scenarios where cost-insensitive models might be appropriate. If the cost associated with misclassifications is relatively uniform across different classes, or if the focus is solely on overall accuracy without considering the consequences of specific errors, a cost-insensitive approach may be suitable.

Cost-sensitive models: Cost-sensitive models are machine learning models that explicitly consider the costs associated with different types of errors during the training process. In many real-world scenarios, the consequences of false positives and false negatives can vary, and it's important to build models that prioritize minimizing the total cost rather than simply optimizing for overall accuracy.

3.5 CNN model

Figure 5 shows the CNN 2D image processing architecture. An ordinary CNN classifies the visible spam and ham data. CNNs are best suited for text and image spam classification. CNNs mimic the way the human brain interprets images. By not integrating all the nodes, the approach reduces the processing time required and increases efficiency. Also, it successfully discriminates and emphasizes data as features with nearby images with the help of spatial data. This facilitates comparison. CNN models' components are in charge of extracting and classifying image features.

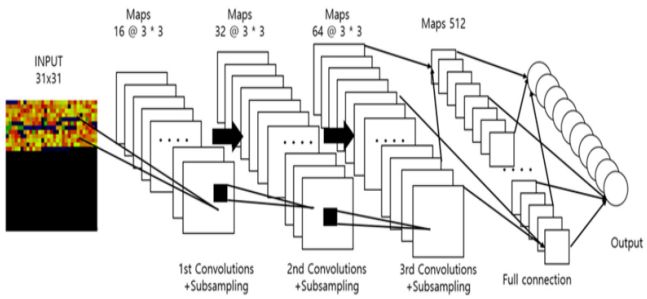

Figure 5. Illustration of the architecture of CNN 2D

3.6 Proposed cost matrix

We denote the proposed cost matrix with the symbol θ, which is appropriate for training the input data for feature extraction. However, as shown in Figure 6, it can be used to alter the output of a CNN's last layer for activation of layers compressed amid 0 and 1 in advance, determining the loss of the model.

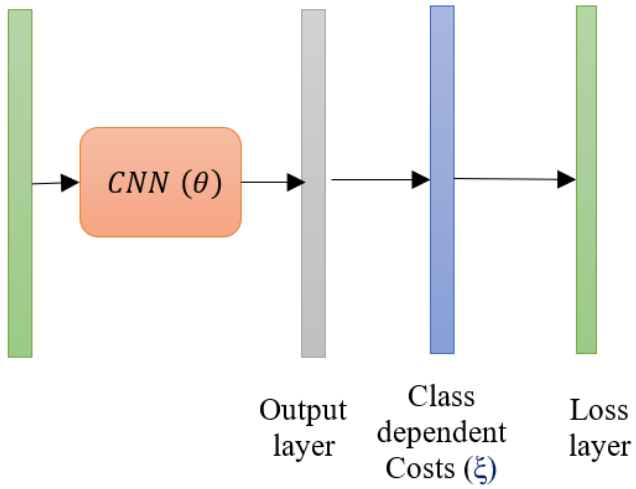

Figure 6. Simplified overview of CNN parameters

The CNN classifies data based on the highest latent score during the classification procedure. During training, the classifier weights are adjusted to alter the confidences or probabilities of the classifier. This is done to ensure that the targeted class receives the highest latent score while the other sequences have significantly lower scores. To enhance classification, "score-level costs" are added to the trainset, affecting the CNN outputs (o) through a mathematically indicated cost matrix (ξ).

$$y^{(i)} = \mathcal{F}\left(\xi_p, o^{(i)}\right), \quad ; y_p^{(i)} \geq y_j^{(i)}, \forall j \neq p, \tag{3}$$

where, y represents the adjusted output, and p indicates the desired class. The confidence of a classifier is significantly influenced by its "score-level costs." This type of perturbation enables the classifier to prioritize "less common" and "challenging-to-separate classes."

3.7 Cost-sensitive surrogate losses

This CNN training method overcomes class imbalance. To achieve this, we propose a cost-sensitive error function, the mean loss across the training set:

$$E\left((\theta, \xi)\right) = \frac{1}{M}\sum_{i=1}^{M} \ell\left(d^{(i)}, y_{\theta,\xi}^{(i)}\right) \tag{4}$$

where, ξ denotes class-sensitive costs, M and N denote is the numbers of training features and output layer neurons, y designates the predicted output, and $d \in \{0,1\}^{1\times N}$ denotes the desired output when $\sum_n d_n = 1$. We will describe a single data instance and not directly address y dependence on parameters (θ; ξ). The optimization goal is to find ideal parameters ($\theta*$; $\xi*$) for the lowest cost $E*$, as the error increases as the prediction system performs catastrophically in training.

$$(\boldsymbol{\theta}^*, \boldsymbol{\xi}^*) = arg\min_{\theta,\xi} E\,(\theta, \boldsymbol{\xi}) \tag{5}$$

In Eq. (4), the loss function ℓ (.) is any variant of the three Cost-Sensitive (CS) losses:

(i) This MSE loss minimizes the squared error between expected output and ground-truth.

$$l(d,y) = \frac{1}{2}\sum_{n}(d_n - y_n)^2 \tag{6}$$

where, y_n represents the previous layer output:

$$y_n = \frac{1}{1 + exp(-o_n\xi_{p,n})} \tag{7}$$

(ii) The hinge loss function exploits margins among classes, expressed as:

$$l(d,y) = -\sum_{n} max(0,1) - (2d_n - 1)y_n \tag{8}$$

where, y_n is the "previous layer output" and ξ is the cost.

$$y_n = o_n\xi_{p,n} \tag{9}$$

(iii) The CE loss function exploits the prediction's accuracy, as shown by:

$$l(d,y) = -\sum_{n}(d_n log y_n) \tag{10}$$

where, y_n includes the "class-dependent cost" (ξ) and is associated to the output through the "soft-max function".

$$y_n = \frac{\xi_{p,n} exp(o_n)}{\sum_k \xi_{p,k}\, exp(o_n)} \tag{11}$$

3.8 Cost-sensitive classifier

The ability of cost-sensitive classifiers to distinguish between treatments for majority and minority classes is their most significant benefit. They also assess the financial costs of misclassification. As stated in the study [35], the negative (majority) and positive (minority) classes are represented by the values 0 and 1, respectively. The rows represent the current classes, while the columns represent the predicted classes. The letters TP and TN indicate correctly classified samples as positive and negative, while FP and FN indicate incorrectly classified samples as positive and negative. From the confusion matrix, the accuracy and geometric mean were determined as follows: Eqs. (12) and (13).

$$G_{mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{FP}{TN + FP}} \tag{12}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

The estimated risk is shown in Eq. (14) according to the least expected cost principle.

$$R(i|S) = \sum_{j} P(j|S)C(j,i) \tag{14}$$

In Eq. (14), $R(i|S)$denotes estimated risk of classifying for the posterior probability is $P(j|S)$ and the misclassification cost is denoted as C (j, i).

Also, posterior probability calculation is always challenging. Thus, empirical risk in mathematical form, such as Eqs. (15) and (16).

$$\hat{R}_l(\boldsymbol{o}) = \boldsymbol{E}_{S,Y}[\boldsymbol{l}] = \frac{1}{n}\sum_{i=1}^{n} l\left(C, d^{(i)}, o^{(i)}\right) \tag{15}$$

$$C = \begin{cases} C_{p,q} = 1, p = q \\ C_{p,q} = IR, p \neq q \end{cases} \tag{16}$$

The relevant class labels are represented by *Y*. The total number of multivariate time series occurrences is given by *n*. When the projected class *q* matches the actual class *p*, the cost will be set as the imbalance ratio. The network's loss function is represented by *l* ().

3.9 Proposed cost-sensitive learning strategy

Cost-sensitive models are useful for text and image spam classification by accounting for the specific costs associated with misclassifications. They help address the imbalance between false positives and false negatives in spam classification. By customizing the loss function to reflect these costs, the model can minimize overall costs rather than focusing solely on accuracy. Cost-sensitive learning allows the model to adapt to new types of spam by considering the specific costs of misclassifying unseen instances. Adjusting the classification threshold based on the trade-off between false positives and false negatives can help minimize the cost of misclassifications in spam classification.

If a penalty for cost-sensitive misclassification entails using an imbalanced ratio, this may help alleviate the overall class imbalance problem. Nonetheless, the fixed cost matrix could not account for the uneven dissemination such as trained CNN data. As a result, the proposed system used a misclassification cost weight that could be updated iteratively and changed dynamically. This was constructed for uneven dissemination of both trained data and minibatches. The previously stated convolutional classifiers were modified, utilizing the cost-sensitive learning technique to deal with ITSC concerns.

The *n*th training instance's cross-entropy loss could be expressed as follows:

$$\begin{aligned} LOSS(\theta) = \lambda \times d_n \times \big(&-ln(y_n) \\ &+ (1 - d_n) \times \big(-ln(1 - y_n)\big)\big) \end{aligned} \tag{17}$$

where, $\boldsymbol{\theta}$ denotes weight parameters.

The cost of misclassification is assigned a cost value. The desired output, denoted by d_n, and the predicted output, denoted by y_n, are both listed in the *n*th training instance. The global loss optimization is represented by Eqs. (18)-(20).

$$\begin{aligned} E(\theta) = &\frac{1}{n^{pos}}\sum_{i=1}^{n^{pos}} LOSS^{pos}\left(\theta^{pos}, \lambda_n^{pos}\right) \\ &+ \frac{1}{n^{neg}}\sum_{i=1}^{n^{pos}} LOSS^{neg}\left(\theta^{neg}, \lambda_n^{neg}\right) \end{aligned} \tag{18}$$

$$\lambda_n = \begin{cases} IR^{normal} \times exp\left(-\frac{G_{mean}^{batch}}{2}\right) \\ \quad \times exp\left(\frac{Acc^{batch}}{2}\right) \quad , if\ n \epsilon neg \\ 1, if\ n \epsilon\ pos \end{cases} \tag{19}$$

$$(\theta^*) = argminE(\theta) \quad (20)$$

IR^{normal} denotes the overall imbalance ratio. The local metrics G_{mean}^{batch} and Acc^{batch} are updated after each minibatch. For input training unbalanced temporal sequence sets, a random shuffle method was utilized to allocate minibatches from scrambled time series data. This strategy avoided minibatch deficits in minority data and improved classifier generalization.

4. RESULTS AND DISCUSSIONS

This paper examines the proposed technique on Spambase and ISH datasets using the Python IDE on a Google Colab with an Intel I5 CPU and 8GB of RAM. This section showcases cost-sensitive models that enhance the accuracy of spam and ham text and image classification. The section demonstrated improvement using metrics such as accuracy, sensitivity, recall, precision, F1-score, AUC, and processing time, along with Receiver Operating Characteristics (ROC). Accuracy is a statistic used to judge how well a classifier works, as stated in Eqs. (21)-(24).

$$Acc = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (21)$$

$$Pre = \frac{TP}{(TP + FP)} \quad (22)$$

$$en\ or\ Rec = \frac{TP}{(TP + FN)} \quad (23)$$

$$F1 = 2 * \frac{(Pre \times Sen)}{(Pre + Sen)} \quad (24)$$

where, *FP* stands for false positive, *FN* stands for false negative, *TP* stands for true positive, and *TN* denotes true negative. Tables 2 and 3 list classification results with Cost-insensitive for different models for the ISH and Spambase datasets, respectively. All four-performance metrics, accuracy, precision, recall, and F1-score, are calculated.

Table 2. Classification results with cost-insensitive for different models for the ISH dataset

Model	Acc	Pre	Rec	F1
LR	0.9688	0.9688	0.9688	0.9688
RF	0.9668	0.9668	0.9668	0.9668
DT	0.9668	0.9668	0.9668	0.9668
KNN	0.9649	0.9649	0.9649	0.9649
GaussianNB	0.9668	0.9669	0.9668	0.9668
AdaBoost	0.9746	0.9746	0.9746	0.9746
LSVM	0.9688	0.9688	0.9688	0.9688
RSVM	0.9532	0.9540	0.9532	0.9532

In general, cost-sensitive models can achieve better performance than cost-insensitive models in terms of these metrics. This is because cost-sensitive models consider the cost of different types of errors, while cost-insensitive models do not. Figure 7 shows the performance comparison with cost-insensitive for eight different models for the ISH dataset, in which AdaBoost is showing the highest accuracy, precision, recall, and F1-score of 97.46%.

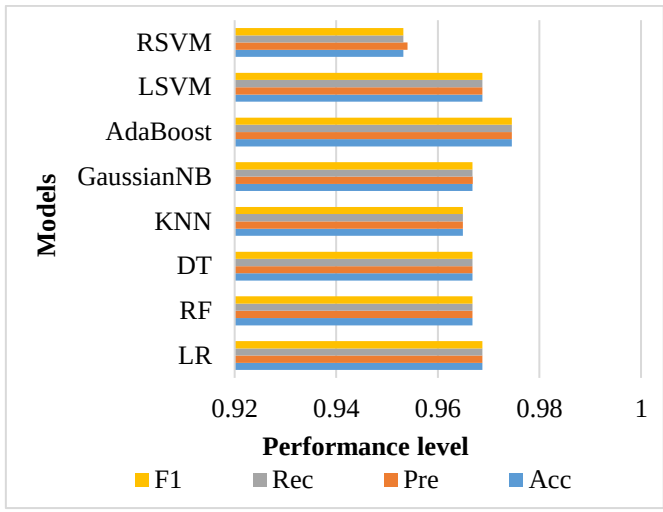

Figure 7. Performance comparison with cost-insensitive for different models for the ISH dataset

Table 3. Classification results with cost-insensitive for different models for the Spambase dataset

Model	Acc	Pre	Rec	F1
LR	0.9732	0.9357	0.9784	0.9691
RF	0.9678	0.9688	0.9672	0.9684
DT	0.9678	0.9678	0.9682	0.9687
KNN	0.9658	0.9665	0.9684	0.9653
GaussianNB	0.9675	0.9678	0.9687	0.9684
AdaBoost	0.9789	0.9777	0.9789	0.9798
LSVM	0.9699	0.9741	0.9741	0.9699
RSVM	0.9656	0.9602	0.9601	0.9600

Figure 8 shows the performance comparison with cost-insensitive for eight different models for the Spambase dataset, in which AdaBoost is showing the highest accuracy, precision, recall, and F1-score of 97.98%.

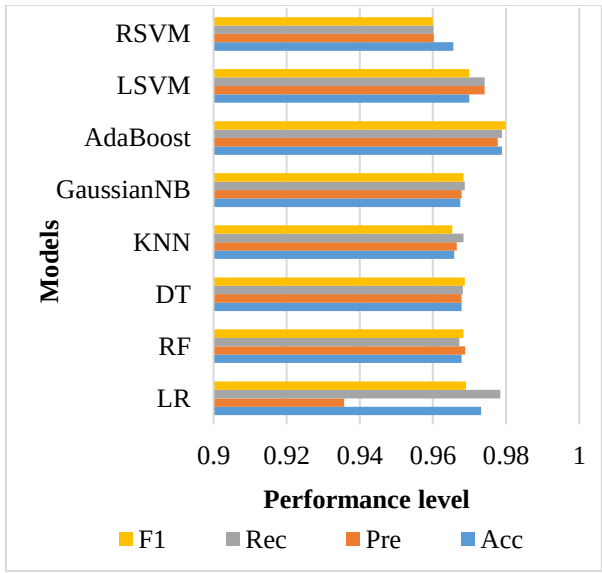

Figure 8. Performance comparison with cost-insensitive for different models for the Spambase dataset

Tables 4 and 5 list classification results with Cost-sensitive for different models for the ISH and Spambase datasets, respectively. All four-performance metrics-accuracy, precision, recall, and F1-score, are calculated.

Table 4. Classification results with cost-sensitive for different models for the ISH dataset

Model	Acc	Pre	Rec	F1
LR	0.9746	0.9746	0.9746	0.9746
RF	0.9785	0.9786	0.9785	0.9785
DT	0.9668	0.9668	0.9668	0.9668
KNN	0.9805	0.9805	0.9805	0.9805
GaussianNB	0.9668	0.9670	0.9668	0.9668
AdaBoost	0.9785	0.9785	0.9785	0.9785
LSVM	0.9805	0.9805	0.9805	0.9805
RSVM	0.9805	0.9805	0.9805	0.9805
DT [36]	0.85	-	-	-
CNN [37]	0.92	-	-	-
NN [38]	0.96	-	-	-

Figure 9 shows the performance comparison with cost-sensitive for eight different models for the ISH dataset in which RSVM is showing the highest accuracy, precision, recall, and F1-score of 98.05%.

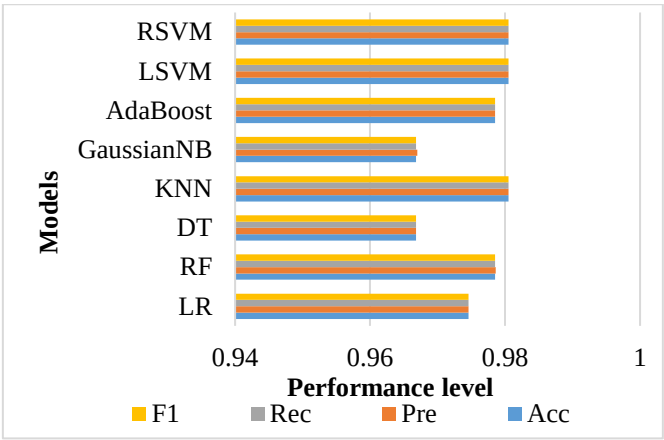

Figure 9. Performance comparison with cost-sensitive for different models for the ISH dataset

Table 5. Classification results with cost-sensitive for different models for the Spambase dataset

Model	Acc	Pre	Rec	F1
LR	0.9777	0.9786	0.9784	0.9753
RF	0.9788	0.9789	0.9791	0.9791
Improved DT	0.9777	0.9888	0.9858	0.9759
KNN	0.9874	0.9853	0.9844	0.9833
GaussianNB	0.9732	0.9725	0.9744	0.9432
AdaBoost	0.9820	0.9813	0.9823	0.9845
LSVM	0.9845	0.9855	0.9866	0.9866
RSVM	0.9899	0.9887	0.9893	0.9893
GA+LR [39]	0.89	-	-	-
LR [40]	0.93	-	-	-
LR + Gradient Boost Tree [41]	0.95	-	-	-

GA+LR: Genetic algorithm+ Logistic regression

Figure 10 shows the performance comparison with cost-sensitive for eight different models for the Spambase dataset, in which RSVM is showing the highest accuracy of 98.99%, precision of 98.87, recall of 98.93, and F1-score of 98.93%.

Figures 11 and 12 show the ROC curves obtained with and without the cost-sensitive approach for the ISH dataset using the examined ML and DL models. Clearly, the curves corroborate our earlier conclusions about the superiority of the CNN-RSVM model's performance.

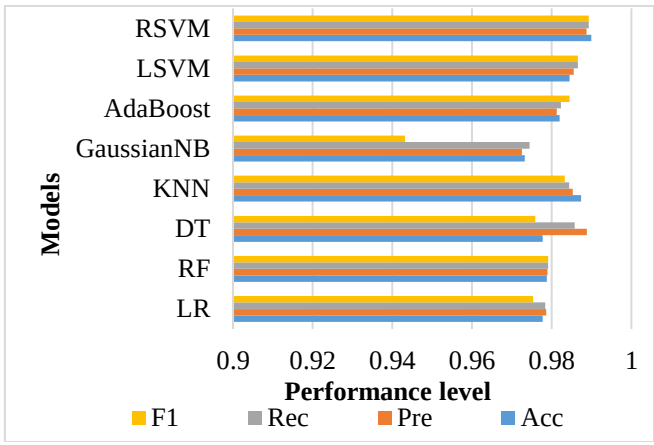

Figure 10. Performance comparison with cost-sensitive for different models for the Spambase dataset

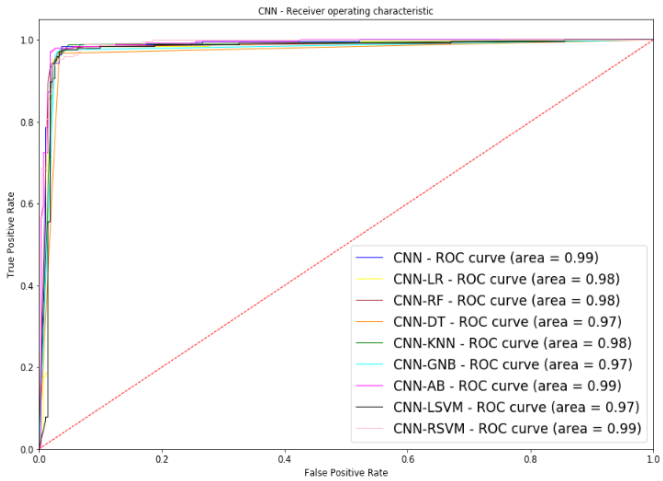

Figure 11. ROC curve for hybrid models with cost-insensitive

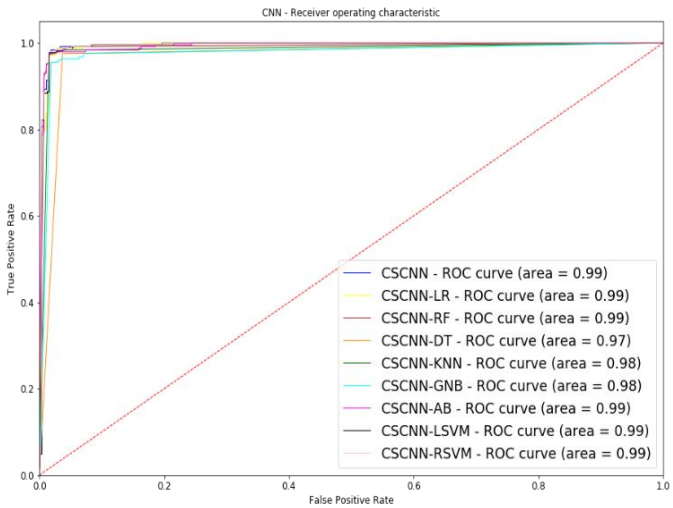

Figure 12. ROC curve for hybrid models with cost-sensitive

5. CONCLUSIONS AND FUTURE SCOPE

This paper addresses the issue of classifying text and images as spam using various classification models with a hybrid model based on CNN being presented for detecting spam. The study utilized various machine learning algorithms like SVM,

KNN, NB, DT, LR, RF, and AB to identify text and image spam based on the collected data. To address the challenge of class imbalance, issue in real-world datasets, a cost-sensitive deep CNN approach was proposed. However, we use cost functions that were analyzed, and class-specific cost estimates were derived for each case. Experimental evaluation of the proposed methodology demonstrated that the CNN-RSVM model outperformed other methods in text and image spam classification. The evaluation results showed that the CNN-RSVM model achieved an accuracy, precision, recall, F1-score, and area under the curve (AUC) of 98.05%, indicating its effectiveness. Obtaining labeled datasets that contain both text and image data for email spam classification might be challenging. In the same way, both text and image data can be computationally expensive, especially for large datasets. This can lead to longer training times and higher resource requirements. The remarkable performance of transformer-based architectures like BERT and RoBERTa in text classification tasks opens new avenues for their application in email spam filtering. Leveraging these advanced models in the future holds the promise of significantly enhancing the accuracy and efficiency of spam detection systems. Leveraging large-scale pre-trained VLMs (such as ViT-BERT or LXMERT) can enable effective feature extraction and understanding of image content within spam emails. As VLMs continue to evolve, they may become more adept at recognizing subtle visual cues indicative of spam content, thereby improving the accuracy of image spam filtering.

REFERENCES

[1] Kumar, N., Sonowal, S., Nishant. (2020). Email spam detection using machine learning algorithms. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp. 108-113. https://doi.org/10.1109/ICIRCA48905.2020.9183098

[2] Jain, G., Sharma, M., Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. International Journal of Information Technology, 11: 239-250. https://doi.org/10.1007/s41870-018-0157-5

[3] Masood, F., Ammad, G., Almogren, A., Abbas, A., Khattak, H.A., Din, I.U., Guizani, M., Zuair, M. (2019). Spammer detection and fake user identification on social networks. IEEE Access, 7: 68140-68152. https://doi.org/10.1109/ACCESS.2019.2918196

[4] Akhtar, A., Tahir, G.R., Shakeel, K. (2017). A mechanism to detect Urdu spam emails. In 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, pp. 168-172. https://doi.org/10.1109/UEMCON.2017.8249019

[5] Yang, Y.M., Yoo, S., Lin, F., Moon, I.C. (2010). Personalized email prioritization based on content and social network analysis. IEEE Intelligent Systems, 25(4): 12-18. https://doi.org/10.1109/MIS.2010.56

[6] Thooyamani, K.P., Khanaa, V., Udayakumar, R. (2013). An integrated agent system for e-mail coordination using jade. Indian Journal of Science and Technology, 6(6): 4758-4761. https://doi.org/10.17485/ijst/2013/v6isp6.22

[7] Jain, G., Sharma, M., Agarwal, B. (2019). Spam detection in social media using convolutional and long short-term memory neural network. Annals of Mathematics and Artificial Intelligence, 85: 21-44. https://doi.org/10.1007/s10472-018-9612-z

[8] Spirin, N., Han, J.W. (2012). Survey on web spam detection: principles and algorithms. ACM SIGKDD Explorations Newsletter, 13(2): 50-64. https://doi.org/10.1145/2207243.2207252

[9] Abuwardih, L.A. (2018). Towards evaluating web spam threats and countermeasures. International Journal of Advanced Computer Science and Applications (IJACSA), 9(10): 538-543. https://doi.org/10.14569/IJACSA.2018.091065

[10] Hasan, Y.M.Y., Karam, L.J. (2000). Morphological text extraction from images. IEEE Transactions on Image Processing, 9(11): 1978-1983. https://doi.org/10.1109/83.877220

[11] Rusland, N.F., Wahid, N., Kasim, S., Hafit, H. (2017). Analysis of naïve bayes algorithm for email spam filtering across multiple datasets. IOP Conference Series Materials Science and Engineering, 226(1): 012091. https://doi.org/10.1088/1757-899X/226/1/012091

[12] Santhi, G., Wenisch, S.M., Sengutuvan, P. (2013). A content based classification of spam mails with fuzzy word ranking. IJCSI International Journal of Computer Science, 10(3): 48-58.

[13] Shrestha, A., Mahmood, A. (2019). Review of deep learning algorithms and architectures. IEEE Access, 7: 53040-53065. https://doi.org/10.1109/ACCESS.2019.2912200

[14] Aiwan, F., Yang, Z.F. (2018). Image spam filtering using convolutional neural networks. Personal and Ubiquitous Computing, 22: 1029-1037. https://doi.org/10.1007/s00779-018-1168-8

[15] Xie, Q.Z., Dai, Z.H., Hovy, E., Luong, M.T., Le, Q.V. (2020). Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems, 33(8): 6256-6268.

[16] Wu, B.Z., Liu, Z.C., Yuan, Z.H., Sun, G.Y., Wu, C. (2017). Reducing overfitting in deep convolutional neural networks using redundancy regularizer. In: Lintas, A., Rovetta, S., Verschure, P., Villa, A. (eds) Artificial Neural Networks and Machine Learning – ICANN 2017. ICANN 2017. Lecture Notes in Computer Science, vol 10614. Springer, Cham. https://doi.org/10.1007/978-3-319-68612-7_6

[17] Chattopadhyay, A., Subel, A., Hassanzadeh, P. (2020). Data-driven super-parameterization using deep learning: Experimentation with multiscale Lorenz 96 systems and transfer learning. Journal of Advances in Modeling Earth Systems, 12(11): e2020MS002084. https://doi.org/10.1029/2020MS002084

[18] Image Spam Dataset. https://www.cs.jhu.edu/~mdredze/datasets/image_spam/, accessed on 29 June 2021.

[19] https://www.dropbox.com/s/7zh7r9dopuh554e/New Spam.zip? dl=0, accessed on 29 June 2021.

[20] Kaggle. https://www.kaggle.com/, accessed on 29 June 2021.

[21] Singh, A.B., Singh, K.M., Chanu, J., Thongam, K., Johnson, K. (2022). An improved image spam classification model based on deep learning techniques. Security and Communication Networks, 2022: 1-11. https://doi.org/10.1155/2022/8905424

[22] Abari, O.J., Sani, N.F., Khalid, F., Sharum, M.Y., Ariffin, N.A.M. (2020). Phishing image spam classification

research trends: Survey and open issues. International Journal of Advanced Computer Science and Applications, 11(11). https://doi.org/10.14569/IJACSA.2020.0111196

[23] Kadam, V., Rohokale, V.M. (2021). Enhancement of email spam detection using improved deep learning algorithms for cyber security. Journal of Computer Security, 30(1): 1-34. https://doi.org/10.3233/JCS-200111

[24] Imam, N., Vassilakis, V. (2019). Detecting spam images with embedded Arabic text in Twitter. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), Sydney, NSW, Australia, pp. 1-6. https://doi.org/10.1109/ICDARW.2019.50107

[25] Srinivasan, S., Vinayakumar, R., Sowmya, V., Krichen, M., Noureddine, D.B., Shashank, A., Soman, K.P. (2020). Deep convolutional neural network based image spam classification. TechRxiv. https://doi.org/10.36227/techrxiv.11999736.v1

[26] Singh, A.P., Potika, K. (2020). Image spam classification with deep neural networks. Malware Analysis Using Artificial Intelligence and Deep Learning, 605-631. https://doi.org/10.1007/978-3-030-62582-5_24

[27] UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/94/spambase, accessed on 29 June 2021.

[28] Image Spam Hunter. https://users.cs.northwestern.edu/~yga751/ML/ISH.htm, accessed on 29 June 2021.

[29] Mallampati, D., Hedge, N.P. (2022). Feature extraction and classification of email spam detection using IMTF-IDF+Skip-thought vectors. Ingénierie des Systèmes d'Information, 27(6): 941-948. https://doi.org/10.18280/isi.270610

[30] Deepika, M. Hegde, N.P. (2021). Framework for spam detection using multi-objective optimization algorithm. Smart Computing Techniques and Applications, 345-355. https://doi.org/10.1007/978-981-16-0878-0_34

[31] Adnan, M., Imam, M.O., Javed, M.F., Murtza, I. (2024). Improving spam email classification accuracy using ensemble techniques: A stacking approach. International Journal of Information Security, 23: 505-517. https://doi.org/10.1007/s10207-023-00756-1

[32] Kumaresan, T., Saravanakumar, S., Balamurugan, R. (2019). Visual and textual features-based email spam classification using S-Cuckoo search and hybrid kernel support vector machine. Cluster Computing, 22: 33-46. https://doi.org/10.1007/s10586-017-1615-8

[33] Hao, L.Y., Awang, N. (2022). The performance of logistic regression and discriminant analysis in spam e-mail classification. Intelligent Systems Modeling and Simulation II, 467-478. https://doi.org/10.1007/978-3-031-04028-3_30

[34] Salau, I.T.T., Adjunct, O.B.S., Oyelakin, A.M. (2023). Spam email detection scheme based on random forest algorithm. LAUTECH Journal of Computing and Informatics (LAUJCI), 3(1): 97-107.

[35] Sammut, C., Webb, G.I. (2011). Encyclopedia of Machine Learning. Springer Publishing Company, Incorporated. https://dl.acm.org/doi/10.5555/2011878

[36] Dredze, M., Gevaryahu, R., Elias-Bachrach, A. (2007). Learning fast classifiers for image spam. In CEAS 2007 - The Fourth Conference on Email and Anti-Spam, Mountain View, California, USA.

[37] Sriram, S., Vinayakumar, R., Sowmya, V., Krichen, M., Noureddine, D.B., Anivilla, S., Soman, K. (2020). Deep convolutional neural networks for image spam classification. In 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, pp. 112-117, https://doi.org/10.1109/CDMA47397.2020.00025

[38] Singh, A.P. (2018). Image spam classification using deep learning. Master's Projects, 641. https://doi.org/10.31979/etd.wehw-dq4h

[39] Shah, N.F., Kumar, P. (2017). A comparative analysis of various spam classifications. In Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, pp. 265-271. http://doi.org/10.1007/978-981-10-3376-6_29

[40] Nandhini, S., Marseline, K.S. (2020). Performance evaluation of machine learning algorithms for e-mail spam detection. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, pp. 1-4. https://doi.org/10.1109/ic-etite47903.2020.312

[41] Anggraina, A., Primartha, R., Wijaya, A. (2019). The combination of logistic regression and gradient boost tree for email spam detection. Journal of Physics Conference Series, 1196: 012013. https://doi.org/10.1088/1742-6596/1196/1/012013