applied sciences

MDPI

Article

MMPC-RF: A Deep Multimodal Feature-Level Fusion Architecture for Hybrid Spam E-mail Detection

Ghizlane Hnini *, Jamal Riffi, Mohamed Adnane Mahraz, Ali Yahyaouy and Hamid Tairi

LISAC Laboratory, Faculty of Sciences Dhar El Mahraz, University Sidi Mohamed Ben Abdellah,
Fez 30000, Morocco; riffi.jamal@gmail.com (J.R.); adnane_1@yahoo.fr (M.A.M.); ayahyaouy@yahoo.fr (A.Y.);
htairi@yahoo.fr (H.T.)
* Correspondence: ghizlane.hnini@usmba.ac.ma

Abstract: Hybrid spam is an undesirable e-mail (electronic mail) that contains both image and text parts. It is more harmful and complex as compared to image-based and text-based spam e-mail. Thus, an efficient and intelligent approach is required to distinguish between spam and ham. To our knowledge, a small number of studies have been aimed at detecting hybrid spam e-mails. Most of these multimodal architectures adopted the decision-level fusion method, whereby the classification scores of each modality were concatenated and fed to another classification model to make a final decision. Unfortunately, this method not only demands many learning steps, but it also loses correlation in mixed feature space. In this paper, we propose a deep multimodal feature-level fusion architecture that concatenates two embedding vectors to have a strong representation of e-mails and increase the performance of the classification. The paragraph vector distributed bag of words (PV-DBOW) and the convolutional neural network (CNN) were used as feature extraction techniques for text and image parts, respectively, of the same e-mail. The extracted feature vectors were concatenated and fed to the random forest (RF) model to classify a hybrid e-mail as either spam or ham. The experiments were conducted on three hybrid datasets made using three publicly available corpora: Enron, Dredze, and TREC 2007. According to the obtained results, the proposed model provides a higher accuracy of 99.16% compared to recent state-of-the-art methods.

Keywords: PV-DBOW; feature-level fusion; CNN; deep neural network; hybrid spam e-mail; multimodal architecture

check for updates

Citation: Hnini, G.; Riffi, J.; Mahraz, M.A.; Yahyaouy, A.; Tairi, H. MMPC-RF: A Deep Multimodal Feature-Level Fusion Architecture for Hybrid Spam E-mail Detection. *Appl. Sci.* **2021**, *11*, 11968. https://doi.org/10.3390/app112411968

Academic Editor: Paolino Di Felice

Received: 8 October 2021
Accepted: 19 November 2021
Published: 16 December 2021

1. Introduction

Spamming, which is defined as the behavior of sending unsolicited messages to a large number of people, is currently spreading rapidly. The spam overflow has led academic researchers to consider this phenomenon as an interesting research area because reputational damage and financial disruption are among the most important activities of spammers [1]. Accordingly, efficient and reliable spam e-mail filtering systems are required for businesses, as well as individuals. These systems should necessarily employ powerful techniques to deal with the increasing number of undesirable e-mails.

Spam e-mail detection has been extensively studied for a long time. The existing spam e-mail filtering systems are divided into two main categories: unimodal and multimodal. The first category comprises two types: text-based [2] and image-based [3]. The former is developed to detect spam e-mails containing only text, while the latter is built to deal with embedded text within images, knowing that it was created by spammers to evade text-based systems. The second category, multimodal systems, is designed for detecting spam e-mails containing both image and text, known as hybrid spam e-mails [4]. The hybrid spam e-mail is more harmful and complex as compared to image-based and text-based spam e-mail due to the amount of information within. This makes it a challenging task. Consequently, the hybrid spam e-mail demands efficient algorithms for treating both text and image content. The multimodal system is aimed at generating feature vectors from both text and image modalities, before combining them at the feature level or decision level.

To the best of our knowledge, a small number of studies have been aimed at detecting hybrid spam e-mails. The authors of [5] used the term frequency inverse document frequency (TF-IDF) to extract text features, while image features represented the color, image, and file attributes. First, the P-SVM (support vector machine with a confidence P) was adopted to separately learn text and image modalities. Then, the output of the multi-classifier was fed to the SVM (support vector machine) model to classify an e-mail as either spam or ham. Additionally, Ref. [4] fused two classification probability scores. These values were generated from image and text parts by consecutively using a convolutional neural network (CNN) and a long short-term memory (LSTM) model. The authors of [6] used the CNN model to construct two different multimodal architectures to identify an e-mail as spam or ham. The first concatenated the last layer of CNN of both image and text features to feed them to a fully connected layer using the softmax function. Using the same features, the second architecture applied decision-level fusion to classify an e-mail as either spam or ham.

However, despite the significant performance of the previous models, they still suffer from many drawbacks that are not yet solved. Firstly, the decision-level fusion not only demands many learning steps but also loses correlation in mixed feature space [7]. Indeed, each modality is separately trained using different classification models before the fusion step. In addition, the output classification scores are fused to make a final decision. Secondly, the TF-IDF feature extraction technique has very limited applicability. This method is only based on the frequency of a word in a document. However, it does not take the word order into account, which leads to misclassification. Feature extraction is a crucial step in spam detection. Indeed, it has a negative impact on the performance of the model. Recently, many improvements of great importance have been accomplished while using machine learning (ML) and deep learning (DL) techniques.

The main contribution of this paper is the proposition of a new multimodal architecture based on PV-DBOW, CNN, and RF. It consists of generating feature vectors from both text and image of the same e-mail by consecutively using the PV-DBOW and CNN models. Then, the two generated vectors are concatenated at the feature level before feeding them into the RF model to classify an e-mail as either spam or ham. Regarding the text features, the adoption of the PV-DBOW method proves its importance in our system by obtaining a highly semantic representation, while the CNN model also ensures important features that are extracted from the image. Moreover, the concatenation of these vectors feeding the RF classifier increases the performance of the proposed architecture compared to the state-of-the-art methods.

Section 2 includes a background of related work and earlier research techniques. Section 3 provides the methods adopted and presents the proposed deep multimodal feature-level fusion architecture. The comparative study and the experimental results are presented in Section 4. Section 5 concludes our paper.

2. Related Works

Generally, the accuracy rate of spam detection system is often affected by the feature extraction techniques adopted. Therefore, in order to enhance the performance of multimodal spam e-mail systems, more efficient and powerful image-based and text-based feature extraction techniques are required. Below, we describe some of these techniques.

2.1. Text-Based Feature Extraction Techniques

Recently, many feature representation techniques have been suggested in order to train various machine-learning classification models such as naïve Bayes (NB), support vector machine (SVM), and random forest (RF). Therefore, most of these techniques implement the vector space model (VSM) to represent an e-mail. In VSM, the (i, j)-th value in a term-by-document matrix generally represents the occurrence of the i-th term within the *j*-th document, where this term can reflect an individual word or text unit [8]. For instance, the bag of words (BoW), the *n*-gram, and the term frequency inverse document frequency (TF-IDF) are the most popular algorithms used for feature representation. The authors of [9] used these feature techniques to transform incoming e-mails into a set of feature vectors. Then, these vectors were fed to evaluate different nearest neighbor classifiers, including k-NN, weighted KNN, and k-d tree, in order to classify e-mails as either spam or ham. In addition, Ref. [10] used the term frequency technique to extract the features that are employed to train SVM for spam detection. Despite the success of the SVM model, other authors [11] have suggested the use of naïve Bayes in the field of spam detection.

However, despite the good results obtained through using such feature representation techniques, they have some limitations. The authors of [12] stated that the BoW and the TF-IDF techniques lead to overfitting classifiers because they have a large sparse feature space that is caused by their large vocabulary. In addition, Ref. [13] confirmed that the higher dimensionality of features is one of the biggest issues of spam detection. Thus, several dimensionality reduction techniques have been suggested in order to overcome the above problem. Many studies suggested the use of a singular value decomposition (SVD)-based approach for feature reduction. There are two well-known SVD-based techniques: principal component analysis (PCA) and latent semantic analysis (LSA). They are used to convert the original space to a new lower-dimensional feature space. The authors of [13] proposed the use of PCA and LSA for the pre-processing step of the e-mails, while using the TF-IDF for building the feature vectors which were fed to a linear SVM classifier, called sequential minimal optimization (SMO). Similarly, Ref. [14] provided a new model using a classifier based on PCA document reconstruction (PCADR) which was compared with the popular SMO classifier. The authors of [15] declared that, because LSA is a term co-occurrence-based technique, it does not work for noisy data. In addition, Ref. [12] stated that LSA does not capture word order although it achieved good performance. In recent years, many techniques have been adopted for spam e-mail detection such as PV-DBOW. This technique takes into consideration the context of e-mails. The authors of [16] proposed a new architecture to detect spam e-mails using the PV-DBOW model. This technique was used to extract features from e-mails that were fed to various classification models in order to distinguish between spam and ham.

The authors of [17] compared various deep learning techniques to the classical machine learning classifiers for detecting spam e-mails. In addition, different word embedding methods were used in order to generate a dense feature vector from an e-mail. The authors of [17] proposed the DeepSpamNet architecture that uses the CNN and the LSTM methods.

In the work of [18], various machine learning techniques such as support vector machine, naïve Bayes, decision tree, random forest, and multilayer perceptron, which were optimized using the PSO (particle Swarm Optimization) and the GA (genetic algorithm), were adopted in order to detect spam e-mails. The experiments were conducted on seven datasets to select the most suitable model.

The pretrained transformer model BERT (bidirectional encoder representations from transformers) was used in the work of [19]. This model, which takes into account the context of the text, was fine-tuned to classify an e-mail as either spam or ham. The performance of the BERT technique was compared to various machine learning techniques. The experiments were conducted on two available datasets, obtaining the highest accuracy of 98.67% and 98.66% in terms of *F*1-*score*.

2.2. Image-Based Feature Extraction Techniques

Image spam was introduced by spammers in 2006 to evade text-based spam filters. It generally consists of embedding texts within images. Thus, various computer vision and pattern recognition techniques have been recently suggested to detect image spam.

Text area (TA), low-level (LL), image similarity (IS), image regions similarity (RS), image metadata (M), and text obfuscation (T) are the most discriminative features extracted from an image. These features are used by many studies to train different machine learning models for classifying image-based e-mail as either ham or spam. The authors of [20] used the TA and LL as features to train the one-class SVM classifier, whereas Refs. [21,22] adopted SVM for the same features. In addition, Refs. [23–25] suggested the use of LL and IS features to train different classifiers including one-class SVM, SVM, and probabilistic boosting tree. Furthermore Ref. [26] used the LL and M features to train three models: max entropy, naïve Bayes, and decision tree.

The authors of [27] used two approaches to classify an image as either spam or ham: principal component analysis (PCA) and support vector machines (SVM). The first approach was used to construct the eigenspace from a training set of spam images. In the testing phase, a new image was projected onto this constructed eigenspace, and then this image was classified as spam if its score, which was the Euclidean distance between the scoring matrix of the training set and the weight vector of the test image, was below a certain threshold. In the second approach, they extracted 21 different features from images used as input to a linear SVM classifier. They obtained good results using PCA and SVM on the Image Spam Hunter (ISH) dataset, but these techniques did not detect spam images in their new improved dataset, referred to as Challenge Dataset 1. Moreover, Ref. [28] used a linear SVM to classify the image as ham or spam using 38 features including metadata, color, texture, shape, and noise. They achieved good performance on the ISH and Dredze datasets, but they failed to detect spam images on their improved dataset, referred to as Challenge Dataset 2. Despite the good results, Ref. [29] stated that the features used by these previous works were computationally extensive to extract. The authors of [29] proposed the use of the Canny edge detector combined with the raw image as features of images, which were then fed to a CNN model. Their experiment showed good results on the ISH dataset compared to improved Challenge Dataset 1 and 2. In addition, Ref. [30] suggested a new image-based spam filtering architecture based on the convolutional neural network (CNN). The image features were generated from the last layer of CNN and fed to the support vector machine (SVM) classifier.

In addition, Ref. [31] proposed the DeepCapture architecture for image spam detection, which was based on the convolutional neural network (CNN) model and the XGBoost classifier. This method achieved the highest *F*1-*score* compared to existing spam detection models. Furthermore, Ref. [32] used the DCNN (deep convolutional neural network) and pretrained CNN for spam image detection. Three different datasets were used to prove the effectiveness of their proposed models.

Despite the improvements achieved in both image-based and text-based feature extraction techniques for distinguishing spam from ham, there is an important lack of multimodal systems which combine the two feature extraction techniques to deal with hybrid spam e-mails. Concatenating both image and text features into one vector, before the classification step, provides a robust representation for each e-mail.

3. Methodology

Three techniques were adopted in this paper: PV-DBOW, CNN, and RF. This section describes each of these techniques and the proposed multimodal architecture (MMPC-RF) for detecting hybrid spam e-mails.

3.1. The PV-DBOW Model

Traditional methods such as bag of words (BoW) or term frequency inverse document frequency (TF-IDF) ignore the semantic features of words. Paragraph vector, also known as Doc2vec, is a new unsupervised neural network model which was proposed by [33] for representing a document. The model provides a continuous distributed feature vector in a fixed length of a given document while taking into consideration the relationships between its words. Doc2vec is an extension of the Word2vec model that takes two forms: the continuous bag of words (CBOW) and the skip-gram [34]. The former is used to predict a target word from a given context, whereas the skip-gram is employed to conduct the inverse task. It predicts the neighbor words of a given word. Let us consider $w_1, w_2, w_3, \ldots, w_M$

as the words which are represented in a document and c as the size of the window. The skip-gram model aims to maximize the average log probability as follows:

$$\frac{1}{M}\sum_{m\,=\,1}^{M}\left(\sum_{-c\,\leq\,j\,\leq\,c,\,j\,\neq\,0}\log p\left(w_{m\,+\,j}\middle|w_m\right)\right). \tag{1}$$

To calculate the conditional probability $p\left(w_{m+j}\middle|w_m\right)$, several functions are employed, including the softmax function, the hierarchical softmax, or negative sampling. In this paper, the softmax function was used as indicated in the following equation:

$$p\left(w_{m+j}\middle|w_m\right)=\frac{\exp\left(u_{w_{m+j}}v_{w_m}\right)}{\sum_{m=1}^{M}exp\left(u_m v_{w_m}\right)}. \tag{2}$$

The input and output vector representations of a given word w represent, respectively, the two parameters u_w and v_w. M denotes the size of the vocabulary.

As with Word2vec, the Doc2vec model comprises two different models: the paragraph vector distributed memory (PV-DM) and the distributed bag of words of paragraph vector (PV-DBOW). The first is an extension of the CBOW model, while the second is based on the skip-gram neural network model.

Let v_d be the vector representation that is generated from a document d using the PV-DBOW model. The model trains the vector v_d to learn the probability of a word w in the document d. Therefore, this probability $p(\mathrm{w}|\mathrm{d})$ is calculated as follows in the skip-gram model:

$$p(\mathrm{w}|\mathrm{d})=\frac{\exp(u_\mathrm{w}v_\mathrm{d})}{\sum_{\hat{w}\in W}exp(u_{\hat{w}}v_\mathrm{d})}. \tag{3}$$

For each document d that is composed of a certain number $N(d)$ of words w_1^d, w_2^d, $\ldots$, $w_{N(d)}^d$ and belonging to D, a set of documents, the PV-DBOW aims to maximize the log probability as indicated below.

$$\sum_{d\in D}\sum_{i=1}^{N(d)}logp\left(w_i^d\middle|\mathrm{d}\right). \tag{4}$$

3.2. The CNN Model

The convolutional neural network (CNN) architecture was developed to solve the problems of the classic artificial neural network (ANN) related to the cost, time, number of parameters, and selected features. The most important benefits of the CNN model are extracting the most relevant features, minimizing the number of parameters, training massive data, and decreasing the computation in the network [35]. The CNN model has achieved high performance in a variety of fields such as image recognition. It is composed of three main layers: convolution, pooling, and fully connected.

First, the convolution layer is designed to extract features using the convolution operation. The number of feature maps and the size of the kernels are two hyperparameters defining the convolution operation. The kernel of a selected size 3 × 3 or 5 × 5 is passed in stride over the input tensor. This operation is repeated as many times as the feature map number. The (m,n)-th feature map value, which is generated after applying the convolution operation on the input image f using a kernel h, is calculated according to the following equation:

$$\sum_j\sum_k h(i,j)f(m-j,n-k). \tag{5}$$

Set N as the size of the input image, K as the size of the kernel, P as the number of layers of zero-padding, and S as the stride size. The size F of the feature map is obtained using the following equation:

$$F = 1 + \frac{N + 2P - K}{S}. \tag{6}$$

Second, the pooling layer consists of reducing the dimension of feature maps and controlling overfitting. It is a required task in the CNN model, and it is situated after the convolutional layer. The pooling operation is done by selecting the maximum value in the convolution layer on each region.

Third, the last operation in the convolution neural network is the fully connected layer. It is considered a trainable classifier, and it takes the pooling layer as input to transform it into one vector with the desired size.

3.3. Random Forest Classifier

The authors of [36] introduced an efficient algorithm that is mainly based on the ensemble learning model, called the random forest (RF) classifier. It consists of a combination of two popular strategies: bagging and random subspace techniques. The strongest benefit of the RF classifier includes its ability to indicate the feature importance efficiency. Furthermore, the robustness of this classifier is generally determined by two parameters: the number of trees and the feature number of each node.

Let us consider a dataset $L = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ that contains n observations. Each observation (X, Y) contains two elements which are variables, and the predictor is denoted consecutively by $X = \left(X^1, \ldots, X^p\right)$ and $Y \in \mathcal{Y}$, where $\mathcal{Y}$ is either a class label or a numerical response and $X \in \mathbb{R}^p$. The function $t : \mathbb{R}^p \to Y$ is a mapping for a classification problem, where $Y = \mathcal{Y}$, but is not the same in the regression problem, where Y is supposed to be $s(X) + \varepsilon$, where $E[\varepsilon|X] = 0$ and s denotes the so-called regression function.

The aim of the random forest algorithm is to form and combine several classifications and regression trees (CARTs). These trees are made randomly by combining many decision trees in two main steps using two strategies. As a first step, the RF classifier uses the bootstrap sampling method to create b sets of samples from the initial learning sample. Second, it selects k variables randomly from the original features p for a given node, with $k < p$, in order to make a decision tree for each bootstrap sample set. Finally, for a new input sample, the prediction is made using the vote from the classification trees.

3.4. The Proposed Approach

3.4.1. Dataset

The experiments of the proposed MMPC-RF architecture were conducted on three different datasets, as shown in Table 1. The authors of [4] used two publicly available datasets, Enron and Dredze, to build hybrid e-mails. The Enron dataset [37] contains 17,108 text ham e-mails and 16,537 text spam e-mails, whereas the Dredze dataset [26] has 2021 personal image ham, 3298 personal image spam, and 16,031 SpamArchive image spam. After removing duplicates from these datasets, they constructed two mixed datasets. The first, which we refer to as Dataset 1, has 600 hybrid ham e-mails (each e-mail contains text and image ham) and 600 hybrid spam e-mails (each e-mail contains text and image spam). The second, named Dataset 2, contains 600 hybrid ham e-mails (600 text ham, 300 image ham) and 600 hybrid spam e-mails (600 text spam, 300 image spam). On the other hand, in order to make the evaluation more acceptable and valuable, we tested our model on the same dataset used by [5]. It contains 2792 hybrid e-mails where 414 are hybrid ham e-mails and 2378 are hybrid spam e-mails from two different datasets: TREC 2007 and Dredze [26].

Table 1. The datasets used in the experiments.

Title 1	Dataset 1	Dataset 2	Dataset 3
Original dataset	[4]	[4]	[5]
Total e-mails	1200	1200	2792
Total spam e-mails	600	600	2378
Total ham e-mails	600	600	414

3.4.2. The MMPC-RF Architecture

The proposed MMPC-RF model for spam filtering, as illustrated in Figure 1, includes three main parts: preprocessing, feature learning, and classification.

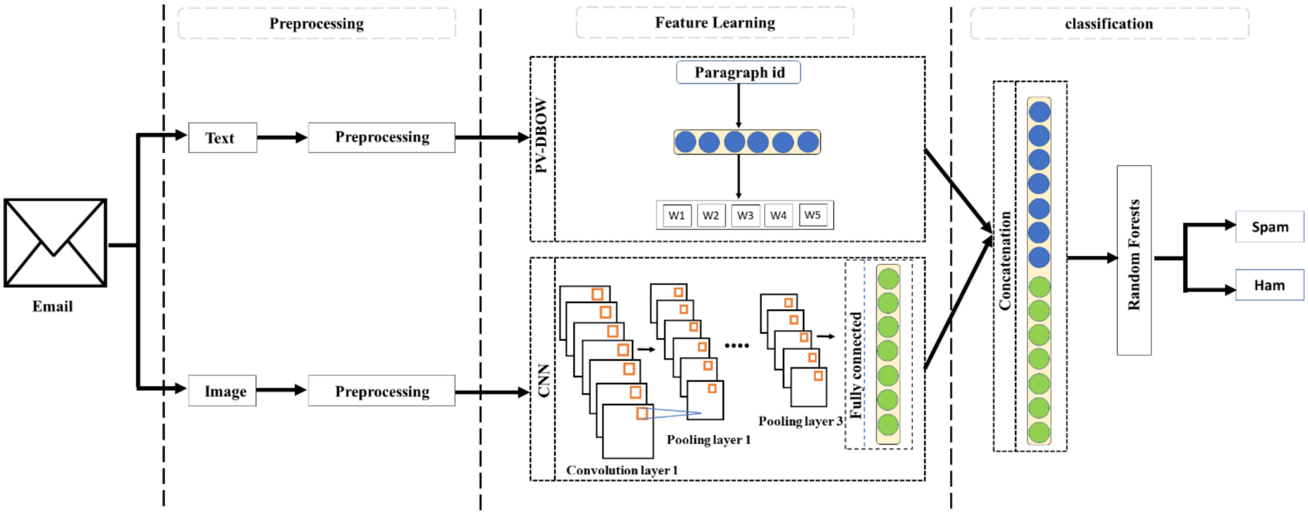


Figure 1. The architecture of the proposed MMPC-RF model.

The preprocessing part consists of cleaning both text and image data. On one hand, the text data preprocessing step adopts several techniques to enhance the classification accuracy of the e-mails. First, the e-mail text is converted into a set of words, known as tokens, using the tokenization technique. Second, these tokens contain useless words (so-called stop words, such as “at”, “for”, and “of”) which lead to misclassification. In order to perform the classification, these stop words are removed. Third, to use these e-mails in the PV-DBOW model, a unique id is assigned to each e-mail. On the other hand, the image data preprocessing step consisted of normalizing and resizing these images to 128 × 128 RGB size.

The feature learning part generates the feature vectors from each modality of the e-mail using two models: the PV-DBOW model and the CNN model for text and image data, respectively. On one hand, for the text data, the PV-DBOW model learns and generates the vector representations in a low-dimensional space of a fixed length for each tagged e-mail. Let E_i be the vector representation of an e-mail, which is represented by one-hot vector; $E = \{E_1, E_2, E_3, \ldots, E_n\}$ is a set of e-mails, and W is the weight matrix of the network between the input and the hidden layer of the PV-DBOW model. W is $K \times M$, where M is the number of e-mails, and K represents the dimension of the hidden layer. The activation function for the hidden layer $\hat{E}$ uses a weighted sum of the input layer given by $\hat{E} = WE$. The output layer of the model is fully connected to the hidden layer. The feature vector for document ID d of the e-mail $\hat{E}_{id}$ is generated from the paragraph vector (hidden layer) $\hat{E}$ in d dimensions given by the d-th column in $\hat{E}$. After training the model, the hidden layer produces fixed-size paragraph vectors (embedding vectors).

On the other hand, the CNN model was used to extract high-quality features from image e-mails. The CNN model encompasses three main layers: an input layer, followed by many convolution layers and pooling layers, before ending with a fully connected layer. The input of the CNN model is an image of 128 × 128 RGB size. In this paper, three convolution layers were used in order to extract the abstract features from the image data. Let $I_{c,i,j}$ be the pixel element in row i and column j of the c channel image; a set of filters K of dimension $k_1 \times k_2$, and $K_{c,m,n}$ represents the weight of row m and column n of the channel c filter. The j-th column element in row i of the feature map is given as follows:

$$(I * K)_{i,j} = f\left(\sum_{m\,=\,0}^{k_1-1}\sum_{n\,=\,0}^{k_2-1}\sum_{c\,=\,1}^{C} K_{c,m,n}.I_{c,i\,+\,m,j\,+\,n}\right). \tag{7}$$

In this paper, the rectified linear unit (ReLU) was used to calculate the activation function f, while C represents the number of channels. The most important information is represented by the maximum value of each region in the feature map extracted using the max pooling operation. The generated feature maps were then converted into a one-dimensional vector to make a prediction using the softmax activation function. Moreover, batch normalization was adopted in order to prevent over-fitting. The CNN was trained in order to generate the embedding vector of an image using the binary cross-entropy loss function. The trained CNN architecture contained layers from the input to the first dense layer, which had 64 features (neurons).

The classification part concatenated the two embedding vectors to have a rich e-mail representation and increase the performance of the classification, using the PV-DBOW and the CNN models for text and image of the same e-mail. T_n is the embedding vector of the n-th text e-mail, which came from the PV-DBOW model, and I_n is the feature vector of the image of the n-th e-mail, which was generated using the trained CNN model. The concatenation of these two representations T_n and I_n produced a vector which was fed to the RF classifier in order to distinguish between spam and ham.

4. Experimental Results and Comparative Study

In order to evaluate our proposed architecture, a preprocessing task was required. First, we resized and normalized the image data to 128 × 128 RGB size. Then, we prepared the text e-mail data for the PV-DBOW model by removing stop words and tokenizing these e-mails. On one hand, the hyperparameters of the PV-DBOW used in this experiment included the size of the text e-mail vector set to 300, an initial learning rate of 0.002, and a context window equal to 30. On the other hand, the input of the CNN model was a 128 × 128 RGB image. The model contained three convolutional layers, each of them followed by a maximum pooling layer. In addition, the numbers of filters used in the first, second, and third convolutional layers were 32, 64, and 64, respectively. The kernel sizes used in the convolutional layers were set to 5 × 5, 3 × 3, and 3 × 3, while 2 × 2 was the kernel size used in the three pooling layers.

The rectified linear unit (ReLu) and sigmoid were used as the activation functions. The last pooling layer was flattened to feed it into two hidden layers with 64 and 32 neurons. The model was trained using the hyperparameters cited in Table 2, and then we used the first hidden layer with 64 neurons as a feature vector to describe an image e-mail. The two feature vectors, coming from the PV-DBOW model and the last layer of the trained CNN model, were concatenated to form a robust feature vector describing a hybrid e-mail. This vector was fed to the RF classifier to distinguish between spam and ham e-mails.

Table 2. The hyperparameters used in the experiments.

Optimizer	Adam
Learning rate	0.001
Epochs	100
Batch size	20

Table 3 shows the experimental results conducted on Dataset 1, Dataset 2, and Dataset 3. The performance of our proposed architecture was verified using the k-fold cross-validation method. In this paper, we used fivefold cross-validation to divide each of these datasets. One set was used for validation, while the remaining sets were used for the training. The average values of *accuracy*, *recall*, *F*1-*score*, and *precision* were calculated after using the fivefold cross-validation method as follows:

$$Accuracy = \frac{\sum_{i=1}^{k} Accuracy_i}{k}, \quad (8)$$

$$Recall = \frac{\sum_{i=1}^{k} Recall_i}{k}, \quad (9)$$

$$F1 - score = \frac{\sum_{i=1}^{k} F1 - score_i}{k}, \quad (10)$$

$$Precision = \frac{\sum_{i=1}^{k} Precision_i}{k}, \quad (11)$$

where $Precision_i$, $Recall_i$, $Accuracy_i$, and $F1 - score_i$ for the *i*-th set are computed as follows:

$$Precision = \frac{TP}{TP + FP}, \quad (12)$$

$$Recall = \frac{TP}{TP + FN}, \quad (13)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (14)$$

$$F1 - score = \frac{2.Precision.Recall}{Precision + Recall}. \quad (15)$$

Table 3. The experimental results of our method using fivefold cross-validation technique.

Dataset	*Accuracy*	*F1-Score*	*Recall*	*Precision*	ROC
Dataset 1	99.16%	99%	99.83%	98.52%	99.97%
Dataset 2	98.91%	98.92%	99.49%	98.20%	99.96%
Dataset 3	99%	99.08%	99.33%	98.84%	99.95%

The number of ham e-mails that were misclassified (FP), the number of ham e-mails that were correctly classified (TN), the number of spam e-mails that were misclassified (FN), and the number of spam e-mails that were correctly classified (TP) were obtained after classification. The performance results of the MMPC-RF architecture in terms of *accuracy*, *F*1-*score*, *recall*, and *precision* are shown in Table 3. The proposed model provided high performance on the different datasets (Dataset 1, Dataset 2, and Dataset 3).

After applying the fivefold cross-validation to Dataset 1, Table 4 presents a comparative study with the existing state-of-the-art models. The SVM, classical k-NN, and MMA-MF models achieved an accuracy of 98.25%, 97.83%, and 98.42%, whereas our model achieved the best accuracy with 99.16%. In addition, as demonstrated in Table 4, our model had the best *F*1-*score* as compared to other models. On the other hand, the experiments conducted on the Dataset 2 are shown in Table 4, showing that the results obtained using our method exceeded those of all existing models in terms of *accuracy* and *F*1-*score*. In addition, Figures 2 and 3 provide the confusion matrix for Dataset 1 and Dataset 2. As in Dataset 3, our model outperformed the P-SVM model, reaching an accuracy of 99%. In addition, Figures 4–6 show the receiver operating characteristic (ROC) curves of different datasets using cross-validation technique. These figures demonstrate that our proposed architecture achieved the best area under the curve (AUC), as indicated in Table 3. We deduce that our multimodal architecture is efficient in the detection of hybrid spam e-mail as compared to the existing models.

Table 4. The comparative study.

Dataset	Model	*Accuracy*	*F1-Score*	*Recall*	*Precision*
Dataset 1	SVM (Yang et al., 2019)	98.25%	98.25%	98.25%	98.10%
	Classical k-NN [4]	97.83%	97.83%	97.81%	97.70%
	MMA-MF [4]	98.42%	98.41%	98.45%	98.40%
	Our method	**99.16%**	**99%**	**99.83%**	**98.52%**
Dataset 2	SVM [4]	98.42%	98.41%	98.44%	98.30%
	Classical k-NN [4]	98.08%	98.08%	98.13%	97.90%
	MMA-MF [4]	98.46%	98.45%	98.52%	98.30%
	Our method	**98.91%**	**98.92%**	**99.49%**	**98.20%**
Dataset 3	P-SVM [5]	90%	-	-	-
	Our method	**99%**	**99.08%**	**99.33%**	**98.84%**

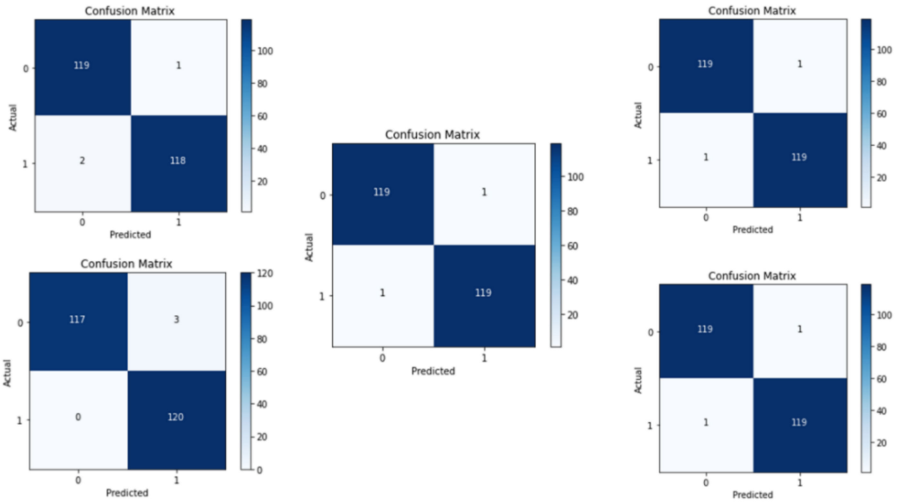
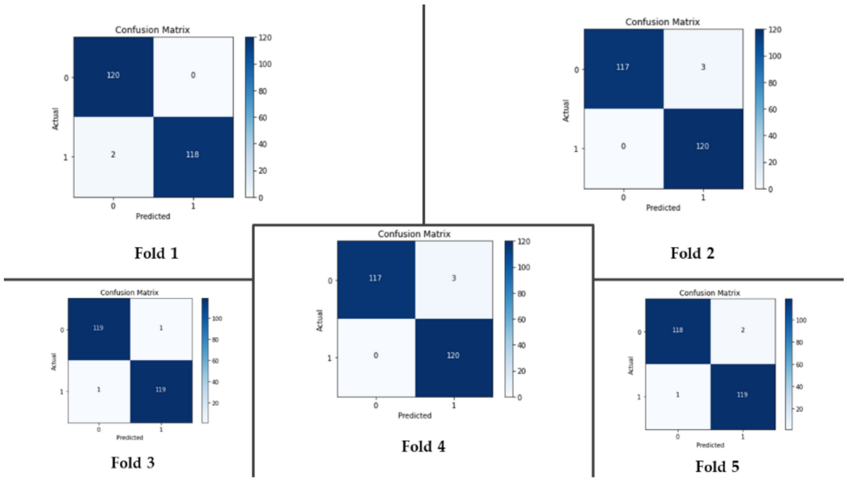


Figure 2. Confusion matrix for Dataset 1 for each set (using fivefold cross-validation).



Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Figure 3. Confusion matrix for Dataset 2 for each set (using fivefold cross-validation).

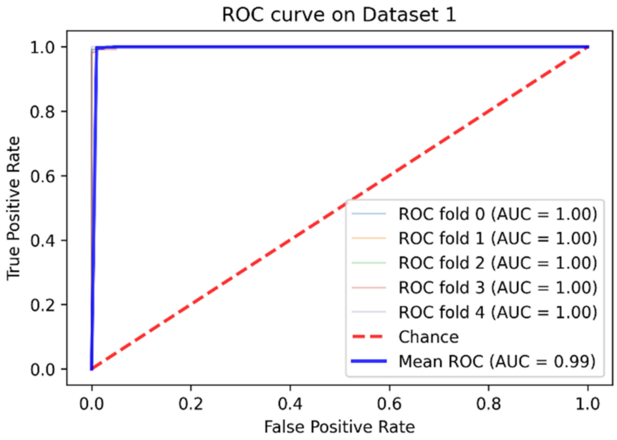


Figure 4. The ROC curve for Dataset 1 using cross-validation.

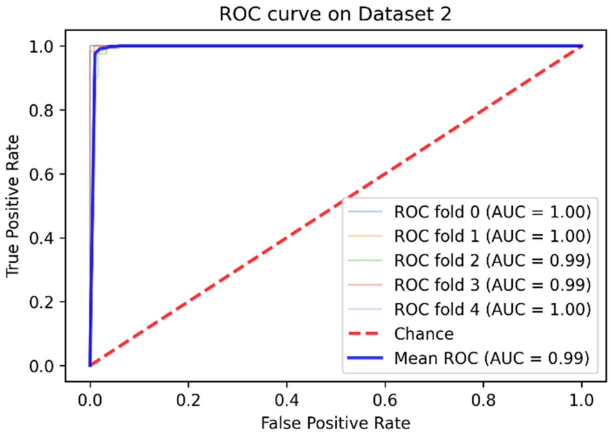


Figure 5. The ROC curve for Dataset 2 using cross-validation.

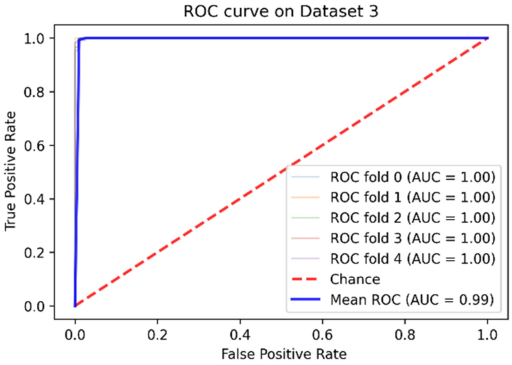


Figure 6. The ROC curve for Dataset 3 using cross-validation.

5. Conclusions

This paper proposed a multimodal architecture based on PV-DBOW and CNN models for hybrid spam e-mail detection. The PV-DBOW model was used to generate the feature vector from the text of an e-mail while preserving its semantic features, whereas the CNN was used as a feature extraction technique to extract the important features from the image of the same e-mail. The two representations were concatenated and fed to a random forest classifier to distinguish spam from ham. The experiments, which were conducted on three different hybrid datasets, showed that our proposed architecture is more efficient and outperforms the existing models according to four metrics: *precision*, *recall*, *accuracy*, and *F*1-*score*. To the best of our knowledge, there is a remarkable lack of public datasets containing hybrid spam e-mails. Furthermore, these datasets do not include the challenging image spam e-mails. For these reasons, new datasets are needed to evaluate the effectiveness of existing models. In the future, we plan to create a hybrid dataset containing challenging image spam and more difficult text spam e-mails. In addition, we will improve the detection performance of the multimodal architecture by using transformer models in order to enrich the representation of both image and text e-mails.

Author Contributions: Conceptualization, G.H., J.R. and H.T.; methodology, G.H., J.R. and H.T.; software, G.H.; validation, G.H., J.R., M.A.M., A.Y. and H.T.; formal analysis, G.H., J.R. and H.T.; investigation, G.H., J.R. and H.T.; resources, G.H., J.R. and H.T.; data curation, G.H., M.A.M. and A.Y.; writing—original draft preparation, G.H.; writing—review and editing, G.H., J.R., A.Y. and H.T.; visualization, G.H., M.A.M. and A.Y.; supervision, J.R. and H.T.; project administration, J.R. and H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We used Enron [37], Dredze, and TREC 2007 [26] datasets in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Karim, A.; Azam, S.; Shanmugam, B.; Kannoorpatti, K.; Alazab, M. A comprehensive survey for intelligent spam email detection. *IEEE Access* **2019**, *7*, 168261–168295. [CrossRef]
2. Dada, E.G.; Bassi, J.S.; Chiroma, H.; Abdulhamid, S.M.; Adetunmbi, A.O.; Ajibuwa, O.E. Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon* **2019**, *5*, e01802. [CrossRef] [PubMed]
3. Biggio, B.; Fumera, G.; Pillai, I.; Roli, F. A survey and experimental evaluation of image spam filtering techniques. *Pattern Recognit. Lett.* **2011**, *32*, 1436–1446. [CrossRef]
4. Yang, H.; Liu, Q.; Zhou, S.; Luo, Y. A spam filtering method based on multi-modal fusion. *Appl. Sci.* **2019**, *9*, 1152. [CrossRef]
5. Feng, H.; Yang, X.; Liu, B.; Chao, J. A spam filtering method based on multi-modal features fusion. In *Proceedings of the 2011 7th International Conference on Computational Intelligence and Security, CIS 2011, Sanya, China, 3–4 December 2011*; IEEE: Piscataway, NJ, USA, 2011; pp. 421–426.
6. Seth, S.; Biswas, S. Multimodal Spam Classification Using Deep Learning Techniques. In *Proceedings of the 13th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2017, Jaipur, India, 4–7 December 2017*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2018; pp. 346–349.
7. Worring, M.; Smeulders, A.W.M.; Snoek, C.G.M. Early versus late fusion in semantic video analysis. In Proceedings of the 13th annual ACM international conference on Multimedia, Singapore, 6–11 November 2005. [CrossRef]
8. Santos, I.; Laorden, C.; Sanz, B.; Bringas, P.G. Enhanced Topic-based Vector Space Model for semantics-aware spam filtering. *Expert Syst. Appl.* **2012**, *39*, 437–444. [CrossRef]
9. Hnini, G.; Riffi, J.; Mahraz, M.A.; Yahyaouy, A.; Tairi, H. Spam filtering system based on nearest neighbor algorithms. In *Proceedings of the International Conference on Artificial Intelligence & Industrial Applications, Meknes, Morocco, 19–20 March 2020*; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2020; pp. 36–46.
10. Amayri, O.; Bouguila, N. A study of spam filtering using support vector machines. *Artif. Intell. Rev.* **2010**, *34*, 73–108. [CrossRef]
11. Metsis, V.; Androutsopoulos, I.; Paliouras, G. Spam Filtering with Naive Bayes—Which Naive Bayes? In Proceedings of the CEAS 2006—Third Conference on Email and Anti-Spam, Mountain View, CA, USA, 27–28 July 2006.
12. Diale, M.; Celik, T.; Van Der Walt, C. Unsupervised feature learning for spam email filtering. *Comput. Electr. Eng.* **2019**, *74*, 89–104. [CrossRef]
13. Zareapoor, M.; Shamsolmoali, P.; Afshar Alam, M. Highly discriminative features for phishing email classification by SVD. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 649–656.
14. Gomez, J.C.; Moens, M.F. PCA document reconstruction for email classification. *Comput. Stat. Data Anal.* **2012**, *56*, 741–751. [CrossRef]
15. Cai, D.; Chang, L.; Ji, D. Latent semantic analysis based on space integration. In Proceedings of the 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, IEEE CCIS 2012, Hangzhou, China, 30 December 2012; pp. 1430–1434.
16. Ghizlane, H.; Anass, F.; Jamal, R.; Mohamed Adnane, M.; Ali, Y.; Hamid, T. Spam Filtering based on PV-DBOW model. *Int. J. Data Anal. Tech. Strateg.* **2021**, in press.
17. Srinivasan, S.; Ravi, V.; Alazab, M.; Ketha, S.; Al-Zoubi, A.M.; Kotti Padannayil, S. Spam Emails Detection Based on Distributed Word Embedding with Deep Learning. *Stud. Comput. Intell.* **2021**, *919*, 161–189. [CrossRef]
18. Gibson, S.; Issac, B.; Zhang, L.; Jacob, S.M. Detecting spam email with machine learning optimized with bio-inspired metaheuristic algorithms. *IEEE Access* **2020**, *8*, 187914–187932. [CrossRef]

19. AbdulNabi, I.; Yaseen, Q. ScienceDirect The 2nd International Workshop on Data-Driven Security (DDSW 2021) Spam Email Detection Using Deep Learning Techniques. *Procedia Comput. Sci.* **2021**, *184*, 853–858. [CrossRef]
20. Wu, C.T.; Cheng, K.T.; Zhu, Q.; Wu, Y.L. Using visual features for anti-spam filtering. In *Proceedings of the International Conference on Image Processing, ICIP, Genoa, Italy, 11–14 September 2005*; IEEE: Piscataway, NJ, USA, 2005; pp. 509–512.
21. Aradhye, H.B.; Myers, G.K.; Herson, J.A. Image analysis for efficient categorization of image-based spam E-mail. In *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Seoul, South Korea, 31 August–1 September 2005*; IEEE: Piscataway, NJ, USA, 2005; pp. 914–918.
22. Liu, Q.; Qin, Z.; Cheng, H.; Wan, M. Efficient modeling of Spam Images. In Proceedings of the Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2–4 April 2010. [CrossRef]
23. Hsia, J.H.; Chen, M.S. Language-model-based detection cascade for efficient classification of image-based spam e-mail. In *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, ICME 2009, New York, NY, USA, 28 June–3 July 2009*; IEEE: Piscataway, NJ, USA, 2009; pp. 1182–1185.
24. Gao, Y.; Yang, M.; Zhao, X.; Pardo, B.; Wu, Y.; Pappas, T.N.; Choudhary, A. Image spam hunter. In *ICASSP, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings, Las Vegas, NV, USA, 31 March–4 April 2008*; IEEE: Piscataway, NJ, USA, 2008; pp. 1765–1768.
25. Zuo, H.; Li, X.; Wu, O.; Hu, W.; Luo, G. Image Spam Filtering Using Fourier-Mellin Invariant Features. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009.
26. Dredze, M.; Gevaryahu, R.; Elias-Bachrach, A. Learning Fast Classifiers for Image Spam. In Proceedings of the Fourth Conference on Email and Anti-Spam, Mountain View, CA, USA, 2–3 August 2007.
27. Annapurna, A.; Mark, S. Image spam analysis and detection. *Artic. J. Comput. Virol. Hacking Tech.* **2018**, *14*, 39–52. [CrossRef]
28. Chavda, A.; Potika, K.; Di Troia, F.; Stamp, M. Support Vector Machines for Image Spam Analysis. In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018)—Volume 1: DCNET, ICE-B, OPTICS, SIGMAP and WINSYS, Porto, Portugal, 26–28 July 2018*; SciTePress: Setúbal, Portugal, 2018; Volume 1. [CrossRef]
29. Sharmin, T.; Di Troia, F.; Potika, K.; Stamp, M. Convolutional neural networks for image spam detection. *Inf. Secur. J.* **2020**, *29*, 103–117. [CrossRef]
30. Shang, E.X.; Zhang, H.G. Image spam classification based on convolutional neural network. In *Proceedings of the International Conference on Machine Learning and Cybernetics, Jeju, Korea, 10–13 July 2016*; IEEE Computer Society: Washington, DC, USA, 2016; pp. 398–403.
31. Kim, B.; Abuadbba, S.; Kim, H. DeepCapture: Image Spam Detection Using Deep Learning and Data Augmentation. In *Proceedings of the Australasian Conference on Information Security and Privacy, Perth, WA, Australia, 30 November–2 December 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12248, pp. 461–475. [CrossRef]
32. Srinivasan, S.; Ravi, V.; Sowmya, V.; Krichen, M.; Ben Noureddine, D.; Anivilla, S.; Soman, K.P. Deep Convolutional Neural Network Based Image Spam Classification. In *Proceedings of the 2020 6th Conference on Data Science and Machine Learning Applications, CDMA 2020, Riyadh, Saudi Arabia, 4–5 March 2020*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2020; pp. 112–117.
33. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, PMLR, Beijing, China, 21–26 June 2014; Volume 32.
34. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2013; pp. 3111–3119.
35. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef] [PubMed]
36. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Klimt, B.; Yang, Y. The Enron Corpus: A New Dataset for Email Classification Research. In *Proceedings of the European Conference on Machine Learning, Pisa, Italy, 20–24 September 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 217–226.