Email Spam Detection Using Convolutional Neural Network

1st Babu Pallam
P2849288
MSc. Artificial Intelligence
De Montfort University
p2849288@my365.dmu.ac.uk

***Abstract*—Email providers have been trying to solve the issues with spam emails for decades. As technology changes and advancement happens, the techniques being used have been revamped according to the severity of the threats and productivity loss. This paper presents an approach to detect email spams more accurately, compared to the traditional spam detection methods available today. The proposed approach uses Convolutional Neural Networks (CNNs). It is a well known deep learning algorithm, among several algorithms available today. Though CNN is known at present for problems in image segmentation, and computer vision, where complex analysis and computations are required, this paper uses CNN for text based email spam detection. Based on the labeled email dataset chosen, the design and training of three models have been provided in this paper, where each model is different from one another based on complexity and functionality in layers of CNN. In addition to that, the performance analysis done based on the three models have been detailed in this paper. Based on the evaluation matrix and Hyper parameters such as learning rate, batch size, optimizer function, and activation function, the best model that performs efficiently has been determined. This research highlights the potential of CNN in enhancing the effectiveness of spam detection algorithms, along with the possibility of further research in this specific domain.**

***Index Terms*—Email spam detection, Convolutional Neural Networks, Performance Analysis, deep learning.**

I. Introduction

Emails have been classified into ‘spam’ or ‘ham’, depends upon the relevancy of content of the emails. Spam emails are the irrelevant emails sent for various purposes, like commercial advertisements, fraudulent, and malicious or phishing purposes. Whereas the ‘ham’ emails are those which are legitimate and relevant to the recipient. The content of those emails would be solicited and personalized, and the purpose of those emails are well defined. Isolating spam emails from ham emails have been a big challenge on the Internet. Including higher resource consumption, security risks, financial loss, the damage of reputation, and other ethical issues, the email service providers on the Internet have implemented several techniques, and revamped those techniques by adapting the state of art technologies and innovation over time. Introduction of Artificial Intelligence took these spam detection algorithms into another level by introducing automatic detection mechanisms. Before machine learning algorithm were in use, but now Neural network associated deep learning algorithms have been discovered to increase the effectiveness of the implementation. This project is an effort made to implement this email spam detection with the help of Convolutional Neural Networks (CNNs), and check performance analysis by creating three different models for comparison and analysis. The selection of CNN for email spam detection system has been motivated by several factors, including its complex structure to handle image processing tasks, the effectiveness in use for feature learning, scalability in terms of handling large datasets and several others.

To conclude, the purpose of this paper is to propose an approach, then design and implement that approach to email spam detection system using CNN. An introduction of CNN architecture has been included in the next section. Followed by that, implementation of the models have been discussed. Later, in the following paragraph, the results of the experiment are done, which is more like tuning of parameters and computational approaches (functions) in search of an improved algorithm. In that next section, the analysis of the result has been detailed; which includes, the inferences made after a comprehensive comparison of evolution matrix, and hyper parameters, associated with each model. Later conclusion and future work has been included.

II. Background Work

A. Convolutional Neural Network (CNN)

1) Introduction: CNN is a type of artificial neural network(ANN) [1] in deep learning [2] which is primarily used for image processing and computer vision. This neural network is inspired from how the brain processes visual data. CNN is widely used for pattern recognition, feature extraction, and several others. This makes CNN more popular in applications like object detection, face recognition, real time stream analysis and several others. Real world examples include, self driving cars by Tesla [4], where CNN is used for identifying traffic, other vehicles, objects like pedestrians and traffic signs.

2) CNN Model Architecture: CNN architecture is designed based on how the data is being processed. The basic CNN contain three fundamental layers, which are as follows: 1) Convolutional layers, 2) pooling layers, and 3) fully connected layers.

1) **Convolutional Layers:** It is the core of CNN. It does convolution operation to the inputs and pass the output

resulted to the adjacent layer next to it. The convolution operation has a set of kernels, which is considered as filters, which works with the input data and construct a feature map. Each filter is associated with a specific task, which can be extraction of textures, patterns, or others. Several convolutional layers are available now depends upon the data to be handled. For text analysis the Conv1D layer [8] is being used. For image analysis, Conv2D or Conv3D [9] and so many others with different dimensions and functionalities are available in TensorFlow [10]

2) **Pooling Layers:** These are the adjacent layers of convolutional layers. These layers take the feature maps created by convolutional layers and reduce the spatial dimensions. This helps to reduce computational time, and memory usage. The common type of pooling operation is max pooling; in which reduction of dimension, extraction of features from feature map by taking maximum value from each region of feature map, noise reduction and several other functionalities are embedded. Main objective is to extract key features of the data and minimize the complexity as possible.
3) **Fully-Connected Layers:** These layers are also known as Dense layers. These layers receives input from pooling layers, which would be in multidimensional vector. The main objective of this layer is to convert the feature extracted by the pooling layers or convolutional layers into probabilities with different classes or predictions [2]. The steps involved are as follows. First, the vectors are flattened into single vector. Then, the matrix multiplication would be performed over this vectors with a weight, and apply activation function to introduce non linearity. This layer connects each neuron of one layer to the neurons of the next layer. In common, the dense layer would be the final layer of a CNN model.

B. Tools and Terminologies Used in This Report

1) TensorFlow and Keras:: TensorFlow [10] is a well known machine learning framework. It is developed by Google for the design and implementation of machine learning models. This library provides diverse of in build functions that implements machine learning algorithms and deep learning algorithms effectively. In addition, this library consists of multiple tools that can be used easily to test and tune the performance of the model created with least amount of time. This project uses TensorFlow over anaconda platform [11] with python language with the help of Visual Studio Code Toolbox [12].

2) Optimization Algorithms:: Optimization algorithm play a significant role in improving efficiency of the neural network model. These are functions that are used to update the parameter of the model to reduce the loss while training the algorithm. Commonly used Optimization algorithms in CNN are Stochastic Gradient Descent (SGD), Nesterov Accelerated Gradient, Momentum, Adam (Adaptive Moment Estimation) and so on.

3) Activation Functions:: An activation function is a computation unit associated with a neuron, which takes input from the previous layer and determine the importance of the information lies in that data to the next immediate layer. It helps to learn the patterns as well as the relationships of points lies in the data. Several activation functions which are used commonly in real world problems. Some of them are Sigmoid, Tanh, and ReLU (Rectified Linear Unit), ELU(Exponential Linear Unit), and so on [13].

4) Loss Function:: The loss function play a significant role in training of the model by quantifying prediction errors, aligning learning objectives with task requirements, and improving model evaluation. Depends upon the problem loss function can be different. For instance, in the case of regression problems, Mean Squared Error (MSE) is being preferred. And more, Binary Cross-Entropy Loss is preferred mainly for binary classification problem.

C. Related Research

Recently several research has been conducted in detecting phishing emails that use CNN. The model proposed in 2021 by Sergio for phishing Email Classification [3] lighten the possibility of CNN for text analysis. Later in 2022, Yuxin Liu and others [5]reviews CNN with machine learning algorithms (CNN and Bi-LSTM) for the same problem. Another recent work done by G. M. Shaharia [6], compares the efficiency of CNN and other machine learning algorithms such as Naıve Bayes(NB), Linear Regression, and Support Vector Machine(SVM) for a system which filters email spams. Similar to the idea of email spam detection, a research has been conducted by Fanjun Meng [7] which proposes a system for network spam detection using CNN Incorporated with Attention Model.

III. Implementation and Variants

The implementation of basic model has been described in the first section. Followed by how the models differ each other has been detailed.

A. Implementation

1) Data Splitting: The dataset selected for email classification that contain spam emails as well as ham emails has been prepossessed collaboratively, removing unnecessary punctuation, and special characters, and cleaning by converting all characters into lowercase letters. The prepossessed data has been taken into further by applying label encoding, in which spam and ham labels are converted into numerical values such as 1 for spam and 0 for ham. The splitting of database has been carried out into two for training, validation and testing of the model. In which 80% has been selected for training and 20% has been selected for testing.

2) Model Architecture Design: The model has been created based on different layers in the order which is specified in the section II A. The sequential API of TensorFlow has been used to stack the layers sequentially. The layers used are listed below.

- *Embedding Layer:* This layer has been used to initialize the model. It converts each word into numerical vectors, represented by a vector in a high-dimensional space.
- *Convolutional Layer:* It is used for convolution operations. Here in this model, it does extract features from sequences of words.
- *A Bidirectional Long Short-Term Memory (Bi-LSTM) layer:* It processes input data in both directions sequentially, capturing dependencies in both directions. LSTM technique is used for find the long term dependencies between the words which are involved in the email.
- *GlobalMaxPooling Layer:* This layer is used to minimize the dimension of the input data. There are several layers similar to this layer available in use.
- *Dropout Layer:* This layer is used to reduce the over-fitting problem. Since the literature review done found out the research [14] done by Dishara highlighted over-fitting as a problem in CNN. The use of this layer is considerable.
- *Dense Layer:* This has been used to increase high learning capability of the model.

The above layers are combined in different combinations and three model have been created for comparison and performance analysis, which has been described in section III B.

The specification which is static in all the variants implemented are given in Table I.

TABLE I
SPECIFICATIONS OF THE IMPLEMENTATION

Common Hyperparameters and Parameters	
Parameter	**Value**
Embedding Dimension	50
Max Sequence Length	100
Number of Filters	64
Filter Size	5
Dropout Rate	0.5
Optimizer	Adam
Loss Function	Binary Crossentropy
Number of Epochs	5
Final Dense Layer units	1
Final Dense Layer Activation Function	sigmoid

3) Model Compilation: In this step, the model created in the above step has been compiled. The compilation involves configuring three things which is important in the learning capability of the model. Which are as follows.

- *Optimization Algorithm:* Among several optimization algorithms available, the initial model has been implemented using ”adam” optimizer function for all the variants.
- *Loss Function:* The problem which is being solved here is binary classification problem. That is two classification, such as spam or ham. So ”Binary Cross-Entropy Loss” has been chosen and made it static through out the project.
- *Evaluation Metrics:* The problem to be solved in this research is a binary classification problem, matrix parameters such that accuracy, precision, recall, and F1-score are selected from the evaluation matrix. This helps to monitor the training of the model as well as the testing of the model which should be done after training phase.

The model has been compiled with the specification mentioned above.

4) Model Training: The compiled model has been trained using the training data which has been selected in the first phase of the implementation. The parameter specification used are as follows.

- Epoch(the number of iteration model has to perform through the entire dataset) = 5
- batch_size(count of sample processed at a time) = 64

The models have been trained with the above specification and the training progress have been depicted as a graph. Then the models have been investigated further for further analysis.

B. Different Variations

Three models have been created based on the complexity. This has been accomplished by introducing one or more layers into the simple model. As shows in the Table II, Variant 1 uses simple layers along with Conv1D layer, which is mainly used for text analysis or semantic analysis. Variant 2 provides a model where it is more complex compared to variant 1, which is because of introduction of Dense layer with 128 neurons and the Dropout layer to avoid over-fitting problem. Variant 3 is an extension of Variant 2, by adding an extra layer named Bidirectional LSTM.

TABLE II
COMPARISON OF MODEL ARCHITECTURES

Layer Name	**Variant 1**	**Variant 2**	**Variant 3**
Embedding	✓	✓	✓
Conv1D	✓	✓	✓
GlobalMaxPooling1D	✓	✓	✓
Bidirectional LSTM			✓
Dense (128, ReLU)		✓	✓
Dropout		✓	✓
Dense (1, sigmoid)	✓	✓	✓

IV. EXPERIMENTAL RESULTS

The test has been carried out with the three different variants implemented. The result obtained have been summarised in Table III.

As shows in Table III, several tests have been done on the three Variants, determining evaluation matrix associated with the model. In addition to that, evaluation matrix has been determined for different values of hyper parameters that have significant impact on the model. Finally, the result obtained on resource utilization parameters after several tests have been provided.

V. ANALYSIS AND DISCUSSION

This section begins with a comprehensive analysis of the model architecture created based on the experiment result summarized in the above section. Then Detailed discussion of some of the significant areas which need to be focused for an effective solution has been included.

TABLE III
EXPERIMENTAL RESULTS OF THREE VARIANTS

Parameter	Variant		
	1	2	3
Evaluation Matrix			
Accuracy (Ac)	0.983	0.980	0.986
Precision	0.96	0.91	0.95
Recall	0.90	0.93	0.93
F1-score	1	1	1
Hyper Parameters			
Learning Rate	0.03 (Ac:0.935)	0.001 (Ac:0.988)	0.001 (Ac:0.988)
Batch Size	64 (Ac:0.988)	128 (Ac:0.988)	128 (Ac:0.989)
Optimizer	Adam (Ac:0.986)	Adam (Ac:0.989)	Adam (Ac:0.988)
Activation Function	sigmoid (Ac:0.987)	elu (Ac:0.988)	sigmoid (Ac:0.988)
Resource Utilization			
Training Time (seconds)	4.78	3.36	32.97
Inference Time (seconds)	0.235	0.167	1.144

A. Analysis of Results

This section presents the observations found after the analysis of the result obtained from several experiments.

1) Evaluation Metrics: The main observation found after analysis are listed below

1) *Accuracy:* Model 3 achieves the highest accuracy (98.6%), followed by Model 1 (98.3%) and Model 2 (98.0%). Model 3 is slightly more accurate in making correct predictions compared to the other two models.
2) *Precision:* Model 1 exhibits the highest precision (96%). which indicates that it accurately identifies positive for majority of the positive predictions. Where as Model 3 follows closely with a precision of 95%. Model 2 shows the lowest of all which is (91%).
3) *Recall:* Model 2 demonstrates the highest recall (93%). Which means that this model can effectively identifies all relevant instances of the positive class. Both Model 1 and Model 3 have a recall of 90%, slightly lower than Model 2.
4) *F1-Score:* All models achieve a perfect F1-score of 1, indicating an excellent fairness between precision and recall. There is an optimum balance between making accurate positive predictions and identification of the right class for the input.

2) Hyper-parameters: Based on some of the important parameters analysed the following conclusions has been made.

- *Learning Rate:* Variant 1 utilized a higher learning rate of 0.03, but accuracy found to be 93.5%. Variant 2 and 3, with a lower learning rate of 0.001, achieved significantly higher accuracy's of 98.8
- *Batch Size:* Variant 1 had a batch size, which is equal to 64, with an accuracy of 98.8%. Variants 2 and 3, with a larger batch size of 128, also achieved high accuracy of 98.8% and 98.9%, respectively.
- *Optimizer:* All three Variants found the Adam optimizer as the best. Variant 2 achieved the highest accuracy (98.9%), followed closely by Models 1 and 3, both at 98.8
- *Activation Function:* Variant 1 found the sigmoid activation function as best with the accuracy of 98.7%. Variant 2 uses the ELU as the activation function and resulted into the highest accuracy of 98.8%. Model 3 also used the sigmoid activation function and received an accuracy of 98.8

Overall, Variant 2 found to be the best after analysis of hyper parameters in terms of achieving the highest accuracy with a lower learning rate, the ELU activation function, and the Adam optimizer.

3) Model Architecture: Based on the whole modeling, the following conclusions are made about each Variant.

- Variant 1: Achieves high accuracy and F1-score shows high capability of the model towards the prediction. The precision and recall scores suggest a well-balanced model in terms of classification. As can been seen from Table III, Larger batch size (64) found to be best with high accuracy, but learning rate is high (0.03) for best accuracy.
- Variant 2: Slight decrease in accuracy compared to Variant 1. But, the Variant 2 found best in terms of precision and recall. Which shows classification would be better.
- Variant 3: Highest accuracy among the three variants, with good precision and recall(, since F1-score is 1). The Bidirectional LSTM used in this model can capture sequential dependencies effectively. Which resulted an improved performance. Dropout layer helps in regularization, preventing over-fitting. But this model found to be more complex compared to rest of the models.

B. Detailed Discussion

The following provides different perceptions observed while doing the comparison and performance analysis.

1) Selection of Model: Choosing the best model from the above three is significant in order to solve the problem efficiently. Based on the evaluation matrix discussed in section V A, a strategy can be found for choosing a model. Choosing a model depends on the requirement and constraints of the problem. Variant 3 can be selected, if maximizing accuracy is preferred. Variant 1 can be selected, if precision is more important.

2) Computational Efficiency: The computational efficiency was one of the main focus during the research. The reason was the complexity of the model was different from Variant 1 to Variant3. So the time taken for training the model and inference time has been calculated for each model. Training time is the time taken by the CNN model during the training phase, including the pattern recognition, and the whole process. Whereas, Inference time is the time taken by the trained model to perform prediction of the output with new data. Based on the result, shown in Table III, the following observations have been found.

- Training time: Variant 1 and Variant 2 have relatively shorter training times compared to Variant 3. The reason can be drawn that Variant 3 uses complex architecture, so more parameters need to be trained. Bidirectional LSTM are more complex models compared to simple CNN layers. So, it does require more computations.
- Inference time: Inference time found to be different for all variants. Variant 2 has the shortest inference time, followed by Variant 1. Variant 3 has the longest inference time. This difference in inference time can be attributed to the architectural differences between the variants. Variant 2 has fewer layers compared to Variant 3, resulting in faster inference. Same reason can be applied here as that has been discussed about increase of training time in Variant 3.

Variant 2 found to a balanced model in terms of complexity and computational efficiency. It does have shorter training and inference times compared to both Variant 1 and Variant 3. Variant 3, although providing the best performance in terms of evaluation metrics. But in terms of complexity, this model is inefficient. Depends upon the requirements, and number of constraints put into, the model can be selected.

3) Best Model Proposed: Selection of a model as the best among the three models implemented depends on several factors. The different case where one model can be preferred over another has been discussed in the above sections. Variant 1 could be a best option for resource-constrained environments. Variant 2 is best for applications where model complexity and over-fitting need to be addressed. Variant 3 is preferable were high predictive accuracy is significant.

Though variant3, found to be the best among all, the scalability of the model found to be a challenge in this research. This research found it because of two reasons. First reason is its difficulty while dealing with large datasets. Second reason is the deployment challenge in the real world because of the requirement of high computations.

To conclude, this research put forward **Variant 3 as the best model** due to its stability in terms of evaluation matrix, and other performance parameters. The graph showing the model accuracy and model loss has been provided in Fig. 1.

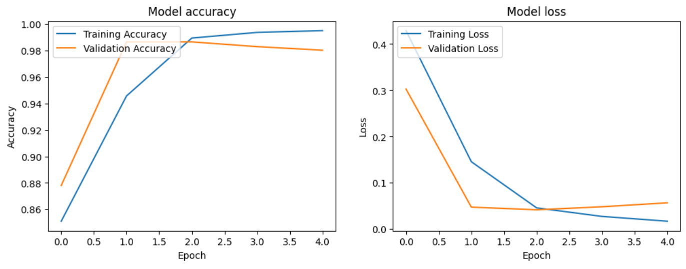

Fig. 1. Model Accuracy-Loss Graph

From Fig. 1, it is clearly seen that the model does have high training accuracy and high validation accuracy. In an ideal model, the curves would be completely overlapped. here training accuracy curve sits above validation accuracy curve, so slight over-fitting exist. So with unseen data, the performance might be a bit less, but reasonably well because the precision was 95 (based on Table III).

4) Challenges and Space for Further Research: This section presents a detailed discussion on the challenges observed and areas where further research is mandatory.

1) *Computational Tools:* The requirement of high computational resources for training and evaluation of these models is found to be important. For instance, large dataset is important for effective training, but the computational effort would also be very high. Another thing is the number of epoch parameter chosen while training of the model. This research has to put epoch as 5 to minimize the effect of computational challenge. The performance analysis of the models based on effect of epoch parameter should need to be further investigated. In addition to that, this research has seen the importance of exploring to development of new algorithms which reduces the computational cost irrespective of scalability in the future.
2) *Over-fitting Problem:* Over-fitting is a persistent challenge where model has to deal with large number of parameters. Variant 2 does has a dropout layer to regulate the over-fitting, but there is still space for improvement. The research should explore further with different regularization techniques.
3) *Use of Large Dataset:* The dataset chosen here is small, which is intensionaly chosen to minimize the computational efforts. Further research could investigate on large datasets with different types of emails, type of contents for fine tuning.
4) *Application of Different CNN Models:* Application of different CNN layers and do evaluate the performance of the model obtained is found to be important to ensure the robustness of the model. Investigating the possibility of combining CNN model with different Neural network algorithms, such as Recurrent Neural Networks(RNNs), is significantly appreciated.

To conclude, the analysis of the models has formulated several observations. The observations formulated put forward several hints in which the research should be focused or to be extended. Exploring new directions envision new advancement in the area of deep learning and natural language processing implementations.

VI. Conclusion

This paper has presented design, implementation, and then evaluation of three variants of Convolutional Neural Networks for email spam detection. The architecture of the CNN architecture followed during the research has been provided at the beginning. Followed by, the tools, concepts, and terminologies used for a successful performance analysis had been highlighted. Then the results obtained during the performance analysis has been provided based on evaluation metrics, hyper-parameters, and parameters associated with the resource utilization. The detailed discussion based on the results has drawn several observations about how to select a model, the complexity and computational efficiency, along with the best model selected during the research. Several hints has been presented where the challenges faced during the research and for several necessity of further research found.

REFERENCES

[1] McCulloch, W. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin Of Mathematical Biophysics*. **5**, 115-133 (1943)

[2] Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning. (MIT Press,2016), http://www.deeplearningbook.org

[3] McGinley, C. & Monroy, S. Convolutional Neural Network Optimization for Phishing Email Classification. *2021 IEEE International Conference On Big Data (Big Data)*. pp. 5609-5613 (2021)

[4] Bojarski, M., Testa, D., Dworakowski, D., Firner, B., Flepp, M., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J. & Al. End to End Learning for Self-Driving Cars. *CoRR*. **abs/1604.07316** (2016)

[5] Liu, Y., Wang, L., Shi, T. & Li, J. Detection of spam reviews through a hierarchical attention architecture with N-gram CNN and Bi-LSTM. *Information Systems*. **103** pp. 101865 (2022)

[6] Shahariar, G., Biswas, S., Omar, F., Shah, F. & Hassan, S. Spam review detection using deep learning. *2019 IEEE 10th Annual Information Technology, Electronics And Mobile Communication Conference (IEMCON)*. pp. 0027-0033 (2019)

[7] Meng, F., Pan, Y. & Feng, R. Network Spam Detection Based on CNN Incorporated with Attention Model. *2022 8th Annual International Conference On Network And Information Systems For Computers (ICNISC)*. pp. 111-116 (2022)

[8] Developers, T. tf.keras.layers.Conv1D. (TensorFlow,2015). https://www.tensorflow.org/apidocs/python/tf/keras/layers/Conv1D

[9] Chollet, F. Deep Learning with Python. (Manning Publications Co.,2018)

[10] Developers, T. TensorFlow. (https://www.tensorflow.org/,2024)

[11] Continuum Analytics, I. Anaconda Platform. (Continuum Analytics, Inc.,2024), https://www.anaconda.com/products/distribution

[12] Microsoft Visual Studio Code. (Microsoft,2015), https://code.visualstudio.com/

[13] Towards Data Science Activation Functions in Neural Networks. *Towards Data Science*. (2019), https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[14] Dishara, H. & Muhammed, L. A Review of the Overfitting Problem in Convolution Neural Network and Remedy Approaches. *Journal Of Physics: Conference Series*. **1432**, 012002 (2020)