Journal of Advanced College of Engineering and Management

Real-Time Spam Detection in Chat Systems Using Naive Bayes, CNN, and OCR Techniques

[1]Prasanna Adhikari, [2]Anku Jaiswal, [3]Aakash Shrestha, [4]Amar Sunar, [5]Ankit Pradhan

[1,3,4,5]*Department of Computer Engineering, Khwopa College Of Engineering, Nepal*

[2]*Department of Electronics and Computer , Institute of engineering, Pulchowk campus*

Email: anku.jaiswal@pcampus.edu.np

DOI: 10.3126/jacem.v11i1.84523

Abstract

In the modern era of digital communication, platforms such as email, SMS, and instant messaging have become integral to daily interactions. However, this widespread adoption has also led to an increase in spam content, which has evolved from traditional text-based messages to more sophisticated image-based formats in an effort to bypass conventional filters. This paper presents a hybrid spam detection approach integrated into a custom chat application, utilizing both text and image classification techniques. For text-based spam detection, a Naive Bayes classifier is employed, achieving an accuracy of 97%. To address image-based spam, a Deep Convolutional Neural Network (CNN) is used, attaining an accuracy of 96.87%. Additionally, an OCR-based method using Tesseract is implemented to extract textual content from images, which is then analyzed using the text classifier. The proposed approach demonstrates efficient processing times, with text messages classified in under one second and image messages within five seconds on web platforms. On mobile devices, image spam classification required approximately one minute and ten seconds. The effectiveness and responsiveness of these techniques for real-time spam detection are thoroughly evaluated in this paper.

Keywords*—Text Spam, Image Spam, CNN, Naive Bayes, Deep Neural Network, Chats*

1. INTRODUCTION

In today's digital era, messaging systems such as email, chat applications, and SMS play a crucial role in communication. However, the widespread use of these platforms has led to a significant increase in unsolicited and irrelevant messages, commonly known as spam. Spam messages, particularly prevalent in email systems, are often used for advertising purposes. These unwanted communications not only occupy valuable storage space but also consume significant network bandwidth, power, and user time ultimately impacting server efficiency and communication quality. Studies suggest that spam constitutes approximately 85% of all email traffic [1], emphasizing the urgent need for effective spam detection mechanisms.

While textual spam has existed since the early days of the internet, modern spammers increasingly exploit images to bypass traditional text-based filters. As a result, spam detection has evolved from simple rule-based approaches to advanced machine learning techniques. This paper explores both text-based and image-based spam detection methods. Specifically, it examines the use of the Naive Bayes algorithm for detecting textual spam and Convolutional Neural Networks (CNNs) for identifying image-based spam. Furthermore, a prototype application has been developed to demonstrate the practical implementation and effectiveness of the proposed models.

2. LITERATURE REVIEW

Choudhary et al [2] proposed a model based on ten features.A binary classifier was proposed by applying the feature vectors of spam and ham messages. Also features like presence of URLs, dots, mathematical symbols, emotions, mobile numbers, etc were extracted and these features were used in SMS Spam Corpus dataset. Na¨ıve Bayes, Logistic Regression, J48, Decision table and Random Forest algorithm were used for testing. Also results were best achieved with Random Forest Classifier v with highest TP rate (96.5%) and lowest FP rate (1.02%).

Karami et al. [3] proposed a content based SMS spam detection Method where two features i.e. LIWC features and SMS specific features were extracted and then classified. This proposed method was used in a dataset collected from Grumbletext website, the National university of Singapore, a Phd Thesis and another website. This paper proposed supervised ML algorithms and around 40 classification algorithms were tested with different testing settings by using Weka. The conclusions were that boosting of Random Forest and SV showed the best performances with 98.47% and 97.99% accuracy respectively.

Shirani-Mehr et al. [4] experimented with a different machine learning algorithm to SMS spam classification problem. The database used was a UCI machine learning repository. Different techniques like Bayes, SVM and other methods were applied. Feature extraction and analysis were done in MATLAB and algorithms were done in python using scikit-learn library. The paper concluded that SVN was best used for the dataset with 97.64% accuracy.

Xu et al [5] proposed features like static, temporal and network features which were feeded to the classification algorithm (SVM and K-NN). The idea in this paper is to use noncontent features where SVM was used to pay attention to margins and cases near hyperplanes while K-NN focused on typical positive and negative cases. In the Telco dataset, the paper shows a comparison between SVM and K-NN classifiers where it showed that SVM classifiers can achieve better performance. The paper also has shown comparison among different feature categories using ROC curve.

El-Alfy et al. [6] suggested using part-of-speech and recognized-entities tags while pre-processing and used features like likely spam words, URLs, special characters, emotion symbols, gappy words, javascript codes, metadata, function words etc. The paper proposed a dendritic cell algorithm (DCA) on combined datasets from Spambase, SpamAssassin, TREC2005, Corpus V.0,1 Big and Spam Collection V.1. In the paper, it concluded that using five benchmark datasets, the empirical results showed improvement is achievable over base classifiers (close to 100% accuracy).

Srinivasan et al [7] used three dataset ISH, Improved and Dredze ImageSpam dataset. He suggested pre-processing of the dataset like removal of duplicate images and corrupt files. He suggested his two CNN networks adding them with cost-sensitive learning and transfer learning models like VGG19, DenseNet201, Xception, and ResNet152V2. He concluded that his CNN2 model performed better than all pretrained models except VGG19 giving the highest accuracy of 97.4% in Dredze PSpam and Pham dataset.

Chowdhury et al. [8] proposed an architecture for image spam detection which consisted of feature extraction, feature selection and a BPNN classifier.File feature extraction like aspect ratio, file size, image area, etc and also visual features extractions from HIS color histogram were suggested . Also the performance using BPNN on Corpora dataset with accuracy of 97.89 was measured

Dinesh K, Amara [9] suggested using a convolutional network with Fully Connected layer. Also an accuracy of 91.7% on the dataset was achieved which was collected from different sources all being RGB images in various dimensions.

Sharmin et al. [10] suggested using a canny edge detector for extracting edge related information of the images. Raw and canny images were suggested and augmented features were used. SVM, MLP and CNN models were experimented in ISH dataset, Challenge dataset1 and Challenge dataset 2. As a conclusion, CNN achieved best accuracy among ISH and challenged 1 dataset with accuracy being 99.02% and 83.13% respectively.

Kim et al. [11] achieved 85% accuracy using CNN-XGBoost classification in Deep Capture in Public Spam and SpamArchive plus Personal Ham and Normal Image ham Dataset. Data augmentation in DeepCapture was proposed. Also methods like splitting spam images and combining with other images was used to create a new augmented data similar to that of the dataset. And searching google Image search API for ham dataset for adding new data in dataset.

Pednekar M, Ashutosh [12] collected the data manually and labelled them himself. He used conv-nets to train the images over his dataset using Nvidia's CUDA platform and the awesome Tensor Flow Framework. He achieved a training accuracy of 79.4%

accuracy.

Singh Pal, Ajay [13] experimented on CNN, DNN, VGG19 on different datasets like ISH, improved, dredze personalized spam archives and improved (combined too). The paper further concludes with use of GAN for the sole purpose, with a combination of CNN, RNN or RFE and UFS .

3. METHODOLOGY

3.1. SYSTEM DESIGN AND MODELLING

The basic overview of the whole system is shown in figure 3.1.It is a basic client server architecture framework where clients can send/recieve messages with help of socket. These messages are classified as spam or ham in respective machine learning models deployed in application.

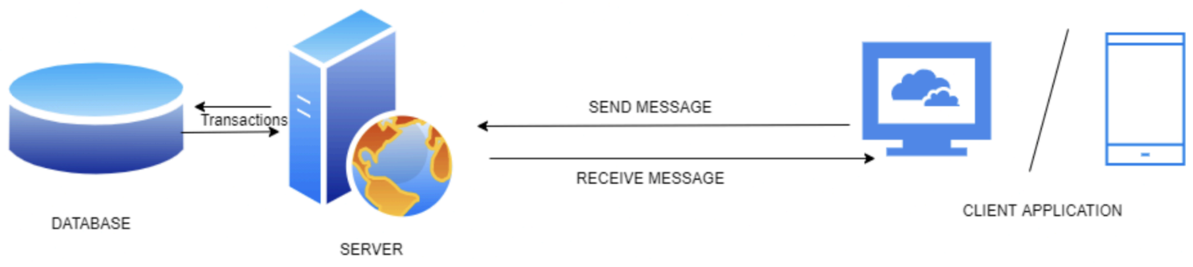

Figure 3.1: System Overview of Chat Application

3.1.1 System Flow

The basic system flow is shown in 3.1.1. User can send both text and image messages. The text messages are classified as spam and ham directly, while the images are classified in two ways, using CNN and Naive Bayes. For text, Naive Bayes is used. Hence to detect Spam or Ham in images, the text is extracted from tesseract.js and that text is fed to Naive Bayes Model for prediction. According to the prediction the database is updated and hence rendered in the application.

The system flow is

1. User send a message
2. If the message is image then

a Extract Text using OCR and feed the text to Naive Bayes for prediction

b Feed the image to CNN model for prediction else Feed the text to Naive Bayes

3. Update the database according to the prediction i.e. spam or ham

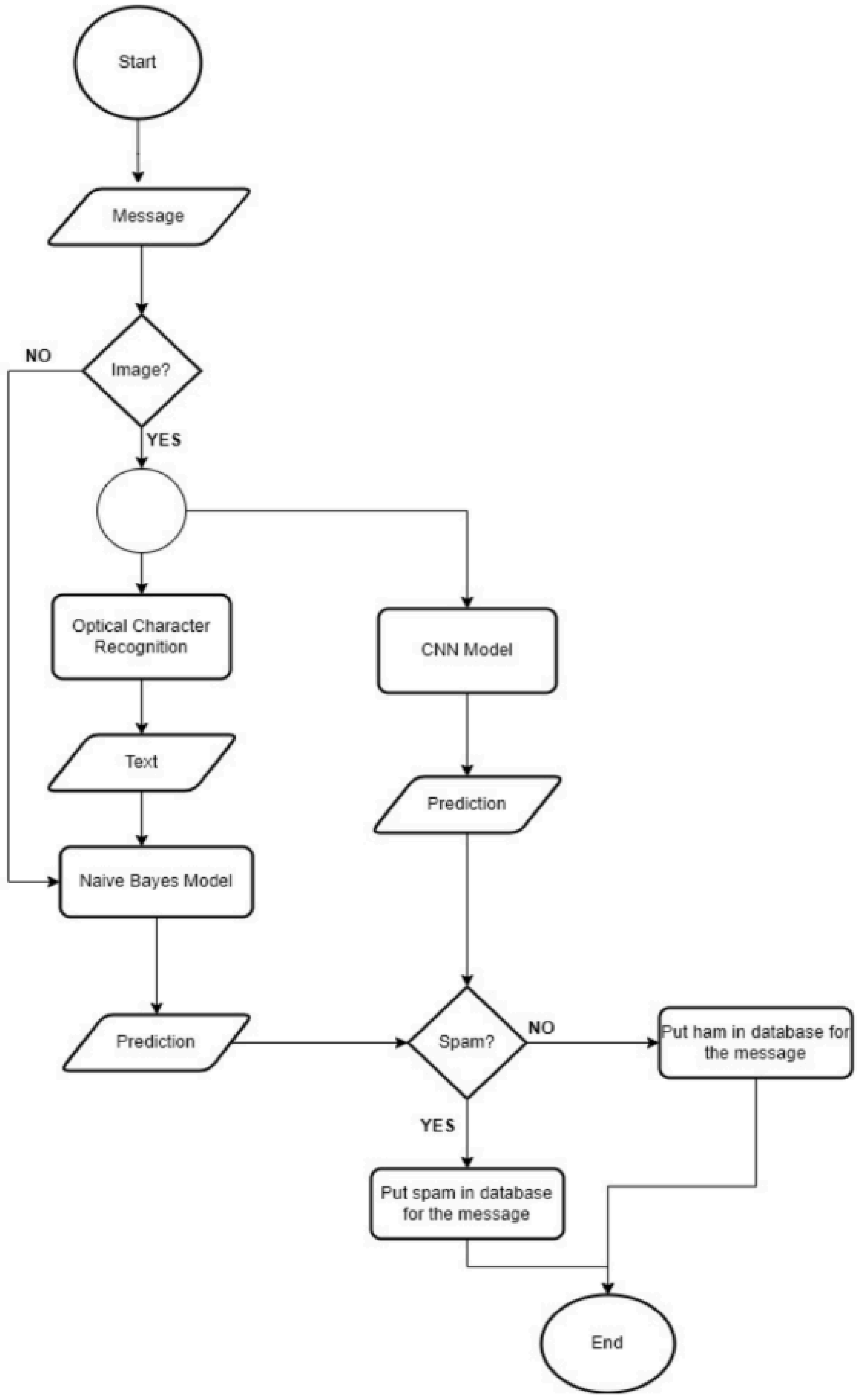
Start
Message
Image?
NO
YES
Optical Character Recognition
CNN Model
Text
Prediction
Naive Bayes Model
Prediction
Spam?
NO
Put ham in database for the message
YES
Put spam in database for the message
End

Figure 3.2. Flow diagram of Spam Classification

3.2. MODEL DESIGN AND ARCHITECTURE

3.2.1 Pipeline for Text classification

The pipeline for classification of text message as spam or ham proceeds as follows:

1. Count Vectorizer: It converts the word present in training data-set to vector form so that mathematical calculations can be carried further.
2. TF-IDF: It is a feature extraction mechanism that quantifies a word in documents, which computes a weight to each word and signifies the importance of the word in the document or corpus.
3. Multinomial Naive Bayes Model: It is a bayesian approach to calculate likelihood of the word to be spam or ham.

3.2.2 CNN Model for Image classification

The CNN architecture for image spam classification is shown in figure 6.6

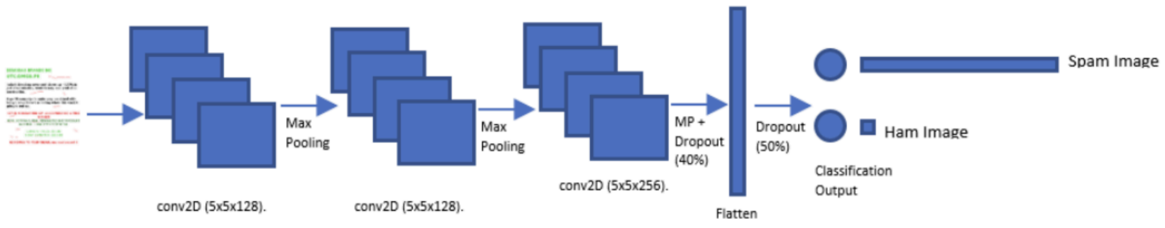

Figure 3.3. CNN Model used for Image Spam classification

4. DATA COLLECTION AND CLASSIFICATION

4.1. Data Acquisition

For Image Spam Classification, we used two dataset as below.

• *Image Spam Hunter Dataset (ISH) :*

This data was collected from [15] It is a publicly available dataset. There are 810 ham having 810 unique images and 929 spam having 870 unique images.

• *Dredze ImageSpam Dataset :*

The dredze image spam dataset was collected from [16] and work was done on 3 sets of Images. Personal Ham has 2021 images having 1517 unique images. Personal Spam has 3298 images having 1274 unique images.Spam Archive has 16028 of various formats like JPEG, BMP, PNG etc in which there are 3039 unique images. It is also available freely.

4.2. Data Pre-Processing

Initially, we merge the above listed separate datasets into a single dataset within their respective class. Some images were corrupted and had unsupported format so to bring uniformity we removed some un necessary images. After this, we generate batches of tensor image data with real time data augmentation using Image data generator where we re-scale the input image, shearing in counter clockwise direction, rotation and horizontal flippings. Then we implement callback functions for early stopping, reducing learning rate, and creating checkpoints.

4.3. Modelling

Finally we create our CNN model for classification. The CNN model consists of three convolutional layers and three max pooling layers with addition of dropout to fit the model properly. The sequential CNN model takes input images of size (224,224,3) with activation function as relu at all times. The convolutional layers extracts the features from the input images which is then appended with a dense layer, a dropout and a dense layer of single filter at the very end to output status of image to be either Spam or Ham i.e. Binary Classification.

4.4. Model Testing and Saving

When a model summary is obtained for the given neural network, we provide the training and testing data that we obtained by splitting the merged data into 70:30 split. Thus, we obtain the statistics of the model while training and testing and obtain the required metrics for judging the model. Then we save the model in .h5 format so that we use this model on the backend side for our chat application.

5. RESULTS

5.1 Model Outcomes

5.1.1 Naive Bayes Model Performance

The Performance Metrics of Naive Bayes Model is shown in table 5.1:

Heads	Precision	Recall	F1- Score	Support
Ham	0.96	1.00	0.98	1450
Spam	1.00	0.7	0.82	222

Overall Accuracy =0.97

Table 5.1: Performance Metrics for Naive Bayes Model

5.1.2 CNN Model Performance

The Performance Metrics of CNN model for image spam classification is shown in table 5.2:

Metrics	Values
Accuracy	0.968750
Precision	1.000000
Recall	0.933333
F1 Score	0.965517

Table 5.2: Performance Metrics

Hence the results summarized are shown in table 5.3

Subject under Study	Performance
Naive Bayes Model	97%
CNN Model	96.87%
Spam Classification in Web	Around 5 seconds for image and less than 1 second for text
Spam Classification in An droid	Around 1 minute 10 seconds for image and less than 1 seecond for text

Table 5.3: Summary of results

6. CONCLUSION

Therefore, we developed the chat application project which can classify spam and ham messages using machine learning models i.e. Multinomial Naive Bayes and Deep learning model i.e. CNN model. From the merged Image spam dataset of Dredze and ISH, we achieved the accuracy of 96.87% via the CNN model. Similarly using UCI repository dataset for text, we achieved 97% accuracy using Naive Bayes Model. After integration of prediction system, we achieved efficient response for classification from the applications as well. Some tradeoffs includes slow image spam classification and limitation of using tesseract in mobile application. Despite the tradeoffs, the application is fully functioning, but requires further optimizations.

REFERENCES

1. N. Choudhary and A. K. Jain, “Towards filtering of sms spam messages using machine learning based technique,” in *Advanced Informatics for Computing Research*, D. Singh, B. Raman, A. K. Luhach, and P. Lingras, Eds. Singa pore: Springer Singapore, 2017, pp. 18–30.
2. A. Karami and L. Zhou, “Improving static sms spam detection by using new content-based features,” 08 2014.
3. H. Shirani-Mehr, “Sms spam detection using machine learning ap proach,” *unpublished) http://cs229. stanford. edu/proj2013/Shir aniMeh r SMSSpamDetectionUsingMachineLearningApproach. pdf*, 2013.
4. Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong, “Sms spam detection using noncontent features,” *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 44– 51, 2012.
5. E.-S. M. El-Alfy and A. A. AlHasan, “Spam filtering framework for mul timodal mobile communication based on dendritic cell algorithm,” *Future Generation Computer Systems*, vol. 64, pp. 98–107, 2016.
6. S. Srinivasan, V. Ravi, V. Sowmya, M. Krichen, D. B. Noureddine, S. Anivilla, and K. Soman, “Deep convolutional neural network based image spam clas sification,” in *2020 6th conference on data science and machine learning ap plications (CDMA)*. IEEE, 2020, pp. 112–117.
7. M. Chowdhury, J. Gao, and M. Chowdhury, “Image spam classification us ing neural network,” in *International Conference on Security and Privacy in Communication Systems*. Springer, 2015, pp. 622–632.
8. A. D. Kumar, S. KP *et al.*, “Deepimagespam: Deep learning based image spam detection,” *arXiv preprint arXiv:1810.03977*, 2018.
9. T. Sharmin, F. Di Troia, K. Potika, and M. Stamp, “Convolutional neural networks for image spam detection,” *Information Security Journal: A Global Perspective*, vol. 29, no. 3, pp. 103–117, 2020.
10. B. Kim, S. Abuadbba, and H. Kim, “Deepcapture: Image spam detection using deep learning and data augmentation,” in *Australasian Conference on Information Security and Privacy*. Springer, 2020, pp. 461–475.
11. A. M. Pednekar, “Spam detection in im images using convolutional neural networks,” *International Journal of Current Research*, vol. 10, no. 07, p. 71786–71791, Jul 2018.
12. A. P. Singh, “Image spam classification using deep learning,” 2018.
13. T. A. Almeida and J. M. G. Hidalgo. [Online]. Available: https: //www.dt.fee.unicamp.br/˜tiago/smsspamcollection
14. Y. Gao, X. Zhao, and M. Yang. [Online]. Available: https://users.cs. northwestern.edu/˜yga751/ML/ISH.htm
15. M. Dredze, R. Gevaryahu, and A. E. Bachrach. [Online]. Available: https://www.cs.jhu.edu/˜mdredze/datasets/image spam/