Research Article

Performance Comparison for Spam Detection in Social Media Using Deep Learning Algorithms

Rushali Deshmukh[1], Vikram Bhalerao[2]

[1]Savitribai Phule Pune Univresity, faculty, Maharhatra/India
[2]Savitribai Phule Pune Univresity, PG research Scholar, Maharhatra/India

Article History: Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27 January 2021; Published online: 05 April 2021

Abstract: Social media applications like Twitter, Instagram, Facebook have helped people to connect to each other. This has been eased due to high-speed internet. However, this has invited various spam messages through tweets or Facebook. The sole purpose of such messages is aggregation or exploitation of personal data in terms of finances or medical records, political benefit's or community violence. This makes spam detection an extreme value-added service. We tend to recommend a 1D CNN algorithmic technique and compare results with variants of CNN and with boosting algorithms. The model is braced with linguistics data in the illustration of the words with the assistance of knowledge-bases such as Word2vec and fast ext. This improves the end to end performance, by providing higher linguistics vector illustration of input testing words. Projected Experimental results show the efficiency of the projected approach from the point of view of accuracy, F1-score and response time.

Keywords: Convolutional Neural Network, fast text, Sentiment Analysis, Word2Vec

1. Introduction

Recently with the dawn of the social media platforms, people are enabled to grow and to communicate efficiently. This opportunity has been threatened through spams, malicious links, and malware [1]. This makes Spam detection extremely required value-added service for any social platform. Initially the spam messages used to get detected by manual process or by simple filter rules for commons properties. Automation in spam detection advance of basic machine-learning algorithms which do not produce spam detection models. Initially, spam started spreading with email spams. Additionally, the SMS is a price effective technique used for converting individual messages to the vector form. Possible purchasers, encompass a higher rate of response as compared to email spam. In conjunction with emails and SMS, social networking platforms like Twitter, Facebook and an instant traveler like WhatsApp, etc., also are tributary to a greater portion of spam on the network. It will be a complex activity of spam detection without any filter at the receiving node. One of the initial classifiers is rule-based, having a lot of formally written principles. These classifiers were used to get deployed to an ample space of purchasers. It comprises a set of pre-defined rules that are applied to associate degree of incoming messages and these messages were labelled as spam if their check score exceeds the threshold value. Even after the spam is detected, the success of these ways is restricted and needs to be combined with different machine learning methods so as to give fairly sensible results. Naive Bayes, Radom forests etc., are few of the standard classifiers. These classifiers are complicated, thanks to feature extracting options from the text, which helps to identify the pattern of spam and ham messages [2]. Most frequently used feature extracting models are bag of words models with token frequency as a common factor. Convolutional Neural network is the deep learning algorithm that addresses the accurate classification of the text messages as spam or ham.

2. Literature Survey

Gauri Jain et al. [1] had proposed an architecture focused on short spam content on SMP like Twitter. This is in contrast with earlier long spams emails detections and deletion. Using basic configurations. of CNN algorithms, the outcomes showed that the proposed model proved to be efficient with the use of Twitter and SMS text datasets.

Thayakorn Dangkesee et al. [3] has proposed a model that was used for spam detection by the victimization of spam word lists using a billboard URL-based security tool. Naive Bayes algorithm has been used to analyses the data using data types such as all data and specific data. It has boosted the performance of the spam detector than usual. One can show their methods fulfills the experimental result.

Rutuja Katpatal [4] has formed an additional input training dataset to classify unlabeled tweets using another dataset. Author has proposed a scheme that adjusts training data sets. Dropping too old samples after a specific time has helped to eliminate unusual information saving space.

Ms. Sayali Kamble et al. [5] has exhibited the plan of ongoing vector space denotation of words. Evaluation of a novel AI-based way to deal with specialization Social spam detection. Their general research goal for consequently shifting and recognizing spammers who point social destinations was to discover methodology ie.SAND, to find compelling devices.

Guanjun Lin et al. [6] has detected the nine mainstream algorithms that were compared to understand the most suitable algorithm. The stability of each algorithm had been studied thoroughly. It had indicated the variation of the training time according to CPU core.

Tingmin Wu, et al. [7] has proposed a system of twitter spam detection. While the paper had addressed the then-existing challenges like low speed and feature extraction difficulties thoroughly; the paper written had experimentally proven the comparisons between achieved results and existing accuracies through the indication of graphs

.

3. Proposed Methodology

The System consists of a model that is trained on Convolutional Neural Network rule. On general terms, CNN is most well-liked for image classification. The variation of CNN 1D has been used for spam text information classification. CNN contains various types of layers that are typically improved in terms of accelerating accuracy during and after the implementation. Here, in the projected system, CNN will work as a classifier to investigate whether or not, the text statement is spam. The model permits to form associate unattended learning as well as supervised mode of learning rule for getting vector representations for input texts. Models like fast texts makes us of neural network for word embeddings The proposed framework comprises of Word2Vec and fast text advancements along with Convolutional Neural Network (CNN) to complete the model. The proposed framework will also be comprised of varieties of CNN models in terms of the number of filters and convolutional layers of the CNN algorithm. These can be used for performance comparisons between different variations of CNN. The research will work around CNN varieties and layers.

3.1. Architecture

1. The Proposed system has the advantage of multiple platforms for computer files for model development.
2. The System has been developed with quite one algorithmic program, so Prediction guarantees are inflated.
3. Live updates area unit involved in prediction so its area unit typically used for live recommendation.
4. The proposed system varies in the filter and convolutional layer combination.
5. The proposed system will compare the results of the CNN algorithms with that of the boosting algorithms and hence will try to achieve the maximum accuracy with better precision.
6. We are also trying to train the module through the vernacular language data sets like Hindi or Marathi.
7. Fig.3.1 shows an overview of the proposed system architecture.
8. Out of the complete training data set, 80% of the data will be used for the training the model under supervised learning technique, and the remaining 20% of the data set will be used as a test set to generate the accuracy and response time calculation tests.

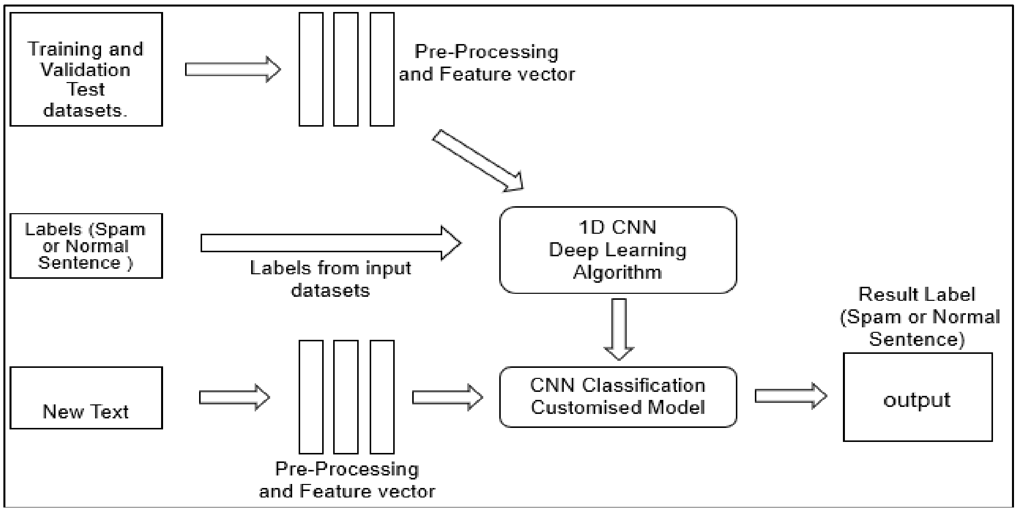

Training and Validation Test datasets.
Pre-Processing and Feature vector
Labels (Spam or Normal Sentence)
Labels from input datasets
1D CNN Deep Learning Algorithm
Result Label (Spam or Normal Sentence)
New Text
CNN Classification Customised Model
output
Pre-Processing and Feature vector

Figure 1. Proposed System Architecture

3.2. Algorithms

Fast text and Word2Vec algorithms are used for implementation.

3.2.1 Scope of fast text:

The scope of fast text is to work as a runtime library for classification and vector conversion of input texts. It is used for processing the number of tasks. It is based on the C++ platform. Fast text allows the end-user or the developer to coach supervised and unsupervised classification of input texts. Generally, it trains the models of a Skip-gram model or continuous bag of words (CBOW). It makes use of function such as Soft Max or negative sampling loss functions. Huge number of linguistic embeddings can be trained with the use of fast text.

3.2.2 Scope of Word2Vec:

Word2Vec may be a shallow, two-layered neural network which is trained to reconstruct linguistic contexts of words. Every unique word gets assigned a vector notation in vector space. This conversion is taken place across the given classes. Words which are share the common class context are placed in close proximity in the vector space. Word2vec thus has proved to be efficient models in the word embeddings techniques [8]. Like fast text model, the word2vec models are majorly applicable to CBOW or Skip gram models [9].

3.3 1D CNN algorithm

The process of feature learning has been used in 1D CNN algorithm. The algorithm maps the extracted features of the input text sequence.1D algorithm is especially useful in datasets which have segments of fixed length.

Size of Data Set: ~12k records (80% training,20% validation)
Attributes: Tweet Type, Tweet

1D CNN Algorithm

The Algorithm of a 1D-CNN is formed through the following important steps:

```
Input: With filter size F, Input matrix x(l*d),
Process:
Assign Weightage W and process filter F
For {    …
           # first CNN variation ‘V1’ For {                              …# each epoch N
CNN (hidden) and       MLP neurons
CNN layer each with kernel size =3
Subsampling factor (wj) = [xj + xj +1 + …+ xj + k-1]
activation functions. (ReLU (wjn +b))
probability Output
Errors possibility
BP- Backward propagation.
}
}      #end for CNN variation ‘V1’
Output: The trained model of CNN classifier is produced.
```

In each CNN-layer, equation (1) represents the 1D forward propagation (1D-FP) [9]:

$$x_k^l = b_k^l + \sum_{i=1}^{N_{l-1}} conv1D\left(w_{ik}^{l-1}, s_i^{l-1}\right) \quad \text{…. (1)}$$

where, x_k^l is bias k^{th} neuron bias at layer l, s_i^{l-1} is i^{th} neuron output of layer l-1 , w_i^{l-1} is the ik^{th} kernel of layer l–1 .$conv1D$ (.,.) is used to perform ‘in-valid’ 1D convolution without zero-padding.

4. Results and Discussions

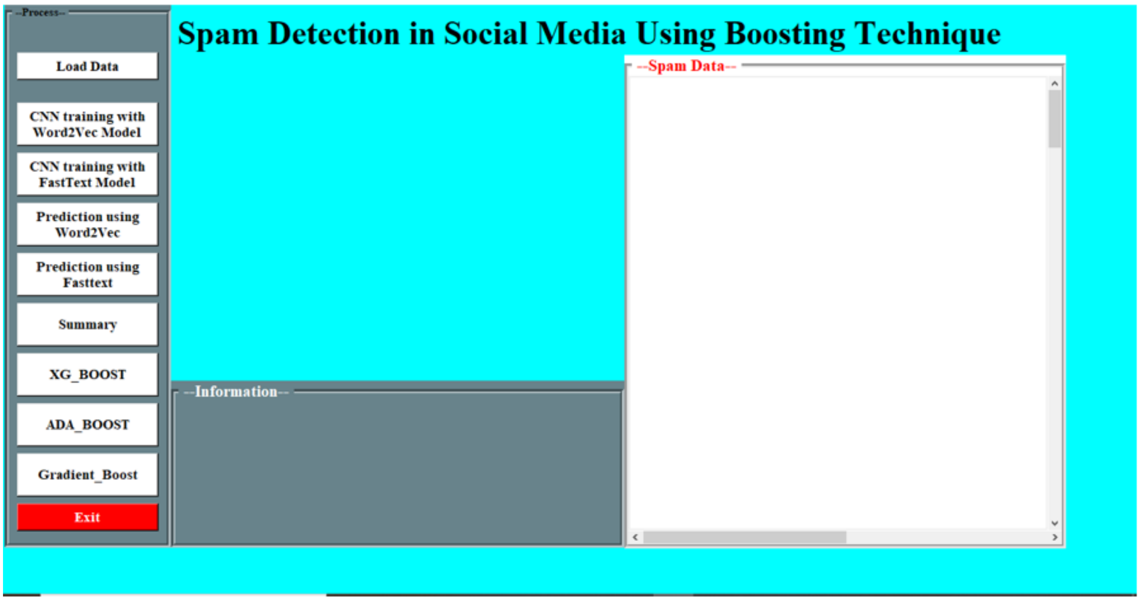

--Process--
Load Data
CNN training with Word2Vec Model
CNN training with FastText Model
Prediction using Word2Vec
Prediction using Fasttext
Summary
XG_BOOST
ADA_BOOST
Gradient_Boost
Exit
Spam Detection in Social Media Using Boosting Technique
--Spam Data--
--Information--

Figure 2. Home Page of proposed model

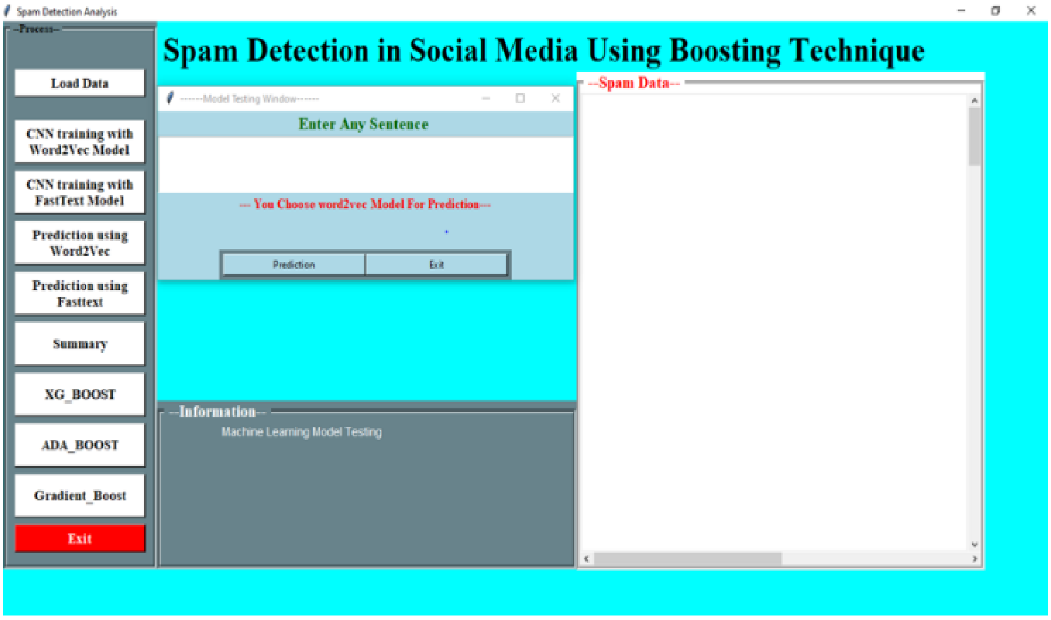
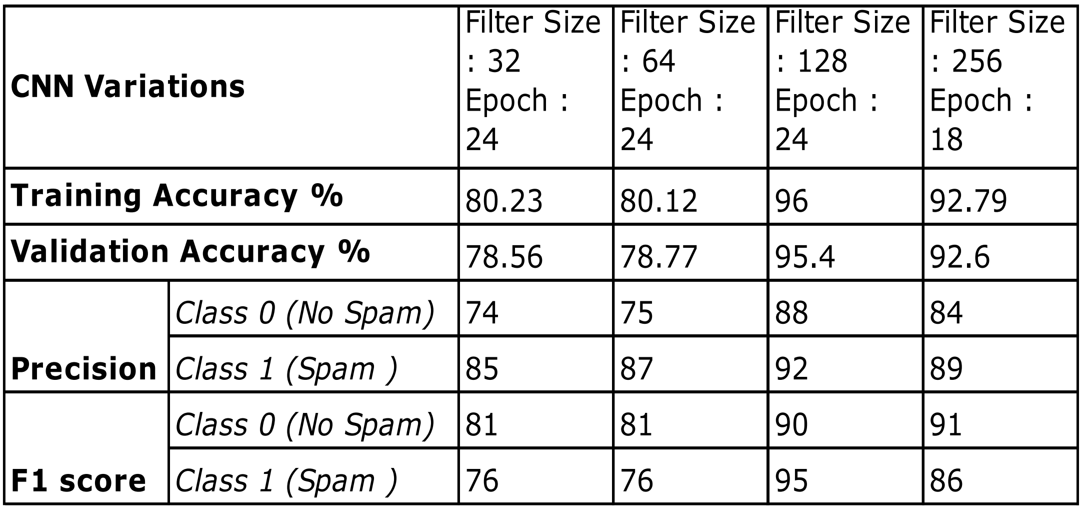

Spam Detection Analysis
--Process--
Load Data
CNN training with Word2Vec Model
CNN training with FastText Model
Prediction using Word2Vec
Prediction using Fasttext
Summary
XG_BOOST
ADA_BOOST
Gradient_Boost
Exit
Spam Detection in Social Media Using Boosting Technique
-------Model Testing Window------
Enter Any Sentence
--- You Choose word2vec Model For Prediction---
Prediction
Exit
--Spam Data--
--Information--
Machine Learning Model Testing

Figure 3. The Prediction from Word2Vec model

1. Figure 2. indicates the overview of the CNN based structure model. This window contains the main modules of the final model, which are summary tab, prediction using word2vec tab, XG boost tab, ADA boost tab, gradient boost tab which in turn at the backend runs the CNN algorithms

2. Figure 3. represents the window where, the input words are converted to vector forms and then are fed to the CNN algorithm for the classification using filters and convolutional layers. An authenticated user needs to give input in the form of text and then the model will give output. In this case, the model will process the input text into vector form and then will feed as the input to CNN algorithm and based on the labeled learning, CNN will classify the sentence into spam or ham

3. Figure 4. shows the results of various CNN variations (filter size and iterations or epoch) in terms of Training accuracy, validation accuracy, precision, and F1 score.



CNN Variations		Filter Size : 32 Epoch : 24	Filter Size : 64 Epoch : 24	Filter Size : 128 Epoch : 24	Filter Size : 256 Epoch : 18
Training Accuracy %		80.23	80.12	96	92.79
Validation Accuracy %		78.56	78.77	95.4	92.6
Precision	*Class 0 (No Spam)*	74	75	88	84
	Class 1 (Spam)	85	87	92	89
F1 score	*Class 0 (No Spam)*	81	81	90	91
	Class 1 (Spam)	76	76	95	86

Figure 4. Results summary of CNN variations

4. Figure 5.4.1 to Figure 5.4.4 represents the actual readings CNN variations with all the iterations or epochs in terms of accuracy of the CNN classification using word2vec. This can be seen when we click on the prediction tab. Here the epoch value will decide the number of iterations. The number of points mentioned in the graph indicates the number of iterations.

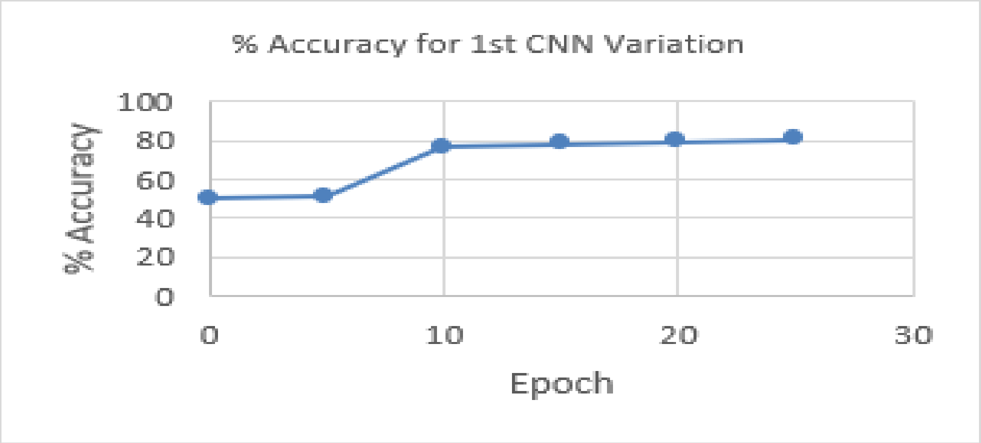


Figure 5.4.1

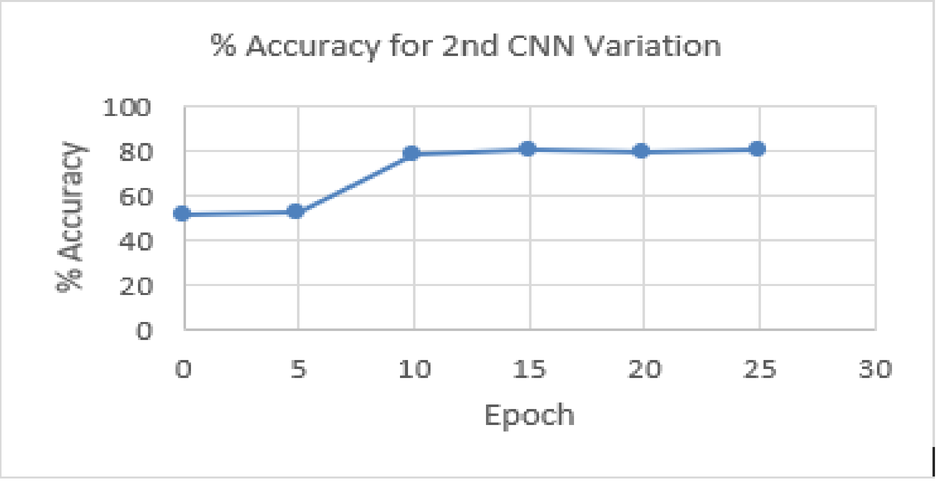


Figure 5.4.2

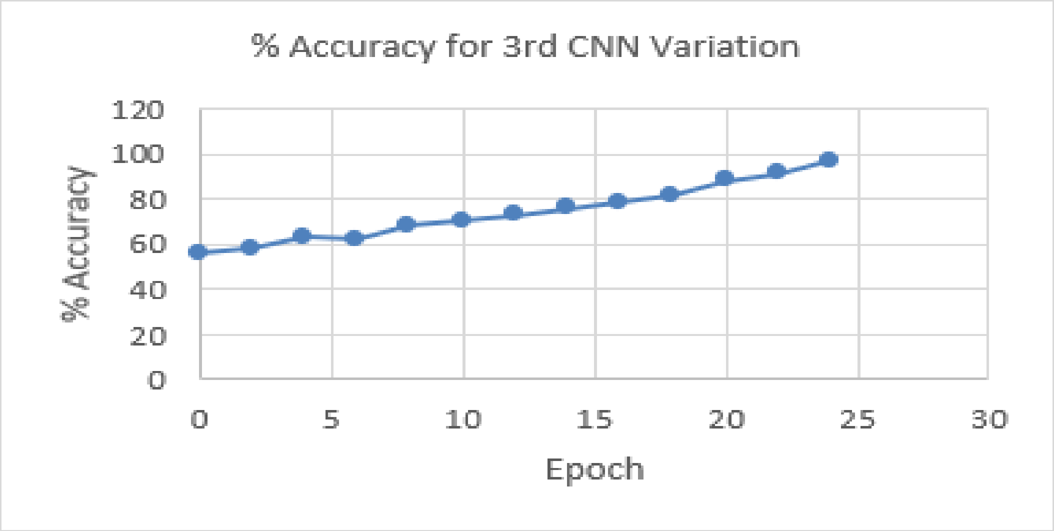


Figure 5.4.3

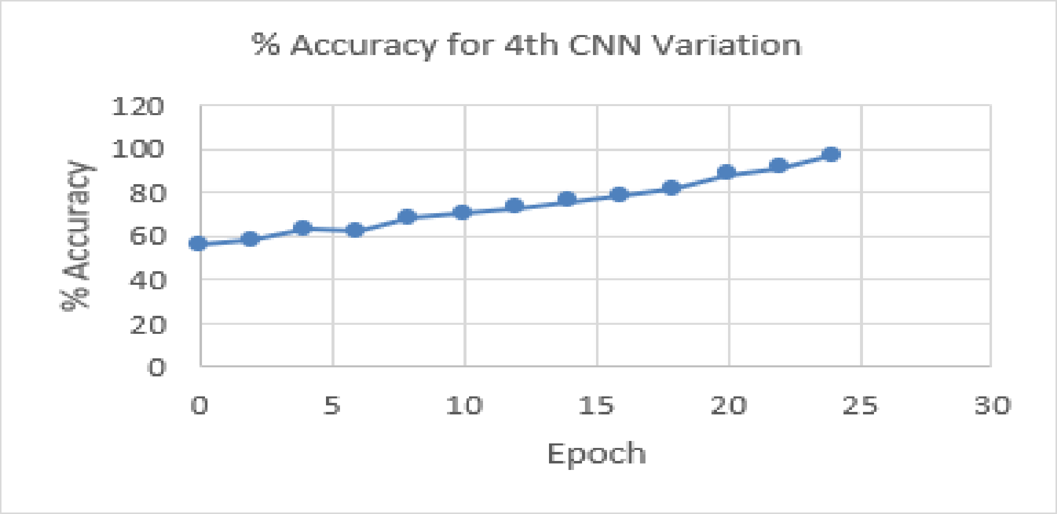


Figure 5.4.4

Figure 5.4.1 to Figure 5.4.4: The actual readings CNN variations with all the iterations or epochs in terms of accuracy of the CNN classification using word2vec

5. The accuracy of the prediction, based on the 80% learning data set and 20% of validation can be shown through the graphical representation of the model. Refer to Figure 5.4.3. This accuracy has been reached to 96% when we have used CNN and LSTM algorithms as endpoints along with word2vec module.

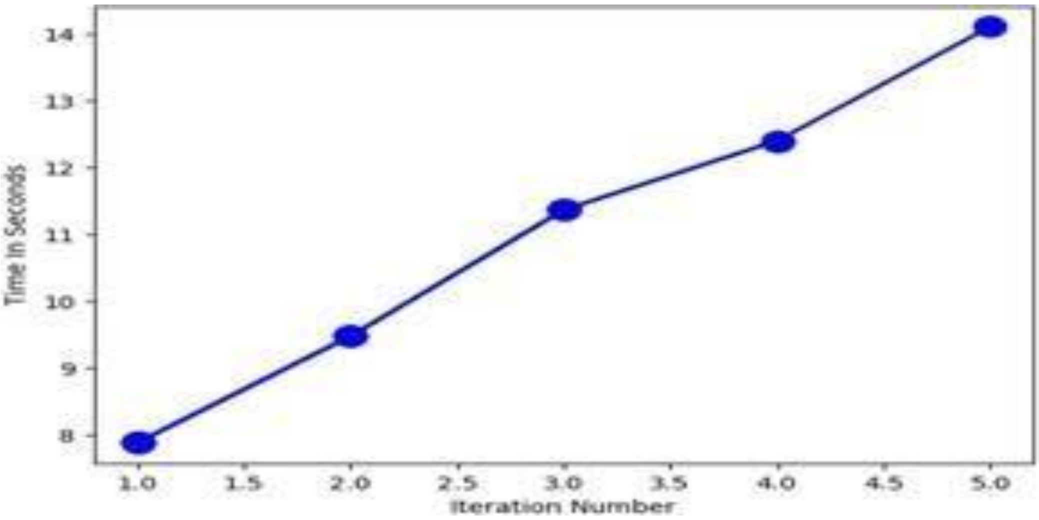
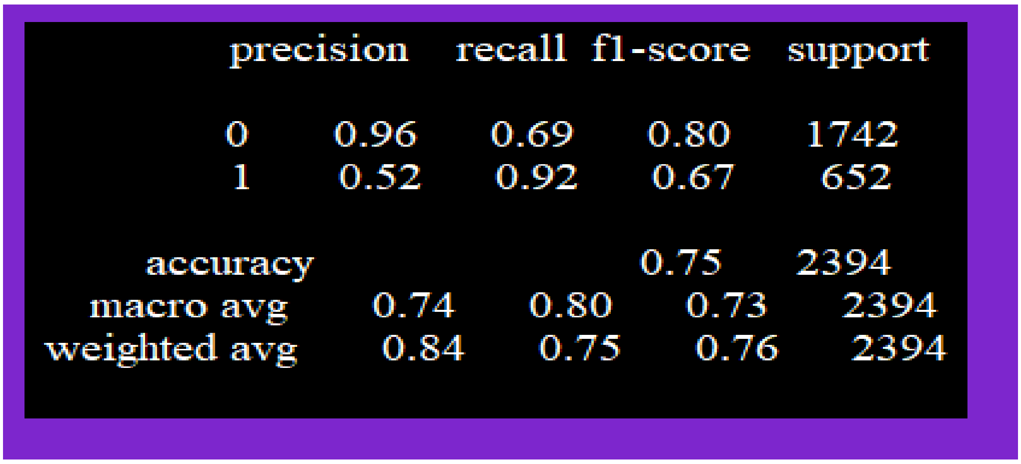


Figure 6. Performance graph of response time from Word2Vec model.

6. We have also tested the twitter dataset for spams detection using machine learning algorithms such as the ADA boost, XG boost and gradient boost and have compared the results of the same in terms of performance and accuracy of the output with CNN -LSTM algorithmic model readings as indicated in the above Figure 5.6.1 to Figure 5.6.3 represents the training reports.

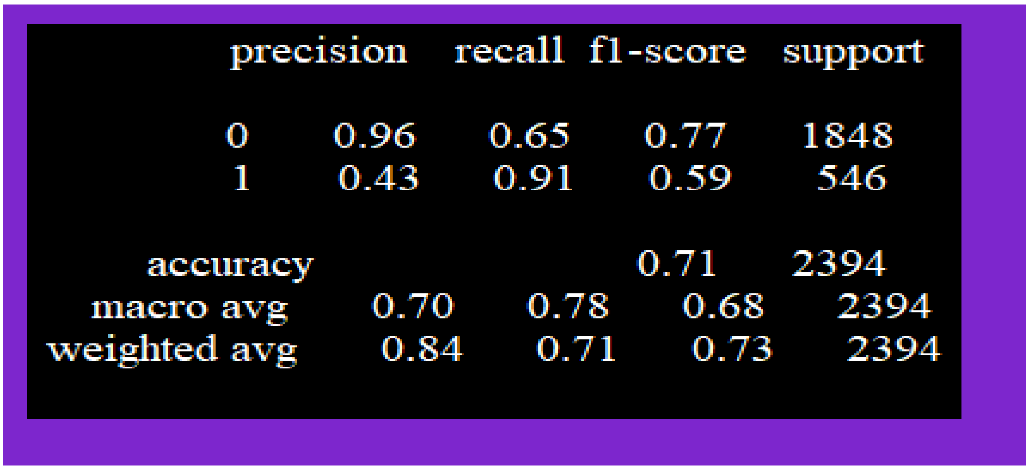


```
              precision    recall  f1-score   support

           0       0.96      0.69      0.80      1742
           1       0.52      0.92      0.67       652

    accuracy                           0.75      2394
   macro avg       0.74      0.80      0.73      2394
weighted avg       0.84      0.75      0.76      2394
```

Figure. 5.6.1. Illustrated the training report using gradient boosting

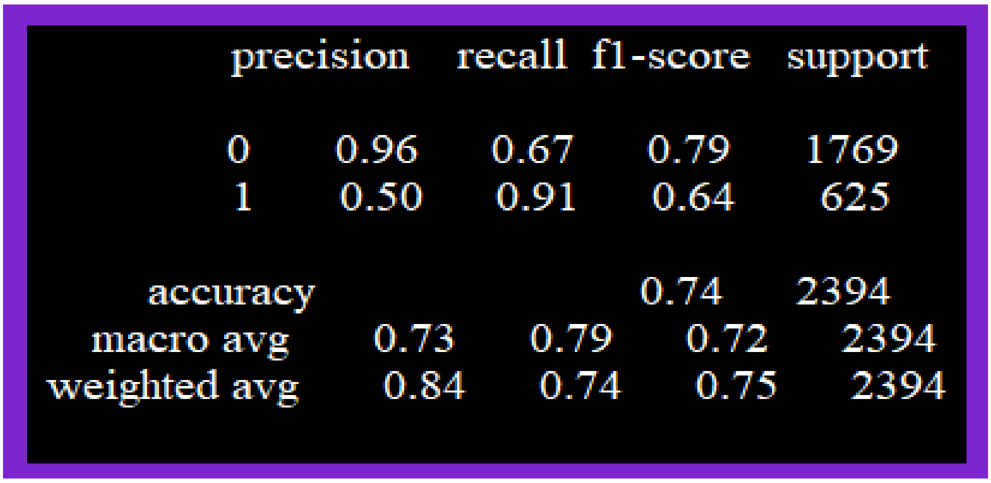


```
              precision    recall  f1-score   support

           0       0.96      0.65      0.77      1848
           1       0.43      0.91      0.59       546

    accuracy                           0.71      2394
   macro avg       0.70      0.78      0.68      2394
weighted avg       0.84      0.71      0.73      2394
```

Figure. 5.6.2. Illustrated the training report using ADA boosting

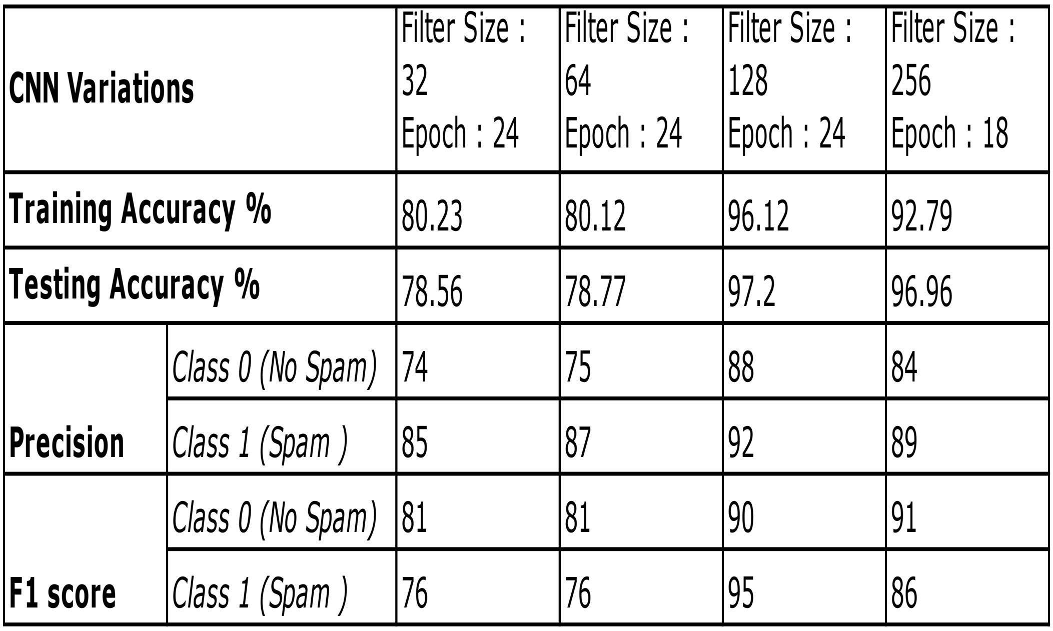


```
              precision    recall  f1-score   support

           0       0.96      0.67      0.79      1769
           1       0.50      0.91      0.64       625

    accuracy                           0.74      2394
   macro avg       0.73      0.79      0.72      2394
weighted avg       0.84      0.74      0.75      2394
```

Figure. 5.6.3. Illustrated the training report using XG boosting



CNN Variations		Filter Size : 32 Epoch : 24	Filter Size : 64 Epoch : 24	Filter Size : 128 Epoch : 24	Filter Size : 256 Epoch : 18
Training Accuracy %		80.23	80.12	96.12	92.79
Testing Accuracy %		78.56	78.77	97.2	96.96
Precision	*Class 0 (No Spam)*	74	75	88	84
	Class 1 (Spam)	85	87	92	89
F1 score	*Class 0 (No Spam)*	81	81	90	91
	Class 1 (Spam)	76	76	95	86

Figure 7. The performance comparison chart in terms of % accuracy, F1 score and precision

7. Figure 7 represents the performance comparison chart in terms of % accuracy, F1 score and precision.

5. Conclusion

After research implementation of the proposed project and idea, the system reflects the conclusion that the CNN-LSTM variant model (III) along with Word2Vec performs efficiently in terms of % accuracy, F1 score, and precision [Figure 7]. The % accuracy has been up to 96% on average. As indicated above, we have conducted several rounds of training and validation on the CNN algorithm and on machine learning algorithms such as the ADA boost and the XG boost. In comparison, it has been concluded that the Word2Vec model takes around 8 to 12 seconds to load and respond, as shown in the graphical representation in the results and discussion section. However, with CNN-LSTM fast text model takes much more time to load and respond. Fast text takes around 1 to 2 minutes. Both are relying on the earlier stage of model training. 1D Convolutional Neural Network algorithm is used for the training model. The number of iterations of the execution has been based on the epoch.

As a part of future work, of the proposed model, multilingual twitter dataset like Hindi, Marathi can be used to arrive at the efficient performance delivery. After deriving accuracy, later on, these individual results will be performance compared with the results of the ADA/XG boosting algorithm. Changed spams can be added in the input datasets to reduce the spam drifts. [7].

References

1. Jain, G., Sharma, M. and Agarwal, B. (2009). Spam detection in social media using convolutional and long short-term memory neural network. Annals of Mathematics and Artificial Intelligence, 85(1), pp.21-44.
2. Lee, K., Caverlee, J. and Webb, S. (2010). Uncovering social spammers: social honeypots+ machine learning. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (pp. 435-442). ACM.
3. Dangkesee, T. and Puntheeranurak, S. (2017). Adaptive Classification for Spam Detection on Twitter with Specific Data. In 2017 21st International Computer Science and Engineering Conference (ICSEC) (pp. 1- 4). IEEE.
4. Katpatal, R. and Junnarkar, A., (2018). An Efficient Approach of Spam Detection in Twitter. In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1240-1243). IEEE.
5. Kamble, S. and Sangve, S.M. (2018). Real Time Detection of Drifted Twitter Spam Based on Statistical Features. In 2018 International Conference on Information, Communication, Engineering and Technology (ICICET) (pp. 1-3). IEEE.
6. Lin, G., Sun, N., Nepal, S., Zhang, J., Xiang, Y. and Hassan, H.(2017). Statistical twitter spam detection demystified: Performance, stability and scalability. IEEE access, 5, pp.11142-11154.
7. Wu, T., Liu, S., Zhang, J. and Xiang, Y. (2017). Twitter spam detection based on deep learning. In Proceedings of the Australasian computer science week multiconference (p. 3). ACM.
8. Chen, C., Wang, Y., Zhang, J., Xiang, Y., Zhou, W. and Min, G. (2016). Statistical features-based real-time detection of drifted Twitter spam. IEEE Transactions on Information Forensics and Security, 12(4), pp.914-925.
9. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. and Inman, D.J., 2019. 1D Convolutional Neural Networks and Applications: A Survey. arXiv preprint arXiv:1905.03554.