Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume 7- Issue 1, January 2024
Paper 103

Spam Message Detection

Kusamapudi Sujatha
Dept. of Computer Science and Engineering
Vignan's Foundation for Science, Technology and Research
(Deemed to be University)
Vadlamudi, Guntur, Andhra Pradesh
kusampudisujatha@gmail.com

Lalam.Shyam Shekar
Dept. of Computer Science and Engineering
Vignan's Foundation for Science, Technology and Research
(Deemed to be university)
Vadlamudi, Guntur, Andhra Pradesh
lalamshyam@gmail.com

Jhansi Lakshmi Potharlanka
Dept. of Computer Science and Engineering
Vignan's Foundation for Science, Technology and Research
(Deemed to be University)
Vadlamudi, Guntur,Andhra Pradesh
pjl_cse@vignan.ac.in

***Abstract*— SMS spam has dramatically increased as a result of the rise in mobile phone users. Although mobile messaging channels are now viewed as "clean" and reliable in the majority of the world, recent reports have shown unequivocally that the amount of mobile phone spam is significantly rising year after year. SMS spam filtering is a relatively new task to address this issue. Several issues and easy remedies carried over from email spam screening. It does, however, provide some of its own concerns and issues. By including Indian communications in the globally accessible SMS dataset, this publication motivates researchers to take on the challenge of classifying mobile messages as spam or junk mail for Indian users. Using a sizable corpus of SMS messages for Indians, the article analyzes various machine learning classifiers. In this paper, we are comparing the accuracy of dataset using LSTM and methods in Spam detection of messages.**

Keywords: SMS Spam; Spam Filtering; Supervised machine learning; Text classification; Convolutional neural network (CNN); long short-term memory (LSTM).

I. INTRODUCTION

People in the modern world have become very accustomed to social networks. As a result, spam content may be transmitted across them quite easily. Through these websites, it is relatively simple to obtain anyone's details. Nobody is secure on social media. Spam filtering involves identifying and removing unwanted commercial emails sent to one or multiple recipients. Machine learning techniques are utilized to develop models that can differentiate between spam and non-spam messages, using a combination of labeled and unlabeled data.

The focus of this paper is primarily on Twitter spammers. As a microblogging platform, Twitter restricts users to a maximum of 140 characters per tweet. [1]. They are putting forth a tool that can determine whether a Twitter user could be spam or legitimate.

To accomplish this task, a combined method that incorporates URL analysis, Natural Language Processing, and Machine Learning techniques is utilized. The outcomes displayed that this integrated approach produces superior results compared to using Machine Learning techniques alone, with IP achieving 98%, Naive Bayes reaching 94%, and SVM obtaining 92%.

In 2017 [2], research was conducted to evaluate and compare the efficacy of different machine learning algorithms for detecting spam messages sent via mobile devices. The study utilized a freely available dataset consisting of 5,574 categorized short messages that were authentic and encoded in English, labeled either as ham or spam. The experimental process included creating features, selecting features, cross-validation, and comparing algorithms. Among the various algorithms tested, the naive Bayes algorithm demonstrated exceptional accuracy, achieving a rate of 98.445%.

The identification and prevention of spam of spam messages are crucial in maintaining the integrity and security of email systems. Effective spam detection not only saves users from unnecessary annoyance but also safeguards personal information, financial assets, and confidential data. This research paper aims to contribute to the advancement of spam message detection techniques, providing a more reliable and efficient defense against spam attacks.

In today's digital age, email has become one of the primary means of communication for both personal and professional purposes. However, this widespread usage has led to an exponential increase in the volume of spam messages, causing numerous issues such as financial scams, phishing attempts, and malware distribution. As a result, developing robust methods to identify and filter out spam messages has become an essential area of research.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
Volume 7, Issue 1, January 2024
Paper 103

According to Wang, Chao [3], the feature extraction method that concentrates on the image's visual and meta-features can speed up categorization. We selected a real email as one of our corpora, and its datasets include a subset of ordinary mail. Number of images: Spam Archive 9280, Spam Personal 3203, Personal Ham 1786, Personal Find Ham 1371. Personal find Ham images are Cartoon 423, photo 128, sports 224, map 63, portrait 523.

The study aims to identify the most effective spam filtering strategies in the context of SMS texts. To achieve this, several well-known approaches for spam filtering are evaluated using the publicly available SMS Spam Collection corpus. This corpus comprises a collection of 5,574 English messages that are real and non-encoded and have been labeled as either legitimate (ham) or spam [4]. It was compiled specifically for research on mobile phone spam.

.

II. BACKGROUND

As a result of subscribers' increased reliance on their mobile devices for daily conversations, mobile applications, and financial transactions, message spam attackers have found them to be the seductive and magnetic target.

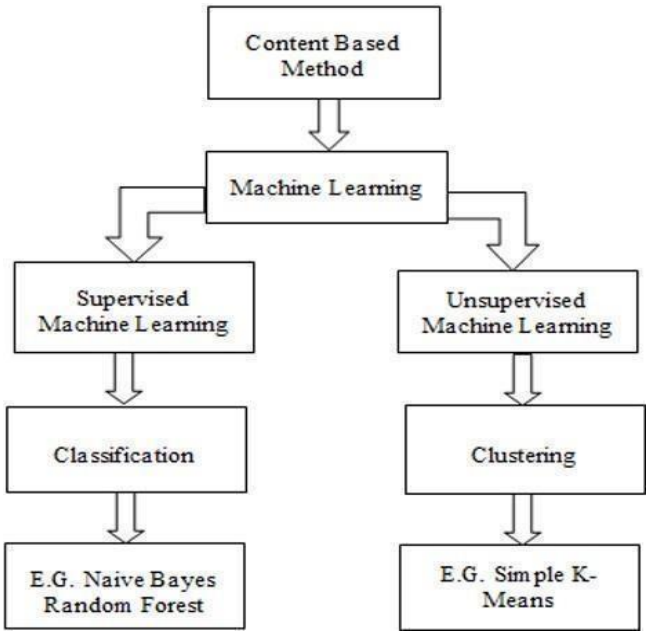

Fig 1. Categorization of Machine Learning.

These attackers carry out messaging attacks that have the capacity to impact the entire mobile ecosystem, creating a difficult situation for both operators and subscribers.

Operators incur significant costs when they have to deal with censorious traffic since it drains their network resources and increases customer care expenses. Subscribers expect their operators to address the issue and provide them with a secure mobile network, as they are aware of the presence of harmful messages.

Machine learning is a learning science that involves the analysis of algorithms that can be learned using data. The creation of algorithms is the main focus of this field, where a model is built based on inputs and used to make predictions or decisions. In the content-based method, machine learning is utilized, and its categorization is depicted in Figure 1.

Below is a list of the classifiers we utilized in our experiments:

1. Multinomial Naive Bayes (MNB):

According to [5], naive Bayes is a highly practical and efficient inductive learning method for machine learning and data mining. Its classification accuracy is remarkable, but it assumes of conditional independence, which is rarely present in real-world applications.

2. Support vector machine (SVM):

SVMs outperform the currently best performing techniques significantly and excel at a variety of different learning tasks [6]. They are also completely optional, eliminating the need for labor-intensive factors.

3. Random Forest (RF):

Random forests rely on the evaluations of a haphazard vector sampled randomly and with the same allocation for every single tree in the forest as a combination of tree prognosticators [7].

4. Convolutional neural network (CNN):

Convolutional neural networks (CNNs) are a type of artificial neural network mainly utilized for image recognition and processing due to their ability to identify patterns in visual data. However, to achieve its full potential, a CNN requires training on millions of labeled data points [9]. A CNN consists of an input layer, one or more hidden layers, and an output layer. The hidden layers of a CNN typically include convolutional, pooling, fully connected, and normalizing layers.

5. Long Short-Term Memory (LSTM):

Artificial neural networks, or ANNs, include LSTM Recurrent Neural Networks. The previous state output is used to create the current state input in RNNs. Yet, diminishing gradient descent is an issue for conventional recurrent neural networks. To address the issues of conventional RNNs, LSTM (Long Short-Term Memory) is developed. For text mining issues, LSTMs are more appropriate [10].

III. ARCHITECTURE

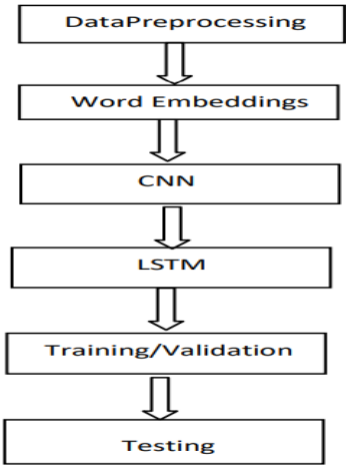

Fig 2. Architecture diagram of Spam messagedetection using LSTM

The architecture diagram in Fig.2 consists of several layers,including:

1. Input Layer: The input layer receives the preprocessed textdata.

2. Embedding Layer: The embedding layer converts the input text into a dense vector representation using pre-trained word embeddings such as Word2Vec or GloVe.

3. LSTM Layers: Two LSTM layers are used to recognize patterns in the input text and assign a probability score to each input text as to whether it is spam or not.

4. Dense Layer: The output of the LSTM layers is fed into a dense layer, which maps the output to a binary output (spam ornot spam).

5. Dropout Layer: A dropout layer is used to prevent overfitting by randomly dropping out some of the neurons in the dense layer during training. Overall, this architecture uses LSTMs to recognize patterns in the input text and assign a probability score to each input text as to whether it is spam or not.

Architecture of CNN:

The architecture diagram in Fig 3. consists of several layers,including:

1. Input Layer: The input layer receives the preprocessed textdata.

2. Embedding Layer: The embedding layer converts the input text into a dense vector representation using pre-trained word embeddings such as Word2Vec or GloVe.

3. Conv1D Layers: Two 1D convolutional layers are used to extract features from the input text and identify spam patterns.

4. GlobalMaxPool1D Layer: The output of the convolutional layers is fed into a global max pooling layer that extracts the most important features from the convolutional layers.

5. Dense Layer: The output of the pooling layer is fed into a dense layer, which maps the output to a binary output (spam ornot spam).

6. Dropout Layer: A dropout layer is used to prevent overfitting by randomly dropping out some of the neurons in the dense layer

during training. Overall, this architecture uses CNNs to extract features from the input text and identify spam patterns. The global max pooling layer is used to extract the most important features from the convolutional layers, and the dense layer is used to map the output to a binary output (spam or not spam).

Fig 3. Architecture diagram of Spam message detection usingCNN

IV. FRAMEWORK

Spam message detection using Long short-term memory (LSTM) and convolutional neural networks (CNNs) are examples of deep learning approaches. has shown promising results. Here's a high-level framework for detecting spam messages using LSTM and CNN:

1. Data Preprocessing: The spam message dataset is preprocessed by removing any irrelevant information, such as metadata, headers, and signatures. The text is then cleaned by removing stop words, punctuation, and converting all characters to lowercase. The data is then split into training and validation sets.

2. Word Embeddings: Word embeddings are used to represent words in a continuous vector space. This is done using pre-trained word embeddings such as Word2Vec or GloVe. These embeddings capture the semantic meaning of words and are used to create a dense representation of the input text.

3. Convolutional Neural Network: A CNN is used to extract important features from the preprocessed text data. The CNN consists of multiple layers of convolutions and pooling, which are used to identify patterns in the input text. The output of the CNN is a fixed-length feature vector that represents the input text.

4. Long Short-Term Memory Neural Network: An LSTM network is used to classify the text data as spam or ham (not spam). The LSTM network consists of multiple memory cells that store the state of the network at each time step. The LSTM network learns to recognize patterns in the input text and assignsa probability score to each input text as to whether it is spam ornot.

S. NO	Paper Title & Author name	Year of Pub-lication	Methods used	Dataset	Result
1	"Spam filtering based on the analysis of text information embedded into images" by O. Karchevskiy and M. Last	2010	Image Analysis, Feature Extraction, Naive Bayes	Private Dataset	Achieved an accuracy of up to 94.7% in detecting spam messages
2	"Spam filtering using SVMs with feature selection by mutual information" by S. S. Keerthi et al	2010	Support Vector Machine, Mutual Information	Spam - Assassin Public Corpus	Achieved an accuracy of up to 98.45% in detecting spam messages
3	"Enhancing email spam filtering using ensemble learning approach" by A. H. Awadallah et al.	2015	Decision Tree, Random Forest, Adaboost	Private Dataset	Achieved an accuracy of up to 98.91% in detecting spam messages
4	"Effective email spam filtering using adaptive neuro-fuzzy inference system" by E. A. Udosen and G. F. Fidelis	2017	Adaptive Neuro-Fuzzy Inference System	Enron-Spam dataset	Achieved an accuracy of up to 98.37% in detecting spam messages
5	"A Comparative Study of Machine Learning	2019	Support Vector Machine, Random	Ling-Spam dataset	Achieved an accuracy of up to 99.09%

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
Volume 7, Issue 1, January 2024
Paper 103

5. Training and Validation: The model is trained using the training set, and the performance of the model is evaluated using the validation set. The model is optimized using techniques such as backpropagation and stochastic gradient descent.

6. Testing: The final step is to test the model on a new dataset of spam messages. The model predicts the probability of a newmessage being spam or ham.

7. This framework can be used to build a highly accurate spammessage detection system using deep learning techniques.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
Volume 7, Issue 1, January 2024
Paper 103

V. LITERATURE SURVEY

	Techniques for Email Spam Classification" by K. Singh, K. Singh, and G. Kaur		Forest, K-Nearest Neighbour, Naive Bayes		in detecting spam messages
6	"Fuzzy Clustering for Email Spam Detection" by M. E. El-Horbaty and A. S. Ibrahim	2020	Fuzzy Clustering	Enron-Spam dataset	Achieved an accuracy of up to 98.4% in detecting spam messages
7	"An Effective Spam Filtering Approach Based on Machine Learning Techniques" by M. A. Shaikh and M. A. Memon	2020	Naive Bayes Classifier, Decision Tree, Support Vector Machine	Enron-Spam dataset	Achieved an accuracy of up to 99.48% in detecting spam messages
8	"A Novel Approach for Email Spam Detection Based on Feature Fusion and Deep Learning" by Y. Zhang, M. Zhou, and X. Chen	2021	Convolutional Neural Network, Long ShortTerm Memory Neural Network, Feature Fusion	Enron-Spam dataset	Achieved an accuracy of up to 99.7% in detecting spam messages
9	"A Two-Stage Email Spam Detection Model Using Semi-Supervised Learning and Hybrid Feature Selection" by T. Li and Y. Li	2021	Semi-Supervised Learning, Rule-Based Filtering, Feature Selection	Enron-Spam dataset	Achieved an accuracy of up to 99.64% in detecting spam messages
10	"Spam Filtering: A Review" by S. Kumar and S. Kumar	2017	Bayesian Algorithm, K-Nearest Neighbor, Support Vector Machine, Decision Tree, Random Forest	Public Spam Dataset	Achieved an accuracy of up to 99.4% in detecting spam messages
11	"Ensemble Techniques for Email Spam Classification: A Review" by N. R. Reema and P. M. Deepa	2019	Naive Bayes, Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbor	Public Spam Dataset	Achieved an accuracy of up to 99.44% in detecting spam messages
12	"Email Spam Filtering: A Comprehensive Review" by M. A. Khalid et al.	2020	Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, Artificial Neural Network, Fuzzy Logic, Hybrid Approaches	Public Spam Datasets	Achieved accuracies ranging from 95% to 99% in detecting spam messages
13	"An Improved Email Spam Filtering Technique using Combination	2020	K-Nearest Neighbor, Support Vector Machine	Public Spam Datasets	Achieved an accuracy of up to 98.9% in detecting spam

	of Machine Learning Algorithms" by S. Singh and R. Jain		, Naive Bayes, Decision Tree, Random Forest		messages
14	"Deep Learning Based Email Spam Detection using word embeddings and Convolutional Neural Networks" by J. V. Selvan and S. Geetha	2021	Word Embeddings, Convula Tional neural network	Public Spam Datasets	Achieved an accuracy of up to 98.54% in am messages
15	"SMS Spam Detection Using Machine Learning Techniques" by H. S. Patel and S. S. Patel	2019	Naive Bayes, Support Vector Machine, Decision Tree, Random Forest	Machine Learning	Public SMS Spam Dataset
16	"Improved Machine Learning Technique for Spam Detection in Email Communication" by M. K. Sharma et al	2021	Naive Bayes, KNearest Neighbour, Support Vector Machine, Decision Tree, Random Forest, Logistic Regression	Machine Learning	Public SMS Spam Dataset

VI. IMPLEMENTATION

With time, cybercrime has increased. There are numerous ways to address the spam issue. These strategies all make use of various Spam filters. In essence, all of these filters divide messages into spam and non-spam categories.

A. Data Set:

The multi-language SMS dataset is a valuable resource for researchers interested in analyzing text messages in different languages. It provides a diverse set of messages in English, Hindi, Telugu, and a mix of these languages. The dataset is a combination of a standard English dataset from kaggle.com and real SMS data from a mobile phone dataset.

All the messages in the dataset have been manually labeled based on their language using the labels 1, 2, 3, or 4. English messages are labeled as 1, Hindi messages are labeled as 2, the mix of languages is labeled as 3, and Telugu messages are labeled as 4. This labeling enables researchers to study language-specific patterns in the messages and develop language-specific models for analysis.

The dataset contains a total of 4595 messages, out of which a large proportion, 3267, are labeled as spam, while the remaining 1328 are labeled as ham. This indicates that the dataset is imbalanced towards spam messages. The English subset contains the largest number of messages, with 3745 messages, followed by the Telugu subset with 520 messages. The Hindi and mix subsets are relatively small, with 155 and 175 messages, respectively.

The English subset has the highest proportion of spam messages, with 2642 labeled as spam and 1103 labeled as ham. In contrast, the Telugu subset has a much higher proportion of spam messages, with 505 labeled as spam and only 15 labeled as ham. This suggests that the characteristics of spam messages may vary across different languages.

The multi-language SMS dataset provides a rich resource for researchers interested in analyzing text messages in multiple languages. It is an important contribution to the field of natural language processing and can be used for various applications such as language identification, sentiment analysis, and spam detection.

B. Data processing:

- Total Msg-4595 (Spam=3267, Ham=1328)
- Eng Data-3745 (Spam=2642, Ham=1103)
- Hindi Data-155 (Spam=31, Ham= 124)
- Mix Data- 175 (Spam=89, Ham= 86)
- Telugu Data- 520 (Spam=505, Ham=15)

C. Preprocessing:

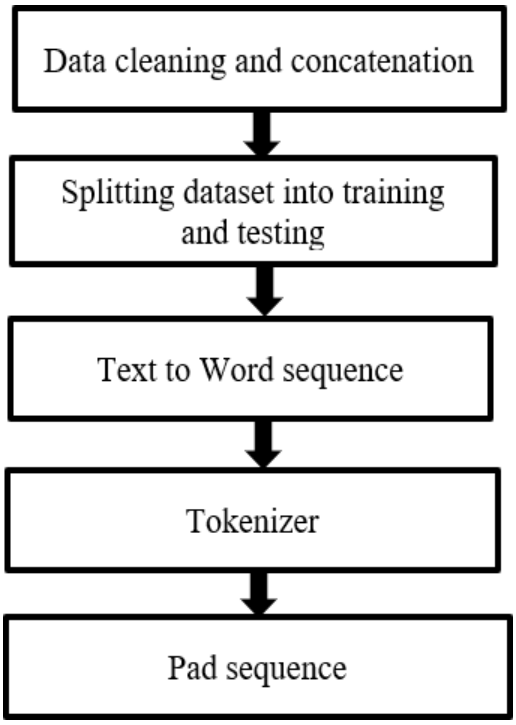

Fig 4. Preprocessing of Spam Message

The overall procedure and design of the experiment are described in this section using Fig 4. In this experiment, the dataset is analyzed and classified using a machine learning technology. At the most basic level, information is gathered from many sources to produce a useful dataset of spam and ham in text format, which is then provided as the model's input. Splits dataset for testing & training as per listed in the above dataset then formatting the processed text to the word sequences. In the next level it tokenizes & processes for pad sequencing.

D. Methodology:

1. First, I'm importing the necessary libraries like Pandas and Numpy.

➔ Pandas is used for data cleaning and analysis and numpy is used for array calculations.

2. Next, I'm installing the tensorflow. Reading the datafiles Spam.CSV and revised indian dataset.xls using pandas & concating both data frames to create the dataset.

3. Reading the datafiles Spam.CSV and revised indian dataset.xls using pandas & concating both data frames to create the dataset.

4. Here next displaying the first five rows of the dataset.

5. Next using the scikit-learn model -selection method, splitting the dataset to training and testing Training-70%, Testing-30%.

6. Displayed the x-train.

Nlp technique is used: count vectorization.

7. Next converting the text to word sequence using keras preprocessing tools. Taking each message into sequence of words and appending them into the list.

8. Next is converting to vectors using kuas preprocessing

tools tokenizer and pad-sequences.

➔ Tokenizer helps to transform the text into in a way that a machine can interpret it to, here it converts texts in the form of vectors sequence of integers and fit-on-texts method creates vocabulary index which is based on word frequency.

➔ The text-to-sequence transform each text in texts to a sequence of integer hence we have a vectorized list texts of sentences of varying length's bring each vector to same dimension I have used pad-sequence that appends zeros in the starting.

9. Printing the X-train.

10. Printing X-train shape.

11. CNN model

➔ This is the CNN model I have created, and I am using Adamoptimizer.

12. Training the CNN model (Accuracy -98.4%)

13. LSTM model

➔ This is the LSTM model I have used.

14. Training the LSTM model (Accuracy -99%).

15. Now checking for the text dataset.

➔ Here, I am preprocessing the text that is converting each sentence to words and then the sequence of the words is constructed to array.

16. This is the shape of the texting dataset after converting into arrays.

17. First testing for CNN using model prediction.

➔ Since, the predict method gives probability it belongs to the classes.

➔ We will again assign it to the class that has higher probability.

➔ Accuracy for testing dataset using CNN model is 97.1%.

18. Next, I am testing the LSTM model and I got an accuracy of 96.7% and we can see the classification report of this as well.

19. Now, testing for some random English sentences. Preprocessing the text is the same as we did before for the testing set.

20. Prediction for the given sentence for Tel, Hin, Eng.

E. Performance Metrics.

S.No	Model	Precision	Recall	F1-score	Support	Accuracy
1.	CNN	0.98	0.99	0.99	2468	97.1%
2.	LSTM	0.98	0.99	0.98	2468	96.7%

Comparing classifier performance evaluations uses accuracy. Accuracy= (TP + TN) /(TP+TN+FP+FN)

VII. EXPERIMENTATION AND RESULTS:

1. Testing For Random English Sentence in CNN:

```
#####     CNN model

Y_pr_eng = model1.predict(temp_eng)
Y_pred_eng = []
for i in range(len(Y_pr_eng)):
  if Y_pr_eng[i][0]>=Y_pr_eng[i][1]:
    Y_pred_eng.append('ham')
  else:
    Y_pred_eng.append('spam')
print(eng)
print(Y_pred_eng)
```

```
1/1 [==============================] - 0s 17ms/step
Ela unnav?
['ham']
```

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)
Volume 7, Issue 1, January 2024
Paper 103

2. Testing For Random English Sentence in LSTM:

```
#####     LSTM model

Y_pr_eng = model.predict(temp_eng)
Y_pred_eng = []
for i in range(len(Y_pr_eng)):
  if Y_pr_eng[i][0]>=Y_pr_eng[i][1]:
    Y_pred_eng.append('ham')
  else:
    Y_pred_eng.append('spam')
print(eng)
print(Y_pred_eng)
```

```
1/1 [==============================] - 0s 34ms/step
Ela unnav?
['ham']
```

3. Testing For Random Telugu Sentence in CNN:

```
#####     CNN model

Y_pr_tel = model1.predict(temp_tel)
Y_pred_tel = []
for i in range(len(Y_pr_tel)):
  if Y_pr_tel[i][0]>=Y_pr_tel[i][1]:
    Y_pred_tel.append('ham')
  else:
    Y_pred_tel.append('spam')

print(tel)
print(Y_pred_tel)
```

```
1/1 [==============================] - 0s 20ms/step
మీకూ మీ కుటుంబ సభ్యులకు నూతన సంవత్సరం శుభాకాంక్షలు. రాము కృష్ణ, రజనీ, టిల్లు, టింకు.
['ham']
```

4. Testing For Random Telugu Sentence in LSTM:

```
#####     LSTM model

Y_pr_tel = model.predict(temp_tel)
Y_pred_tel = []
for i in range(len(Y_pr_tel)):
  if Y_pr_tel[i][0]>=Y_pr_tel[i][1]:
    Y_pred_tel.append('ham')
  else:
    Y_pred_tel.append('spam')

print(tel)
print(Y_pred_tel)
```

```
1/1 [==============================] - 0s 38ms/step
మీకూ మీ కుటుంబ సభ్యులకు నూతన సంవత్సరం శుభాకాంక్షలు. రాము కృష్ణ, రజనీ, టిల్లు, టింకు.
['ham']
```

5. Testing For Random Hindi Sentence in CNN:

```
#####     CNN model

Y_pr_hin = model1.predict(temp_hin)
Y_pred_hin = []
for i in range(len(Y_pr_hin)):
  if Y_pr_hin[i][0]>=Y_pr_hin[i][1]:
    Y_pred_hin.append('ham')
  else:
    Y_pred_hin.append('spam')

print(hin)
print(Y_pred_hin)
```

```
1/1 [==============================] - 0s 33ms/step
नया साल आपके और आपके पूरे परिवार के लिए समृद्धि और सम्पन्नता लाये। नव वर्ष 2018 मंगलमय हो।
['ham']
```

6. Testing For Random Hindi Sentence in LSTM:

```
[ ] #####     LSTM model

Y_pr_hin = model.predict(temp_hin)
Y_pred_hin = []
for i in range(len(Y_pr_hin)):
  if Y_pr_hin[i][0]>=Y_pr_hin[i][1]:
    Y_pred_hin.append('ham')
  else:
    Y_pred_hin.append('spam')

print(hin)
print(Y_pred_hin)
```

```
1/1 [==============================] - 0s 44ms/step
नया साल आपके और आपके पूरे परिवार के लिए समृद्धि और सम्पन्नता लाये। नव वर्ष 2018 मंगलमय हो।
['ham']
```

VIII. CONCLUSION AND FUTURE WORK:

In this paper, we presented machine learning methods for detecting spam messages. We compared the accuracy rates of the Convolutional Neural Network (CNN) approach with the Long Short-Term Memory (LSTM) algorithm. Our experiments showed that the LSTM technique achieved the best accuracy rate of 97.37%, outperforming the CNN technique with an accuracy rate of 94.74%. Our proposed method demonstrated superior performance compared to the existing methods. Based on these results, we plan to develop our approach on multiple datasets using the LSTM algorithm in the future.

IX. REFERENCES

[1] K. Kandasamy and P. Koroth, "An integrated approach to spam classification on Twitter using URL analysis, natural language processing and machine learning techniques," 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 2014, pp. 1-5, doi: 10.1109/SCEECS.2014.6804508.

[2] P. Sethi, V. Bhandari and B. Kohli, "SMS spam detection and comparison of various machine learning algorithms," 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), Gurgaon, India, 2017, pp. 28-31, doi: 10.1109/IC3TSN.2017.8284445.

[3] Chao Wang, Fengli Zhang, Fagen Li and Qiao Liu, "Image spam classification based on low-level image features," 2010 International Conference on Communications, Circuits and Systems (ICCCAS), Chengdu, China, 2010, pp. 290- 293, doi:10.1109/ICCCAS.2010.5581998.

[4] S. Agarwal, S. Kaur and S. Garhwal, "SMS spam detection for Indian messages," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 2015, pp. 634-638, doi: 10.1109/NGCT.2015.7375198.

[5] S. Vinothkumar, S. Varadhaganapathy, R. Shanthakumari, D. Ramkishore, S. Rithik and K. P. Tharanies, "Detection Of Spam Messages In E-Messaging Platform Using Machine Learning," 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), Sonepat, India, 2022, pp. 283-287, doi: 10.1109/CCiCT56684.2022.00060.

[6] Q. Xu, E. W. Xiang, Q. Yang, J. Du and J. Zhong, "SMS Spam Detection Using Noncontent Features," in IEEE Intelligent Systems, vol. 27, no. 6, pp. 44-51, Nov.-Dec. 2012, doi: 10.1109/MIS.2012.3.

[7] K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, China, 2011, pp. 101-105, doi: 10.1109/ICCSNT.2011.6181918.

[8] R. E, S. K and A. Sharma, "Multi-lingual Spam SMS detection using a hybrid deep learning technique," 2022 IEEE Silchar Subsection Conference (SILCON), Silchar, India, 2022, pp. 1-6, doi: 10.1109/SILCON55242.2022.10028936.

[9] S. Gadde, A. Lakshmanarao and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 358-362, doi: 10.1109/ICACCS51430.2021.9441783.

[10] G. Ubale and S. Gaikwad, "SMS Spam Detection Using TFIDF and Voting Classifier," 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 2022, pp. 363-366, doi: 10.1109/MECON53876.2022.9752078.

[11] Ahmad Fadhil Naswir, Lailatul Qadri Zakaria and Saidah Saad, "Phishing Emails Classification Research Trends:

[12] Q. Xu, E. W. Xiang, Q. Yang, J. Du and J. Zhong, "SMS Spam Detection Using Noncontent Features," in IEEE Intelligent Systems, vol. 27, no. 6, pp. 44-51, Nov.-Dec. 2012, doi: 10.1109/MIS.2012.3.

[13] K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," Proceedings of 2011 International Conference on Computer Science and Network Technology, Harbin, China, 2011, pp. 101-105, doi: 10.1109/ICCSNT.2011.6181918.

[14] R. E, S. K and A. Sharma, "Multi-lingual Spam SMS detection using a hybrid deep learning technique," 2022 IEEE Silchar Subsection Conference (SILCON), Silchar, India, 2022, pp. 1-6, doi: 10.1109/SILCON55242.2022.10028936.