ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

Detecting SPAM in Real-Time Chat Messages using LSTM Networks

[1] V.Srinivasarao Suragani, [2] J.A.Paulson,

[1,] PG Student, Department of CSE, Varaprasad Reddy Institute of Technology, Sattenapalli, Kantepudi (Village)

[2] Associate Professor, Department of CSE, Varaprasad Reddy Institute of Technology, Sattenapalli, Kantepudi (Village)

Abstract—

Thanks to recent improvements in mobile technology, SMS texting has become more commonplace than ever before. Because of this, more and more spam messages are being sent via mobile devices. Even while emails still account for the vast majority of spam, short message service (SMS) providers are getting close. Having one's mobile device flooded with spam texts is something no one wants. There are a lot of ways to detect and reduce spam communications, and studies on the topic are always happening. There are tremendous obstacles to overcome when trying to categorize spam in SMS texts. Several machine learning methods, including Support Vector Machine (SVM), Naive Bayes (NB), and Random Forest (RF), have been investigated in this issue. On the other hand, these methods have their limitations, thus they can't accurately categorize all kinds of spam. Therefore, in order to find a more trustworthy and precise method, a comprehensive inquiry is necessary. We present Long Short-Term Memory (LSTM), a state-of-the-art RNN design that uses memory cells in its Gating Mechanism, to tackle this problem.

Machine learning, artificial intelligence, natural language processing, and long short term memory are its keywords.

INTRODUCTION

With billions of people using mobile devices every day, messaging is a crucial method of interpersonal communication. This kind of communication, however, becomes susceptible in the absence of adequate message filtering measures. This vulnerability is further amplified by spam, making SMS conversation very unsafe. The prevalence of spam is one of the main complaints leveled against IM services. Emails sent to people without their permission are known as spam, and they may be quite annoying. Content such as phishing attempts or ads for goods and services are commonplace in these communications. The proliferation of spam communications parallels the meteoric rise in the use of mobile devices as means of communication. These notifications might cause consumers to lose money in certain situations. Spam costs nothing for the sender but a big penny for the receiver. Not only does spam cause people to lose valuable time, but it also interrupts critical conversations, which might lead to the disclosure of sensitive information. Our interactions have been drastically altered by the advent of technology. Through the use of ever-evolving chat applications and other gadgets, we are able to maintain flawless communication even when we are physically apart. Since the introduction of chat programs, the volume of spam messages has skyrocketed. Unwanted, uninvited, and potetially harmful electronic messages are known as spam. B. Issue Description Recurrent neural networks are a kind of artificial neural network. To generate the current state input, RNNs employ the output from the previous state. Current RNNs, however, have issues with diminishing gradient descent. The gradients of the loss function tend to approach 0 when more layers with particular activation functions are added to a neural network, making it difficult to train the model successfully. One kind of Recurrent Neural Networks (RNNs) designed to overcome the shortcomings of more conventional RNNs is the Long Short-Term Memory (LSTM) network. To lessen the impact of training deep networks' drawbacks, they may handle sequential input and gradually learn long-term dependencies. In 1997, Hochreiter and Schmidhuber presented the LSTM design. It solves the vanishing gradient issue by improving the standard RNN model with cell states, which aid in remembering or forgetting information. There is a network of gates, and each one controls one of these cellular states. A memory-cell state gate, an input gate, a forget gate, and an output gate make up the four main gates. If the incoming data is significant enough, the input

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

gate will hold it. To determine how much of the prior concealed state to save in the memory cell, the forget gate is responsible. After the input and forget gates have made their determinations, the information is updated by the memory-cell state gate. Lastly, the output gate determines the network's output by using the state of the memory cells at the moment.

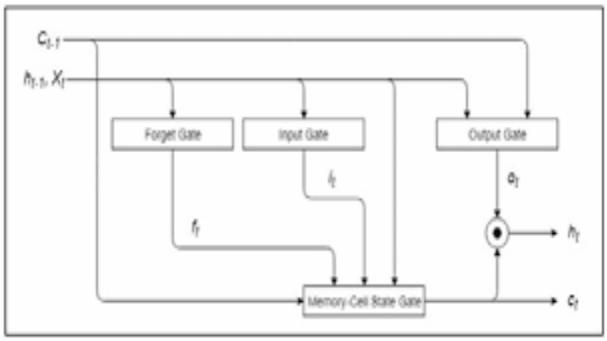


Fig. 1. Architecture of an LSTM Network

C. Related Works

A novel approach to spam email filtering is shown in research [14–16] by combining email context with particular qualities using PV-DM and the TF-IDF framework. Two vectors are used to represent each email, and the final categorization is performed by combining the findings from both vectors. Their results show that this dual-vector method improves the robustness of classifiers against changes in language structure and message coherence, and performs similarly to classic PV-DM and Bag-of-Words (BoW) models. Applying ML and DL methods to the task of Twitter sentiment analysis and spam detection is the subject of an additional research [17]. Spam detection is the process of finding and removing undesirable material, such as false profiles, ads, and unnecessary details. For real-time spam detection and removal, the researchers use a variety of ML and DL algorithms. Also, sentiment analysis may tell you whether a tweet is pleasant, negative, or neutral in tone based on its emotional tone. This research looks at how well ML and DL models can categorize tweets' sentiment in real time. Decision trees, support vector machines (SVMs), naive Bayes, and neural networks are some of the machine learning algorithms that are compared in a later research [18] that focuses on spam identification. Metrics like accuracy, precision, recall, and F1 score are used to assess the performance of these models. They are trained and tested on datasets that include both spam and non-spam samples. Various algorithms' abilities to decrease both false positives and false negatives are also investigated in the study. The article goes on to say that these algorithms utilize things like keywords, email headers, sender information, and language patterns to identify spam from real material. The authors of the study published in [19] investigate the use of machine learning to identify phishing and spam emails. This paper delves into several machine learning algorithms that have been trained on labeled email datasets to identify phishing and spam communications. These techniques include random forests, decision trees, support vector machines (SVMs), Bayes classifiers, and others. Improving the efficiency of email filtering systems is explored in the article, which also uses performance measures like accuracy and precision to assess various solutions. In addition, the paper offers suggestions on how current algorithms might be improved to increase detection rates and decrease false positives.

II. METHODS

Here are the main parts of an LSTM unit: The "cell state," an essential part of an LSTM model that acts as a memory cell and stores data throughout time, is the first component. In this picture, the cell state (the horizontal line at the top) is like a conveyor belt; data may pass across it unaltered. To compensate for the shortcomings of short-term memory, this technique allows crucial information to be preserved as the sequence is processed, guaranteeing that data from earlier time steps may impact subsequent ones. The model is able to regulate what is remembered or forgotten because gates govern the addition or removal of information as the cell state evolves. 2) Gates: The "Forget" gate finds out which bits of data from the cell's state may be ignored.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate: Controls how much new information should be added to the cell state.

$$i_t = \sigma\,(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\bar{C}_t = \tanh(W_c\,.\,[h_{t-1}, x_t] + b_c)$$

Output Gate: Determines the output of the LSTM cell based on the cell state.

$$o_t = \sigma\,(\,W_o \cdot [\,h_{t-1}\,,\,x_t\,] + b_o\,)$$
$$h_t = o_t * \tanh(C_t)$$

*3) **Cell State Update**:*

The cell state is updated using the input gate and the forget gate.

$$C_t = f_t \cdot C_{t-1} + it \cdot \bar{C}_t$$

Here, $\bar{C}_t$ is the candidate cell state created from the current input.

4) Hidden State:

The hidden state is computed from the output gate and the updated cell state.

$$h_t = o_t \cdot \tanh(C_t)$$
$$o_t = \sigma\,(\,W_o \cdot [\,h_{t-1}\,,\,x_t\,] + b_o\,)$$

The architecture of the chat application –

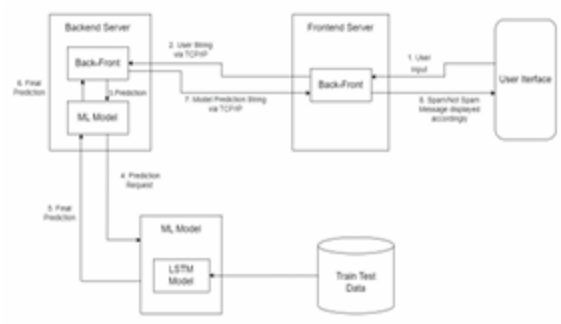


Fig. 2. System Architecture

Forming a server for the backend: 1. Place the model that has been trained for SMS spam detection on the server. 2. Make sure a port is open so the chatbot's frontend may talk to each other. 3. Take in user messages via strings and feed them into the trained SMS spam-detection model. 4. The model will transmit its predictions to the server in the backend, which will then transmit them to the server in the frontend. Building a server for the front end: 1. Establish a connection between the chatbot's (mobile app's or website's) frontend and the frontend server. 2. Make a separate thread for every client so their requests are not mixed up with each other. 3. When the user selects the "send" button, accept their communications. 4. After establishing a communicable port at the backend server, send the messages to that server and wait for a response. 5. Leave if "spam" is the reply. 6. Respond appropriately to the user if the answer is "ham". Details on how to put the LSTM model into action B. Dataset Employed Following the recommendation of Almeida and Hidalgo, we use a dataset consisting of SMS spam. There are about 5,574 records in this collection. Text messaging via SMS is a part of it. Additionally, there are English-language discussions that include numerical and textual elements into sentences of varied lengths. There are no unlabeled entries in this dataset. The number of records labeled as spam is 1, whereas the number of entries labeled as regular communications is 0. Our team has also made any necessary adjustments to the data. Simple greetings like "Good morning," "Good evening," etc., have been flagged as spam. Part C: Dataset Preprocessing To prepare data that is derived from natural language, this technique employs Natural Language Processing (NLP). Natural language processing (NLP) is a method that gives computers human-level comprehension of natural language [15]. To prepare data for computers to understand, a number of methods are used. In this research, we The term The process of converting a sentence's words into their token form is known as tokenization. Using a set of defined, engaging vocabulary words, this process creates a word tokenizer. Once a sentence's words have been constructed, they may be transformed into sequence data using a word tokenizer. When terms are tokenized, they become indices, and for unknown phrases, the index is set to 0. As part of the padding process, we extend all of the dataset's sequences consistently so that LSTM and GRU may train on them. In order to get the optimal message length, we use (1). We add zeros to the beginning of sequences that are less than the needed length to make them fit the intended length after the message length is optimized. Word Embedding—This technique takes a set of pre-processed words and turns them into a more complicated vector representation called embedding space, which has more dimensions than regular word data. All LSTM and GRU model training relies on this vectorization. Following data padding and truncation, word embedding is applied with an embedding size of 32 to raise the input data's dimensionality and provide a better representation for the model's learning. Modeling—Using the Long Short-Term Memory (LSTM) method, a deep learning approach, we construct models to classify SMS spam.

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

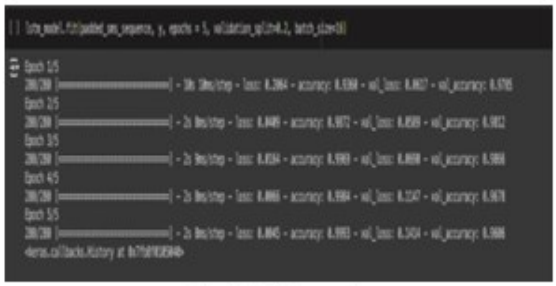


Fig. 3. Training results

E. Verification and Testing The difference between the predicted values and the actual target values is called loss, and it is computed using loss functions. A model's accuracy is a measure of how well it predicts outcomes. Here is one way to put it:

$$Accuracy = \frac{\text{Total number of Predictions}}{\text{Number of Correct Predictions}} * 100$$

The following are the training and validation loss and accuracy graphs: The following inferences are possible from the charts shown above: Errors in Training and Precision: Over the course of the training epochs, the accuracy improves and the loss drops, eventually reaching near 100%. Accuracy and Validation Loss: Potential overfitting is indicated by an increasing validation loss, which starts out low. After the third epoch, validation accuracy begins to decline from its high initial state.

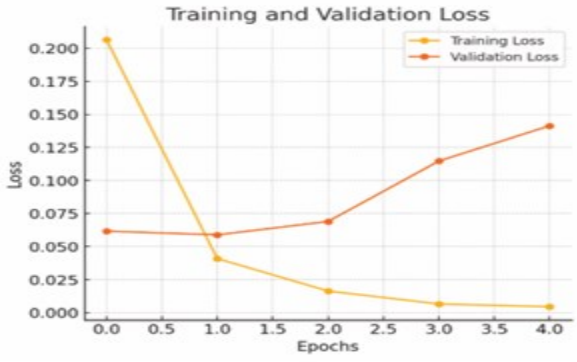


Fig. 4. Training and Validation Loss

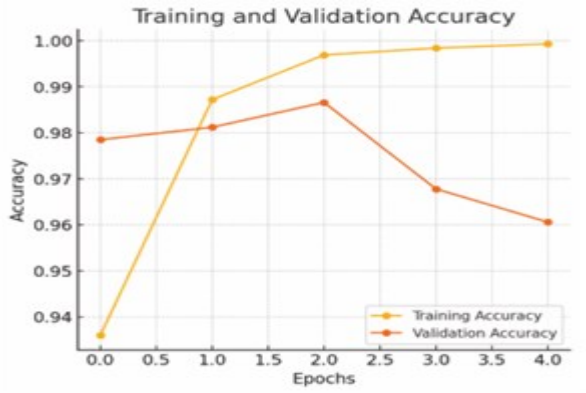


Fig. 5. Training and Validation Accuracy

The model seems to be overfitting, as the validation loss is increasing and the validation accuracy is somewhat decreasing, even if the model does well on the training data. An increase to the dropout rate to 40% prevented the model from being unduly reliant on any one neuron during training, which in turn improved its generalizability and prevented overfitting. To avoid overfitting, the model can't be trained for too long; early halting cuts training when the validation loss stops increasing. After correcting for overfitting, plots reveal a steady decline in training loss and a constant validation loss beyond a certain point (caused by early termination). After a certain point, the validation accuracy stabilizes, and the training accuracy approaches 100%, indicating that overfitting has been successfully addressed.

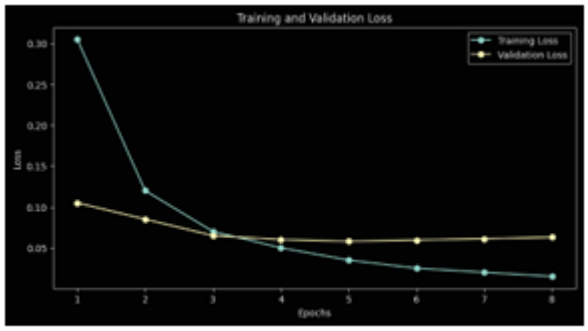


Fig. 6. Figure 1 Training and Validation Loss after addressing overfitting

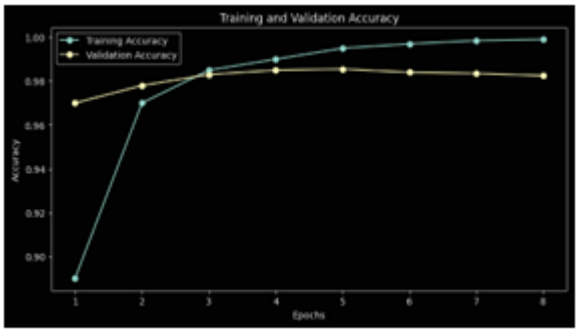


Fig. 7. Training and Validation Accuracy after addressing overfitting

ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

The following figure shows the results obtained after providing different inputs to the model-

```
# check_spam = ["Free entry in 2 a wkly comp to win FA Cup finals"]
# check_spam = ["Nah I don't think he goes to usf"]
check_spam_msg = input()
check_spam_msg = [check_spam_msg]
sms_spam_tokenizer.fit_on_texts(X)
check_spam_tokenized = sms_spam_tokenizer.texts_to_sequences(check_spam_msg)
print(prediction(check_spam_tokenized, sms_spam_tokenizer))
check_ham_msg = input()
check_ham_msg = [check_ham_msg]
# print(check_spam_ham_msg)
check_ham_tokenized = sms_spam_tokenizer.texts_to_sequences(check_ham_msg)
# print("Text          : ", check_spam_ham_msg[0] )
# print("Numerical Sequence : ", check_spam_tokenized[-1])

print(prediction(check_ham_tokenized, sms_spam_tokenizer))
```

```
Free entry in 2 a wkly comp to win FA Cup finals
1/1 [==============================] - 0s 18ms/step
SPAM!
Nah I don't think he goes to usf
1/1 [==============================] - 0s 17ms/step
HAM!
```

Fig. 8. Outputs of the LSTM model

To develop the chat application, the following technologies were used –

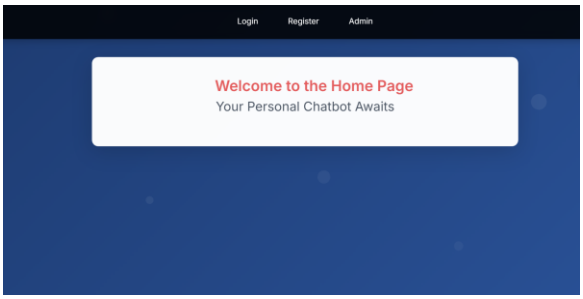
TABLE I.

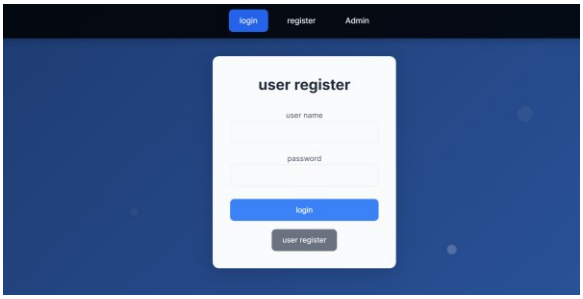
Sr. No	Tools/Technology	Use
1	Python	Backend Development
2	Python Libraries (Scikit-learn, Pandas, etc.)	Spam detection Model
3	ReactJS	Frontend Development(Web)
4	React Native	Frontend Development(mobile)
5	Selenium	Web testing
6	Postman	API Testing

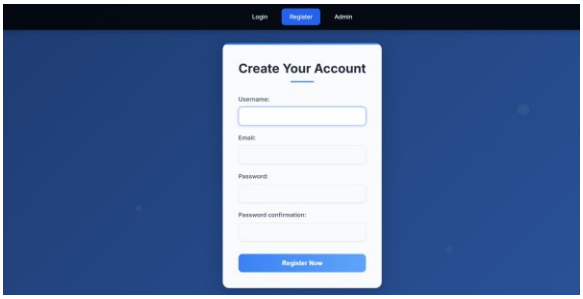
III. RESULTS

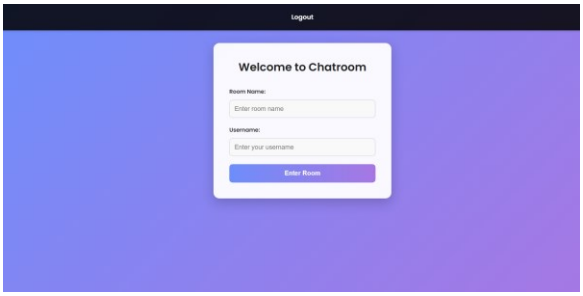
Impressive accuracy in distinguishing between spam and valid communications has been achieved by using an LSTM (Long Short-Term Memory) model for spam detection in a chat application. Following training on a labeled dataset, such the widely-used spam.csv file, the LSTM model successfully captured word relationships over the long run. occurrences, which is critical for comprehending the background of conversations on chat. The capacity to retain context over The model's ability to generate more educated predictions is enhanced in longer sequences, leading to better spam categorization in real-time chat situations. By reducing the number of false positives and correctly recognizing spam messages, the model was able to prevent the incorrect flagging of real interactions. See Figure 6. Illustration 1: Training and Validation Revenue after the removal of overfitting Figure 7: Validation and Training Precision after correction for overfitting Figure 2 illustrates the output from running the model with various inputs: Figure 8: LSTM Model Outputs The following technologies were used in the development of the chat application: Section I: Table of Non-Useful Tools and Technologies Development of a Python backend 2 Python Packages (e.g., Scikit-Learn and Pandas) Web front-end development, spam detection model, ReactJS, and React Native Work on the mobile front end 5. Web testing using Selenium 6. Testing using Postman API Part III: Findings

Impressive accuracy in distinguishing between spam and valid communications has been achieved by using an LSTM (Long Short-Term Memory) model for spam detection in a chat application. After training on a labeled dataset, we introduced features like group chat, media upload (images, videos, gifs, pdf), login, and logout in addition to spam detection. This is what the chat app's web and android versions have to offer in terms of results:



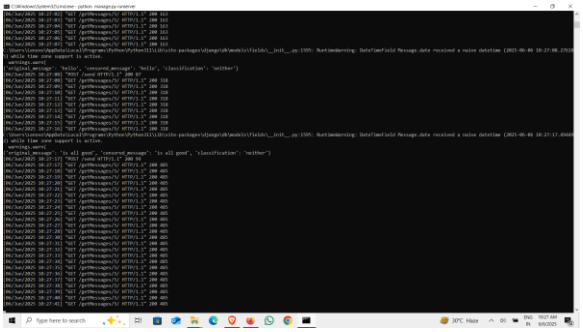


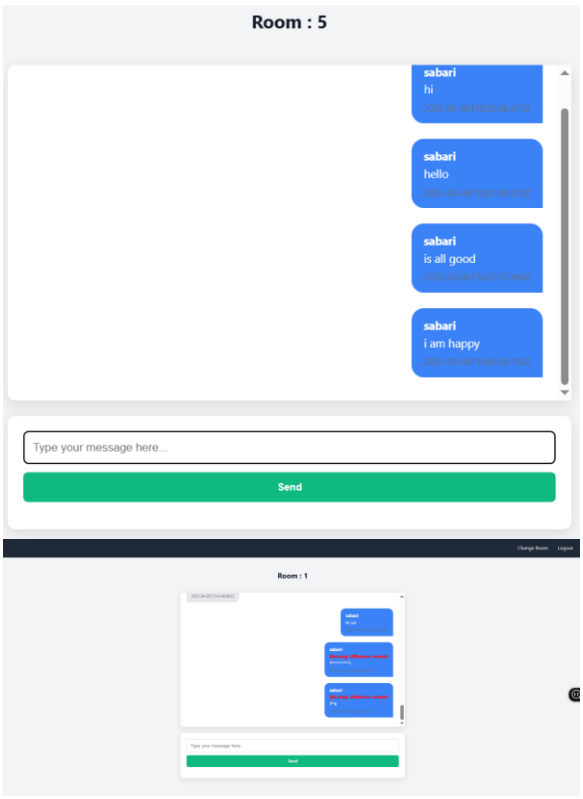




ISSN 2321-2152
www.ijmece.com
Vol 13, Issue 2, 2025

```
import re
import nltk
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))
Windsurf: Refactor | Explain | Generate Docstring | ×
def clean_text(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
df_offensive['tweet']=df_offensive['tweet'].apply(clean_text)
x=df_offensive['tweet']
y=df_offensive['class']
print(df_offensive['class'].unique())
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42)


#print(len(x_train), len(y_train))
#print(len(x_test), len(y_test))
type(x_train)
from sklearn.feature_extraction.text import CountVectorizer

count = CountVectorizer(stop_words='english')
x_train_vectorizer=count.fit_transform(x_train)
x_test_vectorizer=count.transform(x_test)
x_train_vectorizer.toarray()
from sklearn.feature_extraction.text import TfidfTransformer
tfidf = TfidfTransformer()

x_train_tfidf = tfidf.fit_transform(x_train_vectorizer)

x_train_tfidf.toarray()
x_test_tfidf = tfidf.transform(x_test_vectorizer)
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```





IV. CONCLUSION

Because of their ability to grasp contextual information and long-term relationships inside text sequences, LSTM (Long Short-Term Memory) networks perform better than conventional spam detection systems, according to the validation findings. Because of this improvement, LSTM networks are better able to detect spam material in different situations by understanding linguistic subtleties than traditional methods. Long short-term memories (LSTMs) are essential for spam detection in conversational or sequential contexts because they can handle sequences of data and remember previous inputs, in contrast to traditional machine learning algorithms that depend on hand-crafted features and may have trouble accounting for sequential word patterns.

Long short-term memory (LSTM) networks are great at recognizing complex spam communications because they comprehend the structure and flow of language over time, in contrast to approaches that rely on static, independent variables, such as decision trees or logistic regression. Long Short-Term Memory (LSTM) models are able to separate messages that seem identical by learning patterns in the sequence of words. This allows them to detect minor signs of spam that other models may miss. Because of its superior comprehension of both the

current and previous words or phrases, LSTMs excel in processing complicated text data, which is particularly useful in ever-changing contexts like chat apps, where spam patterns might change. Consequently, as compared to other machine learning approaches, LSTMs' sequence processing capabilities provide a more resilient and versatile strategy for spam identification. Our end result is an all-inclusive chat software that works well on mobile devices as well as the web. This program has sophisticated text message spam detection features. Several Natural Language Processing (NLP) methods were used to pre-process the text messages in order to make sure the spam identification is accurate and efficient. Some of these methods include word tokenization, which segments text into its component words; padding and truncating, which adjust the length of input sequences to meet model specifications; and word embedding methods, which convert text data into numerical sequences that the LSTM model can handle more efficiently. The LSTM model has been painstakingly trained using a Kaggle dataset that includes 5,574 messages in total. Additional data that was targeted to those specific spam messages was also included in the training process to further improve the model's effectiveness against certain forms of spam. Overfitting was also addressed by implementing early halting and raising the dropout rate. Consequently, the chat app now has strong anti-spam features without sacrificing functioning. A lot of the functionality that users are used to from more traditional chat apps is available here as well, including the ability to transmit media items like movies, photos, and PDFs. The software is a flexible and easy-to-use platform for communication since it provides emoji responses and allows group conversations.

REFERENCES

[1] M.Rubin Julis, S.Alagesan, “Spam Detection In SMS Using Machine Learning Through Text Mining” 2020.
[2] . F. Kai Petersen, Shahid Mujtaba, Michael Mattsson, "Systematic Mapping Studies in Software Engineering," 2008.
[3] Muhammad Iqbal, Malik Muneeb Abid, Mushtaq Ahmad, Faisal Khurshid, “Study on the Effectiveness of Spam Detection Technologies”, 2016
[4] Luo GuangJun, Shah Nazir, Habib Ullah Khan, Amin Ul Haq, “Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms”, 2020
[5] Nikhil Kumar, Sanket Sonowal, Nishant, “Email Spam Detection Using Machine Learning Algorithms”, 2020
[6] Pumrapee Poomka, Wattana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop “SMS Spam Detection Based on Long Short Term Memory and Gated Recurrent Unit” , 2019
[7] Lingyun Xiang, Guoqing Guo, Qian Li, Chenzhang Zhu, Jiuren Chen, Haoliang Ma, “Spam Detection in reviews using LSTM based multi entity temporal features”, 2020.
[8] https://www.analyticsvidhya.com/blog/2021/05/sms-spam-detection using-lstm-ahands-on-guide/ [9] Sanjiban Shekhar Roy, Saptarshi Chakraborty, Swapnil Sourav, Ajith Abraham, “Rough set theory approach for filtering spams from boundary messages in a chat system”, 2013.
[10] Alberto, T. C., Lochter, J. V., & Almeida, T. A. Tubespam: Comment spam filtering on youtube. In 2015 IEEE 14th international conference on machine learning and applications (ICMLA) (pp. 138–143). IEEE.
[11] Banerjee, S., Chua, A. Y., & Kim, J.-J. Using supervised learning to classify authentic and fake online reviews. In Proceedings of the 9th international conference on ubiquitous information management and communication (p. 88). ACM, 2015.
[12] Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(1), 23. 2015.
[13] Abdallah Ghourabi, Mahmood A. Mahmood, Qusay M. Alzubi, A hybrid CNN-LSTM Model for SMS spam detection in Arabic and English Messages, 2020.
[14] Alrawad, M., Lutfi, A., Almaiah, M. A., Alsyouf, A., Arafa, H. M., Soliman, Y., & Elshaer, I. A. (2023). A Novel Framework of Public Risk Assessment Using an Integrated Approach Based on AHP and Psychometric Paradigm. Sustainability, 15(13), 9965.
[15] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms. Sensors, 24(2), 713.
[16] Ali, A., Pasha, M. F., Fang, O. H., Khan, R., Almaiah, M. A., & K. Al Hwaitat, A. (2022). Big data based smart blockchain for information retrieval in privacy-preserving healthcare system. In Big Data Intelligence for Smart Applications (pp. 279-296). Cham: Springer International Publishing.
[17] Almaiah, M. A. (2020). An Efficient Smart Weighted and Neighborhood-enabled Load Balancing Scheme for Constraint Oriented Networks. International Journal of Advanced Computer Science and Applications, 11(12).

[18] DAWAHDEH, Z. E., ALMAIAH, M. A., ALKHDOUR, T., LUTFI, A., ALDHYANI, T. H., & BSOUL, Q. (2024). A NEW MODIFIED GRAYSCALE IMAGE ENCRYPTION TECHNIQUE USING ELLIPTIC CURVE CRYPTOSYSTEM. Journal of Theoretical and Applied Information Technology, 102(7).
[19] Vijayalakshmi, K., Al-Otaibi, S., Arya, L., Almaiah, M. A., Anithaashri, T. P., Karthik, S. S., & Shishakly, R. (2023). Smart Agricultural–Industrial Crop-Monitoring System Using Unmanned Aerial Vehicle– Internet of Things Classification Techniques. Sustainability, 15(14), 11242.
[20] Almaiah, M. A., & Alkdour, T. (2023). Securing Fog Computing Through Consortium Blockchain Integration: The Proof of Enhanced Concept (PoEC) Approach. In Recent Advancements in Multimedia Data Processing and Security: Issues, Challenges, and Techniques (pp. 107- 140). IGI Global.