

SMS SPAM DETECTION USING DEEPLARNING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Nikhith M (Reg. No - 40110800)
Sai Charan M (Reg. No -40110798)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI, CHENNAI – 600119

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Nikhith M (40110800)** and **Sai Charan M (40110798)** who carried out the Project Phase-2 entitled "**SMS SPAM DETECTION USING DEEP LEARNING**" under my supervision from January 2024 to April 2024.

Internal Guide

D.DEVI, ME., (Ph.D.)

Head of the Department
HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva Voce Examination held on 12/04/2024

Internal Examiner
External Examiner

DECLARATION

I, **Sai Charan M** (Reg.No- 40110798), hereby declare that the Project Report entitled "**SMS Spam Detection Using Deep Learning**" done by me under the guidance of **D.Devi, ME.,(Ph.D.)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 12/04/2024

Sai Charan

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **D.Devi, ME.,Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

SMS also known as Short Message Services plays a major part in our day-to-day digital life that people spend via their mobile phones. Over, the recent years there has been a widespread popularity of mobile phones among the population leading to more exposure to them. This gives a change to the spammers to scam the customers via numerous ways, one of them being the SMS Spam Messages. A SMS spam is the message that hackers develop and send to people via mobile devices targeting to get their important information. The spammer tries to send spam messages for their financial or business benefits like market growth, lottery ticket information, credit card information, etc. For people who are ignorant, if they follow the instruction in the message and fill their important information, such as internet banking account in a faked website or application, the hacker may get the information. Machine learning method comes to be the most popular choice for Spam detection, a number of researchers have utilized supervised machine learning methods for comparative results of spam classification. A plenty of research has been carried out in this direction make use of machine learning techniques such as Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM) etc. Using traditional machine learning methods as mentioned above, the feature engineering is a time consuming process with the extra computational expense. It is also difficult to extract all the information in short length of the text. Hence, we have perceived a technique that involves Deep Learning, that makes use of the Long Short-Term Memory Algorithm. Although these methods have shown adequate performance, but are not efficient enough in terms of spam classification. Hence, a rigorous study is needed to find a more accurate and robust method. To address this, we proposed a novel method Long Short-Term Memory which is an advanced structure of Recurrent Neural Network that has gating mechanism including memory cells. To evaluate the effectiveness of our method, SMS datasets have been used which are freely available and we have updated the current dataset with varied test entries.

TABLE OF CONTENTS

CHAPTER NO:	TITLE	PG NO:
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
	INTRODUCTION	1
1	1.1 Problem Statement	2
	1.2 Objective and Scope	2
	LITERATUTE SURVEY	3
2	2.1 Inferences from Literature Survey	5
	2.2 Existing System	5
	2.2.1Introduction	6
	2.2.2 Drawbacks	
	DESCRIPTION OF PROPOSED SYSTEM	7
	3.1 Description of Research Approach and Methods	7
	3.2 Work Plan of the Project	9
	3.3 Model Workflow	11
	3.4 UML Diagrams	12
3	3.5 Class Diagram	13
	3.6 Object Diagram	14
	3.7 Activity Diagram	14
	3.8 Sequence Diagram	15

	REQUIREMENT ANALYSIS	
4	4.1 Feasibility Study of the Project	17
	4.2 Requirements Specification	18
	DETAILS AND RESULTS	
	5.1 Development and Deployment Setup	23
5	5.2 Algorithms	23
		24
	5.3 Implementation	25
		25
	5.4 Results	
	5.5 Data Analysis and Implementation	
	SUMMARY AND CONCLUSIONS	
	6.1 Conclusion	29
6		30
	6.2 Recommendations for Future Research	30
		32
	6.3 Future Scope	
	6.4 Summary	
	REFERENCES	33
	APPENDIX	35
		35
	A. Source Code	41
	B. Screenshots	46
	C. Research Paper	

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	STRUCTURE OF LONG SHORT-TERM MEMOMRY	8
3.2	SYSTEM ARCHITECTURE	9
3.3	PROJECT DESIGN FLOW	11
3.4	USE CASE DIAGRAM	13
3.5	CLASS DIAGRAM	14
3.6	OBJECT DIAGRAM	14
3.7	ACTIVITY DIAGRAM	15
3.8	SEQUENCE DIAGRAM	16
5.1	IMAGE SHOWING THE MESSAGE IS NOT SPAM	26
5.2	IMAGE SHOWING THE MESSAGE IS SPAM	26
5.3	DISPLAY THE DESCRIPTION OF HAM AND SPAM MESSAGES SEPARETLY	26
5.4	LENGTH COMPARISION BETWEEN SPAM AND HAM	27
5.5	ACCURACY OF LSTM MODEL	27
5.6	LENGTH COMPARISION BETWEEN SPAM AND HAM	28
5.7	CONFUSION MATRIX	28

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
SMS	Short Message Service
NLP	Natural Language Processing
LSTM	Long Short-Term Memory
UML	Unified Modeling Language

CHAPTER1

INTRODUCTION

Human uses short message service (SMS) in mobile as a way of communication or business. Recently SMS is the most used data service in the world. SMS is an important feature for business communications. Recently, SMS spam target mobile phones. SMS spam refers to any illusion text message that is delivered using the mobile network. They are disturbing to users. A survey exposes that 68% of mobile phone users are affected by SMS Spam. In some cases, SMS spam contains malicious activities, such as smishing. Smishing is a cyber-security attack for mobile user aimed at deceiving the user via SMS spam messages that may include a link or malicious software or both. The most victims of smishing and phishing attacks are users that have a smartphone. The attackers attempt to steal user's secret information, like credit card number, bank account details, etc., by sending a link or direct contact with victims by SMS texts.

The filtration of SMS Spam in smartphones is still not very robust as compared to email spams that are supported by advanced methods of spam filtering. In the recent years, several researchers have proposed solutions to identify SMS spam. The most commonly known Machine Learning techniques that have been proposed include the Naive Bayes Algorithm, Random Forest, Decision Tree, Support Vector Machine and AdaBoost etc.

For working with SMS spam, there is a lag of research that applying deep learning algorithm to develop models. Therefore, in this research, we aim to develop SMS spam classify model for preventing people from the effect of SMS spam. we propose to develop SMS spam classify model using Natural Language Process (NLP) techniques for data preparation and developing model based on LSTM.

1.1 PROBLEM STATEMENT

Short Message Service (SMS) spam is a type of the spam messages that hacker send it via mobile devices targeting to get their sensitive information. Nowadays, a SMS spam attack is greatly affecting various people who trust the information in the message and follow the instruction from the hacker. The innocent people who get the message and panic, follow the instruction from the hacker and give their sensitive information to the hacker. This problem can be alleviated if we have a tool that can effectively detect the spam messages. Hence, we have proposed a solution to detect whether a received SMS is spam or not with the help of a Deep Learning technique known as Long Short-Term Memory (LSTM) which can configure the parameters in terms of statistical data.

1.2 OBJECTIVE AND SCOPE

OBJECTIVE

SMS spam detection using deep learning has emerged as a critical technology in safeguarding users from unwanted and potentially harmful messages. This essay explores the current advancements in this field and outlines the promising future scope for further development. In recent years, deep learning techniques have revolutionized the way spam detection is approached. With the ability to automatically extract relevant features from vast amounts of data, deep learning models have shown remarkable accuracy in identifying and filtering out spam messages. These models leverage neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to process text data and detect patterns indicative of spam.

SCOPE

Looking ahead, the future scope of SMS spam detection using deep learning is highly promising. One avenue of advancement lies in improving the accuracy of detection algorithms. This entails further refinement of neural network architectures, optimization of training algorithms, and exploration of novel data preprocessing techniques. By enhancing accuracy, deep learning models can better differentiate between legitimate and spam messages, minimizing false positives and negatives.

CHAPTER 2

LITERATURE SURVEY

1. **Pavas Navaney, Gaurav Dubey, Ajay Rana, "SMS Spam Filtering using Supervised Machine Learning Algorithms" in IEEE 2018** : This paper presents detection of Spam and ham messages using various supervised machine learning algorithms like naïve Bayes Algorithm, support vector machines algorithm, and the maximum entropy algorithm and compares their performance in filtering the Ham and Spam messages. As people indulge more in Web-based activities, and with rising sharing of private – data by companies, SMS spam is very common. As overall utilization of cell phones has grown, another road for e-junk mail has been opened for notorious advertisers. These publicists use instant messages (SMS) to target probable purchasers with undesirable publicizing known as SMS spam. This sort of spam is especially bothersome since, not at all like email spam, numerous PDA clients pay an expense for each SMS got. Building up a classification algorithm that channels SMS spam would give a helpful apparatus for mobile phone suppliers.
2. **Shafi'i Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Haruna Chiroma, Oluwafemi Osho, Gaddafi Abdul-Salaam, Adamu Abubakar, and Tutut Herawan, "A Review on Mobile SMS Spam Filtering Techniques " in IEEE 2017**: Short messaging service (SMS) spam is understood the unsolicited or undesired messages received on mobile phones. These SMS spams constitute a veritable nuisance to the mobile subscribers. This marketing practice also worries service providers in view of the fact that it upsets their clients or even causes them lose subscribers. By way of mitigating this practice, researchers have proposed several solutions for the detection and filtering of SMS spams. In this paper, we present a review of the currently available methods, challenges and future research directions on spam detection techniques, filtering and mitigation of mobile SMS spams. The most popular techniques for SMS spam detection, filtering and mitigation are compared, including the used datasets, and limitations, and the future research directions are discussed. This review is designed to assist expert researchers to identify open areas that need further improve.

3. **Sjarif NNA, Azmi NFM, Chuprat S, Sarkan HM, Yahya Y, Sam SM. SMS spam message detection using term frequency-inverse document frequency and random forest algorithm. Proc Comput Sci. 2019:** The authors presented a feature extraction method based on the term-frequency-inverse document frequency method to assess relevant words. They applied different ML models to evaluate the classifier's overall performance. The study showed that the combination of TF-IDF and random forest algorithm outperforms other state-of-the-art algorithms, thus improving the detection of SMS spam. The study proposed cost-sensitive techniques based on stacking of multilayer perceptron (MLP) algorithm and Bayesian network. The contribution of the study was its ability to minimize misclassification rate and very impressive performance when compared with other ML algorithms.
4. **Gadde S, Lakshmanarao A, Satyanarayana S. SMS spam detection using machine learning and deep learning techniques. Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS); IEEE:** This paper presented the application of deep learning techniques for SMS spam detection using LSTM. Author also applied three different word embedding techniques based on the count, TF-IDF, and hashing vectorizer. The experimental results for LSTM were compared with some of the state-of-the-art ML techniques. However, author in the latter demonstrated the robustness of LSTM topologies with a clonal selection algorithm for text classification. The study was evaluated using three datasets and benchmarked with some of the state-of-the-art ML classifiers. The experimental results showed that their proposed model outperforms other models regarding accuracy, precision, recall, F1- score, and computational time.
5. **Roy PK, Singh JP, Banerjee S. Deep learning to filter SMS spam. Future Gener Comput Syst. 2020:** Roy proposed a deep learning method based on convolutional neural networks and long short-term memory models in the classification of SMS spam. The study showed a significant performance of deep learning models using three configurations, and the addition of regularization parameters such as dropout improved classification accuracy. The model generated therefore has great accuracy and computational speed

2.1 INFERENCE FROM LITERATURE SURVEY

- **Model Performance:** Deep learning models, particularly recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have shown promising performance in SMS spam prediction compared to traditional machine learning methods.
- **Data Preprocessing:** Effective preprocessing techniques such as tokenization, stemming, and removal of stop words are essential for improving the performance of deep learning models on SMS spam prediction tasks.
- **Feature Engineering:** While deep learning models are adept at automatically learning features, incorporating handcrafted features such as message length, presence of specific keywords, and lexical features can further enhance the predictive performance.
- **Model Architecture:** Different architectures such as LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), CNN, and hybrid models have been explored for SMS spam prediction, each with its own advantages and limitations.
- **Transfer Learning:** Transfer learning from pre-trained language models such as BERT or GPT can potentially improve the performance of deep learning models for SMS spam prediction, especially when dealing with limited labeled data.
- **Evaluation Metrics:** Besides traditional metrics like accuracy, precision, recall, and F1-score, additional metrics like receiver operating characteristic (ROC) curve and area under the curve

2.2 EXISTING SYSTEM

2.2.1 Introduction

The existing systems in SMS spam detections are very useful for the consumers, so as to prevent any kind of digital malicious activity from occurring and causing any personal and financial damage to the consumer. They are mostly based upon various machine learning and data mining techniques. These machine learning algorithms include Naïve Bayes, Multinomial Naïve Bayes, Logistic Regression, Support Vector Machine, Random Forrest Classifier, K Nearest Neighbor, AdaBoost and even a rule

based classification etc but they've not given us very accurate classification of SMS as spam or ham. The number of False Positives predicted were more, which reduces the accuracy of the algorithm and will not be able to predict a SMS as spam or ham with highest of accuracy.

2.2.2 Drawbacks

- The accuracy is low.
- The performance and training time was slow.
- Low F1 Scores.
- Similar cosine values indicating the under- performance on the dataset.
- Difficult to configure the parameter in term of statistical data
- Not considering presence of informal languages like idiom, phrases, shortcuts and abbreviations.

CHAPTER 3

DESCRIPTION OF PROPOSED SYSTEM

We propose a model using Deep Learning Technique called LSTM(Long Short Term Memory) which is used to analyse the sequential data. We also use Natural Language Process (NLP) for pre-processing natural language data. NLP is the process to make computer understanding the natural language as same as a human understands. It has many techniques to pre-process data into a format that a computer can be understood. We transform the SMS text data into sequential data using NLP techniques in order to use it for developing SMS classification models using the LSTM algorithms. A function is created that would classify the message as spam or ham when the user gives the message as input. We also use word tokenization, padding data, truncating data and word embedding techniques for pre-processing data.

3.1 DESCRIPTION OF RESEARCH APPROACH AND METHODS

The standard specified in international journal of research publication and reviews ISSN 2582-7421 the form and content of individual test documents is followed to document the methodology. The selected process uses LSTM algorithm.

3.1.1 Deep Learning

Deep learning is a type of machine learning that uses algorithms meant to function in a manner similar to the human brain. While the original goal for AI was broadly to make machines able to do things that would otherwise require human intelligence, the idea has been refined in the decades since. François Chollet, AI researcher at Google and creator of the machine learning software library Keras, says: "Intelligence is not a skill in itself, it's not about what you can do, but how well and how efficiently you can learn new things."

3.1.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs

you will get insight into LSTMs using the words of research scientists that developed the methods and applied them to new and important problems. There are few that are better at clearly and precisely articulating both the promise of LSTMs and how they work than the experts that developed them. (fig 3.1)

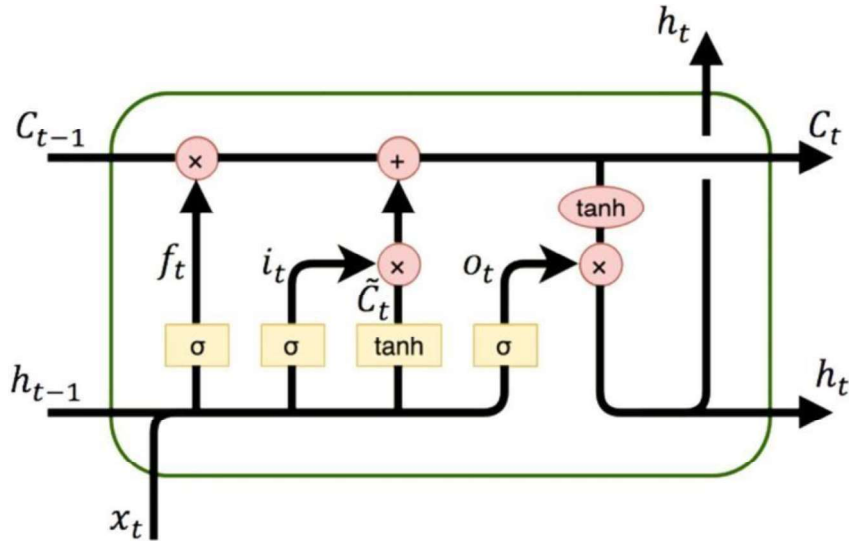


Fig. 3.1 Structure of Long Short-Term Memory

3.1.3 Natural Language Processing

Natural language processing (NLP) is an interdisciplinary subfield of computer science and information retrieval. It is primarily concerned with giving computers the ability to support and manipulate human language. It involves processing natural language datasets, such as text corpora or speech corpora, using either rule-based or probabilistic (i.e. statistical and, most recently, neural network-based) machine learning approaches. The goal is a computer capable of "understanding the contents of documents, including the contextual nuances of the language within them. To this end, natural language processing often borrows ideas from theoretical linguistics.

3.1.4 System Architecture

The system architecture of an SMS spam prediction project utilizing Natural Language Processing (NLP) and Long Short-Term Memory (LSTM) algorithms typically follows a structured approach tailored to exploit the sequential nature of text data. Initially, the raw SMS data is collected and subjected to preprocessing, including steps like tokenization, lowercasing, and removal of

punctuation and stop words. Additionally, feature engineering may involve extracting linguistic features or encoding the text into numerical representations, such as word embeddings.(fig 3.2)

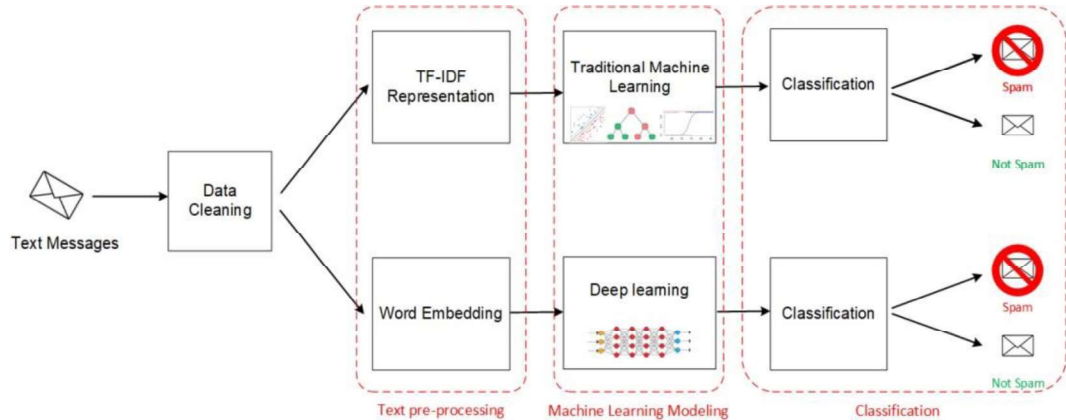


Fig. 3.2 System Architecture

3.2 WORK PLAN OF THE PROJECT

We move to the plan of executing the creation of our model after developing a precise problem specification and limiting down the Algorithms to be used as a part of the model. As part of the work plan, the following phases were created:

Stage1: Dataset Collection:

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. We have taken it from Kaggle (Link: <https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset>) It contains one set of SMS messages in English of 5658 messages, tagged according being ham (legitimate) or spam. The files contain one message per line. Each line is composed by two columns: v1 contains the label (ham or spam) and v2 contains the raw text.

Stage 2: Data Cleansing:

The first step in data cleaning is to remove all the unwanted columns. We also remove stop words like is in this step. This process next involves exploring and comparison of the length of the text messages. It then performs

under sampling on the dataset.

Stage 3: Text Preparation:

The next step involved is text preparation. The first step in text preparation is tokenization of messages. Next one hot implementation is applied on the tokenized messages. Word embedding is performed on the text, using the Word2Vec technique.

Stage 4: Data Preparation/ Data splitting:

The data in the dataset is split into training and validation as 85% and testing data as 15%. The training and validation data is further split into 85% training data and 15% validation data.

Stage 5: Model Building:

A Sequential Model is built which consists of 3 important layers. They are Embedding Layer followed by the LSTM Layer and then concludes with the Dense Output Layer. Then the training data is fit onto the model created and validated by using the validation data.

Stage 6: Model Evaluation:

The model is then evaluated by running the model on the test dataset. The model accuracy score is obtained and also a visualization of the confusion matrix is obtained as the output.

Stage 7: Model Testing:

A function is then created that would classify the message as spam or ham when the user gives the message as input.(fig3.3)

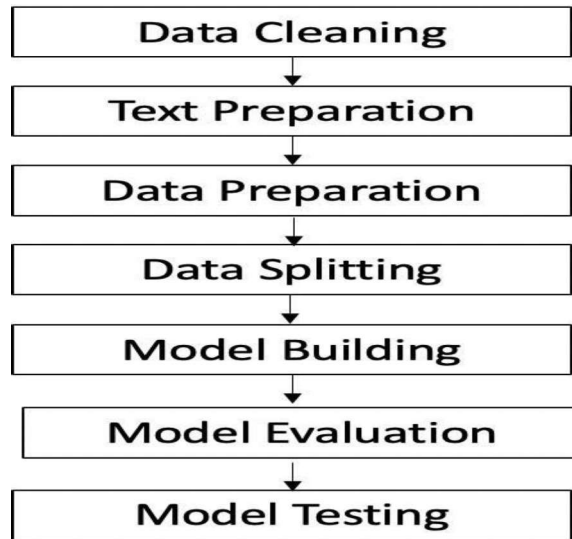


Fig. 3.3 Project Design Flow

3.3 Model WorkFlow

In order to carry out our work plan in a way that is consistent with logic, we start by constructing a sequential model that is capable of using the spam dataset as an input. As a later phase in the process of Validation and Evaluation of the Model, the dataset is subjected to the following steps:

- 1) We build the sequential LSTM model which consists of three layers which are embedding layer, LSTM, and dense layer(output).
- 2) In the embedding layer We declare input_dimesion as our vocabulary size(10,000),input_length as the sentence length andoutput dimension as 100.
- 3) In the LSTM layer we Add 128 units to the layers whose output will be fed as an input to ouroutput Dense layer.
- 4) In the Dense layers we Add 1 unit(neurons) to the dense layers and with a sigmoid activation, since we have binary classification problem.
- 5) Next, we Evaluate the model on test dataset and get the model accuracy score and visualizeconfusion matrix
- 6) Lastly, we do model testing by using the Created a function that would classify the messagesusing the model.

3.4 UML DIAGRAMS

The **unified modeling language (UML)** is a general-purpose visual modeling language that is intended to provide a standard way to visualize the design of a system.^[1] UML provides a standard notation for many types of diagrams which can be roughly divided into three main groups: behavior diagrams, interaction diagrams, and structure diagrams. The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed at Rational Software in 1994–1995, with further development led by them through 1996.^[1] In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005, UML was also published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) as the **ISO/IEC 19501** standard.^[2] Since then the standard has been periodically revised to cover the latest revision of UML.^[3] In software engineering, most practitioners do not use UML, but instead produce informal hand drawn diagrams; these diagrams, however, often include elements from UML.

3.4.1 Advantages of UML Diagrams

- To establish an explicit coupling between concepts and executable code.
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines UML defines several models for representing systems
 - The class model captures the static structure
 - The state model expresses the dynamic behavior of objects
 - The use case model describes the requirements of the user
 - The interaction model represents the scenarios and messages flows
 - The implementation model shows the work units
 - The deployment model provides details that pertain to process allocation. (fig 3.4)

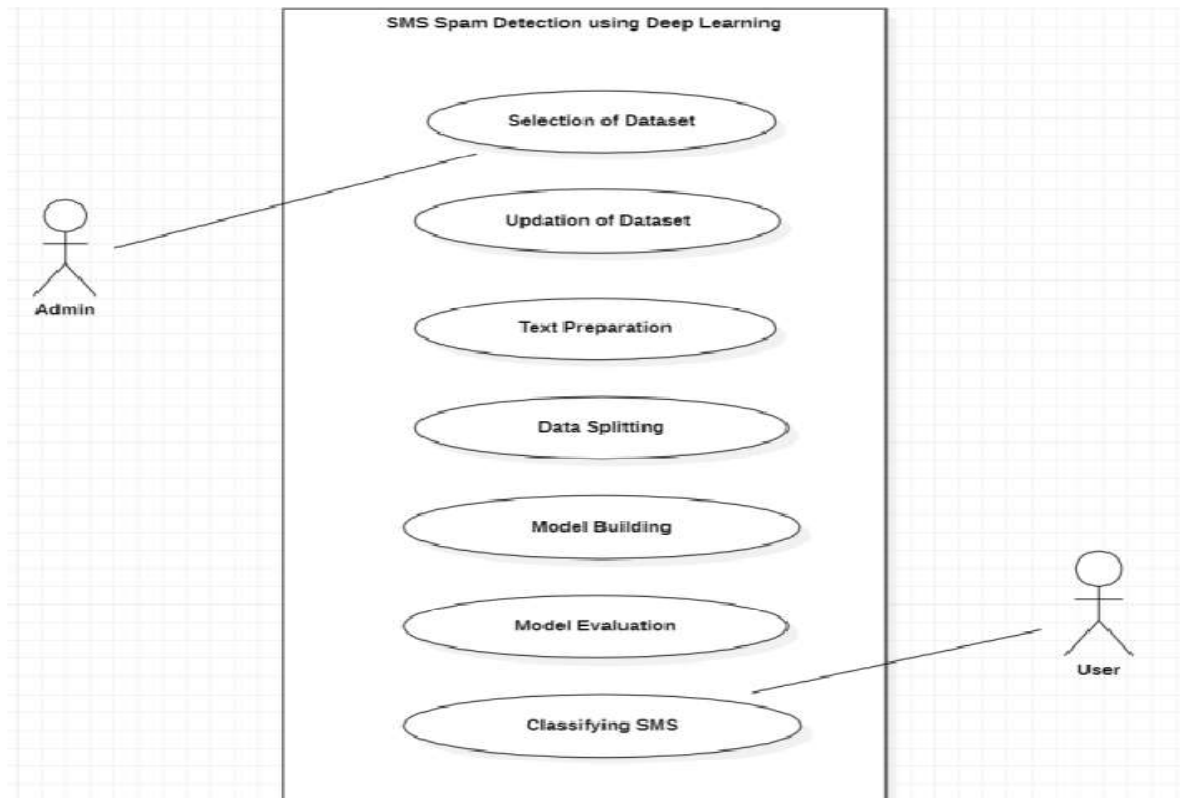


Fig. 3.4 Use Case Diagram

3.5 CLASS DIAGRAM

The use case diagram may be improved with the help of the class diagram, which is also used to define the comprehensive design of the system. The class diagram assigns each of the players that were identified in the use case diagram to one of many classes that are connected to one another. There are two possible ways to describe the connection or link that exists between the classes: "is-a" or "has-a." There is a possibility that each class shown in the class diagram can execute several functions. These built-in capabilities are known as "methods," and they are provided by the class. Aside from that, it's possible for every class to have their own unique "attributes" that help to identify them. (fig 3.5)

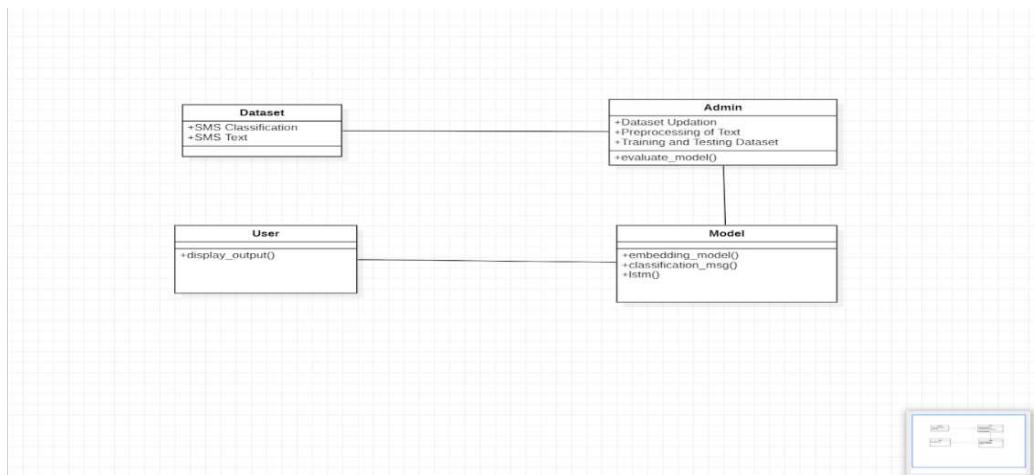


Fig. 3.5 Class Diagram

3.6 OBJECT DIAGRAM

The class diagram is expanded into the object diagram in this way. A class instance is synonymous with an object. This indicates that an object reflects the state that a class was in at a certain instant in time when the system is being executed. The object diagram illustrates the present state of the system's many classes, as well as the ties or linkages between those classes.(fig. 3.6)

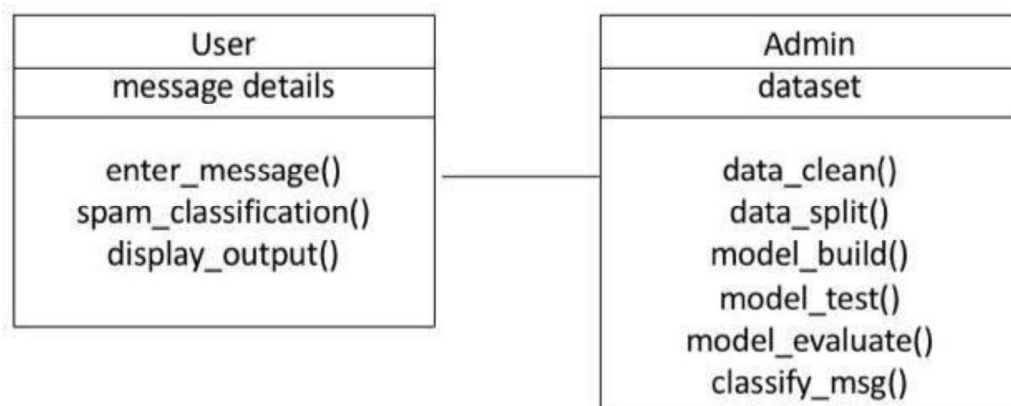


Fig. 3.6 Object Diagram

3.7 ACTIVITY DIAGRAM

The activity diagram presents a representation of the system's processflows. A state diagram is similar to an activity diagram in that it consists

of activities, actions, transitions, beginning and end states, and guard conditions.(fig 3.7)

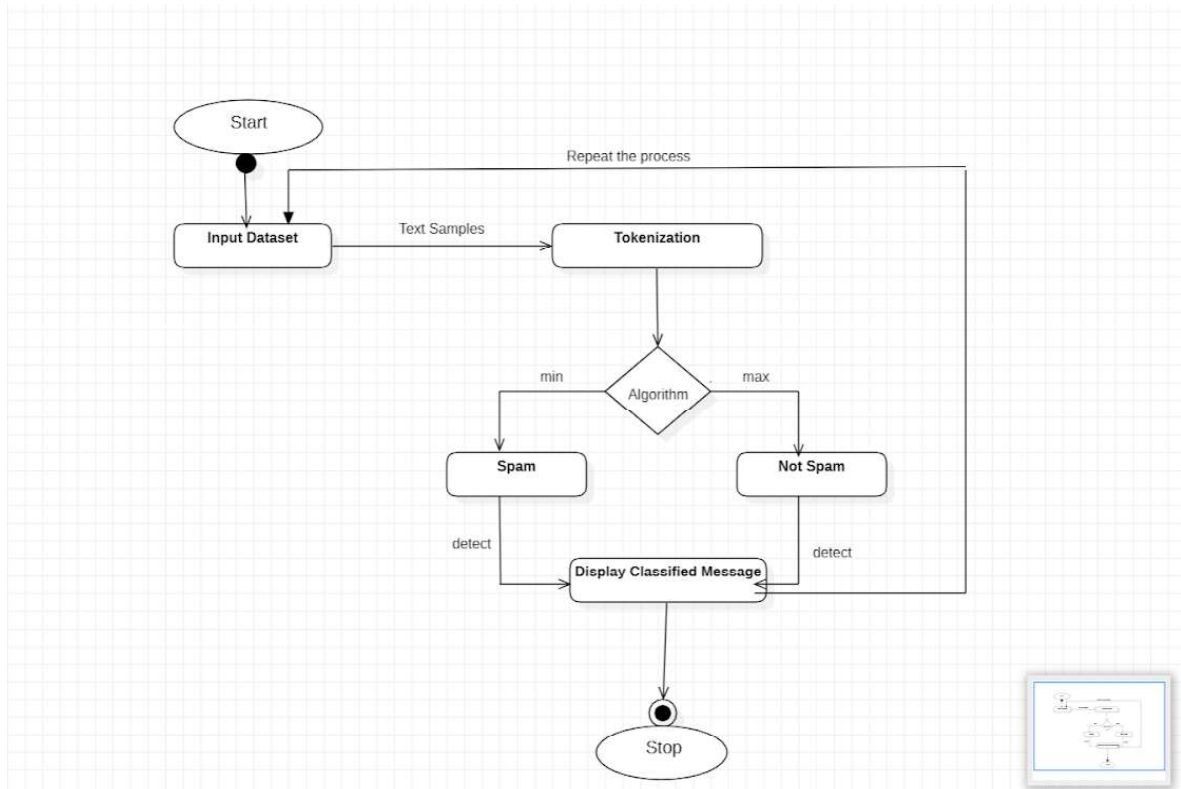


Fig. 3.7 Activity Diagram

3.8 SEQUENCE DIAGRAM

The way in which the various components of the system communicate with one another is shown by a sequence diagram. The fact that a sequence diagram presents information in chronological order is perhaps its most important characteristic. This implies that a step-by-step description is given of the precise order in which the items interact with one another. Throughout the sequence diagram, the various components have conversations with one another via "messaging" one another.(fig 3.8)

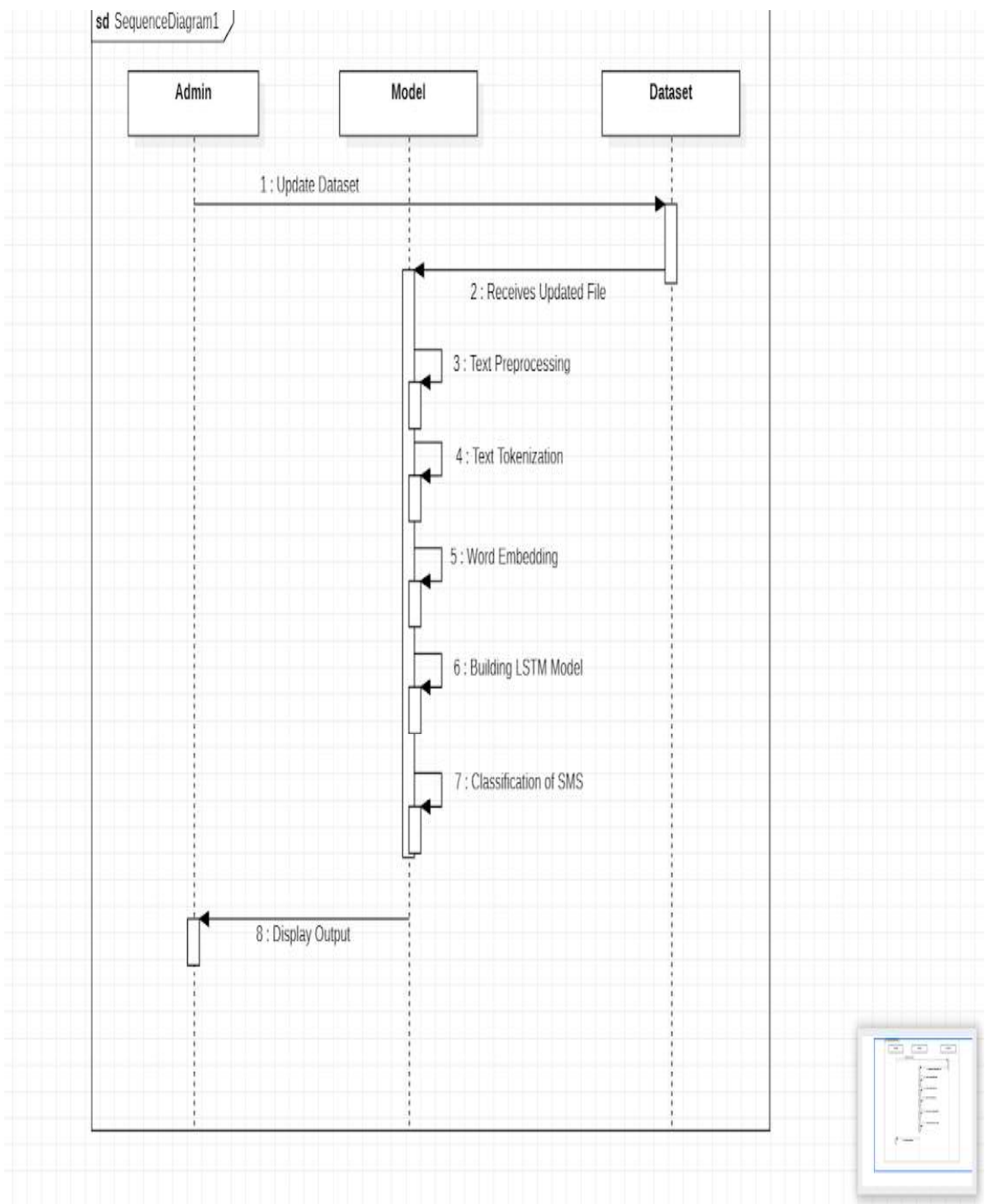


Fig. 3.8 Sequence Diagram

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FEASIBILITY STUDIES OF THE PROJECT

As the name implies, a feasibility analysis is used to determine the viability of a notion, such as ensuring that a project is legally, technically, and commercially feasible. It tells us whether a project is worthwhile—in some cases, completing a project is impossible. There are a number of reasons for this, including the requirement for too many resources, which not only prevents those resources from performing other tasks, but also may cost more than the organisation would make back by taking on a less profitable project. Three significant aspects to consider when doing a feasibility study are as follows:

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility

4.1.1 Economic Feasibility

Both the end user and the creator of this system stand to benefit economically from the implementation of this system. Because the system is based on deep learning and the software is open source and mostly free, the amount of funds required or spent to build this system are very minimal. These types of developed systems are built well within the budget, and this is due to the fact that these types of developed systems are built well within the budget. This is also economically possible for the user, i.e., the consumers, who already uses a smartphone can utilize this feature without any external download of the spam detector. This is not only easy to use and download any third-party app.

4.1.2 Technical Feasibility

This investigation is being carried out in order to assess whether or not the technological feasibility can be met, or more specifically, whether or not the system can meet its own technical requirements. There is little need for

customized software because deep learning techniques involve a little complexity in while they run the program. Furthermore, the majority of the tools are open source and free. Most importantly, we can run this program without any additional software on any machine, making it incredibly portable and versatile.

4.1.3 Operational Feasibility

It is necessary for the system to be able to make spam detection and classification with the maximum level of precision that is humanly possible in order for the consumer to be able to profit from this advantage. Because we have already established that our system is capable of resolving the issues that have arisen as a result of using other systems, the users are also most likely to easily learn how about identifying a spam message. This eliminates the complexity that was previously a problem and makes accessing the system very simple.

4.2 REQUIREMENTS SPECIFICATION

Based on the reference of IEEE Std. 830 – 1998 [19] Recommended Practice for Software Requirements, The Hardware and Software Requirements are described.

4.2.1 Hardware Requirements

The following are the minimum Hardware requirements in order to train and deploy crop recommendation System using Deep Learning:

Processor - Intel® Xeon® CPU E5-2640@2.50GHz or more

RAM - 4 GB or higher

System type - 64-bit Windows 10 Education OS, x64-based processor
Hard

Disk- 500 GB

4.2.2 Software Requirements

Our model is programmed using Python, a popular language of choice for Machine Learning and Deep Learning, owing to its vast support for various Learning tasks by means of leveraging pre-existing packages. We used the following packages in our codebase:

Operating System : Windows 8 and

aboveLanguages : Python

Tools used : Google Collab

4.2.3 Python

Python is a scripting language that is high-level, interpreted, interactive and object-oriented. Python is intended to be extremely readable. It commonly employs English terms rather than punctuation, and it has fewer syntactical structures than other languages.

Python Features

- **Easy-to-learn** - Python has a small number of keywords, a basic structure and a well-defined syntax. This enables the pupil to swiftly learn the language.
- **Easy-to-read** - Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** - Python's source code is relatively simple to maintain.
- **A broad standard library** - The majority of Python's library is portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** - Python features an interactive mode that enables for interactive testing and debugging of code snippets.
- **Portable** - Python can operate on a broad range of hardware devices and has the same interface across them all.
- **Extendable** - The Python interpreter may be extended using low-level modules. These modules allow programmers to enhance or adapt their tools to make them more efficient.
- **Databases** - Python provides interfaces to all major commercial databases.

- **Scalable** - Python provides a better structure and support for large programs than shell scripting.

1. **NumPy**: NumPy is a complete mathematical library that includes features such as random number generators, linear algebra routines, transforms of Fourier series, and more. NumPy is a library for the Python programming language that adds support for huge, multi-dimensional arrays and matrices, in addition to a large number of high-level mathematical functions that can be used to operate on these arrays. NumPy was developed by the Python Software Foundation.

Features:

- The N-dimensional array object with the highest

□ performance. It has resources for combining C/C++

and Fortran programs.

2. **Matplotlib**: Matplotlib is a library for the Python programming language that enables the creation of visualizations that can be either static, animated, or interactive. Matplotlib is a graphing tool for the Python programming language that works in conjunction with NumPy, which is an extension for numerical mathematics in Python.

3. **Pandas**: Pandas is a collection of applications for manipulating and analysing data that was developed specifically for the Python programming language. It has procedures and data structures that allow for the manipulation of numerical tables and time series.

Library features:

- Data Frame object for data manipulation with integrated indexing.

- Instruments for reading and writing data between the various data structures stored in memory and the file formats used by various types of files.

- Transformation of data sets by pivoting and reshaping.

4. **Seaborn**: Seaborn is a Python module for creating statistical visuals.

Features:

- An application programming interface that is focused on datasets, designed for analyzing the connections between many variables.
- Views that are easy to use onto the general organization of complicated datasets.
- Specialized support for using categorical variables to show observations or aggregate statistics.
- Estimation and display of linear regression models that may be done automatically for a variety of various types of dependent variables.

5. Tensor flow: TensorFlow is a library that is used in machine learning and it is an open-source library for numerical computation. It is used for developing machine learning applications and this library was first created by the Google brain team and it is the most common and successfully used library that provides various tools for machine learning applications.

Features:

- It is used as a visualization of graph library for the Python Programming language.
- It is an open-source library for complex analysis and it is easy to build a neural network.
- To handle a large amount of data we can easily use the TensorFlow library. It supports multiple programming languages like Python, c++, and java.
- It is used to display the images, graphs and also deploy the machine learning models. It also works in any environment like Android and IOS.

6. NLTK: NLTK is a string processing library that takes strings as input. The output is in the form of either a string or lists of strings. This library provides a lot of algorithms that helps majorly in the learning purpose. One can compare among different variants of outputs.

Features:

- Part of Speech
- Tagging
- Summarization
- NamedEntity
Recognition
- Sentiment Analysis

4.2.4 Deep Learning

Deep learning is a type of machine learning that uses algorithms meant to function in a manner similar to the human brain. While the original goal for AI was broadly to make machines able to do things that would otherwise require human intelligence the idea has been refined in the decades since. François Chollet, AI researcher at Google and creator of the machine learning software library Keras, says: “Intelligence is not a skill in itself, it’s not about what you can do, but how well and how efficiently you can learn new things.

4.2.5 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to- sequence relate to the field. In this post, you will get insight into LSTMs using the words of research scientists that developed the methods and applied them to new and important problems. There are few that are better at clearly and precisely articulating both the promise of LSTMs and how they work than the experts that developed them.

CHAPTER5

IMPLEMENTATION DETAILS AND RESULTS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

To develop And deploy SMS Spam Prediction using Convolutional Natural Language Processing and Long Short-Term Memory, you can follow the stepsbelow:

- **Data Collection and Pre-processing** - Collect data from various sources and pre-process it to make it suitable for machine learning. You can use various techniques for data pre-processing, such as normalization, data augmentation, and feature extraction.
- **Model Architecture Design** - Design the architecture of the NLP-LSTM model. This architecture should include convolutional layers for feature extraction from the input data, LSTM layers for modeling time-series data, and fully connected layers for classification.
- **Model Training** - Train the NLP-LSTM model using the preprocessed data. You can use various optimization algorithms, such as Adam, RMSprop, or SGD, to minimize the loss function.
- **Model Evaluation** - Evaluate the performance of the trained model using various metrics, such as accuracy, precision, recall, and F1 score. You can also use visualization techniques, such as confusion matrices or ROC curves, to analyze the performance of the model.
- **Deployment** - Deploy the model in a production environment. You can use various frameworks, such as TensorFlow or PyTorch, to create a deployable model. You can also use cloud platforms, such as AWS or Azure, to deploy the model in a scalable and cost-effective way.
- **Continuous Improvement** - Continuously improve the model by retraining it on new data and fine-tuning the hyperparameters. You can also use techniques such as transfer learning to improve the model's performance.

5.2 ALGORITHMS

In SMS spam prediction projects utilizing deep learning, several algorithms and architectures are commonly used to effectively classify messages as either spam or non-spam. Some of the key algorithms include:

5.2.1 Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) that is designed to model sequences of data, such as speech, text, and time series. Unlike traditional RNNs, which suffer from the vanishing gradient problem when modelling long sequences, LSTM uses a gated memory mechanism to store and retrieve information over time. Each LSTM cell has three gates - input gate, forget gate, and output gate - that control the flow of information into, out of, and within the cell. The input gate decides which information to update, the forget gate decides which information to discard, and the output gate decides which information to output.

5.2.2 Attention Mechanisms

Attention mechanisms have been incorporated into deep learning models for SMS spam prediction to dynamically focus on relevant parts of the input text. By assigning different weights to different parts of the input sequence, attention mechanisms allow the model to selectively attend to the most informative tokens, potentially improving classification accuracy.

5.2.3 Word Embedding

Word embeddings represent words as dense vectors in a continuous vector space, where similar words have similar representations. Pre-trained word embeddings like Word2Vec, GloVe, or FastText can be used to convert text data into numerical vectors, which can then be fed into deep learning models.

5.3 IMPLEMENTATION

5.3.1 Data Collection and Preprocessing

Gather a dataset of SMS messages labeled as spam or ham (non-spam). Preprocess the data by tokenizing the messages, removing stop words, and possibly stemming or lemmatizing the words. Split the dataset into training, validation, and test sets.

5.3.2 Evaluation

- Evaluate the trained model using the test set to assess its performance metrics, such as accuracy, precision, recall, and F1 score.
- Conduct error analysis to identify patterns of misclassifications and areas for improvement

5.3.3 Model Building:

A Sequential Model is built which consists of 3 important layers. They are Embedding Layer followed by the LSTM Layer and then concludes with the Dense Output Layer. Then the training data is fit onto the model created and validated by using the validation data.

5.3.4 Monitoring and Maintenance

- Set up monitoring systems to track the model's performance and detect any drift in the data distribution or model behavior.
- Implement mechanisms for retraining the model periodically with new data to adapt to changing trends in spam messages.
- Handle model updates and versioning to maintain compatibility with the deployment environment and APIs.
- Continuously evaluate and refine the deployment pipeline to improve efficiency and reliability.

5.4 RESULTS

We get an accuracy of 98.11% from the trained LSTM Model. The model predicts pretty well on the test data as evident from the accuracy score. The user can type the message that he wants to check for spam detection

5.5 DATA ANALYSIS AND INTERPRETATION

We used a trainval data set which contains both the spam and ham messages.

```
message1="Hey how are you? It was nice meeting you. See you soon"
```

```
import nltk
nltk.download('punkt')
classify_message(model,message1)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
1/1 [=====] - 0s 31ms/step
It is not a spam
```

Fig. 5.1 Image showing that the message is not spam

```
message2="Hello! Pls approach us for urgent need for work. Call on:973894938"
```

```
classify_message(model,message2)
```

```
1/1 [=====] - 0s 27ms/step
It is a spam
```

Fig 5.2 Image showing that the message is spam

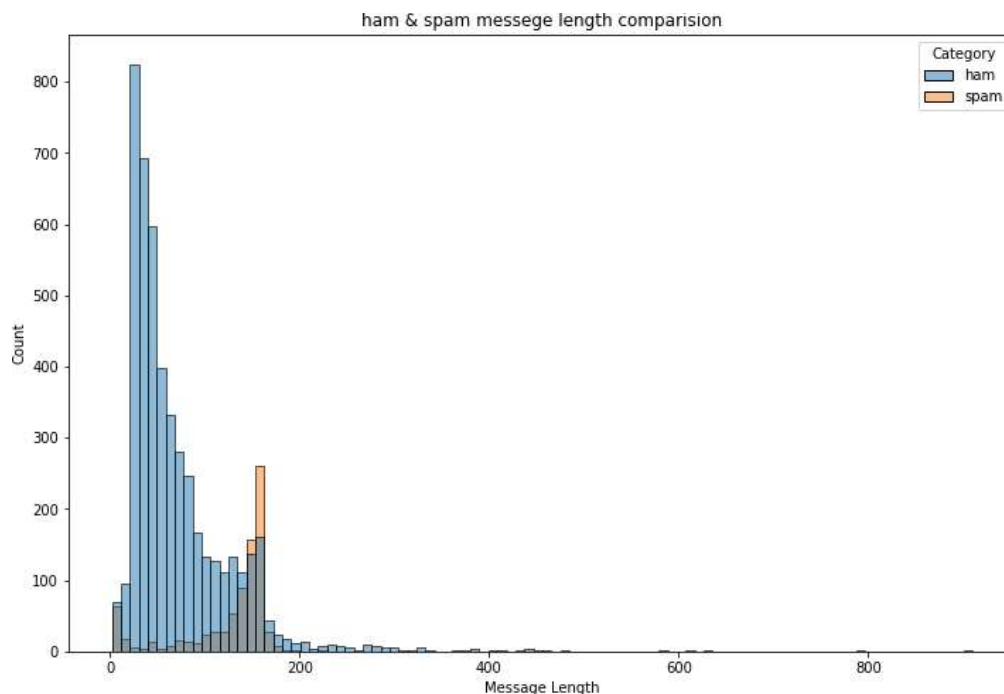


Fig 5.3 Display the description of length of ham and spam messages seperately on an individual series.

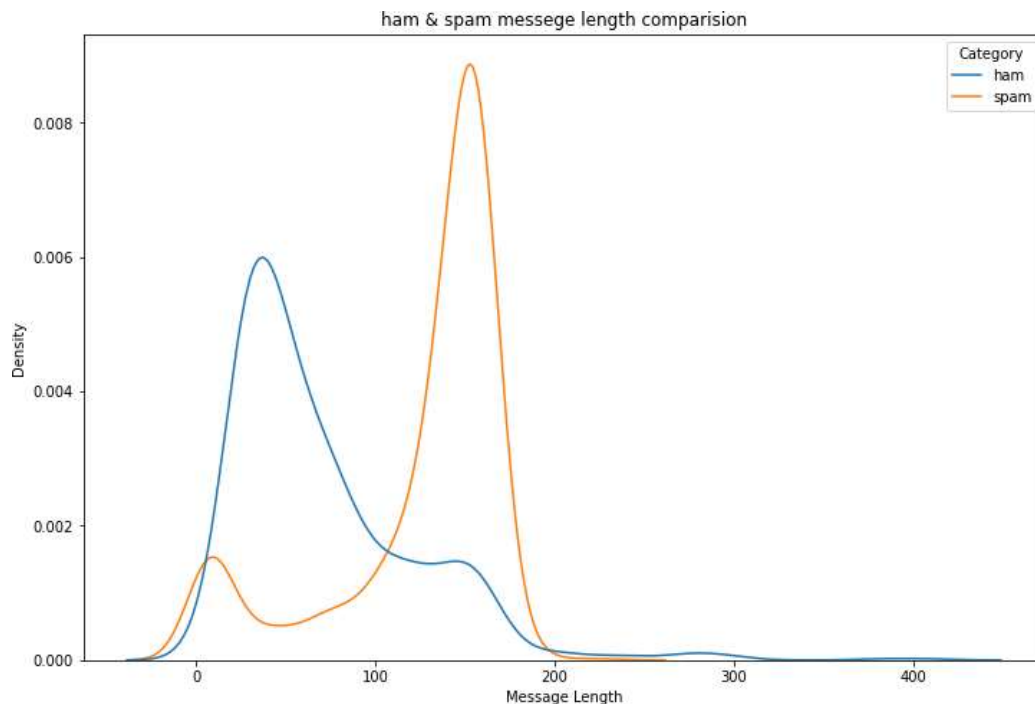


Fig 5.4 Length comparison between Spam and Ham

```
Epoch 1/10
38/38 [=====] - 16s 354ms/step - loss: 0.7451 - accuracy: 0.6958 - val_loss: 0.4723 - val_accuracy: 0.7925
Epoch 2/10
38/38 [=====] - 15s 407ms/step - loss: 0.3816 - accuracy: 0.8500 - val_loss: 0.3113 - val_accuracy: 0.8774
Epoch 3/10
38/38 [=====] - 13s 333ms/step - loss: 0.2056 - accuracy: 0.9308 - val_loss: 0.2265 - val_accuracy: 0.9104
Epoch 4/10
38/38 [=====] - 13s 341ms/step - loss: 0.1209 - accuracy: 0.9583 - val_loss: 0.1975 - val_accuracy: 0.9387
Epoch 5/10
38/38 [=====] - 13s 352ms/step - loss: 0.0848 - accuracy: 0.9675 - val_loss: 0.1931 - val_accuracy: 0.9245
Epoch 6/10
38/38 [=====] - 13s 338ms/step - loss: 0.0571 - accuracy: 0.9925 - val_loss: 0.1843 - val_accuracy: 0.9292
Epoch 7/10
38/38 [=====] - 13s 334ms/step - loss: 0.0392 - accuracy: 0.9942 - val_loss: 0.1836 - val_accuracy: 0.9434
Epoch 8/10
38/38 [=====] - 13s 337ms/step - loss: 0.0283 - accuracy: 0.9958 - val_loss: 0.1870 - val_accuracy: 0.9528
Epoch 9/10
38/38 [=====] - 13s 332ms/step - loss: 0.0213 - accuracy: 0.9983 - val_loss: 0.1887 - val_accuracy: 0.9528
Epoch 10/10
38/38 [=====] - 13s 332ms/step - loss: 0.0158 - accuracy: 0.9992 - val_loss: 0.1942 - val_accuracy: 0.9528
<keras.callbacks.History at 0x7fbc27b041d0>
```

Fig 5.5 Accuracy of LSTM Model

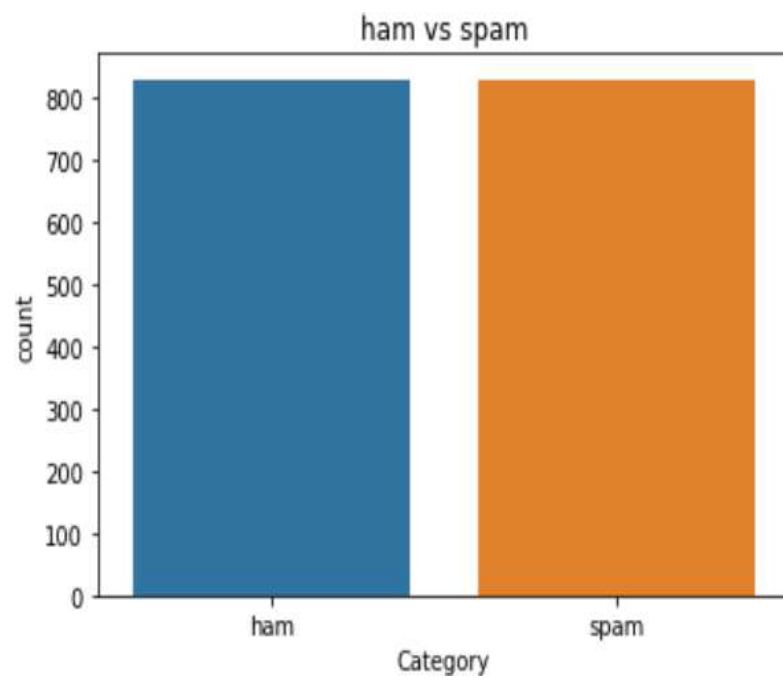


Fig. 5.6 Length comparison between Spam and Ham

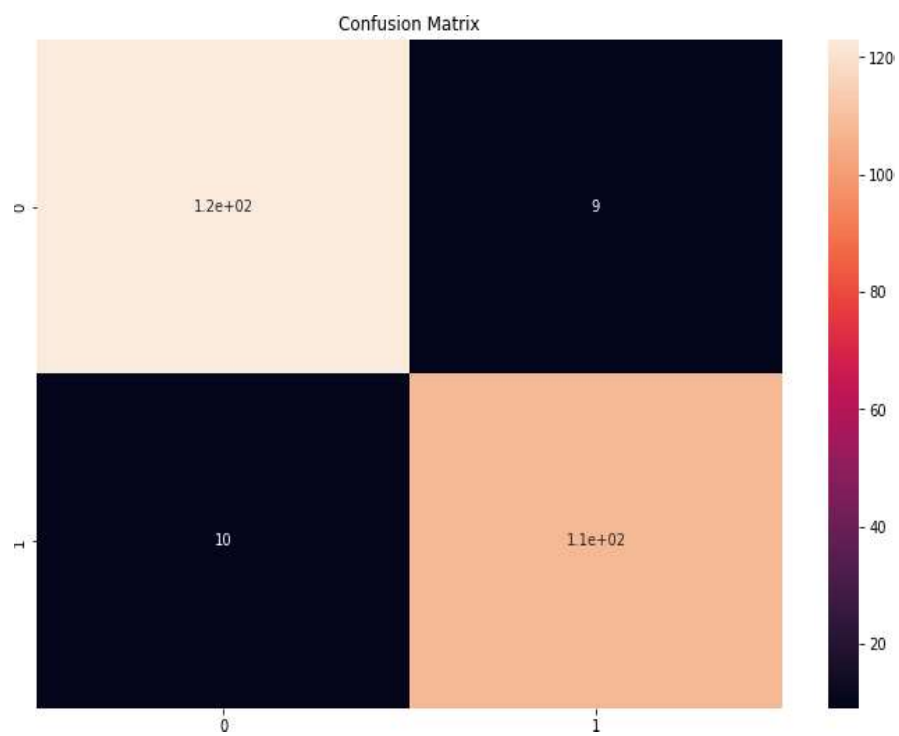


Fig. 5.7 Confusion Matrix

CHAPTER 6

SUMMARY AND CONCLUSIONS

The SMS Spam problem is increasing nowadays with the increase in the use of text messaging. SMS Spam filtering is the big challenge these days. We have seen that deep learning algorithms like Long Short Term Memory play a vital role in identifying SMS Spam. This classifier reduces the overall error by more than half. Adding meaningful features such as the length of messages in number of characters, adding certain thresholds for the length, and analysing the learning curves and misclassified data have been the factors that contributed to this improvement in results.

In future work we can elaborate this topic to prediction by using content and context data like Host address of the SMS, sender, number of times received, URL's in the messages etc. We can also automate the process of dataset updation and also propose hybrid algorithms in the future to achieve in more accuracy.

6.1 CONCLUSION

In conclusion, SMS spam detection using deep learning represents a significant advancement in ensuring user privacy, security, and overall messaging experience. The current state of the art in deep learning techniques has enabled highly accurate detection of spam messages, offering users protection against unwanted and potentially harmful content. As we look to the future, there is immense potential for further improvement in accuracy, real-time detection, and integration of advanced analysis techniques. By harnessing the power of deep learning, researchers and developers can continue to innovate, providing users with ever-improving methods to safeguard against spam messages.

Ultimately, the ongoing development of SMS spam detection using deep learning promises a future where users can communicate freely and securely, without the disruption of unwanted messages, thus enhancing the overall quality of digital communication experiences. The current landscape showcases the effectiveness of deep learning techniques, which have led to highly accurate detection rates and adaptability to evolving spamming strategies. However, the journey doesn't end here; it merely signifies the beginning of an ongoing quest for improvement and

innovation.

6.2 RECOMMENDATIONS FOR FUTURE RESEARCH

- SMS spam detection using deep learning has emerged as a critical technology in safeguarding users from unwanted and potentially harmful messages. This essay explores the current advancements in this field and outlines the promising future scope for further development.
- In recent years, deep learning techniques have revolutionized the way spam detection is approached. With the ability to automatically extract relevant features from vast amounts of data, deep learning models have shown remarkable accuracy in identifying and filtering out spam messages. These models leverage neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to process text data and detect patterns indicative of spam.
- One of the key advantages of deep learning-based spam detection is its adaptability to evolving spamming techniques. As spammers employ increasingly sophisticated methods to evade detection, deep learning models can continuously learn and adapt to new patterns, ensuring that users remain protected. Moreover, these models can be trained on large datasets of labeled spam and non-spam messages, allowing them to generalize well to diverse types of spam content.

6.3 FUTURE SCOPE

- Improved Accuracy: Continuously enhancing the accuracy of spam detection models is a primary objective. Future research can focus on refining LSTM architectures, experimenting with different hyperparameters, and integrating advanced NLP techniques to better understand the context and semantics of text messages.
- Real-time Detection: Advancements in deep learning frameworks and hardware can facilitate the deployment of real-time SMS spam detection systems. Such systems can

analyze messages instantly as they arrive, providing users with immediate protection against spam.

- **Multimodal Spam Detection:** Incorporating multimodal data sources, such as images and audio, along with text, can enhance spam detection capabilities. Integrating techniques like convolutional neural networks (CNNs) for image processing and recurrent neural networks (RNNs) for audio analysis can create more robust spam detection systems.
- **Personalized Filtering:** Tailoring spam detection models to individual users' preferences and behaviors can further improve accuracy. By leveraging user feedback and interaction data, personalized models can adapt to the specific spam patterns encountered by each user, minimizing false positives and negatives.
- **Cross-lingual Spam Detection:** Extending spam detection models to handle multiple languages can address the global nature of spam. Multilingual NLP models, coupled with LSTM networks, can enable effective spam detection across diverse linguistic contexts.
- **Integration with Communication Platforms:** Integrating SMS spam detection models directly into communication platforms and mobile devices can offer seamless protection to users. By partnering with telecom companies or messaging app developers, these models can be deployed at scale, benefiting millions of users worldwide.

6.4 SUMMARY

SMS spam detection using deep learning has become essential in safeguarding users from unwanted and potentially harmful messages. Current advancements in deep learning techniques, such as neural network architectures, have enabled highly accurate detection of spam messages.

These models adapt to evolving spamming techniques, enhance user experience, and contribute to network efficiency. Looking ahead, the future scope includes improving accuracy, real-time detection, multimodal analysis, privacy-preserving techniques, behavioral analysis, and cross-platform integration. The future of SMS spam detection using deep learning promises even more effective and privacy-conscious methods to protect users from unwanted messages. Current advancements include highly accurate detection through neural networks, adaptability to evolving spamming techniques, and contributions to user experience and network efficiency.

Looking forward, opportunities for improvement lie in enhancing accuracy, real-time detection, multimodal analysis, privacy preservation, behavioral analysis, and cross-platform integration. These advancements promise a future where users can enjoy a safer and more personalized messaging experience, free from the intrusion of spam messages, facilitated by innovative deep learning techniques.

REFERENCES

1. K. Yadav, S. K. Saha, P. Kumaraguru, and R. Kumra, "take control of your SMSs: designing a usable spam sms filtering system," in 2012 IEEE 13th International Conference on Mobile Data Management. IEEE, 2012, pp. 352–355.
2. S. J. Warade, P. A. Tijare, and S. N. Sawalkar, "An approach for sms spam detection."
3. J. W. Yoon, H. Kim, and J. H. Huh, "Hybrid spam filtering for mobile communication," *computers & security*, vol. 29, no. 4, pp. 446–459, 2010.
4. Pavas Navaney, Gaurav Dubey, Ajay Rana, "SMS Spam Filtering using Supervised Machine Learning Algorithms" in IEEE 2018.
5. Jindal Nitin, LIU Bing. Review spam detection[C] // Proceedings of the 16th international conference on World Wide Web. Canada. New York: ACM Press, 2007: 1189- 1190.
6. N. Jindal and B. Liu, "Review spam detection," in *Proc. of the 16th International Conference on World Wide Web*, 2007, pp. 1189-1190.
7. T. A. Almeida, J. M. G. Hidalgo, and T. P. Silva, "Towards SMS spam filtering: Results under a new dataset," *International Journal of Information Security Science*, vol. 2, No. 1, pp. 1-18, 2013.
8. J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85-117, 2015.
9. Tiago A. Almeida, Jos Mara G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of SMS spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering (DocEng '11)*. ACM, New York, NY, USA.
10. T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering," in *Proceedings of the 11th ACM symposium on Document engineering - DocEng '11*, 2011.
11. Chaudhari N., Jayvala, Vinitashah, "Survey on Spam SMS filtering using Data Mining Techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 5, Issue 11, November 2016

12. Anchal, Sharma A., "SMS Spam Detection Using Neural Network Classifier," International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 6, June 2014.
13. J. W. Yoon, H. Kim, and J. H. Huh, "Hybrid spam filtering for mobile communication," computers & security, vol. 29, no. 4, pp. 446–459, 2010.
14. S. Onashoga, O. O. Abayomi-Alli, A. S. Sodiya, and D. A.Ojo "An adaptive and collaborative server side sms spam filtering scheme using artificial immune system," Information Security Journal: A Global Perspective, vol. 24, no. 4- 6, pp. 133–145, 2015

APPENDIX

A.SOURCECODE

```
import numpy as np
import pandas as pd
import os
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
data=pd.read_csv("spam.csv",encoding="latin")
data.head()
from google.colab import drive
drive.mount('/content/drive')
data.columns
data=data.drop(columns=["Unnamed: 2","Unnamed: 3","Unnamed: 4"])
data=data.rename(
{
    "v1":"Category",
    "v2":"Message"
},
axis=1
)
data.head()
data.isnull().sum()
data.info()
data["Message Length"]=data["Message"].apply(len)
fig=plt.figure(figsize=(12,8))
sns.histplot(
    x=data["Message Length"],
    hue=data["Category"]
)
plt.title("ham & spam messege length comparision")
plt.show()
ham_desc=data[data["Category"]=="ham"]["Message Length"].describe()
spam_desc=data[data["Category"]=="spam"]["Message Length"].describe()

print("Ham Messege Length Description:\n",ham_desc)
print("*****")
print("Spam Message Length Description:\n",spam_desc)
data.describe(include="all")
data["Category"].value_counts()
sns.countplot(
    data=data,
    x="Category"
)
plt.title("ham vs spam")
plt.show()
```

```

ham_count=data["Category"].value_counts()[0]
spam_count=data["Category"].value_counts()[1]

total_count=data.shape[0]

print("Ham contains:{:.2f}% of total data.".format(ham_count/total_count*100))
print("Spam contains:{:.2f}% of total data.".format(spam_count/total_count*100))
#compute the length of majority & minority class
minority_len=len(data[data["Category"]=="spam"])
majority_len=len(data[data["Category"]=="ham"])

#store the indices of majority and minority class
minority_indices=data[data["Category"]=="spam"].index
majority_indices=data[data["Category"]=="ham"].index

#generate new majority indices from the total majority_indices
#with size equal to minority class length so we obtain equivalent number of indices length
random_majority_indices=np.random.choice(
    majority_indices,
    size=minority_len,
    replace=False
)

#concatenate the two indices to obtain indices of new dataframe
undersampled_indices=np.concatenate([minority_indices,random_majority_indices])

#create df using new indices
df=data.loc[undersampled_indices]

#shuffle the sample
df=df.sample(frac=1)

#reset the index as its all mixed
df=df.reset_index()

#drop the older index
df=df.drop(
    columns=["index"],
)
df.shape
df["Category"].value_counts()
sns.countplot(
    data=df,
    x="Category"
)
plt.title("ham vs spam")
plt.show()
df.head()
df["Label"]=df["Category"].map(
    {
        "ham":0,
        "spam":1
    }
)

```

```

    }
)
df.head()
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

stemmer=PorterStemmer()
#declare empty list to store tokenized message
corpus=[]

import nltk
nltk.download('stopwords')

#iterate through the df["Message"]
for message in df["Message"]:

    #replace every special characters, numbers etc.. with whitespace of message
    #It will help retain only letter/alphabets
    message=re.sub("[^a-zA-Z]", " ",message)

    #convert every letters to its lowercase
    message=message.lower()

    #split the word into individual word list
    message=message.split()

    #perform stemming using PorterStemmer for all non-english-stopwords
    message=[stemmer.stem(words)
              for words in message
              if words not in set(stopwords.words("english"))
            ]
    #join the word lists with the whitespace
    message=" ".join(message)

    #append the message in corpus list
    corpus.append(message)
from tensorflow.keras.preprocessing.text import one_hot
vocab_size=10000

oneHot_doc=[one_hot(words,n=vocab_size)
             for words in corpus
            ]
df["Message Length"].describe()
fig=plt.figure(figsize=(12,8))
sns.kdeplot(
    x=df["Message Length"],
    hue=df["Category"]
)
plt.title("ham & spam messege length comparision")
plt.show()

```

```

from tensorflow.keras.preprocessing.sequence import pad_sequences
sentence_len=200
embedded_doc=pad_sequences(
    oneHot_doc,
    maxlen=sentence_len,
    padding="pre"
)
extract_features=pd.DataFrame(
    data=embedded_doc
)
target=df["Label"]
df_final=pd.concat([extract_features,target],axis=1)
df_final.head()
X=df_final.drop("Label",axis=1)
y=df_final["Label"]
from sklearn.model_selection import train_test_split
from sklearn.model_selection import train_test_split
X_train,X_val,y_train,y_val=train_test_split(
    X_trainval,
    y_trainval,
    random_state=42,
    test_size=0.15
)
from tensorflow.keras.layers import LSTM
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Embedding
from tensorflow.keras.models import Sequential
model=Sequential()
feature_num=100
model.add(
    Embedding(
        input_dim=vocab_size,
        output_dim=feature_num,
        input_length=sentence_len
    )
)
model.add(
    LSTM(
        units=128
    )
)

model.add(
    Dense(
        units=1,
        activation="sigmoid"
    )
)
from tensorflow.keras.optimizers import Adam
model.compile(
    optimizer=Adam(
        learning_rate=0.001
    )
)

```

```

    ),
    loss="binary_crossentropy",
    metrics=["accuracy"]
)
model.fit(
    X_train,
    y_train,
    validation_data=(
        X_val,
        y_val
    ),
    epochs=10
)
y_pred=model.predict(X_test)
y_pred=(y_pred>0.5)
from sklearn.metrics import accuracy_score,confusion_matrix
score=accuracy_score(y_test,y_pred)
print("Test Score:{:.2f}%".format(score*100))
cm=confusion_matrix(y_test,y_pred)
fig=plt.figure(figsize=(12,8))
sns.heatmap(
    cm,
    annot=True,
)
plt.title("Confusion Matrix")
cm
#The function take model and message as parameter
def classify_message(model,message):

    #We will treat message as a paragraphs containing multiple sentences(lines)
    #we will extract individual lines
    for sentences in message:
        sentences=nlk.sent_tokenize(message)

        #Iterate over individual sentences
        for sentence in sentences:
            #replace all special characters
            words=re.sub("[^a-zA-Z]", " ",sentence)

            #perform word tokenization of all non-english-stopwords
            if words not in set(stopwords.words('english')):
                word=nlk.word_tokenize(words)
                word=" ".join(word)

            #perform one_hot on tokenized word
            oneHot=[one_hot(word,n=vocab_size)]

            #create an embedded documnet using pad_sequences
            #this can be fed to our model
            text=pad_sequences(oneHot,maxlen=sentence_len,padding="pre")

            #predict the text using model

```

```
predict=model.predict(text)

#if predict value is greater than 0.5 its a spam
if predict>0.5:
    print("It is a spam")
#else the message is not a spam
else:
    print("It is not a spam")
import nltk
nltk.download('punkt')
classify_message(model,message1)
```

B. SCREENSHOTS

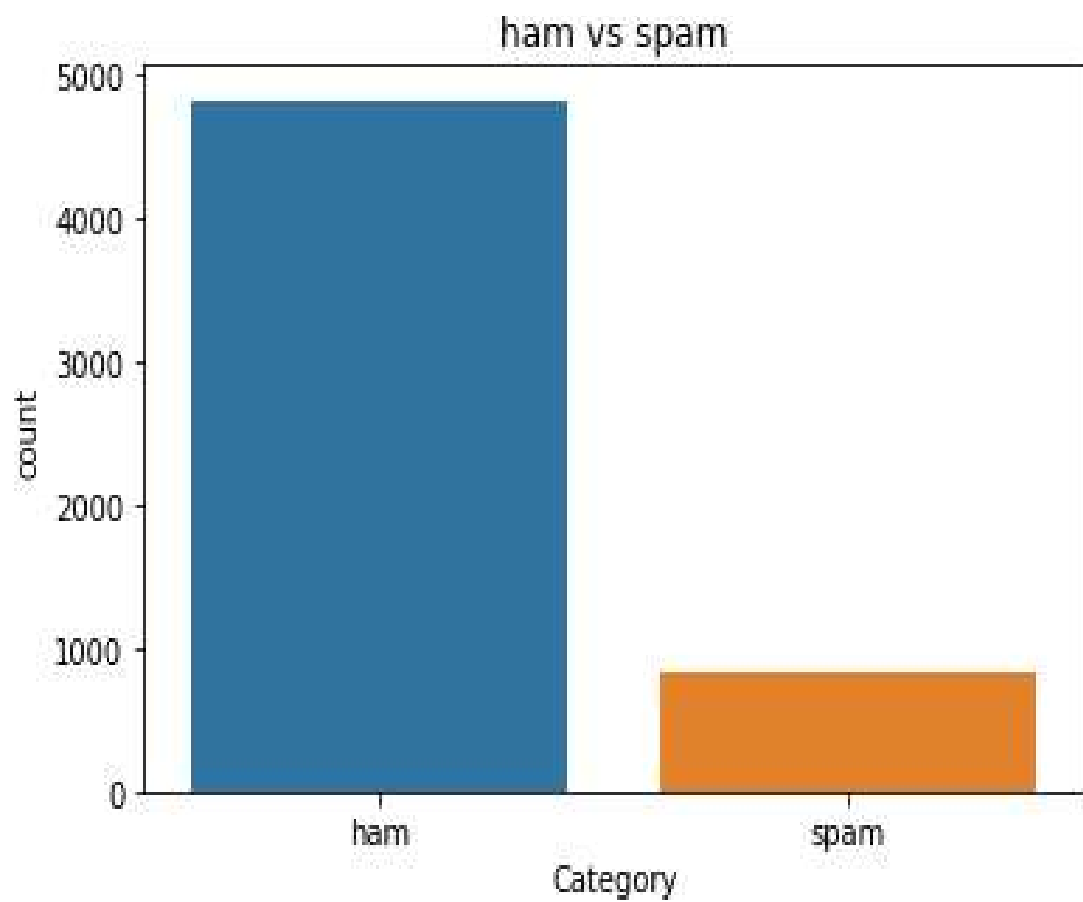
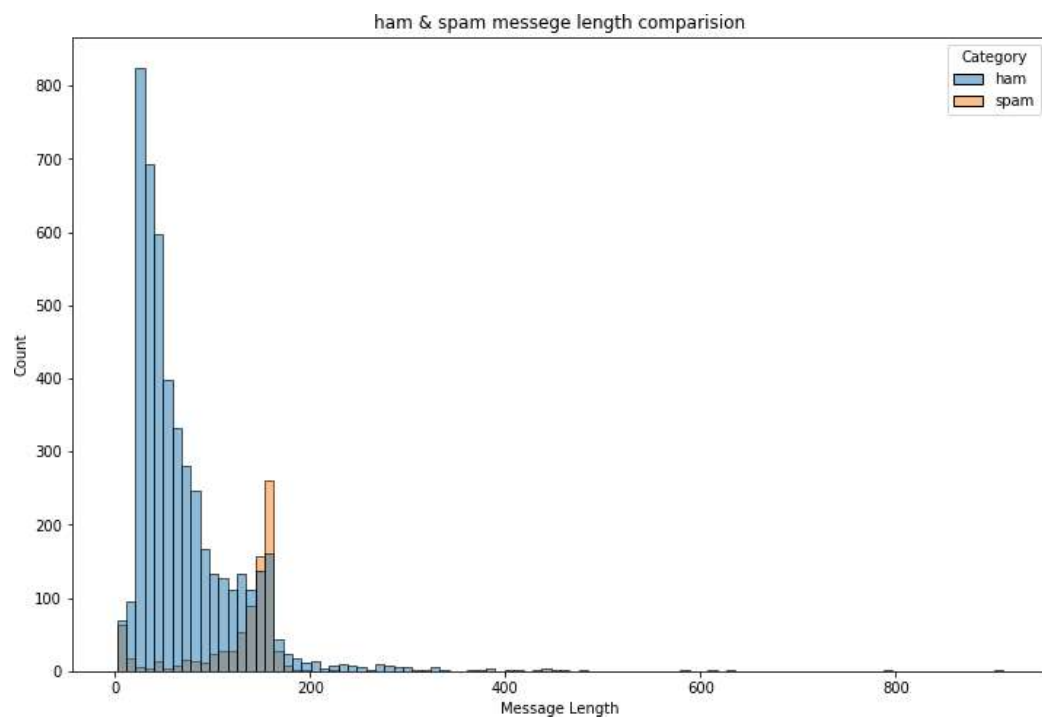
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

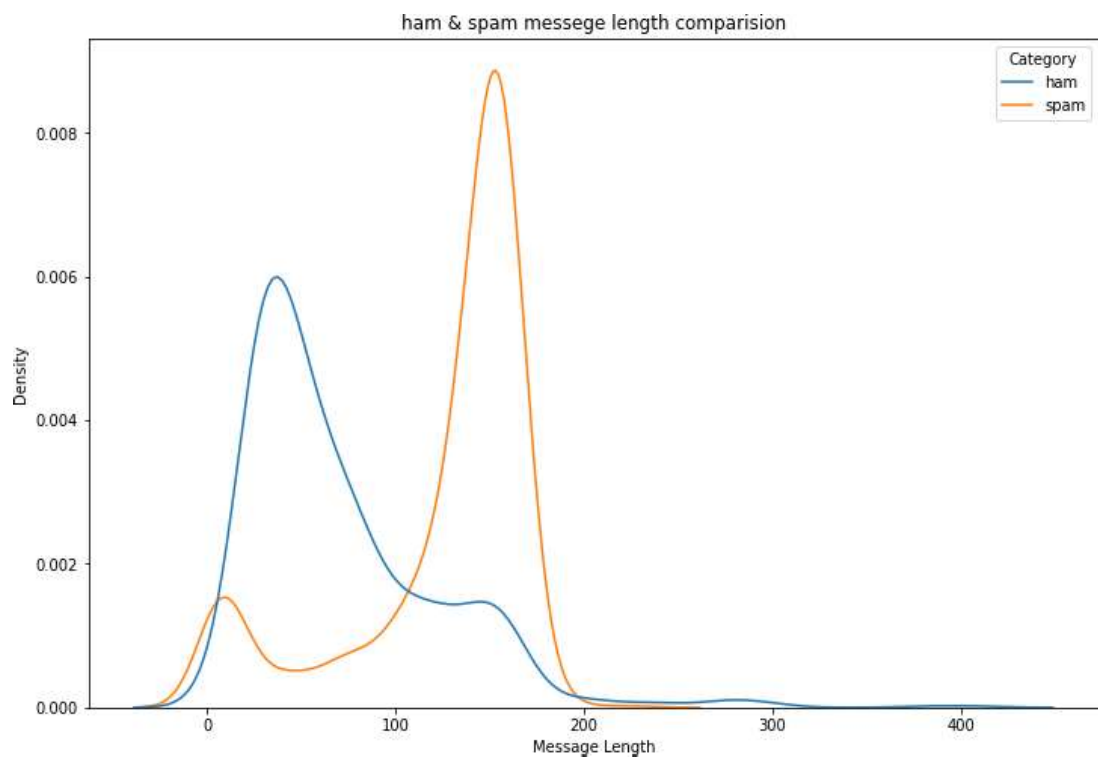
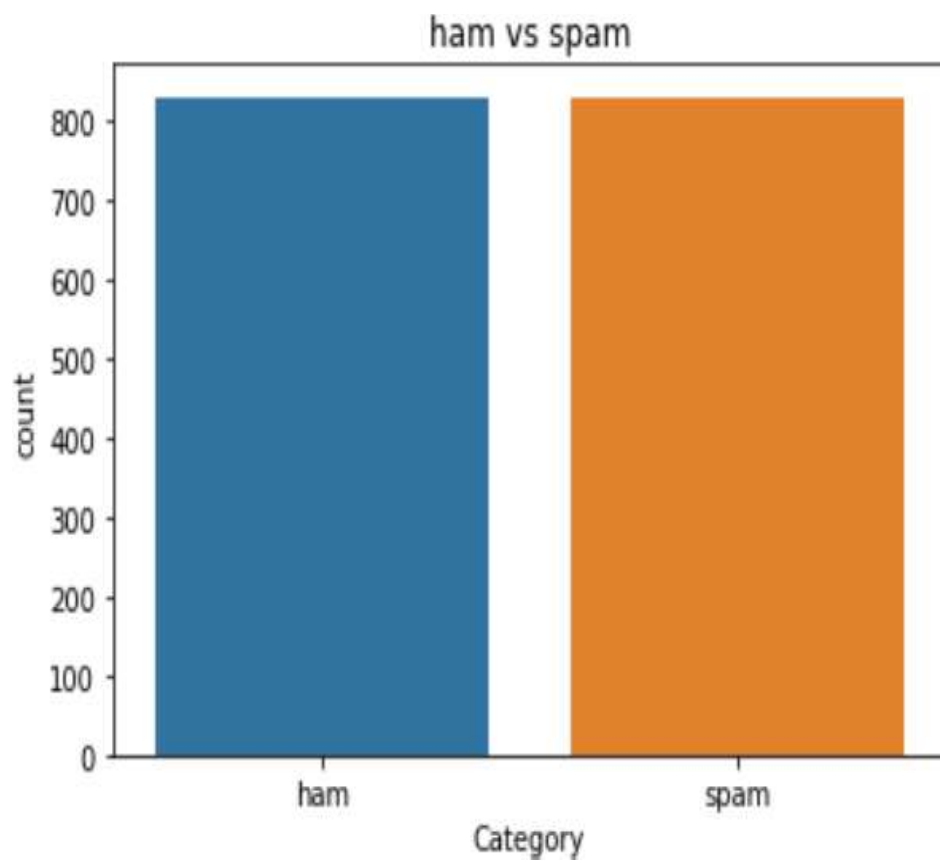
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

↗

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5657 entries, 0 to 5656
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Category    5657 non-null   object
1   Message     5657 non-null   object
dtypes: object(2)
memory usage: 88.5+ KB
```

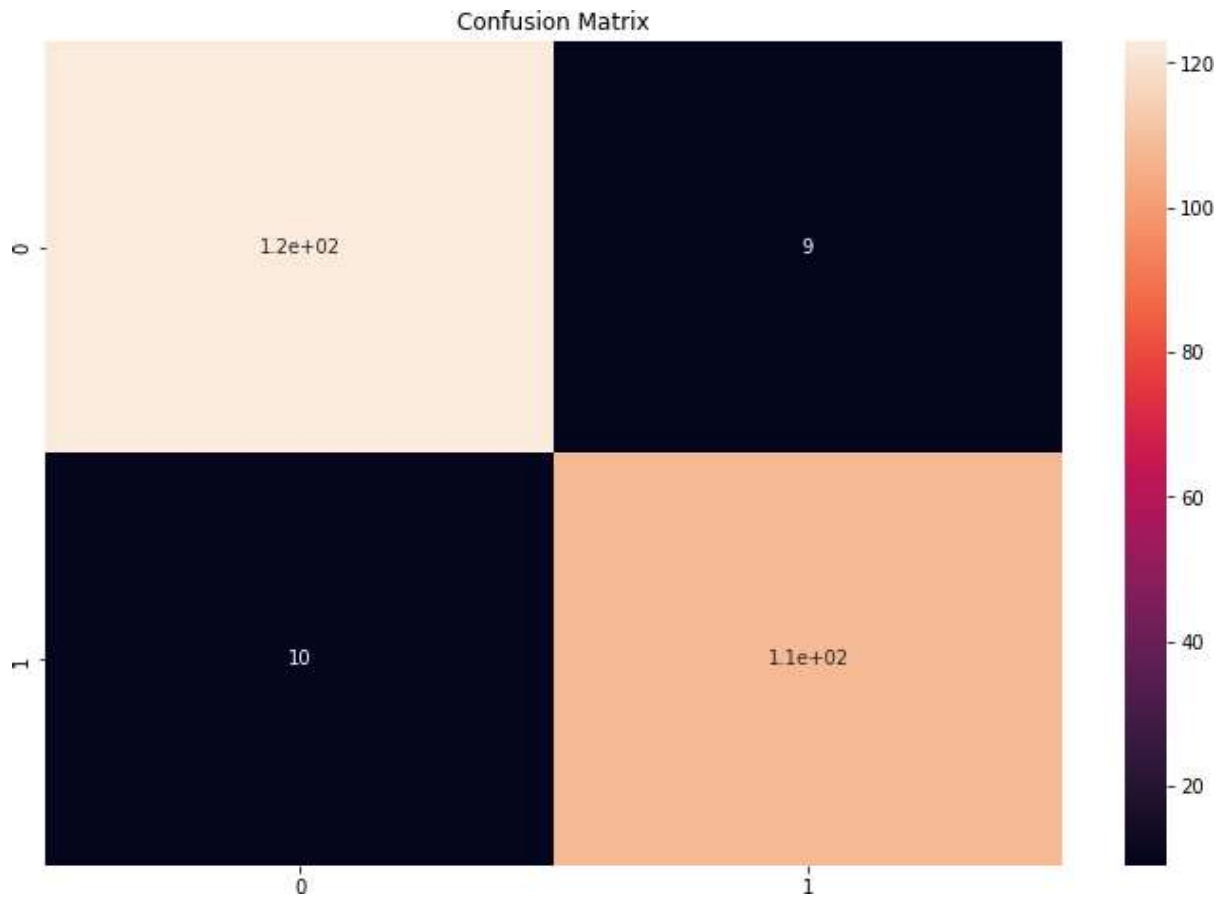




```

Epoch 1/10
38/38 [=====] - 16s 354ms/step - loss: 0.7451 - accuracy: 0.6958 - val_loss: 0.4723 - val_accuracy: 0.7925
Epoch 2/10
38/38 [=====] - 15s 407ms/step - loss: 0.3816 - accuracy: 0.8500 - val_loss: 0.3113 - val_accuracy: 0.8774
Epoch 3/10
38/38 [=====] - 13s 333ms/step - loss: 0.2056 - accuracy: 0.9308 - val_loss: 0.2265 - val_accuracy: 0.9104
Epoch 4/10
38/38 [=====] - 13s 341ms/step - loss: 0.1209 - accuracy: 0.9583 - val_loss: 0.1975 - val_accuracy: 0.9387
Epoch 5/10
38/38 [=====] - 13s 352ms/step - loss: 0.0848 - accuracy: 0.9675 - val_loss: 0.1931 - val_accuracy: 0.9245
Epoch 6/10
38/38 [=====] - 13s 338ms/step - loss: 0.0571 - accuracy: 0.9925 - val_loss: 0.1843 - val_accuracy: 0.9292
Epoch 7/10
38/38 [=====] - 13s 334ms/step - loss: 0.0392 - accuracy: 0.9942 - val_loss: 0.1836 - val_accuracy: 0.9434
Epoch 8/10
38/38 [=====] - 13s 337ms/step - loss: 0.0283 - accuracy: 0.9958 - val_loss: 0.1870 - val_accuracy: 0.9528
Epoch 9/10
38/38 [=====] - 13s 332ms/step - loss: 0.0213 - accuracy: 0.9983 - val_loss: 0.1887 - val_accuracy: 0.9528
Epoch 10/10
38/38 [=====] - 13s 332ms/step - loss: 0.0158 - accuracy: 0.9992 - val_loss: 0.1942 - val_accuracy: 0.9528
<keras.callbacks.History at 0x7fbe27b041d0>

```



C. RESEARCH PAPER

SMS SPAM DETECTION USING DEEP LEARNING

D. Devi. M.Nikhith. M. Sai Charan

Professor, Student

Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology
devi.cse@satyabama.ac.in, suryamuppasani@gmail.com saicharanmupparaju9884@gmail.com

Abstract- In state-of-the-art world wherein digitization is anywhere, SMS has become one of the main sorts of communication, not like other chat structures based on messages like Facebook, What app, and so forth. There's no active demand for SMS. We all recognize that hackers/spammer are trying to infect cellular computing devices and SMS assist of cell gadgets has come to be vulnerable because the attacker is making an attempt to infiltrate the pc and send an unsolicited link, click on that page. May be accessed remotely. , get right of entry to through a cell computing device. The authors consequently advanced a machine to discover such malicious messages and determine whether the message is unsolicited mail or HAM (malicious or non-malicious). The authors advanced a dictionary using the TF-IDF victimizer algorithm, wherein all capabilities of junk mail SMS containing words are primarily based at the content of the message and by means of gaining access to this dictionary, the machine classifies the SMS as junk mail or unwanted facts. No.

Keywords: cell computing devices, SMS aid, junk mail detection.

INTRODUCTION

The quantity of cellphone (cellular smartphone) customers will increase from 1 billion to a few.8 billion in 5 years [1]. The pinnacle 3 international locations the usage of complementary cell phones are China, India and the United States of America. Short Message Service or SMS text messaging is a medium that has been around for decades. The SMS provider also can be used without a community. Therefore, the SMS provider is to be had on smartphones and normal cell telephones. Although there are numerous applications to be had for text

messaging on smartphones, consisting of what's App, this provider can be preferred over ordinary networks. But SMS may be used at any time. As a result, site visitors with SMS website services is increasing every day. Spammer A person / agency this is responsible for spam messages. Spammers send massive sums of cash for company or private gain. Message to customers. These emails are known as junk. Although there are many approaches to clear out SMS unsolicited mail [2], it's miles nonetheless a possibility to remedy this hassle with the assist of first-class practices. The condition of cellular customers is worse

Emails back and forth. Spam messages may be of the following kinds: SMS spam or e mail junk mail. Usually, those spam emails are used by spammers to promote a utility or employer. Many times from those unsolicited mail messages, customers additionally need to go through economic loss. Machine learning is a technique wherein machines analyze past information and make predictions based totally on given regulations. Today, the studies and the profound mastery of devices can be used to clear up the largest real troubles in all fields, including health care, protection, market evaluation, and so forth. There are various strategies of pc-based getting to know wherein gaining knowledge of is invisible and inexcusable. , acquisition of information with partial manage, and so forth. Supervised control offers with labels inside the output records set, at the same time as unsupervised control offers with unstructured facts sets. We used the latent UCI dataset. Therefore, we have tested the detection algorithms for some SMS junk mail officers.

OBJECTIVE:

To increase a version. Build and train a deep learning model the use of neural networks which includes Recurrent Neural Networks (RNN) or Convolutional

Neural Networks (CNN) to effectively distinguish among spam and legitimate SMS messages.

Dataset instruction. They acquire and pre-manner a whole list of SMS messages to provide a robust and various program, cautiously marking them as unsolicited mail or non-spam.

Feature engineering: It includes strategies which includes word embedding or TF-IDF to extract applicable capabilities from text records and represent SMS messages in a shape suitable for deep mastering.

Model assessment and evaluation. Evaluate the overall performance of the model the usage of relevant assessment metrics such as precision, keep in mind, F1 rating, and precision. Fine-tune the version to obtain greatest performance.

Integration and deployment: Integrate the learned model into a consumer-pleasant device that permits customers to send SMS messages to locate junk mail. Develop an account on the platform that customers can get right of entry to.

Real-time detection: The version has been evolved so that incoming SMS messages can be processed in real-time, presenting users with on the spot records approximately capacity junk mail messages obtained.

PROBLEM STATEMENT:

Short Message Service (SMS) junk mail is a sort of unsolicited mail e mail despatched by hackers through cell devices to obtain sensitive facts. We proposed a solution to determine whether or not an SMS message acquired is unsolicited mail the use of a deep learning approach referred to as Long Short Memory (LSTM), that could adjust parameters based on statistical information. Implement a deep getting to know approach to detect SMS spam. We purpose to broaden a version that could as it should be distinguish unsolicited mail from valid messages, offering users with a reliable and green manner to filter out unwanted content.

LITERATURE SURVEY:

1. Pavas Navane, "SMS" Spam Filtering Using Supervised Machine Learning and Maximum Entropy Algorithms, by Gaurav Dubey and Ajay Rana and Comparing Their Effectiveness in Filtering Junk and Spam Messages, in IEEE 2018.

As people share greater personal facts in more on line sports and organizations, SMS spam is turning into more commonplace. As average cellular phone use has grown, unsolicited mail has opened up every other road for malicious classified ads. These advertisers now use immediate messaging (SMS) to send unwanted messages to capacity shoppers which is called SMS junk mail. This form of unsolicited

mail is especially horrific because, unlike email unsolicited mail, many PDA clients pay for each SMS message. Developing a category set of rules that redirects SMS junk mail would be a useful tool for mobile phone vendors.

2. Shafi Muhammad Abdulhamid, Muhammad Shafi Abd Latif, Haruna Siroma, Oluwafemi Osho, Gaddafi Abdul-Salaam, Adamu I. Abubakar and Dudut Herawan, "A Study of Versatile SMS Spam Sifting Procedures" IEEE 2017: Spam in Short Message Services (. SMS) are unwanted or unsolicited mail messages acquired on a cellular telephone. This SMS unsolicited mail is inflicting actual harm to cell customers. This kind of marketing practice annoys service vendors, offends their clients or makes them lose their clients. To reduce this practice, researchers have proposed various answers to detect and filter out SMS unsolicited mail. In this article, we provide an outline of spam detection strategies, to be had strategies, demanding situations, and future research guidelines in cellular SMS unsolicited mail filtering and mitigation. The most famous strategies of detecting, filtering, and mitigating SMS junk mail are as compared, and the datasets used, their implications and boundaries, and destiny research instructions are discussed. The purpose of this evaluation is to assist studies specialists discover open regions that require in addition development.
10

3. Sharif NNA, Azmi NfM, Subrata S, Sarhan HM, Yahya Y, Sam SM. SMS garbage mail location the use of the inverse periodic file frequency and random wooded area algorithm. Educational method in laptop generation. 2019: The authors provided an inverse report frequency function extraction method to estimate related words. Various system learning fashions had been used to evaluate the overall overall performance of the classifier. The review shows that the blend of TF-IDF and Irregular Woods calculation beats different latest algorithms and for that reason improves the detection of SMS spam. The observe proposes consistent estimation thods in view of multi-facet perceptron (MLP) calculations and Bayesian organizations. The study's contribution is its ability to lower the rate of misclassification and superior performance to that of other ML algorithms.

4. Gaddes, Laxmanrao A, Satyanarayana S. SMS Spam Recognition Utilizing AI and Profound Learning Strategies. The seventh International Conference on Advanced Computing and

Communication Systems (ICACCS)'s Proceedings2021; IEEE: This paper offers an utility of contemporary techniques for using LSTM to detect SMS spam. The creator used 3 methods of word embedding in comparison, TF-IDF, vectorizer hashing. Experimental consequences of LSTM are in comparison with a few ultra-modern machine getting to know methods. However, the writer later confirmed the power of LSTM topology with the clonal choice algorithm for textual content category. This examine was evaluated the use of three datasets and comparing them with some today's gadget studying classifiers. Experimental outcomes display that the accuracy of the proposed version is superior to that of other designs, precision, don't forget, F1 score, and computation time.

5. Roy P.K., Singh J.P., Banerjee S. *Profound acquiring knowledge of for SMS unsolicited mail filtering. The subsequent generation of computing systems. 2020:* Roy proposed a deep getting to know approach based totally on convolutional brain organizations and long-present moment reminiscence fashions for SMS spam type. The take a look at indicates the vast performance of deep getting to know fashions using three shapes, and adding regularization parameters to growth the category accuracy c programming language. As a result, the evolved model has higher accuracy and computational pace.

6. Annareti S., Tamina S. *A comparative take a look at of deep gaining knowledge of techniques for spam location. Proceedings of the 2019 I- SMAC (Internet of Things, Mobile, Analytics, andCloud) Third International Conference; January 2020:* The creator used deep studying models to discover SMS junk mail. The first relative investigation of two profound learning styles dependent absolutely upon convolutional brain organizations and RNN. Overall, the consequences display thrilling findings. At the equal time, the latermodel of Jos implemented to the body of the Wikimedia China. The creators additionally tried theimpact of higher boundaries to reap higher results.

EXISTING SYSTEM:

Several techniques are used by huge gadgets to discover SMS as unsolicited mail or junk mail. These plans come through your junk mail email inbox and grow to be to your junk folder. However, amongst those techniques, simple textual techniques were determined to be inadequate to stumble on junk mail e-mail. To stumble on essential junk mail, we

propose the usage of hybrid techniques. A genetic set of rules is used to optimize and define the most important price block, referred to as the confidence element, which controls the tree pruning choice. A important trouble with any text content material software program that involves spam detection is that there are numerous capabilities that appreciably reduce accuracy.

PROPOSED SYSTEM

The proposed tool is designed to obtain the subsequent goals: 1) Study device gaining knowledge of algorithms to clear up the problem of unsolicited mail detection. 2) Use algorithms to check mutants. 3) Engine removal algorithm. 4) Monitor and appropriately examine baseline models. 5) Finish installing Python. The scikit-analyze library can be used so you can test with Python, enhancing examples, pre-processing and doing laptop consequences. The script may be used in addition to optimization strategies and in comparison basic outcomes, i.E. Use default settings. Spamming machines can come across incoming e-mail statistics and use textual content processing and optimization algorithms to identify SMS in unsolicited mail or spam.

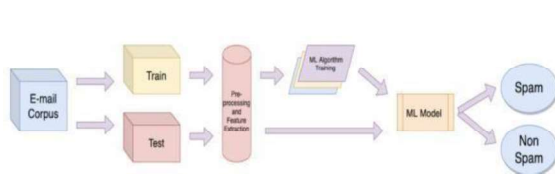
GOALS:

The primary desires of UML improvement are:

1. Provide users with a clean, comprehensible, prepared-to-use format language so that key models can be evolved and allotted.
2. For the expansion and unique schooling of gadget technologies to bolster the principles.
3. Be truthful in the direction of correct programming languages and improvement techniques.
- Four. To offer a appropriate basis for the development of a statistical language.
- 5 Confirm the significance of the market for OOP tools.
6. Support excessive-stage improvement ideas, which include collaboration, structure, style, and collaboration.

SYSTEM ARCHITECTURE

It is terrific to explain the awesome standard abilities of this gadget to discover and configure what you need. In a system structure, several components and their relationships are diagnosed and modeled. Major organizations of programs are diagnosed, damaged down with the aid of methods and device notations and conceptual principles, and connections among modules are diagnosed. The proposed system consists of these modules.



MODULES:

- PYTHON
- NUMPY

- MATPLOTLIB
- PANDAS
- SEABORN
- TENSORFLOW
- NLTK

MODULES USED:

Python: Python is an extensively used programming language for deep learning initiatives due to its wealthy libraries and ease of use.

NumPy: NumPy is an entire math library that includes random range generators, linear algebra workouts, Fourier series transforms and greater.

Matplotlib: Matplotlib is a library for the Python programming language that works with to make compelling static, vivified or intuitive perceptions. Matplotlib is a plotting tool for Python that works with NumPy, Python's numerical extension.

Pandas: Pandas is an application for data technology made specifically for the Python programming language. It includes strategies and records structures that make it possible to manipulate numerical tables and collect time.

Seaborn: Seaborn is a Python module for growing statistical visualizations.

Tensor Flow: Tensor Flow is a library used in gadget learning and an open supply library for numerical computation. It is a tool for growing mastering applications. This library turned into firstly evolved by way of the Google brain team and is a very common and correctly used library that offers diverse gear for device getting to know programs.

NLTK: NLTK is a string manipulation library that accepts enter strings. The output is within the shape of a string or a list of strings. This library provides many gear that are maximum beneficial in accomplishing mastering goals. It compares well with distinctive consequences.

PROPOSED ALGORITHM:

Lstm:

Long-term memory, also called LSTM, is an synthetic neural community used within the fields of artificial intelligence and deep getting to know.

LSTMs have feedback loops and have been applied to discrete, linked problems together with hand reputation, speech popularity, gadget translation, robotic video video games, and fitness care.

A cellular LSTM block includes an average input gate, output gate, and 0 gate. The cellular recollects the values at random durations, and the three gates manage the drift of data into and out of the mobile.

Processing of natural language;

Normal language handling (NLP) is a part of PC innovative expertise, exceptionally a part of man-made brainpower or simulated intelligence, that offers PC frameworks the capacity to figure out text based content and spoken words.

Computational linguistics, which is the rule-based modelling of human language, is combined with statistical models, machine learning, and deep studying models in NLP.

NLP utilizes pc programs that decipher text from one language into some other, answer voice directions, and quick technique a lot of textual content.

SYSTEM REQUIREMENTS

Hardware requirements

System: Pentium i3 Processor.

Hard Disk: 500GB.

Monitor: 15" LED

Input Devices: Keyboard, Mouse

Ram : 4GB

Software requirements

Operating system: Windows 10. Coding

Language: Python

Web Framework: Flask

CONCLUSION:

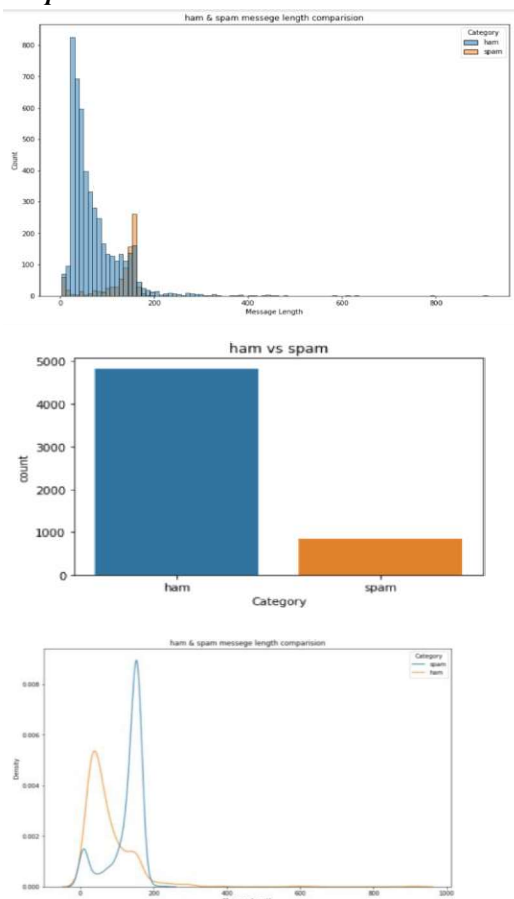
Provides workflow know-how and identifying application violations and unsolicited mail emails. Expanded email and SMS unsolicited mail dataset in five Predefined categories to gain a deeper expertise of their learning. Use it. This shows the end result obtained with the aid of Naive Bayes They are lots higher classifiers than SVM for textual content classification. Parsing messages to understand spam e mail text in Python. Helps an understanding tool for understanding people's personalities and conversations.

FUTURE ENHANCED:

Automate your dataset update method. Increase the size of your dataset. Increasing the wide variety of lines creates a hybrid set of rules. He diligently elevated his dexterity. Multilingual junk mail detection. Real-time detection and reaction. Privacy-safety solutions.

SCREENSHOTS:

Graphs:



REFERENCES

- [1] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based Naive Bayes algorithm for spam filtering," in Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC), Dec. 2016, pp. 1–8, doi: 10.1109/pccc.2016.7820655.
- [2] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for sms spam filtering: Review, approaches and open research problems," Heliyon, vol. 5, no. 6, Jun. 2019, Art. no. e01802, doi: 10.1016/j.heliyon.2019.e01802.
- [3] W. Awad and S. ELseuofi, "Machine learning methods for spam E-Mail classification," Int. J. Comput. Sci. Inf. Technol., vol. 3, no. 1, pp. 173–184, Feb. 2011, doi: 10.5121/ijcsit.2011.3112.
- [4] S. Mohammed, O. Mohammed, and J. Fiaidhi, "Classifying unsolicited bulk sms (UBE) using Python machine learning techniques," Int. J. Hybrid Inf. Technol., vol. 6, no. 1, pp. 43–55, 2013. [Online]. Available: https://www.researchgate.net/publication/236970412_ClassifyingUnsolicited_Bulk_Sms_UB_E_using_Python_Machine_Learning_Techniques
- [5] A. Wijaya and A. Bisri, "Hybrid decision tree and logistic regression classifier for sms spam detection," in Proc. 8th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE), Oct. 2016, pp. 1–4, doi: 10.1109/ICITEED.2016.7863267.
- [6] K. Agarwal and T. Kumar, "Sms spam detection using integrated approach of Naïve Bayes and particle swarm optimization," in Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS), Jun. 2018, pp. 685–690, doi: 10.1109/ICCONS.2018.8662957.
- R. Belkebir and A. Guessoum, "A hybrid BSO-Chi2-SVM approach to arabic text categorization," in Proc. ACS Int. Conf. Comput. Syst. Appl. (AICCSA), Ifran, Morocco, May 2013, pp. 1–7, doi: 10.1109/AICCSA.2013.6616437.
- [7] A. I. Taloba and S. S. I. Ismail, "An intelligent hybrid technique of decision tree and genetic algorithm for E-Mail spam detection," in Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS), Cairo, Egypt, Dec. 2019, pp. 99–104, doi: 10.1109/ICICIS46948.2019.9014756.
- [8] R. Karthika and P. Visalakshi, "A hybrid ACO based feature selection method for sms spam classification," WSEAS Trans. Comput., vol. 14, pp. 171–177, 2015. [Online]. Available: <https://www.wseas.org/multimedia/journals/computers/2015/a365705-553.pdf>
- [9] S. L. Marie-Sainte and N. Alalyani, "Firefly algorithm based feature selection for arabic text classification," J. King Saud Univ.-Comput. Inf. Sci., vol. 32, no. 3, pp. 320–328, Mar. 2020.
- [10] S. Subramanian, and K. Premalatha, "An enhanced cuckoo search for optimization of bloom filter in spam filtering," Global J. Comput. Sci. Technol., vol. 12, no. 1, pp. 75–81, 2012. Accessed: Jan. 18, 2020. [Online]. Available: https://globaljournals.org/GJCST_Volume12/12-An-Enhanced-Cuckoo-Search-for-Optimization.pdf
- [11] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial

and Bernoulli Naïve Bayes for text classification,’’ in Proc. Int. Conf. Autom., Comput. Technol. Manage. (ICACTM), London, U.K., Apr. 2019, pp. 593–596, doi: 10.1109/ICACTM.2019.8776800.