|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **School of Electronics Engineering (SENSE)** | | | | |
| **J COMPONENT – REPORT** | | | | |
| **COURSE CODE / TITLE** | CSE3502– Information Security Management | | | |
| **PROGRAM / YEAR/ SEM** | B.Tech (ECE/ECM)/III Year/ Winter 2021-2022 | | | |
| **DATE OF SUBMISSION** | 20-04-2022 | | | |
| **TEAM MEMBERS**  **DETAILS** | **REGISTER NO.** | | **NAME** | |
| 19BEC1017 | | JAYASOORIYA G | |
| 19BEC1193 | | JEEVARATHINAM N | |
| 19BEC1370 | | SANJAY S | |
| **PROJECT TITLE** | **SMS SPAM DETECTION USING**  **DEEP LEARNING** | | | |
| **COURSE HANDLER’S NAME** | **DR.N.SUBHASHINI** | **REMARKS** | |  |
| **COURSE HANDLER’S SIGN** |  |

**ABSTRACT:**

This paper presents detection of spam and ham using deep learning models such as LSTM (long short-term memory), BiLSTM (bi directional long short-term memory) and Dense text classifier and compare 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. SMS spam filter inherits much functionality from E-mail Spam Filtering. Performance metrics which we use to compare the deep learning models are accuracy, precision, recall, f1 score, confusion matrix, classification report, log loss.

Thanks to advancement in technologies, we are now able to extract meaningful information from such data using various Natural Language Processing (NLP) techniques. NLP, a branch of Artificial Intelligence (AI), makes use of computers and human natural language to output valuable information. NLP is commonly used in text classification task such as spam detection and sentiment analysis, text generation, language translations and document classification.

**KEYWORDS:**

LSTM, BiLSTM, Dense text, Spam detection, Deep Learning, Evaluation metrics.

**INTRODUCTION:**

In the developing period of the Internet, individuals are involving increasingly in free online services. Individuals tend to share their data on different sites, though that data is imparted to different organizations that spam individuals to offer their services.

Mobile message is a way of communication among the people, and billions of mobile device users exchange numerous messages. However, such type of communication is insecure due to lack of proper message filtering mechanisms. One cause of such insecurity is spam, and it makes the mobile message communication insecure. Spam is considered to be one of the serious problems in e-mail and instance message services. Spam is a junk mail or message. Spam e-mails and messages are unwanted for receivers which are sent to the users without their prior permission. The cost of mails and messages are very low for senders but high for receipts of these messages. The cost paid some time by service providers and the cost of spam can be measured in the loss of human time and loss of important messages or mails. A spammer is a person/company which is responsible for unsolicited messages. For their organization benefits or personal benefits, spammers sending a vast number of messages to the users. These messages are called spam messages. Although there are various SMS spam filtering techniques available, still there is a need to handle this problem with advanced techniques. Mobile users may get annoyed because of spam messages. Due to these spam mails and messages, the values able e-mails and messages are affected because each user have limited Internet services, short time, and memory

As we all know that Hackers / Spammer tries to intrude in Mobile Computing Device, and SMS support for mobile devices had become vulnerable, as attacker tries to intrude to the system by sending unwanted link, with which on clicking those link the attacker can gain remote access over the mobile computing device.

SMS Spamming in extremely disappointing for the clients: numerous critical and valuable messages can get lost because of spam messages, Spam messages are additionally used to trap individuals, or bait them into purchasing services. 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.

Spam messages can be classified as redundant messages sent to large number of people at once. The rise of spam messages is based on the following factors:

1) The accessibility to cheap bulk SMS-plans;

2) dependability (since the message comes to the cell phone client);

3) low possibility of accepting reactions from some unaware recipients; and

4) the message can be customized.

5) Free services.

A number of major differences exist between spam-filtering in text messages and emails. Unlike emails, which have a variety of large datasets available, real databases for SMS spams are very limited. Additionally, due to the small length of text messages, the number of features that can be used for their classification is far smaller than the corresponding number in emails. Here, no header exists as well. Additionally, text messages are full of abbreviations and have much less formal language that what one would expect from emails. All of these factors may result in serious degradation in performance of major email spam filtering algorithms applied to short text messages.

So, in order to tackle this problem, an accurate and precise method is needed to detect the spam in mobile message communication. We proposed the applications of the machine learning and deep learning based spam detection method for accurate detection.

In this project, the goal is to apply different machine learning algorithms and deep learning models to SMS spam classification problem, compare their performance to gain insight and further explore the problem, and design an application based on one of these algorithms/models that can filter SMS spams with high accuracy. We use a database of 5574 text messages from UCI Machine Learning repository gathered in 2012. It contains a collection of 425 SMS spam messages manually extracted from the Grumbletext Web site (a UK forum in which cell phone users make public claims about SMS spam), a subset of 3,375 SMS randomly chosen non-spam (ham) messages of the NUS SMS Corpus (NSC), a list of 450 SMS non-spam messages collected from Caroline Tag’s PhD Thesis, and the SMS Spam Corpus v.0.1 Big (1,002 SMS non-spam and 322 spam messages publicly available). The dataset is a large text file, in which each line starts with the label of the message, followed by the text message string. After data exploration pre-processing and extraction of features, deep learning models such as Dense text classifier, LSTM, and other models are applied to the samples, and their performances are compared. Finally, the performance of best classifier from the project is compared against the performance of classifiers applied in the various papers citing this dataset. Feature extraction and initial analysis of data is done in PYTHON using numpy, pandas, matplotlib and plotly libraries, then different deep learning models is created using tensorflow in jupyter notebook. The project report is organized as the follows: **Methodology** which consists of 4 sections. Section 1 explains the research framework of the project, exploration, preprocessing of the data and extraction of features from the main dataset, and explores the result of initial analysis to gain insight. Section 2 explores the application of Dense Text classifier model to the problem. In Section 3, application of Long Short Term Memory model to the classification problem is studied. Section 4 analyses the application of Bidirectional Long Short Term model for the data**. Components and Software used** explores the various IDs used and different packages used in the project. **Evaluation Parameters** describes the various parameters to used evaluate the deep learning models. **Results and Discussions** compares the performances of three deep learning models based a set of evaluation metrics and then finally compares their performance against various papers citing this dataset. **Conclusion** concludes the report.

|  |  |
| --- | --- |
| LABEL | PERCENTAGE IN DATASET |
| HAM | 86.6 |
| SPAM | 13.4 |

**NOVELTY:**

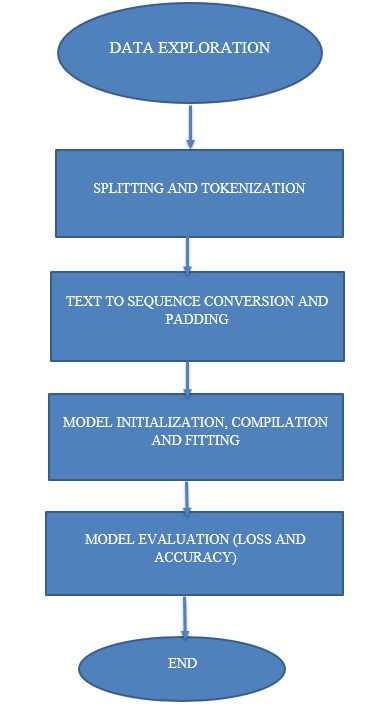
**LITERATURE SURVEY:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Paper title** | **Author** | **Journal Info(Name, Vol, Index, pageno, year)** |
| 1 | SMS Spam Detection using Machine Learning and Deep Learning Techniques | S. Gadde,  A. Lakshmanarao,  S. Satyanarayana. | *7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, pp. 358-362. |
| 2 | SMS Spam Filtering using Supervised Machine Learning Algorithms | P. Navaney,  G. Dubey,  A. Rana. | *8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2018, pp. 43-48. |
| 3 | SMS Spam Detection  using Machine Learning | Suparna DasGupta,  Soumyabrata Saha,  Suman kumar Das. | Journal of Physics: Conference Series, 2021. |
| 4 | Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms | Luo GuangJun,  Shah Nazir,  Habib Ullah Khan, Amin Ul Haq | Security and Communication Networks, vol. 2020, Article ID 8873639, 6 pages, 2020. |
| 5 | SMS spam detection and comparison of  various machine learning algorithms | P. Sethi,  V. Bhandari,  B. Kohli | *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 2017, pp. 28-31. |
| 6 | Comparative Study of Machine Learning Algorithms for SMS Spam Detection | A. Alzahrani,  D. B. Rawat | *2019 SoutheastCon*, 2019, pp. 1-6. |
| 7 | Mobile SMS Spam Detection using Machine Learning Techniques | Samadhan M.Nagre | Journal of emerging technologies and innovative research,2018. |
| 8 | SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm. | Nilam Nur Amir Sjarif,  Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat | Procedia Computer Science, 2019 |
| 9 | SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit | Pumrapee Poomka, Wattana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop | International Journal of Future Computer and Communication, Vol. 8, No. 1, March 2019 |
| 10 | SMS Spam Identification and Risk Assessment Evaluations | Alaa Mohasseb and Benjamin Aziz and Andreas Kanavos | WEBIST, 2020 |
| 11 | A Systematic Literature Review on SMS Spam Detection Techniques | Lutfun Nahar Lota and B. M. Mainul Hossain | International Journal of Information Technology and Computer Science, 2017, vol. 9, pp. 42-50. |
| 12 | A review of soft techniques for SMS spam classification: Methods, approaches and applications | Olusola Oluwakemi Abayomi-Alli | Eng. Appl. Artif.Intell. ,2019, vol.86, pp. 197-212 |
| 13 | A NEW DEFENSE MECHANISM AGAINST SMISHING ATTACKS USING GRAY WOLF OPTIMIZER | Marwan H. Alsammarraie | IU. edu.,2020 |
| 14 | SMS spam detection for Indian messages | S. Agarwal, S. Kaur and S. Garhwal | 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015, pp. 634-638 |
| 15 | SMS Phishing Detection Using Oversampling and Feature Optimization Method | Tong Wu and Kangfeng Zheng and Chunhua Wu and Xiujuan Wang | DEStech Transactions on Computer Science and Engineering,2018 |
| 16 | A proposed data science approach for email spam classification using machine learning techniques | Aakash Atul Alurkar; Sourabh Bharat Ranade; Shreeya Vijay Josh Siddhesh Sanjay Ranade | 2017 Internet of Things Business Models, Users, and Networks |
| 17 | An Improved Spam Detection Method with Weighted Support Vector Machine | V Vishagini Archana K Rajan | 2018 International Conference on Data Science and Engineering (ICDSE) |
| 18 | Performance Evaluation of Machine Learning Algorithms for Email Spam Detection | S. Nandhini Jeen Marseline K.S. | 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) |
| 19 | Comparative Study on Email Spam Classifier  using Data Mining Techniques | R. Kishore Kumar, G. Poonkuzhali, P. Sudhakar | International MultiConference of Engineers and Computer Scientists 2012 Vol I,  IMECS 2012, March 14 - 16, 2012, Hong Kong |
| 20 | Evaluating the Effectiveness of Machine Learning Methods for Spam Detection | YuliyaKontsewaya,  EvgeniyAntonov AlexeyArtamonov | 2020 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial  Intelligence: Eleventh Annual Meeting of the BICA Society |

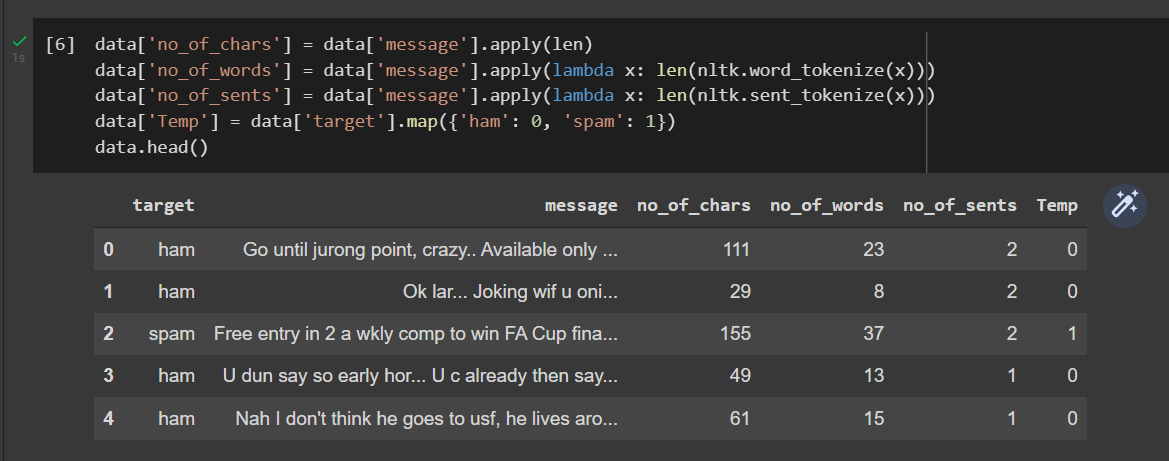
**METHODOLOGY/IMPLEMENTATION:**

**SECTION 1:**

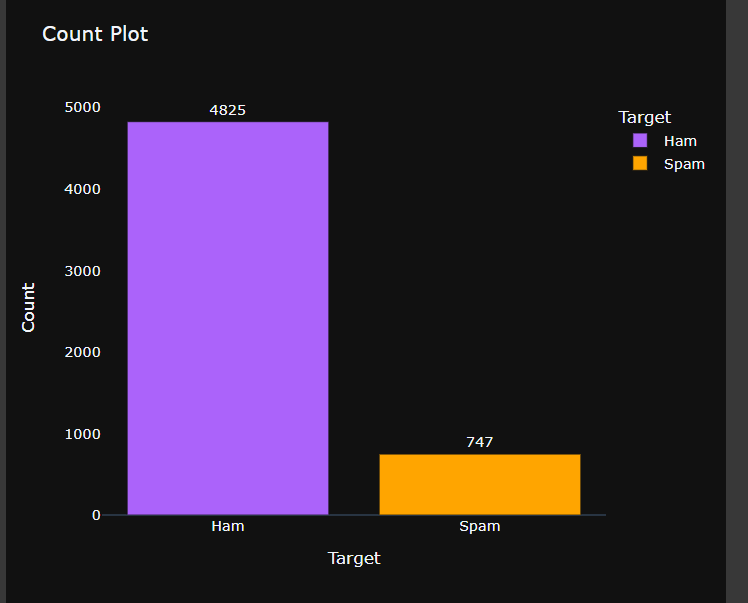
**RESEARCH FRAMEWORK:**



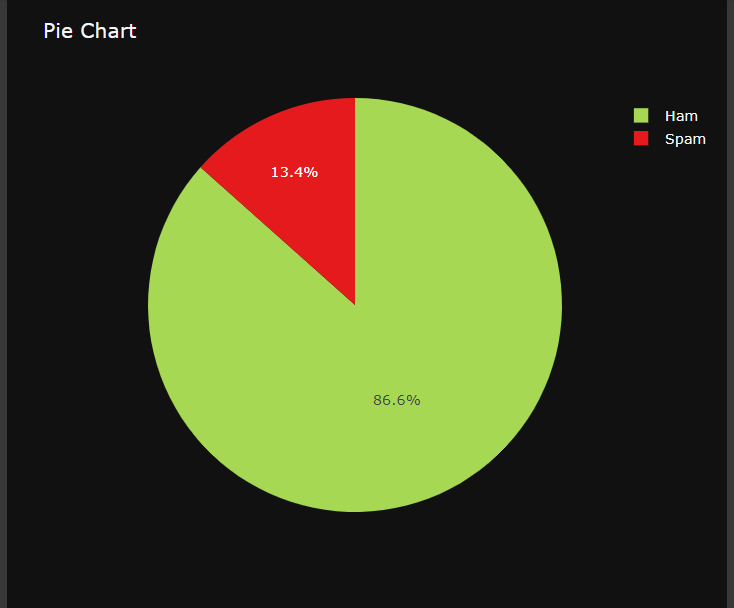
**DATA EXPLORATION:**



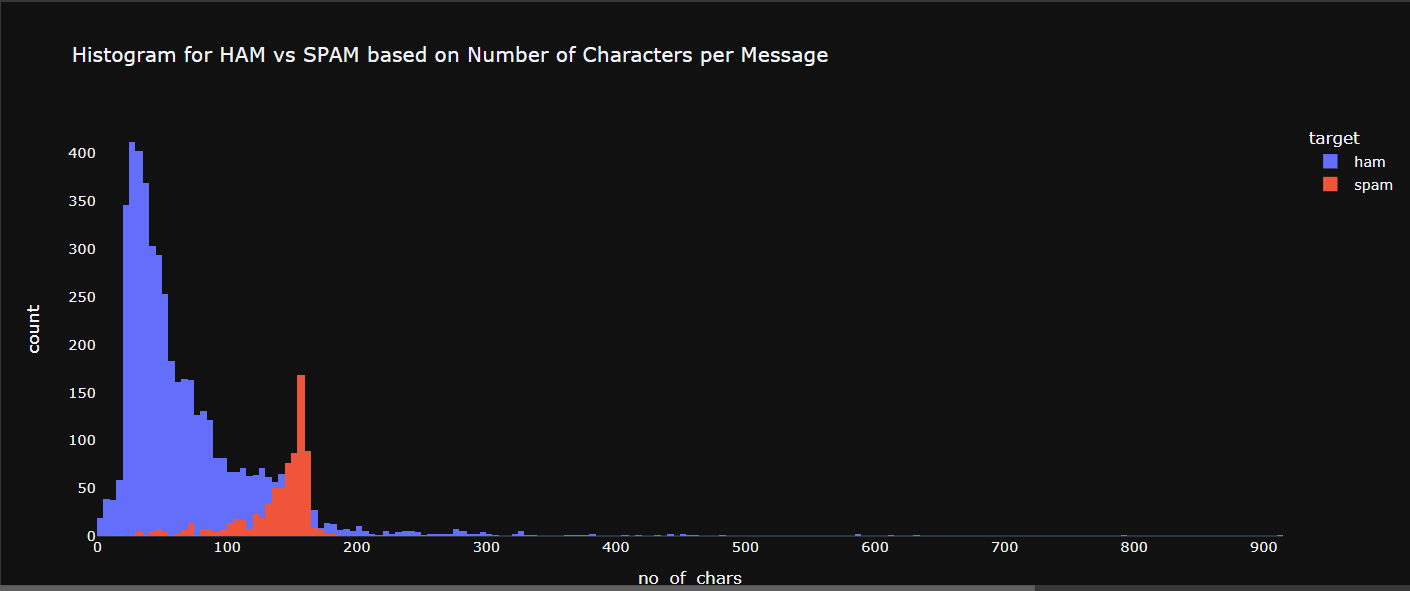
**COUNT PLOT:**



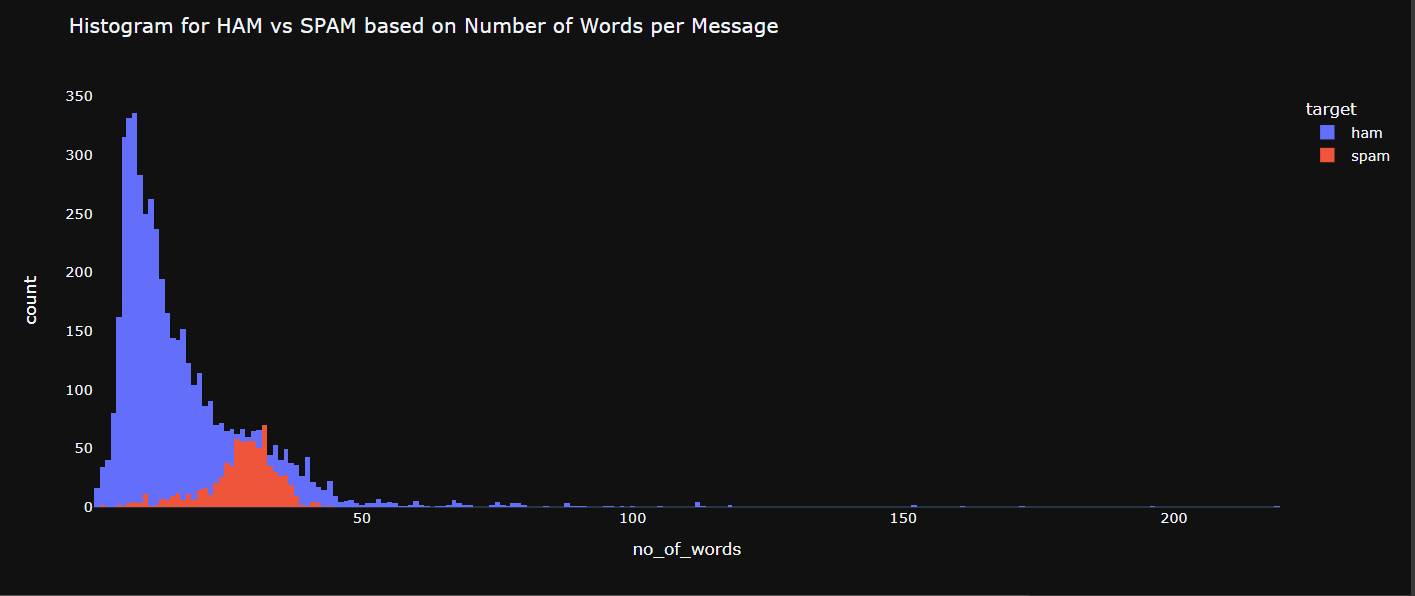
**PIE CHART REPRESENTATION OF HAM AND SPAM MESSAGES:**



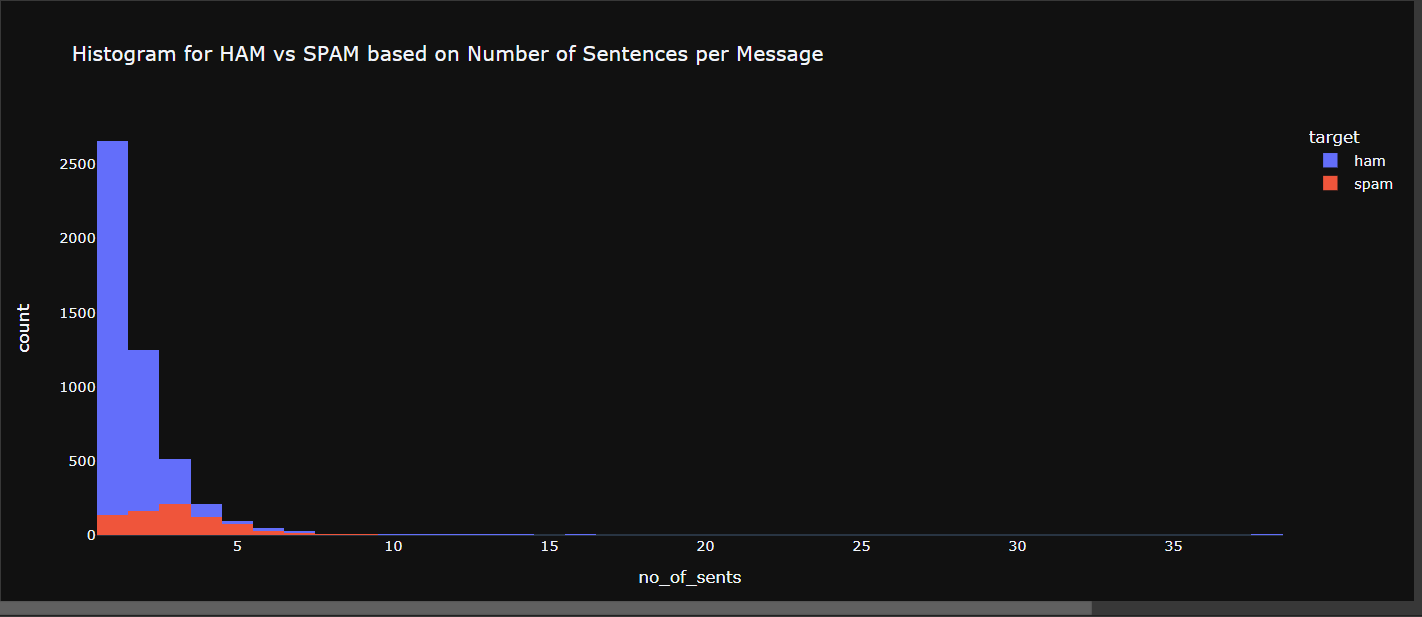
**HISTOGRAM OF HAM VS SPAM BASED ON NUMBER OF CHARACTERS PER MESSAGE:**



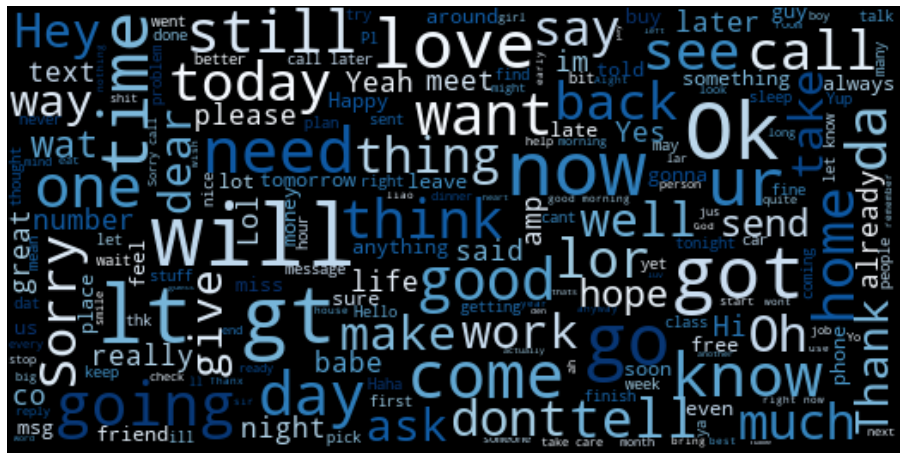
**HISTOGRAM OF HAM VS SPAM BASED ON NUMBER OF WORDS PER MESSAGE:**



**HISTORGRAM OF HAM VS SPAM BASED ON NUMBER OF SENTENCES PER MESSAGE**



**WORD CLOUD OF HAM AND SPAM:**

****

**(i)Ham**

****

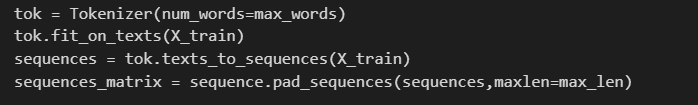
**(ii) Spam**

**DATA PREPROCESSING:**

**FEATURE SELECTION:**



**TOKENIZATION:**



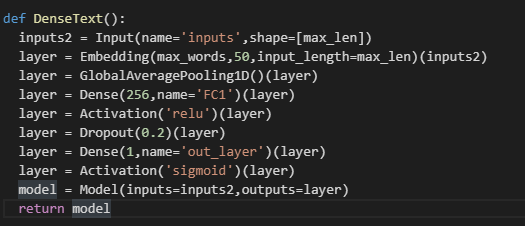
(i)SAMPLE DATA BEFORE AND AFTER TOKENIZATION

|  |  |
| --- | --- |
| MESSAGE | BEFORE TOKENIZATION |
| Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005 | [48, 541, 9, 20, 4, 814, 879, 2, 163, 675, 288] |
| Did you catch the bus? Are you frying an egg? Did you make a tea? Are you eating your mom's left over dinner? Do you feel my Love ? | [112, 3, 5, 466, 22, 3, 116, 112, 3, 144, 4, 22, 3, 13, 422, 189, 345, 27, 3, 204, 11, 65] |
| Thanks for your subscription to Ringtone UK your mobile will be charged å£5/month Please confirm by replying YES or NO. If you reply NO you will not be charged | [182, 12, 13, 2, 391, 190, 13, 95, 33, 30, 848, 348, 103, 842, 76, 136, 26, 39, 32, 3, 93, 39, 3, 33, 24, 30, 848] |
| Didn't you get hep b immunisation in nigeria. | [305, 3, 36, 185, 9] |

**SECTION 2:**

**DENSE TEXT CLASSIFIER:**

MODEL:



Number of Layers = 2

Number of Hidden Layers =1

Activation Function of Hidden Layers = ReLU

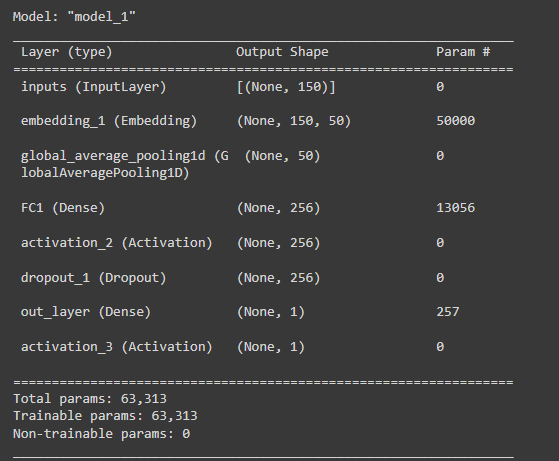
Number of Nodes in Hidden Layer =256

Dropout in Hidden Layer =20%

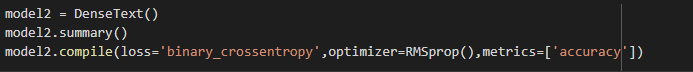
Activation Function of Output Layer = Sigmoid (Preferred for binary classification)

Global average pooling is used to down-sample given input since there is an imbalance in our dataset, ham-86.6% and spam-13.4%.

SUMMARY:



MODEL FITTING:



Loss metric used = Binary cross entropy (log loss)

Optimizer used = RMS propagation

Batch size =128

Number of epochs = 10

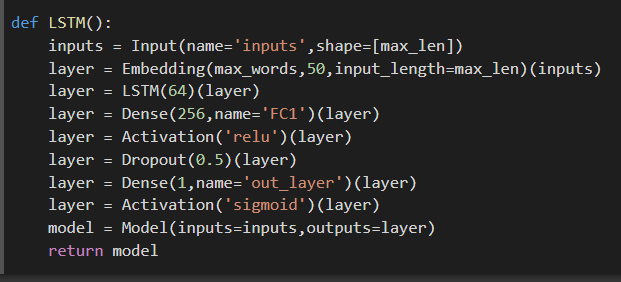
Validation split and Test Split: 20:80



**SECTION 3:**

**LSTM:**

MODEL:



Number of layers: 3

Number of hidden layers: 2

Activation function of hidden layer: ReLU

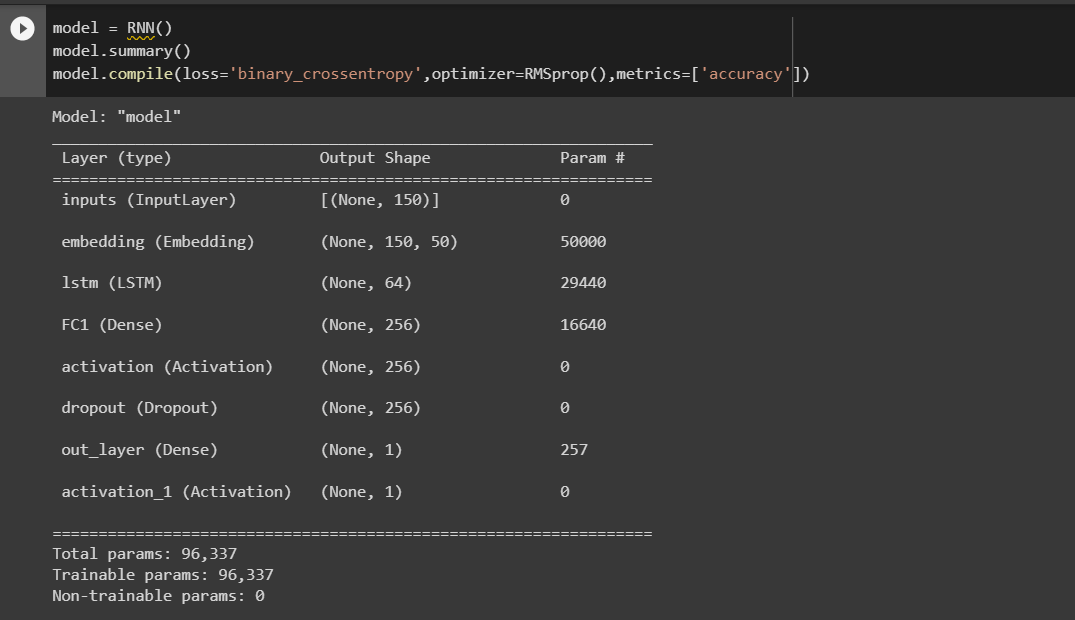
Number of nodes in Hidden layer 1: 64

Number of nodes in Hidden layer 2: 256

Dropout in Hidden layer: 50%

Activation function of outer layer: Sigmoid

MODEL FITTING:



Loss metrics used = binary cross entropy (log loss)

Optimization = RMS propagation

Batch size = 128

Number of epoch = 10

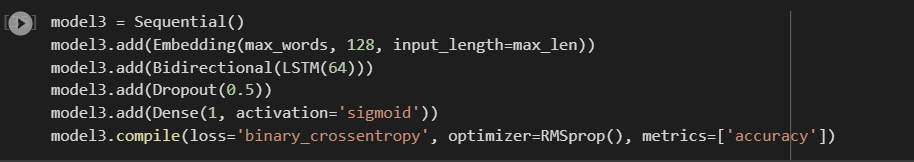
Validation split and Test split = 20:80



**SECTION 4**

**BiLSTM**

**MODEL:**



Number of layers: 2

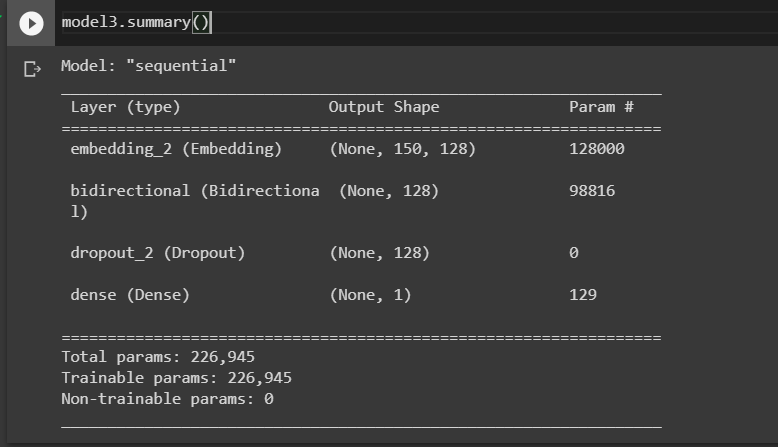
Number of hidden layers: 1

Number of nodes in Hidden layer: 64

Dropout in Hidden layer: 50%

Activation function of outer layer: Sigmoid

MODEL FITTING:



Loss metrics used = binary cross entropy (log loss)

Optimization = RMS propagation

Batch size = 128

Number of epoch = 10

Validation split and Test split = 20:80

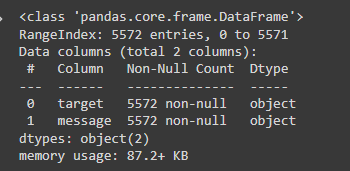


**DATASET DESCRIPTION:**

(i)SAMPLE RECORD OF SMS SPAM DATASET

|  |  |
| --- | --- |
| Message | Spam |
| Go until jurong point, crazy... Available only in bugis n great world la e buffet... Cine there got amore wat... | 0 |
| As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press \*9 to copy your friends Callertune | 0 |
| XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL | 1 |
| Ffffffffff. Alright no way I can meet up with you sooner? | 0 |
| 07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sms for a free nokia mobile + free camcorder. Please call now 08000930705 for delivery tomorrow | 1 |

(ii) DATASET INFO

****

(iii) DATASET ANALYSIS BASED ON NUMBER CHARACTERS, WORDS AND SENTENCES IN A GIVEN MESSAGE

****

**EVALUATION PARAMETERS:**

The following are the parameters used for evaluating deep learning models:

**Confusion matrix:**

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa – both variants are found in the literature.

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).



True Positive:

Interpretation: You predicted positive and it’s true.

True Negative:

Interpretation: You predicted negative and it’s true.

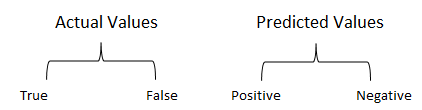
False Positive: (Type 1 Error)

Interpretation: You predicted positive and it’s false.

False Negative: (Type 2 Error)

Interpretation: You predicted negative and it’s false.

Just Remember, we describe predicted values as Positive and Negative and actual values as True and False.



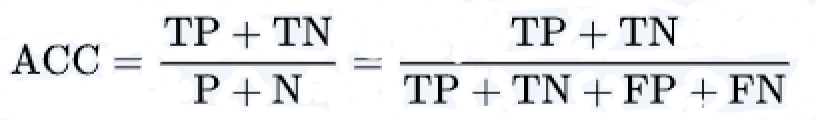
Some of the evaluation metrics used are Precision, Accuracy, Recall, F1 score.

**Precision:**



Precision is the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives

**Accuracy:**



**Recall:**



Recall is the ability of a classifier to find all positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives.

**F1\_score:**



The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F1 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy.

The f1\_score function accepts the following parameters:

**y\_true:** These are the true labels.

**y\_pred:** These are the predicted labels.

**labels:** This parameter identifies the labels to be included when there is a multiclass problem.

**pos\_label:** This is the class to report in case of a binary classification problem.

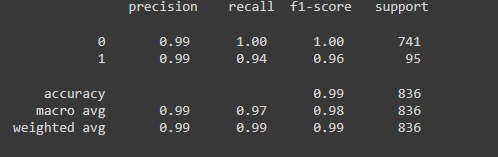
**average:** This is the type of averaging to be performed in the case of multiclass data.

**sample\_weight:** These are any sample weights to be used in the calculation of the F1 score.

**Classification report:**

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report as shown below.

The table will be in the form of



**Log loss: (Should be minimum)**

Log loss also called as logistic loss or cross entropy loss. This is a loss function used in logistic regression and extension of its such as neural networks, defined as negative log – likelihood of a logistic model that returns y-pred probability of its training data. This is only defined in 2 or more labels.



**CONCEPT LEARNED:**

**Recurrent Neural Networks** (RNN) suffer from short-term memory. If a sequence is long enough, they’ll have a hard time carrying information from earlier time steps to later ones. So if you are trying to process a paragraph of text to do predictions, RNN’s may leave out important information from the beginning.

During back propagation, recurrent neural networks suffer from the vanishing gradient problem. Gradients are values used to update a neural networks weight. The vanishing gradient problem is when the gradient shrinks as it back propagates through time. If a gradient value becomes extremely small, it doesn’t contribute too much learning.

**LSTM** (Long short-term memory) were created as the solution to short-term memory. They have internal mechanisms called gates that can regulate the flow of information.

These gates can learn which data in a sequence is important to keep or throw away. By doing that, it can pass relevant information down the long chain of sequences to make predictions. Almost all state of the art results based on recurrent neural networks are achieved with these two networks. LSTM can be found in speech recognition, speech synthesis, and text generation. You can even use them to generate captions for videos.

Sequential machine learning models that input or output data sequences are known as sequence models. Text streams, audio clips, video clips, time-series data, and other types of sequential data are examples of sequential data. Recurrent Neural Networks (RNNs) are a well-known method in sequence models. The analysis of sequential data such as text sentences, time-series, and other discrete sequence data prompted the development of Sequence Models. These models are better suited to handle sequential data, whereas Convolutional Neural Networks are better suited to treat spatial data. The crucial element to remember about sequence models is that the data we’re working with are no longer independently and identically distributed samples, and the data are reliant on one another due to their sequential order. For speech recognition, voice recognition, time series prediction, and natural language processing, sequence models are particularly popular.

**BiLSTM** (Bidirectional LSTM) recurrent neural networks (RNN) are really just putting two independent RNNs together. This structure allows the networks to have both backward and forward information about the sequence at every time step

Using bidirectional will run your inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backward you preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future.

**ReLU** (rectified linear unit) is one of the most popular function which is used as hidden layer activation function in deep neural network. ReLU activation function is defined as

**g ( z ) = max { 0 , z }**

The ReLU activation function **g(z) = max {0, z}** **is not differentiable at z = 0**. A function is differentiable at a particular point if there exist left derivatives and right derivatives and both the derivatives are equal at that point. ReLU is differentiable at all the point except 0. the left derivative at z = 0 is 0 and the right derivative is 1.

This may seem like g is not eligible for use in gradient based optimization algorithm. But in practice, gradient descent still performs well enough for these models to be used for machine learning tasks. This is in part because neural network training algorithms do not usually arrive at a local minimum of the cost function. Hence it is acceptable for the minima of the cost function to correspond to points with undefined gradient. Hidden units that are not differentiable are usually non-differentiable at only a small number of points.

ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.



A **sigmoid** function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve. A common example of a sigmoid function is the logistic function shown in the first figure and defined by the formula



The sigmoid function is used as an activation function in neural networks. Just to review what is an activation function, the figure below shows the role of an activation function in one layer of a neural network. A weighted sum of inputs is passed through an activation function and this output serves as an input to the next layer.

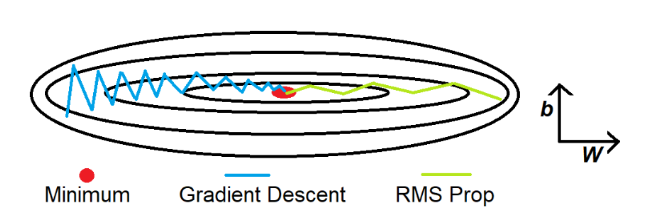
When the activation function for a neuron is a sigmoid function, it is a guarantee that the output of this unit will always be between 0 and 1. Also, as the sigmoid is a non-linear function, the output of this unit would be a non-linear function of the weighted sum of inputs. Such a neuron that employs a sigmoid function as an activation function is termed as a sigmoid unit.

The fact that the sigmoid function is monotonic, continuous and differentiable everywhere, coupled with the property that its derivative can be expressed in terms of itself, makes it easy to derive the update equations for learning the weights in a neural network when using back propagation algorithm.

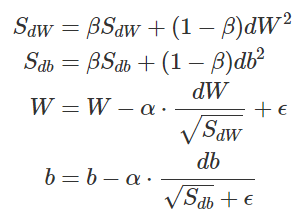


**RMS propagation:**

The Root Mean Square Propagation RMS Prop is similar to Momentum, it is a technique to dampen out the motion in the y-axis and speed up gradient descent. For better understanding, let us denote the Y-axis as the bias b and the X-axis as the weight W. It is called Root Mean Square because we square the derivatives of both w and b parameters.



The intuition is that when we divide a large number by another number, the result becomes small. In our case, the first large number is db and the second large number that we use is the weighted average of db². We introduce two new variables Sdb and SdW, to keep track of the weighted average of db² and dW². The division of db and Sdb results in a smaller value which dampens out the movement in the y-axis. The Ⲉ is introduced to avoid the division by 0 error.



The idea is to slow down the learning on the y-axis direction and speed up the learning on the x-axis direction. On each iteration t the derivatives dw and db are computed on the current mini-batch. It is also performed the exponentially weighted average called SdW and Sdb. Now, the exponentially weighted average parameter SdW is relatively small and so we are dividing by a small number to obtain the weight. The alpha parameter is the learning rate and the Ⲉ is introduced to avoid the division by 0 error. On the contrary, Sdb is relatively large and it helps to slow down the updates vertical dimension b. In fact, looking at the figure above, the derivative in the horizontal dimension is always small and the derivative on the vertical dimension is large. The net effect is to speed up the vertical learning and at the same time slow down the vertical learning. The result is the green line of the figure above.

**CHALLENGES FACED:**

* Lack of knowledge of Recurrent Neural Network models (LSTM and BiLSTM).
* Our initial accuracy for LSTM model was around 85%, so we performed hyperparameter tuning on parameters like batch size, number of epochs, numbers of layers, number of nodes in each layer, activation functions, optimization strategies and loss functions.
* Since LSTM and BiLSTM are complicated neural networks, it took longer runtime. It took about 7-8 min for complete execution of program and the work became tedious while performing hyper parameter tuning.

**APPLICATIONS:**

* The implemented deep learning models can be used for other binary classification problems in natural language processing such Email spam detection, malicious URL detector etc.
* Since the accuracy of our LSTM, BiLSTM and Dense-Text is around 98-99% we can use this as a backend to create a website where people can just post the text message they receive and check whether its spam or ham.
* The website can be converted to a mobile app which access the messages and classifies them as spam or ham.
* This can prevent a lot of frauds such as banking frauds and information theft etc.

**RESULTS AND DISCUSSION IN DETAIL:**

We are going to evaluate our deep learning models using previously mentioned evaluation metrics and compare and find out which is the best model.

**LSTM:**

**CONFUSION MATRIX:**

|  |  |  |
| --- | --- | --- |
| **Predicted/Actual Values** | **Ham** | **Spam** |
| **Ham** | 739 (TP) | 2 (FP) |
| **Spam** | 6 (FN) | 89 (TN) |

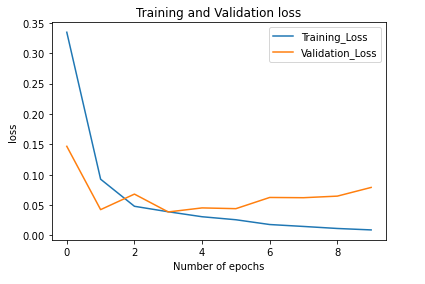
**CLASSIFICATION REPORT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1\_Score** | **Support** |
| **0(Ham)** | 0.99 | 1.00 | 0.99 | 741 |
| **1(Spam)** | 0.98 | 0.94 | 0.96 | 95 |
| **Accuracy** |  |  | 0.99 | 836 |
| **Macro avg** | 0.98 | 0.97 | 0.98 | 836 |
| **Weighted avg** | 0.99 | 0.99 | 0.99 | 836 |

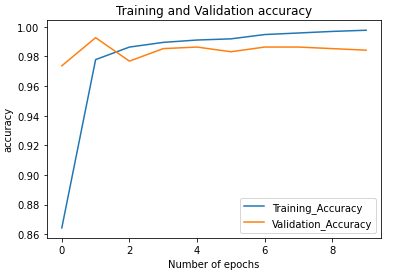
F1 SCORE = 0.95698

LOG LOSS = 0.33051

**GRAPHS:**

****

**(i)**

****

**(ii)**

**BiLSTM:**

**CONFUSION MATRIX:**

|  |  |  |
| --- | --- | --- |
| **Predicted/Actual Values** | **Ham** | **Spam** |
| **Ham** | 740(TP) | 1 (FP) |
| **Spam** | 6 (FN) | 89 (TN) |

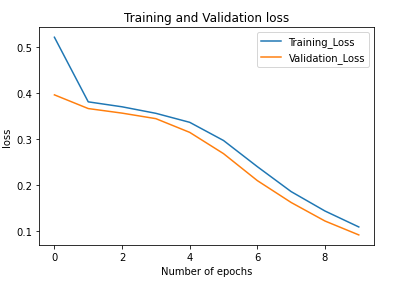
**CLASSIFICATION REPORT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1\_Score** | **Support** |
| **0(Ham)** | 0.99 | 1.00 | 1.00 | 741 |
| **1(Spam)** | 0.99 | 0.94 | 0.96 | 95 |
| **Accuracy** |  |  | 0.99 | 836 |
| **Macro avg** | 0.99 | 0.97 | 0.98 | 836 |
| **Weighted avg** | 0.99 | 0.99 | 0.99 | 836 |

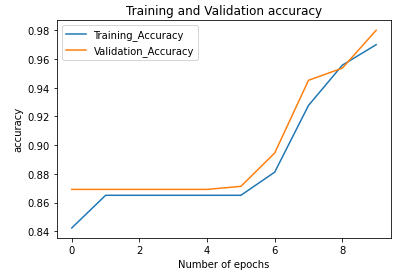
F1 SCORE = 0.96216

LOG LOSS = 0.28920

**GRAPHS:**

****

**(iii)**

****

**(iv)**

**DENSE TEXT CLASSIFER:**

**CONFUSION MATRIX:**

|  |  |  |
| --- | --- | --- |
| **Predicted/Actual Values** | **Ham** | **Spam** |
| **Ham** | 738(TP) | 3 (FP) |
| **Spam** | 17 (FN) | 78 (TN) |

**CLASSIFICATION REPORT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1\_Score** | **Support** |
| **0(Ham)** | 0.98 | 1.00 | 0.99 | 741 |
| **1(Spam)** | 0.96 | 0.82 | 0.89 | 95 |
| **Accuracy** |  |  | 0.98 | 836 |
| **Macro avg** | 0.97 | 0.91 | 0.94 | 836 |
| **Weighted avg** | 0.98 | 0.98 | 0.98 | 836 |

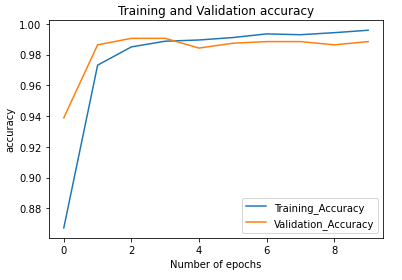
F1 SCORE = 0.88636

LOG LOSS = 0.82628

**GRAPHS:**

****

**(v)**

****

**(vi)**

**COMPARSION OF MODELS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **MODEL** | **Loss** | **Training Accuracy** | **Testing Accuracy** |
| LSTM | 0.056 | 99.997 | 99 |
| BiLSTM | 0.068 | 99.82 | 99.2 |
| Dense-Text | 0.115 | 98.02 | 97.8 |

* From the given tables it clear that Dense text classifier is the worst model out of the 3 models with a test accuracy of 97.6% and an extremely high log loss of 0.82628.
* Both LSTM and BiLSTM have extremely high test accuracy of 99% and 99.2% respectively.
* Since values are too close we cannot decide which is the best model among LSTM and BILSTM based on test accuracy, so we look at other parameters.
* BiLSTM has a better F1 score (0.96216) than LSTM (0.95698).
* BiLSTM has lower log loss (0.28920) than LSTM (0.33051).
* Log loss and F1 score are widely used evaluation parameters to distinguish between models having same or extremely close accuracies.
* So based on Log loss and F1 score values BiLSTM is the best model out of the three deep learning models.

**CONCLUSION:150 Words**

* Summarize the work
* Conclude why your method succeeded

Each model performed well in some performance metrics; Bi LSTM performed the best out of the 3 with 99.2% accuracy. Dense text classifier found to be performing the least with 98% accuracy.

In this research, we propose SMS spam classification models based on deep learning algorithms including LSTM, Bi LSTM and Dense text classifiers . We used NLP techniques for pre-processing SMS text data into sequence using word tokenization, padding data, truncating data and word embedding technique. In addition, we developed model based on deep learning algorithms including LSTM, Bi LSTM and Dense text classifiers . Finally, we evaluated models using test set spilt from SMS spam dataset. The results show that the performance of the Bi LSTM model outperforms other models with 99.2% accuracy. In addition we found LSTM has least loss of 0.056 with 99.997% training accuracy .Moreover, including LSTM, Bi LSTM and Dense text classifiers, which are deep learning algorithms provide a better performance than the model based on machine learning algorithms including Support Vector Machine and Naïve Bayes. This research was study-case about developing SMS spam classification model based on deep learning algorithms. For future works, we aim to enhance the performance of the model by collecting more data from various source to develop model. We expect to develop the model that can be used to help people in real-world.

**REFERENCES:**

* S. Gadde, A. Lakshmanarao and S. Satyanarayana, "SMS Spam Detection using Machine Learning and Deep Learning Techniques," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 358-362, doi: 10.1109/ICACCS51430.2021.9441783.
* P. Navaney, G. Dubey and A. Rana, "SMS Spam Filtering Using Supervised Machine Learning Algorithms," 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018, pp. 43-48, doi: 10.1109/CONFLUENCE.2018.8442564.
* Suparna DasGupta, Soumyabrata Saha, Suman kumar Das, "SMS Spam Detection using Machine Learning, " 2021 Journal of Physics: Conference Series, 2021, doi:10.1088/1742-6596/1797/1/012017
* Luo GuangJun, Shah Nazir, Habib Ullah Khan, Amin Ul Haq, "Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms", Security and Communication Networks, vol. 2020, Article ID 8873639, 6 pages, 2020. <https://doi.org/10.1155/2020/8873639>.
* P. Sethi, V. Bhandari and B. Kohli, "SMS spam detection and comparison of various machine learning algorithms," 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), 2017, pp. 28-31, doi: 10.1109/IC3TSN.2017.8284445.
* Alzahrani and D. B. Rawat, "Comparative Study of Machine Learning Algorithms for SMS Spam Detection," 2019 SoutheastCon, 2019, pp. 1-6, doi: 10.1109/SoutheastCon42311.2019.9020530.
* Samadhan M.Nagre, Mobile SMS Spam Detection using Machine Learning Techniques, Journal of emerging technologies and innovative research,2018.
* Nilam Nur Amir Sjarif, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat , “SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm” , Procedia Computer Science, 2019, DOI:10.1016/j.procs.2019.11.150
* Pumrapee Poomka, Wattana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop, “SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit”, International Journal of Future Computer and Communication, Vol. 8, No. 1, March 2019, doi: 10.18178/ijfcc.2019.8.1.532.
* Alaa Mohasseb and Benjamin Aziz and Andreas Kanavos, “SMS Spam Identification and Risk Assessment Evaluations”, WEBIST, 2020, doi=10.5220/0010022404170424.

BIO DATA:

Name: Jeevarathinam N

Mobile: 91 9962641644

Email: [jeevarathinam.n2019@vitstudent.ac.in](mailto:jeevarathinam.n2019@vitstudent.ac.in)