# VISVESVARAYA TECHNOLOGICAL UNIVERSITY
# BELAGAVI -590002

**A**

**PROJECT REPORT ON**

# "COMBINATIONAL MODELS FROM PERSUASION CUES FOR PHISHING EMAIL DETECTION"

**A Project submitted to Visvesvaraya Technological University in partial fulfillment of requirements**

**For the VIII$^{th}$ Semester**
**Computer Science &Engineering**

**Submitted By**


| NAME | USN |
|---|---|
| AKSHATA HOSKATTA | 2GP18CS002 |
| H DHARANI | 2GP18CS011 |
| KEERTHANA K K | 2GP18CS013 |
| POOJA SHIVANAND SUNADALE | 2GP18CS019 |


**Under the Guidance of**

**Prof. SOMESHA M**
**Assistant Professor &**
**HOD Dept. of CSE**


**Department of Computer Science & Engineering**

**GOVERNMENT ENGINEERING COLLEGE**

**KARWAR, MAJALI-581345, DIST. UTTARA KANNADA,**

**KARNATAKA**

# GOVERNMENT ENGINEERING COLLEGE

# KARWAR, MAJALI–581345, DIST. UTTARA KANNADA, KARNATAKA

## Department of Computer Science and Engineering

# CERTIFICATE

***Certified that the Project work entitled***

## "COMBINATION MODELS FROM PERSUASION CUES FOR PHISHING EMAIL DETECTION"

***This is to certify that***

| | |
|---|---|
| **AKSHATA HOSKATTA** | **2GP18CS002** |
| **H DHARANI** | **2GP18CS011** |
| **KEERTHANA K K** | **2GP18CS013** |
| **POOJA SHIVANAND SUNADALE** | **2GP18CS019** |

*Have satisfactorily completed the project work for the partial fulfillment of Bachelor of Engineering in* ***COMPUTER SCIENCE ENGINEERING*** *of the Visvesvaraya Technological University, Belagavi during the year 2021- 2022*

| | |
|---|---|
| **Prof. SOMESHA M** | **Dr. SHANTHALA B** |
| **Assistant Professor** | **Principal** |
| **HOD Dept. Of CSE** | **GEC, Karwar** |
| **GEC, Karwar** | |

**EXTERNAL**

| **Name of the Examiners** | **Signature with date** |
|---|---|
| **1……..** | **……** |
| **2……..** | **……..** |

# DECLARATION

We hereby declare that the project work entitled "**Combinational Models from Persuasion Cues for Phishing Email Detection**" has been independently carried out by us under the guidance of **Prof. SOMESHA M**, Assistant Professor & HOD, Department of Computer Science and Engineering, Government Engineering College Karwar, in partial fulfillment of the requirements of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi.

We further declare that we have not submitted this report either in part of in full to any other university for the reward of any degree.


| | | |
|---|---|---|
| AKSHATA HOSKATTA | - | 2GP18CS002 |
| H DHARANI | - | 2GP18CS011 |
| KEERTHANA K K | - | 2GP18CS013 |
| POOJA SHIVANAND SUNADALE | - | 2GP18CS019 |


Place: Karwar
Date:

# ACKNOWLEDGEMENT

We give our high, respectful gratitude to our beloved guide **Prof. SOMESHA M,** HOD, Department of Computer Science and Engineering, who has been our source of inspiration. He has been especially enthusiastic in giving his opinions and critical reviews. We have learnt a lot throughout this semester with many challenges yet valuable experience in order to complete this task. We will remember his contribution forever.

We thank our beloved Principal **Dr. SHANTHALA B** for her constant help and support throughout. Our special thanks to Faculty members who have supported us in making this project a successful one.

We also take this opportunity to thank the technician staffs who have helped us a lot in providing the software and any kind of help whenever needed. My thanks and appreciations also go to my friends who have willingly help us with their ability.

| | |
|---|---|
| AKSHATA HOSKATTA | 2GP18CS002 |
| H DHARANI | 2GP18CS011 |
| KEERTHANA K K | 2GP18CS013 |
| POOJA SHIVANAND SUNADALE | 2GP18CS019 |

# ABSTRACT

Phishing is a fraudulent attempt to obtain sensitive information from an unsuspecting victim. According to recent research, phishers commonly use persuasion techniques to generate positive responses from their victims. Now by focusing on persuasion cues, we are building deep learning models with meaningful gain persuasion cues, loss persuasion cues, and combined gain and loss persuasion, and compare the results to a baseline model that ignores the persuasion cues. Our research may reveal that the phishing detection models incorporating relevant persuasion cues beat the baseline model in terms of F-score, accuracy, precision and recall indicating that they are reliable approaches for phishing detection. The goal of this research is to present the best model obtained after comparative estimation of the results of other models. This type of research is required to obtain newest best technique to counterattack new phishing attacking mechanisms emerging every day.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Phishing is an act of social engineering to obtain information from an unsuspecting victim. It involves an attacker who generally masquerades as a legitimate institution to trick users into disclosing sensitive information that can later be used in fraudulent activities. More than 100,000 Internet users around the world are subjected to phishing attacks daily. Phishing emails have become more complicated and harder to detect in recent years. Hackers are using more complex methods of attack to foil their victims, as humans are at the front line of defense in some cases against phishing emails; they are the key point of contact for an attacker to attack.

In our proposed system we modeled the Persuasion cues for phishing email detection using word embedding techniques, with the help of deep learning algorithms. It examines the effectiveness of persuasion cues for phishing email detection. Persuasion cues are signals within phishing emails through which phisher attempt to persuade and influence readers. We take a text mining approach to explore two types of persuasion cues, namely gain and loss persuasion cues. We are proposing word embedding models implemented on phishing and legitimate emails with any of the predicted cues. Our approach is to create different subsamples from a training set, build a classifier on each set, and compare estimates.

We are building a Phishing email detection model on a deep learning algorithm consisting of Convolutional Neural Networks (CNN), long short-term memory (LSTM), and Bi-directional short-term Memory (Bi-LSTM) because deep learning is an accurate way in which data can be categorized and tagged with efficiency. The Confusion matrix is a relevant measure to use to evaluate the performance of the phishing detection model. The evaluation matrices employed are overall accuracy and loss. We aim to estimate the effectiveness of persuasion cues in phishing email detection and predicting legitimate and phishing emails using data visual representation which will be covered in the next chapter with detail.

# CHAPTER 2

# PROBLEM STATEMENT

- Prior phishing research has concentrated on the structural characteristics of phishing messages. For instance, they're concentrating on classifying phishing emails based on the sender's IP addresses, URLs, or domain names being manipulated.
- Researchers analyzed the content of phishing emails, looking for the presence of logos, spelling, and grammatical errors. For interpretability issues recent phishing detection literature has concentrated on content mining rather than theories to explain the detection process.
- Since phishers create phishing content in a variety of ways, data-driven tactics are less likely to be generalizable. This difficulty is exacerbated when phishers manipulate victims' responses while presenting themselves as authentic and genuine, making it difficult for filters to identify phishing information as fake and leaving users exposed to phishing communications.
- As a result, many security analysts at security operations centers must do manual tasks such as reporting emails as phishing content. As a result, automated explainable techniques for phishing detection, as well as qualitative and quantitative phishing data analysis are necessary.

# CHAPTER 3

# OBJECTIVES

The project aims to classify phishing emails by testing different deep learning models. We acquired persuasion labels of gain based on reciprocity, consistency, and liking ability and loss cues based on loss, severity, and immediacy which are thesix basic tactics of persuasion used to motivate the individual with the help of word embedding model (which predicts words based on statistical prediction) are estimated on its effectiveness. Recently, researchers have investigated the role of persuasion cues in phishing susceptibility and victim behavior. We extend this investigation by gauging the effectiveness of persuasion cues for phishing email detection. The effectiveness of persuasion cues for phishing email detection can be measured using word embedding and it explores two types of persuasion cues. And grouped features created by those persuasion cues are applied to the deep learning algorithms. To solve this problem this literature has proposed deep learning algorithms such as 1D-CNN, LSTM, and Bi-LSTM with vectors for phishing email detection models with improved accuracy.

# CHAPTER 4

# LITERATURE SURVEY

## Existing Work

Valecha *et al*. [1] Author proposed persuasion cues for detecting phishing emails. In this, they want to know the effect of Persuasion cues. For this, they used 3 ML models, with relevant gain & loss persuasion and combined gain and loss persuasion cues respectively. They collected the data from Millersmile and Enron corpus. In ML they used Bi-LSTM and W2V models. Used 4 ML algorithms NB, LR, RF, and SVM. ML requires heavy labor for the generation of features. So, they have gone for the DL techniques. Depending upon the types of persuasion cues all the algorithms give accuracy.

Sallom *et al.* [2] these approaches are subjected to comparative assessment and analysis. This problem is its immediate solution space. This survey conveys 4 aspects ofphishing attacks. That communication media, target devices attack tech and counter-measures. Data collection is done by automated data collection and manual data collection techniques are the two types of data collection techniques. The conventional approach is for phishing attacks can recognize only about 20 % of phishing attacks. ML is good.

Ayman *et al.* [3] Author propose a benchmarking framework called 'Phish Bench', study says that how dataset characteristics vary legitimate and phishing ratio increasing the size of an imbalanced dataset. They implement a flexible and customizable benchmarking framework. ML algorithm for supervised learning, deep learning, and imbalanced learning. They collect the dataset from websites like PhishTank, APWG, and openphish. A combination of openphish and Alexa gives high accuracy of 99.23%, but it works accurately with a header only.

Christopher *et al*. [4] Detecting New Forms of Phishing Attacks algorithms use a system called SAFE-PC. To evaluate SAFE-PC, they collect a large corpus of the data set from the central IT organization at a tier-1 research university. SAFE-PC can detect more than 70% of the emails with help of Sophos, its email filtering tool. For the smaller size datasets, accuracy is low.

Abiramasundari *et al.* [5] the author proposed two techniques one is rule-based subject analysis (RBSA) and another semantic based feature selection (SBFS) is integrated

With the ML algorithms. RBSA and SBFS are integrated with four classifiers namely support vector machine, multinomial naive Bayes, gaussian naive Bayes, and Bernoulli naive Bayes. RBSA proposed for analyzed spam terms and SBFS proposed to reduce the number of features required for the classification of phishing emails. NB and SVM yield 96% accuracy.

Isra's *et al.* [6] uses the effectiveness of word embedding. They used a pre-trained transformer model BERT for detecting spam emails from non-spam emails. The outcome of these is compared to the baseline model. This model contains a Bi-LSTM layer and two stacked dense layers. Also, the outcome is compared with KNN and NB. This approach attained the highest accuracy of 98.67% and 98.66% F1 score. It is applied only for 300 sequence lengths of data. In the future results be improved by taking larger input sequences.

Sami. *et al*. [7] author proposed NN and reinforcement learning algorithm. This method successfully classified and identified approximately 98.6% of phishing emails selected from the test data set with a low false positive rate of 1.8%. the neural network has been implemented using DNN. This work reduces the need for feature best performance architecture that can be adapted to new problems easily. DNN is computationally expensive to train data.

Harikrishna *et al*. [8] make use of distributed representation term frequency-inverse dense frequency (TF-IDF) for numerical representation and comparative study of classical machine learning techniques like the random forest, Adboost, Naïve Bayes, Decision tree, and Support vector machine. They have worked on two groups one email with a header dataset and another email with no header. Because of the unbalanced dataset, the test result is over fitted and can enhance by adding an extra data source. Dueto computational constraints, the author could not work on deep learning.

Hiransh *et al*. [9] used Keras word embedding and a convolutional neural network. Keras embedding provides an embedding layer that can be used on test data. The convolutional neural network has several layers which will be having many filters, whose output is combined to get results. The work provides high accuracy of 96.8% for a dataset containing email without a header and an accuracy of 94.4% dataset with a header. This performs well for those without using an external dataset. Hence, he concluded that the proposed work will increase the detection rate of phishing emails by adding more features.

Nguyen *et al*. [10] proposed a deep learning model with hierarchical LSTM and supervised attention for anti-phishing. The author proposed a hierarchical LSTM model at the word level and sentence level. The result of both will be combined using supervised attention it was used to assign the contribution weight to each word in the sentence. The authors have used the dataset in the email body and another dataset with a header, he has proposed a comparative study between H-LSTM and H-LSTM+ supervised attention and it outperforms the baseline model. And this work can be improving its performance by using email frequency ranking of words in the vocabulary of supervised attention.

Rahman *et al*. [11] The authors have used Bi-STM and word embedding to analyze the sentimental and sequential property of text and used CNN to speed up training time and extract high-level text features, they have used two kinds of data set namely ling spam and spam text message classification dataset and they have adopted recall, precision, and f- score for comparing and evaluating performance. And they have improved their performance of accuracy by about 98-99%. A combination of Bi-RNN and LSTM gives the Bi LSTM it performs based on both previous contents and after the context.

Dilhara *et al*. [12] have worked on deep learning models namely CNN (1D), LSTM and GRU. Along with deep learning models namely GRU + LSTM, LSTM –LSTM. Bi (GRU) – LSTM, Bi (LSTM) – LSTM. Dataset has taken from Mendeley data; mainly dataset contains total 111 attributes which indicate whether the specific instance is legitimate or phishing. Confusion matrix for evaluation which gives the count of TP, TN, FP, and FN and enhances the evaluation performance by matrices like accuracy, precision, recall, and F-score. Among all the combinations the Bi (GRU)-LSTM was identified as the best performing model and had an accuracy of 94%. But this combination takes a longer time for the training model; improvement can be done by adding bagging and boosting.

Soni *et al*. [13] proposed a Spam e-mail recognition model called THEMIS, which begins by examining the email structure with an improved RCNN model with staggered vectors. Word2vec is used to get arrangements of vectors with four levels of word-level email header, char-level email header, word-level email and body, and char-level email body which are inputted to word vector models. When this THEMIS is compared to CNN and LSTM, this performed well in TP, TN, FP, FN, and FPR (decreases danger of filtering out genuine emails). But this system produces a model which is restricted to the number. The experimental results show the overall accuracy of THEMIS.

Fette *et al.* [14] the author gives a method for detecting phishing attacks. These methods are present with slight modifications, to the detection of phishing websites, or the emails used to direct victims to these sites. Here the internal and external information is going to be combined and then created into a compact representation called a feature vector. A collection is used to train a model. The picture in an MML-based approach classification. It gives an accuracy of 99.5%.

Amit *et al.* [15] Comparative study between NB and NN classification spam email detection the performance of NB is more accurate compared to others. The implemented form of NN is multi-layer processing. NN approach is more refined, more mathematical, and potential.

Abdul *et al.* [16] provide a literature review of AI techniques ML, DL, Scenario-based techniques, and hybrid learning. They used DL approaches which suggested the classification of phishing websites using DNN. The classifiers which give a good phishing attack detection accuracy are c4.5, KNN and SVM. Scenario-based phishing attack detection; techniques used to detect a phishing attack. ML approach gives better results but with trade-offs and time consuming even on the small size of the database.

Fong *et al*. [17] use THEMIS's RCNN model improved by Bi-LSTM with multi-vectors. They employed the Attention mechanism where its goal is to select information more valuable to the goal of the current task with weights. Char-level embedding model and word-level embedding model are obtained by Word2vec. RCNN has a Long-term dependency problem, and to address that LSTM has been applied. Although n BRNN is the replacement of the original RCNN by Bi-LSTM to obtain left and right semantic information with word embedding, each word is represented as triples. Therefore, the noise will be minimal than the original THEMIS. Although the LSTM model can capture certain semantic information but not deep semantic information, it makes it difficult. Using both word-level and char-level makes the model more comprehensive to capture feature details.

Hajek *et al.* [19] proposed a model DBB- RDNN-REL which worked well with high-dimensional imbalanced and non-linear spam datasets. It uses N-gram and TF-IDF for feature selection whereas it uses unigram and bigram with a two-level hidden layer which was sufficient for our model. The problem is to handle imbalanced datasets and that is done by a modified version of the DBB algorithm which creates a new artificial l dataset by using a probability distribution. This model proves deep learning not only

Increases accuracy but also outperforms state-of-art. The results show DNN model outperforms.

Srinivasam *et al.* [18] proposed the DeepSpamNet framework which is a scalable and robust context-based spam detection frame that can process a large volume of data. Optimal features are extracted from DL algorithms and passed to the next layer consisting of a linear combination of input followed by a 'sigmoid' activation function [18] use database to collect emails that are used by distributed log parser and output is fed to DeepSpamNet framework. Skip-gram model by Word2vec is used for feature extraction. Both FPR and FNR are is measured and got very low values. DeepSpamNet overcomes drawbacks like the need for a domain-level expert for continuous insights into databases. The conclusion is that Deep Learning with Word Embedding performs very well and the proposed DeepSpamNet which uses CNN-LSTM pipelines to detect phishing with the daily email flow. Future work is to develop an optimal cost-sensitive DL architecture.

## SUMMARY

| S L. NO | AUTHOR | APPROACH /METHOD | CONTRIBUTION | DRAWBACK | EVALUATETION RESULT |
|---|---|---|---|---|---|
| 1 | Valecha | NB, SVM, LR, RF with W2V | Persuasion cues give better performance compared to baseline | ML requires more labor work for feature generation | Accuracy depends on thetypes of persuasion cues |
| 2 | Sallom et al. | ML and DL, RF,DT, LR, SVM, Black listing | It provides an organized guide to the current state of the literature. | Threats in phishing emails cause financial losses | THEMIS gives Accuracy of99.84% |
| 3 | Ayman et al. | Random forest | Phish bench which evaluates and compare existing feature for a phishing attack | time changes depending upon the size of the data | Random Forest accuracy 99.23% |
| 4 | Christophe et al. | Machine learning algorithm with SAFE-PC | Can evaluate large corpus of the data set | Cannot suitable for a small number of data set | 70% accuracy using SAFE-PC method |
| 5 | Abhirama sundahari et al. | Machine learning algorithm SVM and NB | analyze the spam and reduce the number of features required | challenging and time-consuming | Accuracy 96% |
| 6 | Isra et al. | Bert, CNN, Bi-LSTM, LSTM | Performance better than baseline | Results can be improved bylarge input sequence | 98.43% and 98.66% score |

| | | | | | |
|---|---|---|---|---|---|
| 7 | Hiranshetal. | Keras word embedding with convolution matrix | The use of CNN provides faster training andgives high accuracy | Cannot perform with the external dataset | Performance accuracy of 94.2% |
| 8 | Nguyen et al. | Hierarchical LSTMs and supervised attention | HLSTM+ supervised attention outperforms baseline | Improve performance with an email frequency ranking | HLSTM +supervised attention performs over HLSTM |
| 9 | Rahman et al | Bi-LSTM with CNN and word embedding. | Improved performance with Bi LSTM | dealing with deep learning models is the overfitting | Performed with high accuracy of98%-99% |
| 10 | Dilhara et al. | Hybrid of GRU-LSTM, LSTM-LSTM, Bi (GRU)-LSTM, Bi (LSTM)-LSTM | Combination of Bi (GRU)-LSTM perform best | A combination of LSTM and GRU takes a longer time to train | Bi(GRU)-LSTM best performing model with 94% |
| 11 | Abdul et al. | DL, ML, Scenariosbased technique, a hybrid technique | Provide literature review of AI techniques | Phishing techniques are not effective | Accuracy of 97% |
| 12 | Amit et al. | NB and NN which implemented MLP-supervised learning | NB give saccurate accuracy | NN and MLP don't have better Performance. | NN approaches are more reliable in accomplishing the task. |
| 13 | Fette et al. | ML and pilfer | Combines internal And external feature | Performance can be increased by adding 5 filter solution | Accuracy of 99.5% |

| | | | | | |
|---|---|---|---|---|---|
| 14 | Soni et al. | RCNN with Multi-level vectors | THEMIS model | Computation is restricted to the number | THEMIS model gave accuracy of 99.8%. With Low FPR. |
| 15 | Fong et al. | Improved RCNN with Bi-LSTM, LSTM, and Attention mechanism. | Improved THEMIS model with triple representation for each word. | To work on the only email body. | High TP and TN<br>Low FP, FN<br>And FPR. |
| 16 | Srinivasan et al. | CNN and LSTM with FastText and Keras | DeepSpamNet | Cost-sensitive models to be build. | 0.1 > FPR and FPN |
| 17 | Hajek et al. | DBB-RDNN-REL<br>With N-gram | Optimal feature selection | High computational<br>l expenses | Accuracy obtained 98.76% |
| 18 | Bagui et al. | LSTM, CNN with Word embedding and one-hot coding. | Stated accuracy is better with word phrasing. | It is just a comparison study of different models. | Accuracy<br>LSTM 96 .64%<br>CNN-97.20%<br>Word Embedding 98.89% |
| 19 | Sami et al. | DNN and Reinforcement learning | architecture can adapt easily, the best performance | expensive to train, work only for large dataset | Phishing emails identify approximately 98.6% with low FP1.8% |
| 20 | Harikrishna et al. | TF-IDF for numeric representation and ML learning | Better performance with higher accuracy than baseline | It takes more time and manpower | The effectiveness of combination results varies |

# CHAPTER 5

# SYSTEM DESIGN

## 5.1 PROPOSED WORK

Even though email detection exists for many years, detecting email is still a very challenging problem because new attacking mechanisms are emerging every day. In the Proposed system, here are introducing deep learning techniques that surpass the rent trend of machine learning methods which rely on feature engineering to generate features represented in the emails, which may require heavy manual labor and domain expertise. To solve this problem this literature has proposed a deep learning algorithm such as CNN, LST, and Bi-LSTM with multilevel vectors for phishing email detection models with improved accuracy. We are focusing more on the context of the words which gives us accurate meaning. We are estimating the results obtained by different combinational models and presenting the best out of all models in terms of accuracy and loss.

## 5.2 MODELS USED

**1. Word2Vector**

Word embeddings is a technique where individual words are transformed into a numerical representation of the word (a vector). Where each word is mapped to one vector, this vector is then learned in a way that resembles a neural network. The vectors try to capture various characteristics of that word about the overall text. These characteristics can include the semantic relationship of the word, definitions, context, etc. With these numerical representations, you can do many things like identify similarities and dissimilarities between words. In this study, we choose the model them. The effectiveness of Word2Vec comes from its ability to group vectors of similar words. Given a large enough dataset, Word2Vec can make strong estimates about a word's meaning based on its occurrences in the text. These estimates yield word associations with other words in the corpus.

**2. Convolutional Neural Network (CNN)**

- The data is squashed down into small sentiments because it learns sentiments more specific.

- Three main features: Filtering, Pooling, and Flattening. This may increase accuracy due to the detail in which the data is being processed.

**3. Long short-term memory (LSTM)**

- Sequentially process any input lengthier with historical memory.
- Learns quickly and takes in historical information from the output and feeds it to the next test data as input.

**4. Bidirectional Long short-term memory (Bi- LSTM)**

- LSTM with both the directions: - gate passes right to left then left to right, giving it the full context of word placement

## 5.3HIGH-LEVEL ARCHITECTURE

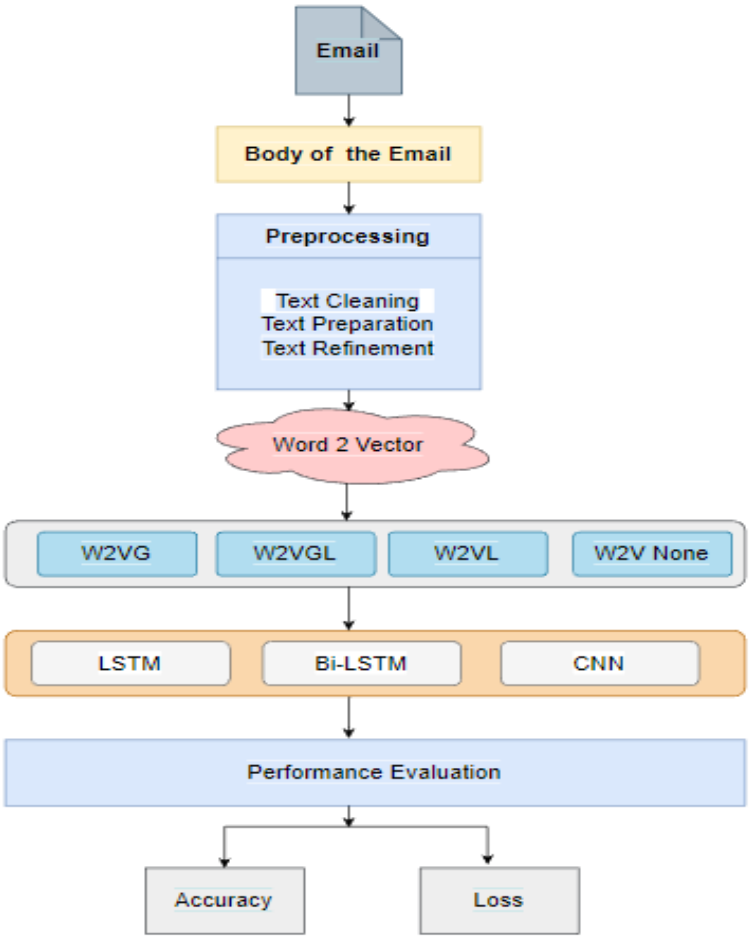


**Fig.1. High-level architecture**

## 5.4 METHODOLOGY OF WORKING

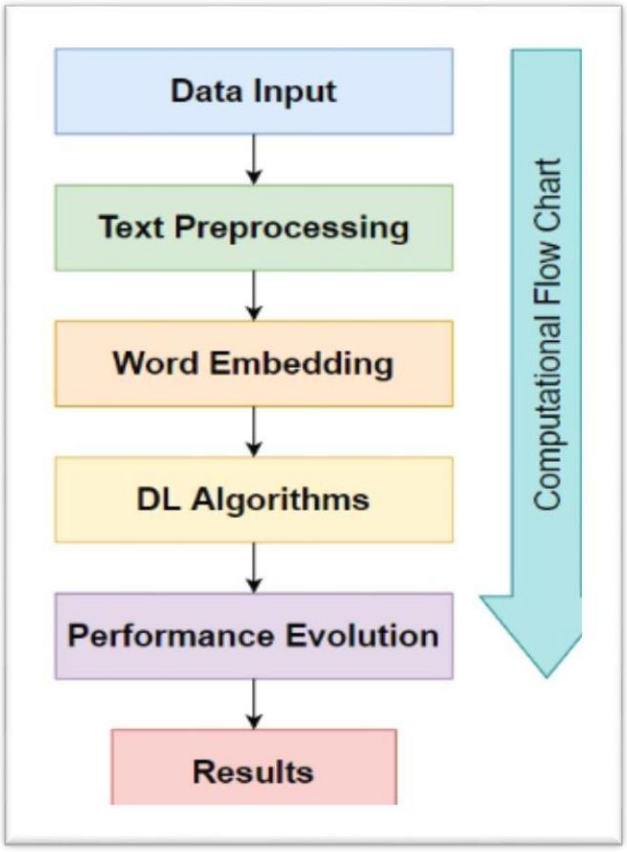


**Fig.2.Working Flow**

- Review the electiveness of different deep learning methods used for natural language processing.
  Create a dataset of phishing and legitimate emails.
- Implementing each model and running on a GPU for maximized performance.
- Review the test data and test the model's detection accuracy. Give the model example email to see if it can detect a phish or a non-phishing email.
- Record and review the overall results during the testing stage of the project.

The four models selected were based on the literature review. Most studies on deep learning compared LSTM models with CNN models and bidirectional models, with different architecture; hence the decision for the Bi-LSTM model. Studies tested the effectiveness of an LSTM model and different types of CNN and RNN models, based on their validation and accuracy. The testing methodology was based on a study of deep learning models for sentiment analysis, which fed the model's data to predict. This study will use phishing emails to see if models can determine they are phishing email or not by analyzing the sentiment from the corpus text. The results gathered will

Be in terms of Loss and Accuracy which have been referencedthrough the literature review.

- For estimation of the results of different combinational models, First create a dataset of legit and phishing emails.
- Word embedding mechanisms.
- Extracting text features.
- Segregation in terms of persuasion cues and developing four text feature models.
- Applying those models into Deep Learning Algorithms in replacement of Machine learning models.
- Classifies emails as phishing emails and legitimate emails when testing data is given.
- Combinational computations are done by combining different feature models with different DL algorithms.
- Performance measure is done to estimate the effectiveness of persuasion cues on our deep learning algorithms.
- Estimation of effectiveness is represented by Loss and Accuracy,values.
- Comparison is done and outputs the results which will be the best combinational model is more accurate.

# CHAPTER 6

# SYSTEM REQUIREMENTS

## 6.1SOFTWARE REQUIREMENTS

### 1. PYTHON

Python as a language has a vast community behind it. Any problem which may be faced is simply resolved with a visit to Stack Overflow. Python is among the foremost standard language on positioning which makes it very likely there will be a straight answer to any question Python has an abundance of powerful tools prepared for scientific computing Packages like NumPy, Pandas, and SciPy area unit are freely available, and, well documented. Packages like these will dramatically scale back, and change the code required to write a given program. This makes iteration fast. Python as a language is forgiving and permits programs that appear as pseudo-code. This can be helpful once the pseudo-code given in tutorial papers must be enforced and tested.

Using python this step is sometimes fairly trivial. However, Python is not without its errors. The language is dynamically written and packages are area units infamous for Duck writing. This may be frustrating once a packaging technique returns one thing that, for instance, looks like an array instead of being an actual array. Plus the fact that standard Python documentation does not clearly state the return type of a method, can lead to a lot of trials and error testing that will not otherwise happen in a powerfully written language. This is a problem that produces learning to use a replacement Python package or library more difficult than it otherwise may be.

### 2. NUMPY

Numpy is a python package that provides scientific and higher-level mathematical abstractions wrapped in python. It is [19] the core library for scientific computing, that contains a strong-dimensional array object, and provides tools for integrating C, C++, etc. It is additionally useful in linear algebra, random number capability etc. NumPy's array type augments the Python language with an efficient data structure used for numerical work. Numpy additionally provides basic numerical routines, like tools for locating Eigenvector. 18CP812 Leaf Disease Detection Using CN 19 .

### 3. TENSORFLOW

TensorFlow is an open-source software library for numerical computation using data flow graphs. Nodes inside the graph represent mathematical formulas, whereas the graph edges represent the multidimensional knowledge arrays (tensors) communicated between them. The versatile architecture permits you to deploy computation to at least one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. . Tensor Flow was originally developed by researchers and engineers acting on the Google Brain Team at intervals Google's Machine Intelligence analysis organization for the needs of conducting machine learning and deep neural networks research, however, the system are general enough to be applicable in a wide range of alternative domains as well.

### 4. PANDAS

A panda is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. A panda is fast and it has high performance & productivity for users.

### 5. KERAS

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with attention to enabling quick experimentation. Having the ability to travel from plan to result with the smallest amount of doable delay is key to doing great research. Keras permits straightforward and quick prototyping (through user-friendliness, modularity, and extensibility). Supports each convolutional network and recurrent network, furthermore as combinations of the two Runs seamlessly on CPU and GPU. The library contains numerous implementations of usually used neural network building blocks like layers, objectives, activation functions, optimizers, and several tools to create operating with image and text data easier. The code is hosted on GitHub, and community support forums embody the GitHub issues page, a Glitter change, and a Slack channel.

## 6.2 SYSTEM SOFTWARE REQUIREMENTS

A software requirements specification (SRS) is a document that details how the program will operate and how it will be expected to perform successfully in a certain environment. Additionally, it will describe the primary functions that the product must provide for all corporate stakeholders as well as any customer needs. The Software which was needed in the development is as given below:

Operating System: Windows family

Technology : Python 3.6

IDE : Spyder/Jupyter Notebook

## 6.3 SYSTEM HARDWARE REQUIREMENTS

Computer hardware requirements outline the functional requirements of a piece of hardware as well as its capability. Additionally, it includes information about the processor's speed, type, and manufacturer. Processor speed, which is typically listed in gigahertz (GHz). The technical descriptions of the parts of computers and their capabilities are expressed in computer hardware specifications. The Hardware which was needed to develop the needed software is as listed as shown below:

Processer: Any Update Processer

RAM: 8GB

Hard disk: 1TB

# CHAPTER 7

# IMPLEMENTATION

In this section, we first go into great depth on the methodology used to obtain the data. After that, we go over our datasets and the steps taken to convert free-form email data into a format suited for critical analysis. Then, we define our coding strategy and level of coding dependability. In the end, we present information on the phishing detection model.

## 7.1 Data Collection

We collected email data sets from the anti-spam platform SpamAssasin, the Apache Software Foundation; SpamAssassin is an open-source email filtering software that applies advanced testing and analytical tools to the headers and body text of messages to determine the likelihood of them being spam. Through which we have downloaded and extracted two open source data sets which are in the form of mbox file format. The first data set is the open source Easy ham data set from the anti-spam platform SpamAssassin. The data set contains 5351 emails.

The second data set is the open source hard ham data set from Spam Assassin which contains 500 emails. A new balanced training data set is created by merging emails from the two data sets Easy Ham and Hard Ham with ensuring no duplicate records. The new data set contains 3758 emails.

## 7.2 Data preprocessing

We go over the preparation procedures used on the dataset in this subsection. Because email content is so noisy, preprocessing is crucial. We extracted the contents of the raw email which was in the mbox format by parsing it over every file and storingit in the database as CSV file. Duplicates from each dataset were relocated again and removed. Three steps—text cleaning, text preparation, and text refinement—were used to preprocess the email content.

The process of text cleaning is deleting duplicate emails that are frequently delivered to numerous recipients in an organizational setting. Using the NLTK Regexp Tokenize software, the email text is segmented into tokens as part of text preparation. The construction and collection of tokens typically neglect grammar. A

``RegexpTokenizer`` splits a string into substrings using a regular expression. In addition to tokenization, punctuations are also removed by Replace the few top punctuation marks (“?”, “.”, “!” etc.). The Next step is converting all the tokens and tokenized sentences into lower-case which helps in reducing the word dimensions in vector model space and while parsing. In addition, stop words like "the" non-alphanumeric this study examines how well gain and loss persuasion cues may identify phishing emails. For this, we need to develop computer-measurable characteristics of persuasion cues for gain and loss. Lexical features are extracted from email content to categorize emails as phishing or legitimate emails. This is done by the process:-The dataset we have got is then searched by every row with the features and if found, we labeled them into the phishing emails, and if not we labeled it into legitimate emails. We segregated the features into basically 2 types, Gain (for instance, click on the link for a $100 gift card) and loss (for instance, the limited-time offer will expire soon).
Letters like "@" are removed during text processing. The pre-processed data is fed into the Word2Vec technology to turn tokens into vectors which can be fed into our Phishing Email Detection model using Deep Learning.

## 7.3Word Embedding

This study examines how well gain and loss persuasion cues may identify phishing emails. For this, we need to develop computer-measurable characteristics of persuasion cues for gain and loss. Lexical features are extracted from email content to categorize emails as phishing or legitimate emails. This is done by the process:- The dataset we have got is then searched by every row with the features and if found, we labeled them into phishing emails, and if not we labeled it into legitimate emails. We segregated the features into basically 2 types, Gain (for instance, click on the link for a
$100 gift card) and loss (for instance, the limited-time offer will expire soon).

We acquired persuasion labels of gain based on reciprocity, consistency, and liking ability and loss cues based on loss, severity, and immediacy which are the six basic tactics of persuasion used to motivate the individual. And respectively these gain persuasion cues and loss persuasion cues were implemented with W2V GAIN and W2V LOSS. On emails containing any of the gain and loss persuasion cues, W2V GAINLOSS was used. Due to the lack of persuasion cues in emails, W2V NONE was implemented as a baseline model. We made sure that the vocabulary used in the W2V models that did not use persuasion cues did not overlap with those used in the W2V

models that did.

The Word2Vector (W2V) model was chosen for the vectorization. W2V learns in advance how to represent words as numeric vectors or word embedding. Each word in the dataset is understood by this trained neural network model in its context. Consequently, W2V performs better than deterministic models like Bag of Words (BOW). In this study, we developed four W2V models: W2V GAIN, W2V LOSS, W2V GAINLOSS, and W2V NONE. W2V GAIN contains relevant gain persuasion cues, W2V LOSS contains relevant loss persuasion cues, and W2V GAINLOSS contains relevant gain and loss persuasion cues. The genuine emails don't contain any relevant gain and loss features and are made as NONE and are modeled in W2V NONE model. Finally, all 4 W2V models experiment with 3 Deep Learning models, and performance is evaluated.

Some of the key features in each of the W2V models are listed table: -

| W2V models | Sample Features extracted and used for W2V models |
|---|---|
| W2V Gain | Free, Reward, Cash, Offer, Fantastic, etc. |
| W2V Loss | Restricted, Blocked, Decline, Error, etc. |
| W2V Gain | Loss Password, Secure, Account, Holder, etc. |
| W2V NONE | Tax, Invoice, Technology, legal, Draft, Letter, etc. |

## 7.4 Deep Learning Models

We investigated and analyzed the effectiveness and applicability of baseline models that use explainable textual elements because we use the structural characteristics of emails. More particular, we demonstrated the effectiveness of persuasion strategies in classifying phishing emails using Deep Learning Models.

In this study, we use to create different subsets of features to build classifiers and compare their estimates. We choose this approach because we want to build a detection model and test the effectiveness of persuasion features. Using this approach we can manipulate the set of input features, namely persuasive vs. non-persuasive,

available to learning algorithms and compare their estimates. If the estimates of the learning algorithm utilizing persuasive features outperform that of the algorithm utilizing non-persuasive features, then we can conclude that persuasion features were effective in phishing email detection. We created four Word2Vector (w2v) models that utilize a different subset of features based on gain and loss persuasion cues.

For generalizability of our findings, we utilized four models within sci-kit- learn, an open source python-based collection of Deep Learning tools, for phishing email classification: Convolutional Neural Networks (CNN), Long short term memory (LSTM), and Bi-directional short term Memory (Bi- LSTM),

**(1) Long short-term memory (LSTM)**: LSTM networks were designed specifically to overcome the long-term dependency problem faced by recurrent neural networks RNNs (due to the vanishing gradient problem). LSTMs have feedback connections which make them different from more traditional feedforward neural networks. This property enables LSTMs to process entire sequences of data (e.g. time series) without treating each point in the sequence independently but rather, retaining useful information about previous data in the sequence to help with the processing of new data points. As a result, LSTMs are particularly good at processing sequences of data such as text, speech, and general time series The LSTM network mainly has four different gates namely, an input gate (it), an output gate (ot), a memory cell mt , a forget gate (ft) and a hidden state (ht). At every timestamp t, a word vector li is given to the LSTM network which processes it and yields an output mi as shown in Figure 3. The first step of the LSTM network is to find the information that is not relevant and throw it away from the cell state. This decision is taken by the very first layer of the network i.e the Sigmoid layer which is called forget gatelayer (ft)

$$ft = \sigma(wf[p(t-1)lt] + bf) \quad (1)$$

Where wf is the weight, p(t−1) is the output from the previous time stamp, lt is the new input message word and bf is the bias. The next step of the network is to decide among the available information what we are going to store for further processing. This is done in two steps i.e., with the help of a Sigmoid layer called input gate (it) and a tanh layer which generate a value (ct) thais t added with the input gate (it) values, the ct and it are calculated by the

$$ct = \tanh(wc[h(t-1), lt] + bc) \quad (2)$$

$$it = \sigma(wi[h(t-1), lt] + bi) \quad (3)$$

Now, the previous cell output p(t−1) is updated to new state pt where pt is defined as (Equation 4)

$pt = ft * p(t-1) + it * ct$ (5)

Finally, with the help of Sigmoid layer, the output ot (Equation 6) of the network is decide, further the cell state ct is pass through the function tanh and multiplied by the output of the Sigmoid function.

$ot = \sigma(wo[h(t-1), lt] + bo)$ (6) $mt = ot * \tanh(ct)$

We use the LSTM network to predict whether the email is a Spam or Not-Spam using the simple model and with the regularization parameter i.e., Dropout..

**(2) Bi-directional short term Memory (Bi-LSTM):** For text sentiment analysis, Recurrent Neural Network(RNN) is very popular and widely used technique. The recurrent neural network (RNN) remembers the previous time step information due to its having memory that facilitates it an advantage over traditional neural networks. The input of RNN is attached with a state vector to make new state vector. The resulting state vector remembers past information back from current. The straight forward RNN follows the following mathematical equations:

$ht = \tanh(Whhht-1 + Wxhxt)$ (1)$yt = Whyht$ (2)

RNN does not have capability of remembering the past sequence much. Apart from it, RNN has vanishing gradient descent problem. Long short-term memory network(LSTM) is a variation of RNN, that is capable of learning long-term dependency and also succeed to solve vanishing gradient descent problem. LSTM is actually developed to resolve long-term dependency difficulty. LSTM has a special property of remembering. Main idea of the LSTM model lies simply in the cell state. The cell state flows straight through the sequence almost unchanged with only some little linear interaction. Another important thing about LSTM is gate. This gates control the information and information is safely added or deleted from the cell stated. LSTM model updates cell using the following equations:

$ft = \sigma(Wf .[ht-1, xt] + bf$

Here, xt refers input, ht refers the hidden state at t time step. The updated cell state Ct is as follows:

$it = \sigma(Wi .[ht-1, xt] + bi)$ (4)

$CT = \tanh(WC [ht-1, xt] + bC )$ (5)$Ct = ft * Ct-1 + it * CT$ (6)

Here, * is a point-wise multiplication operator and we can compute output and hidden state at t time step.

$$ot = \sigma(Wo.[ht-1, xt] + bo) \quad (7)$$

$$ht = ot * \tanh(Ct) \quad (8)$$

LSTM faces difficulty as it only considers the previous contexts from the current. So, both LSTM and RNN can only receive information from the previous time steps. So, to avoid this problem, further improvements are done using the Bidirectional Recurrent Neural Network (Bi-RNN). Bi-RNN [19] can handle two pieces of information both from the front and back. The combination of Bi-RNN and LSTM makes the BiLSTM. Therefore, the benefit of using LSTM is in the form of storage in cell state and Bi-RNN with access to information from the context before and after. Hence, it causes the Bi- LSTM to have the advantage of LSTM with feedback for the next layer. Bi-LSTM has added other important advantages of remembering long-term dependencies

**(3) Convolutional Neural Network**: Convolutional Neural Networks are several layers of convolutions followed by nonlinear activation functions like ReLU. Unlike in additional neural networks where we have fully connected layers, in CNN convolution over input is done to compute the output which results in a local connection. A large number of filters are applied in each layer whose outputs are combined to get the result. Values of filters are learned by CNN during the raining phase. For NLP tasks the input to CNN will be sentences or documents. CNN consists of several parameters such as kernel size, feature map, size of pooling windows, types of pooling such as average, min or max-pooling, activation functions, number of neurons for fully connected dense layer, optimization function, the value of dropout (regularization parameter), learning rate and others. During the training process of the CNN model, the input data was supplied to the network batch-wise. Hence, an epoch consisted of several batches of the training sample. Once an epoch was completed, the loss was computed. If the obtained loss is not desirable, the complete training sample data was again supplied to the network, and the loss was recomputed at the end of the epoch. This process was repeated until the loss was deemed to be acceptable.

Deep learning models for phishing detection have been developed by a number of researchers, including Convoluted Neural Networks(CNN) and Long Short Term Memory (LSTM). Latent features, or autonomously generated features, are produced by these deep learning models and obtained through matrix (tensor) factorization. Although these hidden traits can effectively detect phishing emails, they are frequently challenging to understand. Deep learning algorithms for phishing email detection have improved in explain ability in recent research, which can assist explain why a specific email was flagged as phishing. Deep learning algorithms, however, are unable to link

the persuasive strategies used in phishing emails.

## 7.4 Performance Matrices

To assess prediction algorithms, the confusion matrix is frequently utilized in the literature. A confusion matrix is a useful metric to assess the efficacy of the phishing detection strategy. In the confusion matrix, performance improves with a decrease in the number of misclassifications.

To compare the phishing detection model's output, we created a confusion matrix. The confusion matrix enables us to gauge the proportion of correctly and incorrectly classified phishing emails.

The experiments included evaluation measures from earlier studies, including total accuracy, precision, recall, and F-score. We define the evaluation measures as follows using the confusion matrix: The percentage of emails that were correctly classified out of the total amount is what we refer to as accuracy. The portion of the expected phish that was accurately classified is called precision. The percentage of actual phishes aware correctly categorized is known as recall. The geometric mean of precision and recall is known as the F-measure.

To evaluate the performance of the proposed model, we used the well-known metrics for the classification techniques such as Precision (P), Recall (R), F1-Score (F1), and Accuracy.

**Precision (P):** It is defined as the fraction of circumstances in which the correct phishing email is returned.

$$\text{Precision (P)} = \frac{Tp}{Tp + Fp}$$

**Recall (R):** It is defined as the proportion of actual phishing emails is predicted correctly, Mathematically it is defined in the Equation.

$$\text{Recall (R)} = \frac{Tp}{Tp + Fn}$$

**F1-Score (F1):** It is defined as the harmonic mean of the precision and recall as given in Equation.

$$\text{F1} - \text{S core (F1)} = 2 * \frac{P * R}{P + R}$$

**Accuracy**: It is the fraction of phishing emails Messages that were correctly predicted among the phishing emails.

$$\text{Accuracy (A)} = \frac{Tp + Tn}{Tp + Fp + Tn + F}$$

# CHAPTER 8

# PERFORMANCE EVALUATION

## 8.1 EVALUATION OF THE PHISHING DETECTION MODEL

In this study, we propose that email communications may contain persuasion cues that could offer helpful indications in judging the veracity of the emails. Therefore, we provide a rich feature set of gain persuasion cues such as reciprocity, consistency, likeability, and loss persuasion cues concentrating on loss, severity, and immediacy to evaluate the persuasion cues throughout the phishing emails.

## 8.2 Evaluation Results

In this section, we go into more detail about the three experiments that were done to assess the effectiveness of the gain and loss persuasion cues in phishing detection models based on Loss and Accuracy. We explore the effectiveness of gain and loss persuasion signals in phishing email detection and provide a summary of the evaluation method.

**Experiment 1**

**Word2Vector models with LSTM**

In terms of Loss and Accuracy, the phishing email detection model of W2V_LOSS gives higher accuracy than W2V_NONE. W2V_GAIN gives the least or lower accuracy when compared with other models.

| LSTM | ACCURACY | LOSS | VAL_ACCURACY | VAL_LOSS |
|---|---|---|---|---|
| W2V_GAIN | 0.9609 | 0.1543 | 0.5943 | 0.8203 |
| W2V_GAINLOSS | 0.9442 | 0.1740 | 0.7072 | 0.6198 |
| W2V_LOSS | 0.9779 | 0.997 | 0.8699 | 0.6022 |
| W2V_NONE | 0.8881 | 0.1926 | 0.8658 | 0.3956 |

TABLE.1.W2V_LSTM

**Experiment 2**

**Word2Vector models with BI-LSTM**

According to Table 4, phishing email detection models that considered gain persuasion cues were more accurate overall in terms of accuracy and loss but not more than W2V_NONE model. And W2V gives the least of all.

| BI-LSTM | ACCURACY | LOSS | VAL_ACCURACY | VAL_LOSS |
|---|---|---|---|---|
| W2V_GAIN | 0.8263 | 0.2849 | 0.7970 | 0.3124 |
| W2V_GAINLOSS | 0.8693 | 0.3557 | 0.8891 | 0. 2953 |
| W2V_LOSS | 0.7936 | 0.3786 | 0.7705 | 0.2211 |
| W2V_NONE | 0.9456 | 0.1416 | 0.8275 | 0.3200 |

TABLE.2.W2V_Bi-LSTM

**Experiment 3**

**Word2Vector models with CNN**

In this experiment, The performance of phishing email detection models containing meaningful gain and loss persuasion signals against baselines that did not include any gain or loss persuasion cues have been evaluated and the results shows that baseline model is showing higher accuracy than any other models. And models built on loss persuasion cues are least in the list.

| CNN | ACCURACY | LOSS | VAL_ACCURACY | VAL_LOSS |
|---|---|---|---|---|
| W2V_GAIN | 0.9958 | 0.0367 | 0.6922 | 2.0266 |
| W2V_GAINLOSS | 0.6192 | 0.0386 | 0.5374 | 1.1684 |
| W2V_LOSS | 1.0000 | 3.2137 | 0.8536 | 0.8970 |
| W2V_NONE | 0.9992 | 0.0013 | 0.8786 | 0.6510 |

TABLE.3.W2V _CNN

## 8.3 COMPARISON OF MODELS FOR GAIN AND LOSS PERSUSASION CUES

| EVALUATION TERMS | ACCURACY | LOSS | VAL_ACCURACY | VAL_LOSS |
|---|---|---|---|---|
| W2V-NONE : Word2Vector models without any relevant persuasion cues | | | | |
| LSTM | 0.8881 | 0.1926 | 0.8658 | 0.3956 |
| BI-LSTM | 0.9456 | 0.1416 | 0.8275 | 0.4200 |
| CNN | 0.9992 | 0.0013 | 0.8786 | 0.6510 |
| W2V-GAINLOSS : Word2Vector models with relevant gain persuasion cues | | | | |
| LSTM | 0.9442 | 0.1740 | 0.7072 | 0.6198 |
| BI-LSTM | 0.8693 | 0.3557 | 0. 7953 | 0.4891 |
| CNN | 0.6192 | 0.0386 | 0.5374 | 0.1684 |
| W2V-LOSS : Word2Vector models with relevant loss persuasion cues | | | | |
| LSTM | 0.9779 | 0.997 | 0.8699 | 0.6022 |
| BI-LSTM | 1.0000 | 3.2137 | 0.8536 | 0.8970 |
| CNN | 1.0000 | 3.2137 | 0.8536 | 0.8970 |
| W2V-GAIN : Word2Vector models with relevant gain and loss persuasion cues | | | | |
| LSTM | 0.9609 | 0.1543 | 0.5943 | 0.8203 |
| BI-LSTM | 0.7263 | 0.5849 | 0.7970 | 0.5124 |
| CNN | 0.9958 | 0.0367 | 0.6922 | 2.0266 |

TABLE.4.COMPARISON OF MODELS

# CHAPTER 9

# OUTPUT SNAPSHOTS

## 1.W2V with CNN

### 1. Word2Vector_NONE with CNN

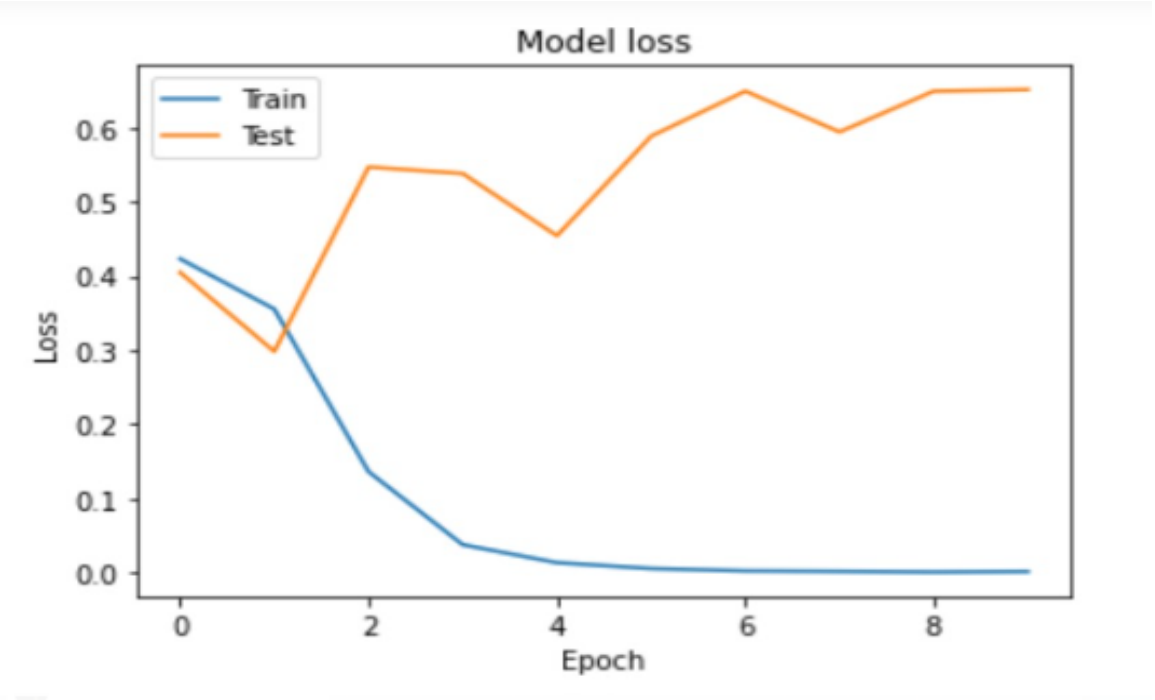


Fig.3.W2V_NONE Model loss

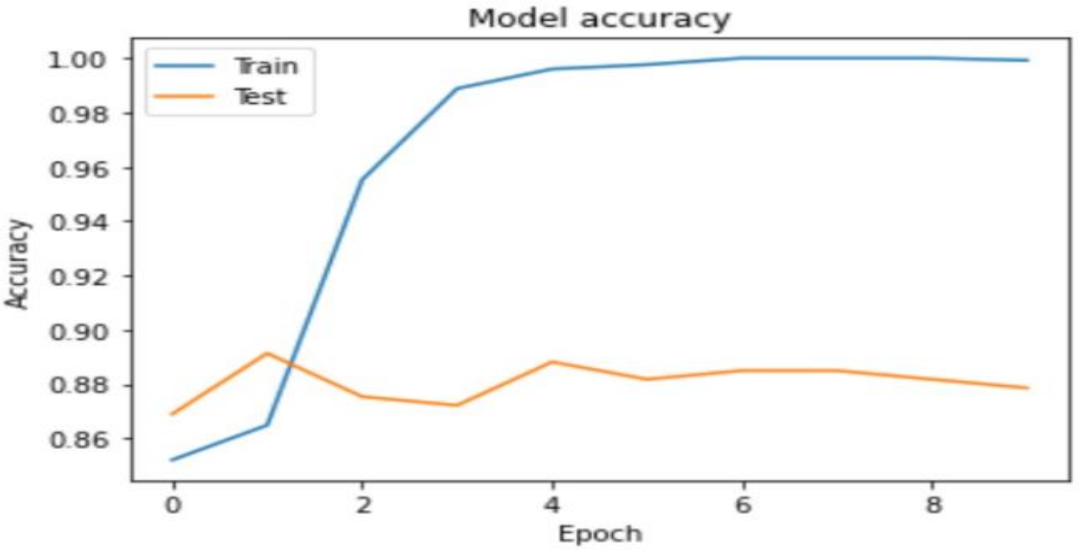


Fig.4. W2V_NONE Model accuracy

## 2.Word2Vector_LOSS with CNN

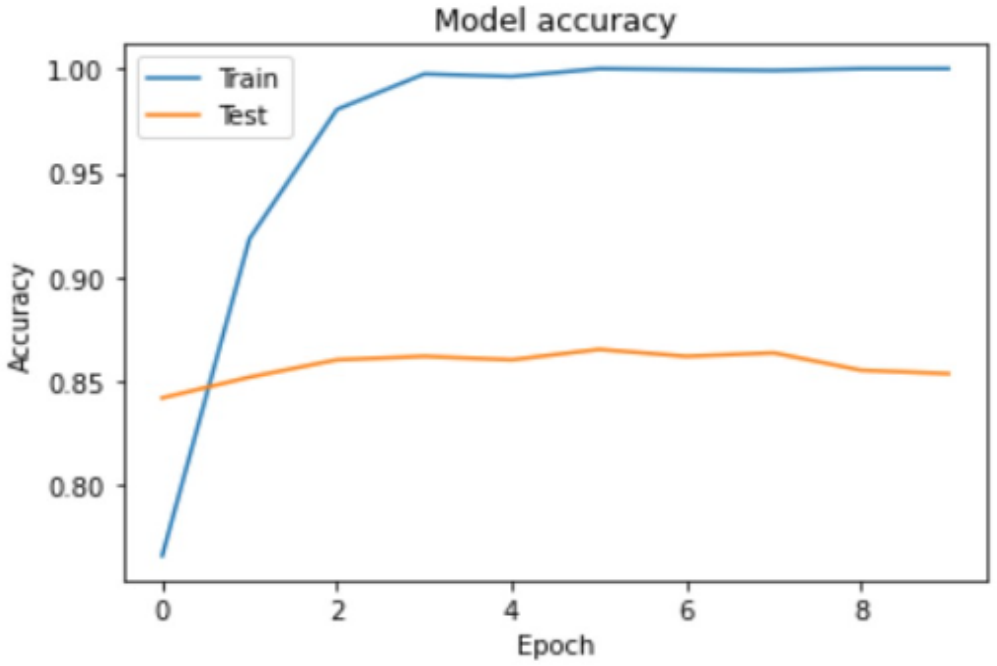


Fig.5.W2V_LOSS Model accuracy

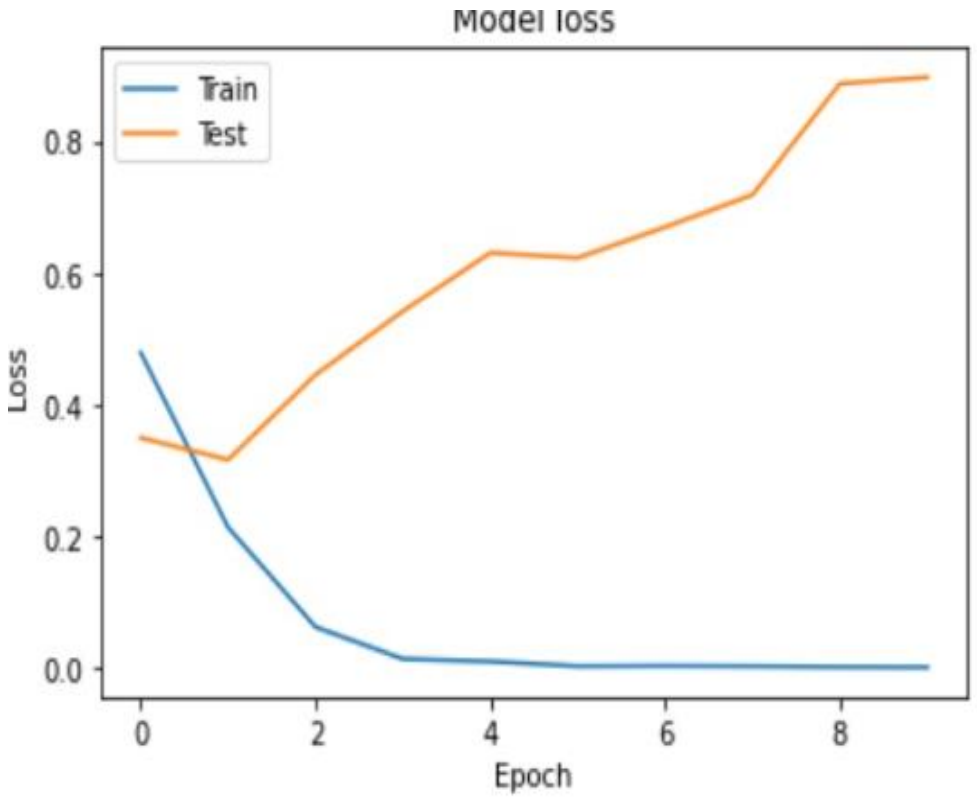


Fig.6.W2V_LOSS Model loss

## 3. Word2Vector_GAINLOSS with CNN

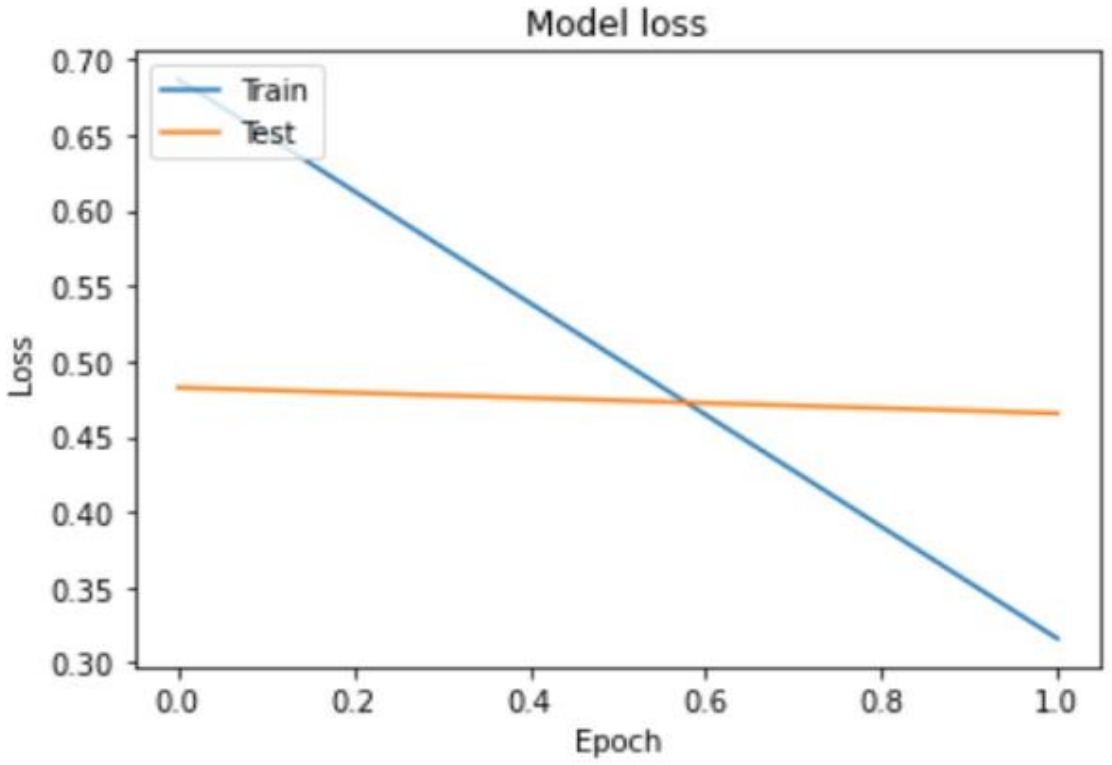


Fig.7.W2V_GN Model loss

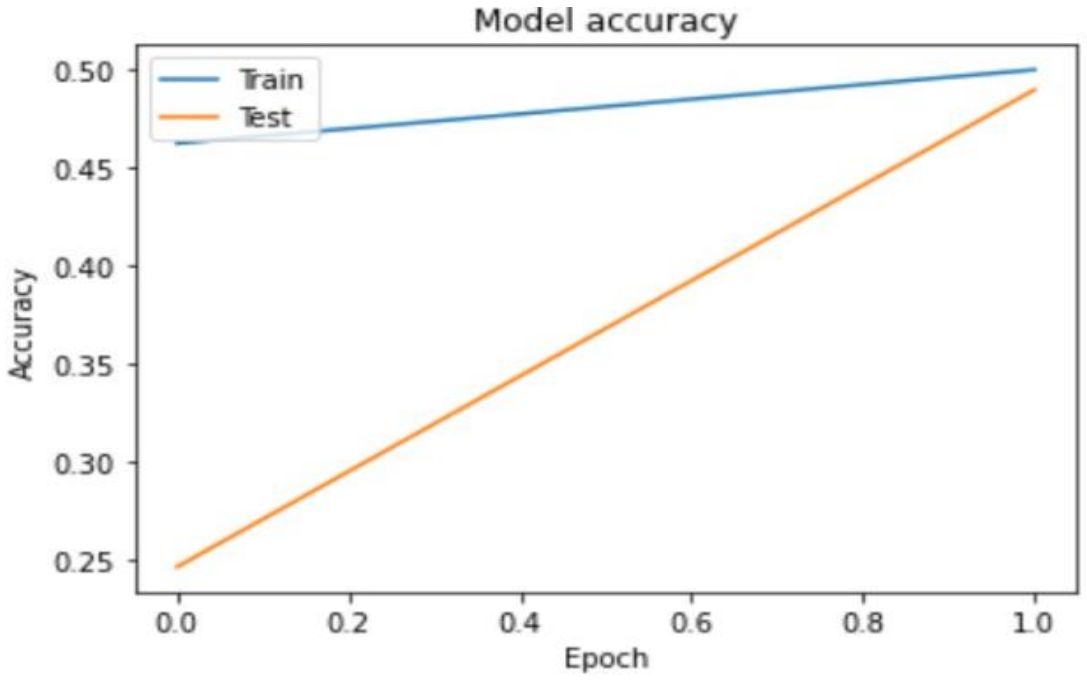


Fig.8.W2V_GN Model accuracy

## 4. Word2Vector_GAIN with CNN

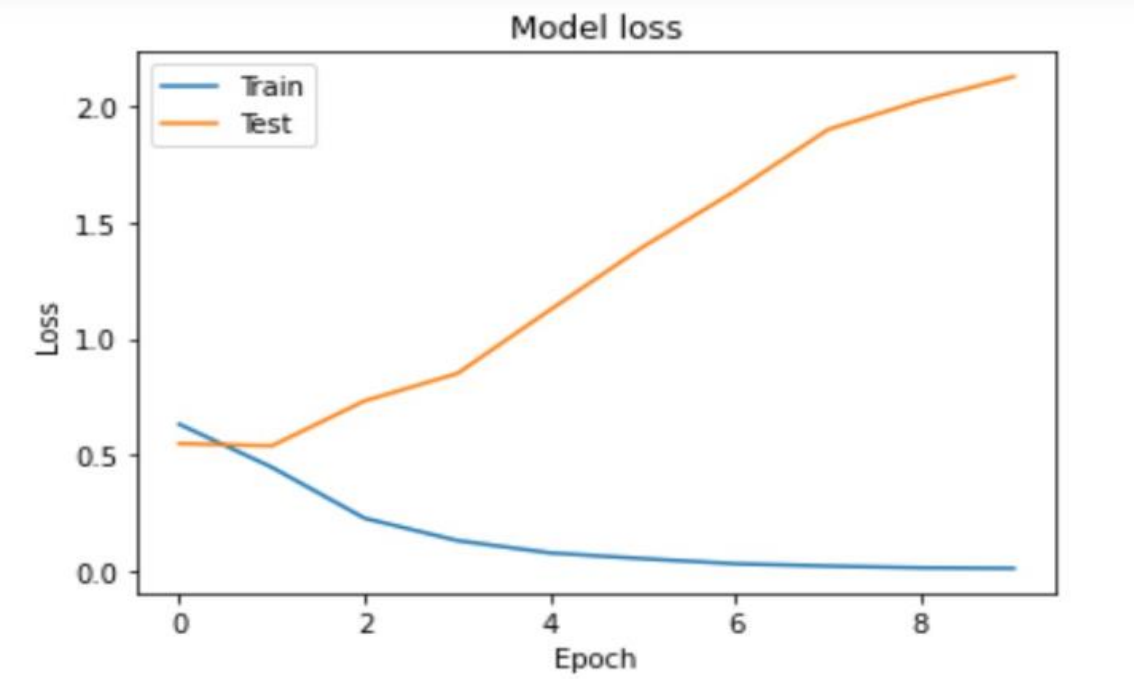


Fig.9.W2V_G Model loss

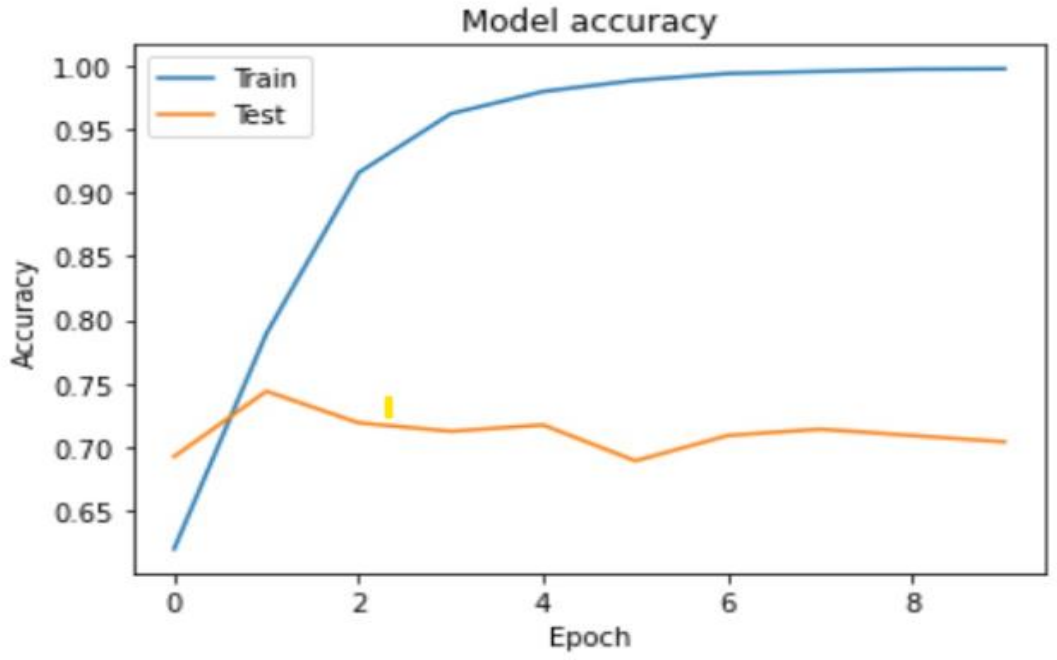


Fig.10.W2V_G Model loss

# 2. W2V with LSTM

## 1. W2V_GAINLOSS

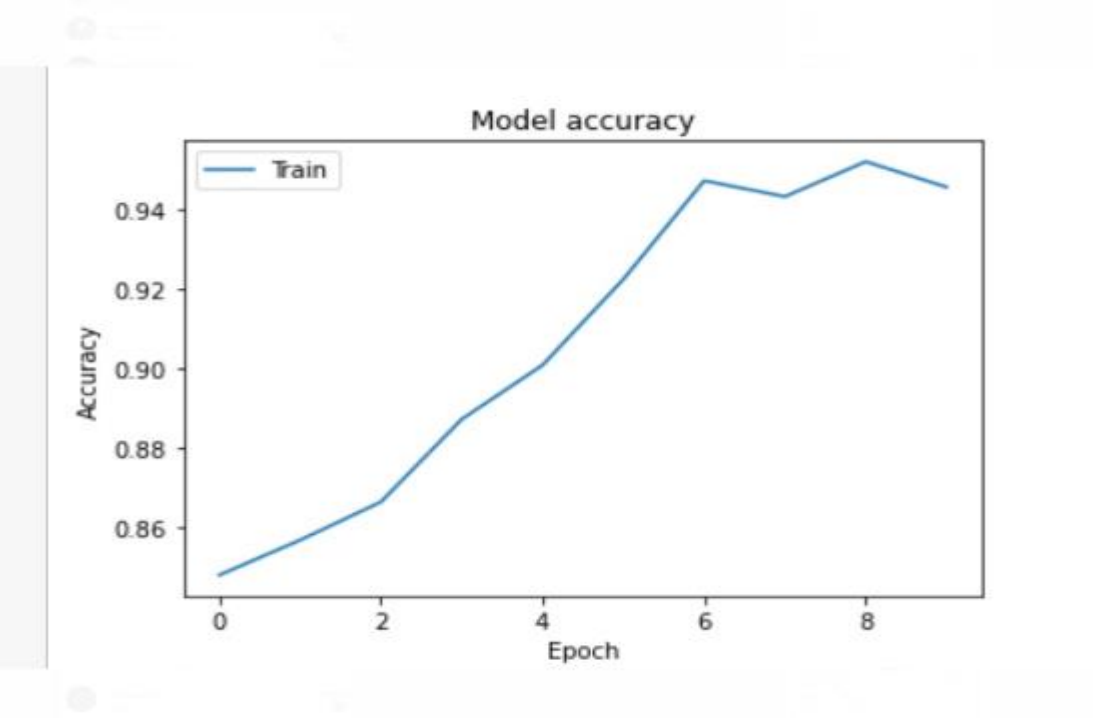


Fig.11.W2V_GN Model accuracy

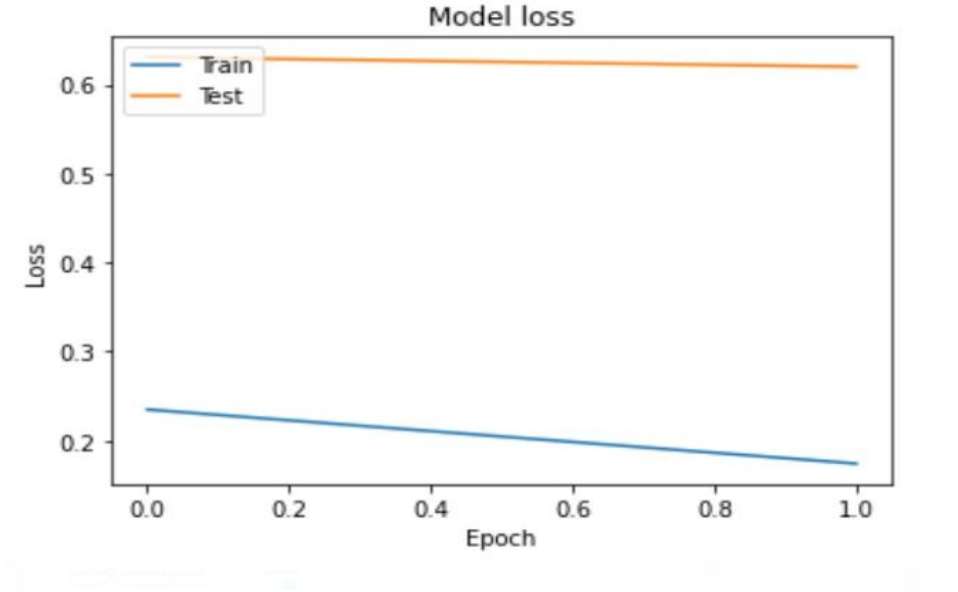


Fig.12 .W2V_GN Model LOSS

## 2. W2V_NONE

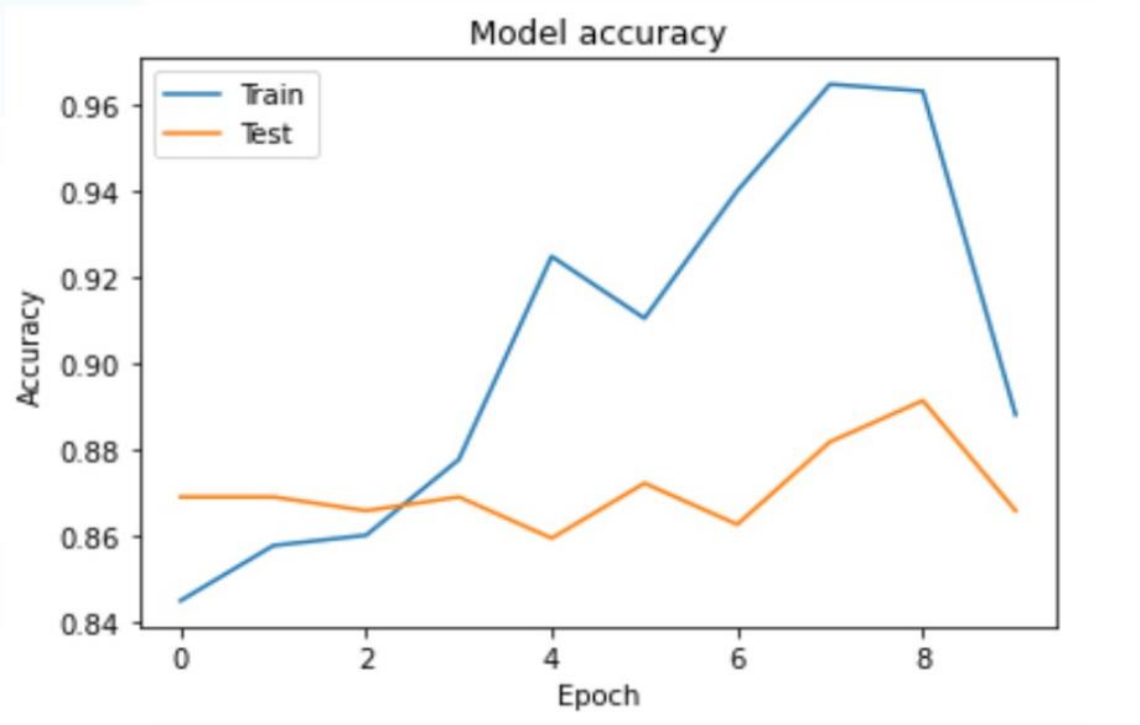


Fig.13.W2V_N Model accuracy

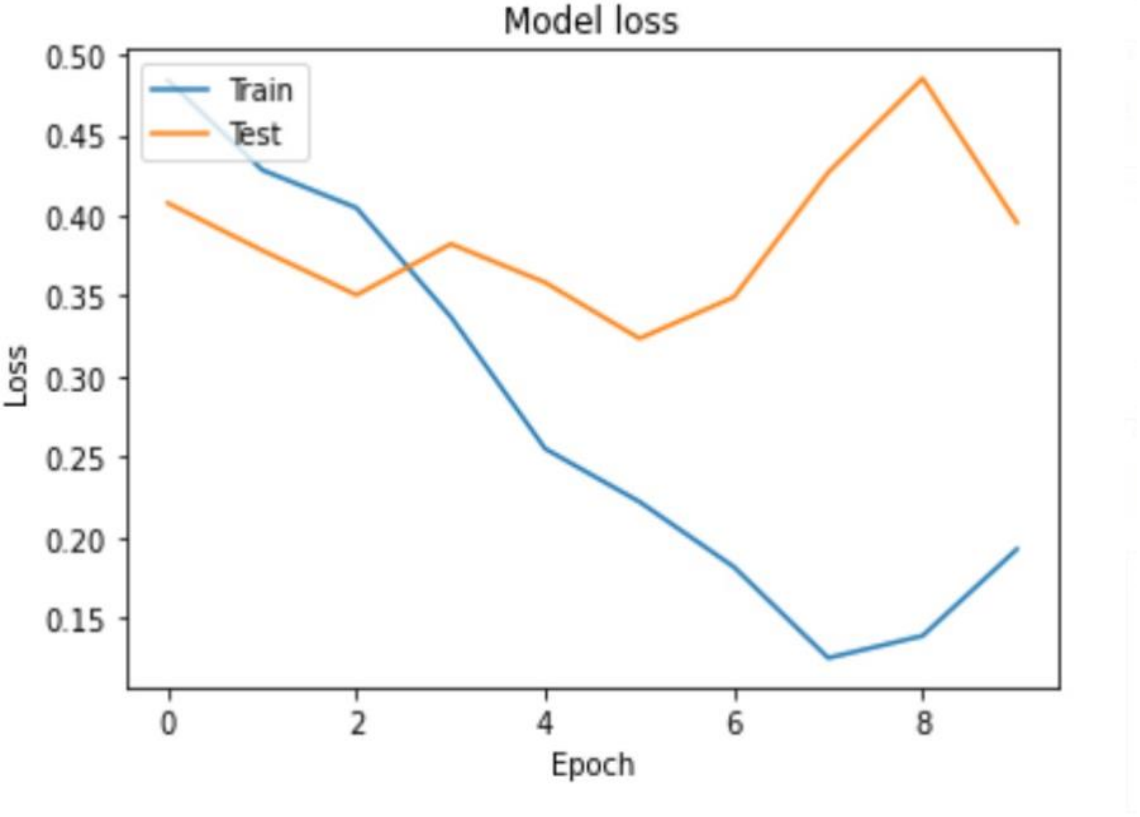


Fig.14.W2V_N Model loss

## 3. W2V_LOSS

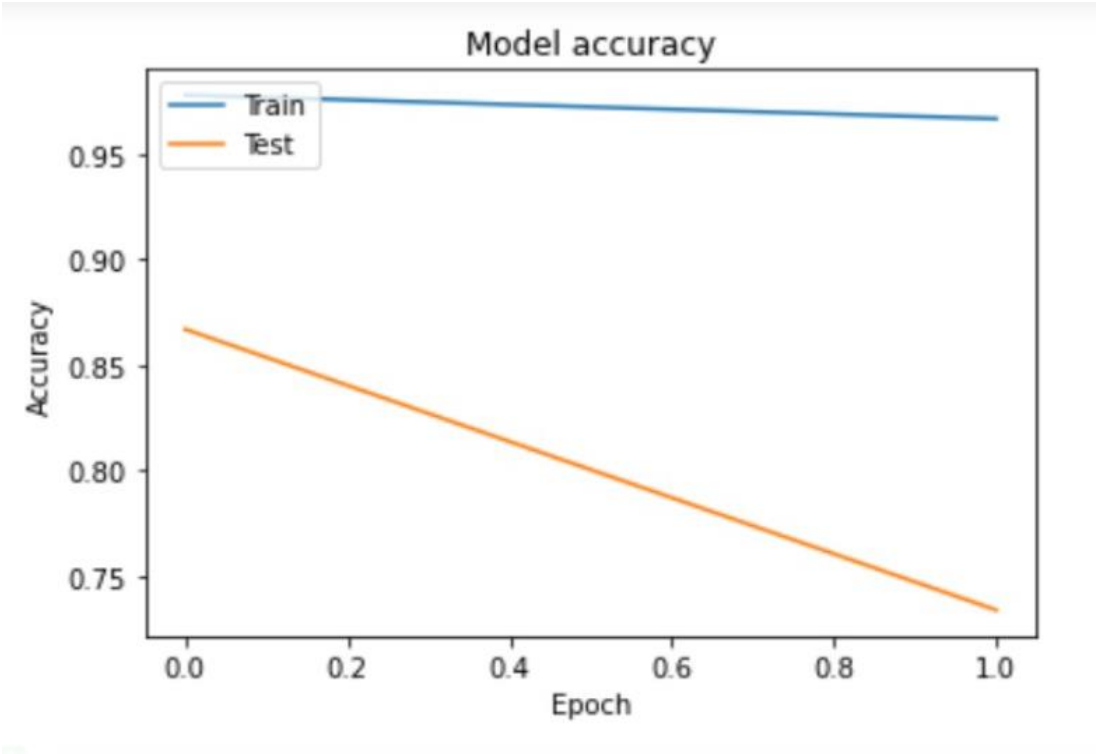


Fig.15.W2V_L Model accuracy

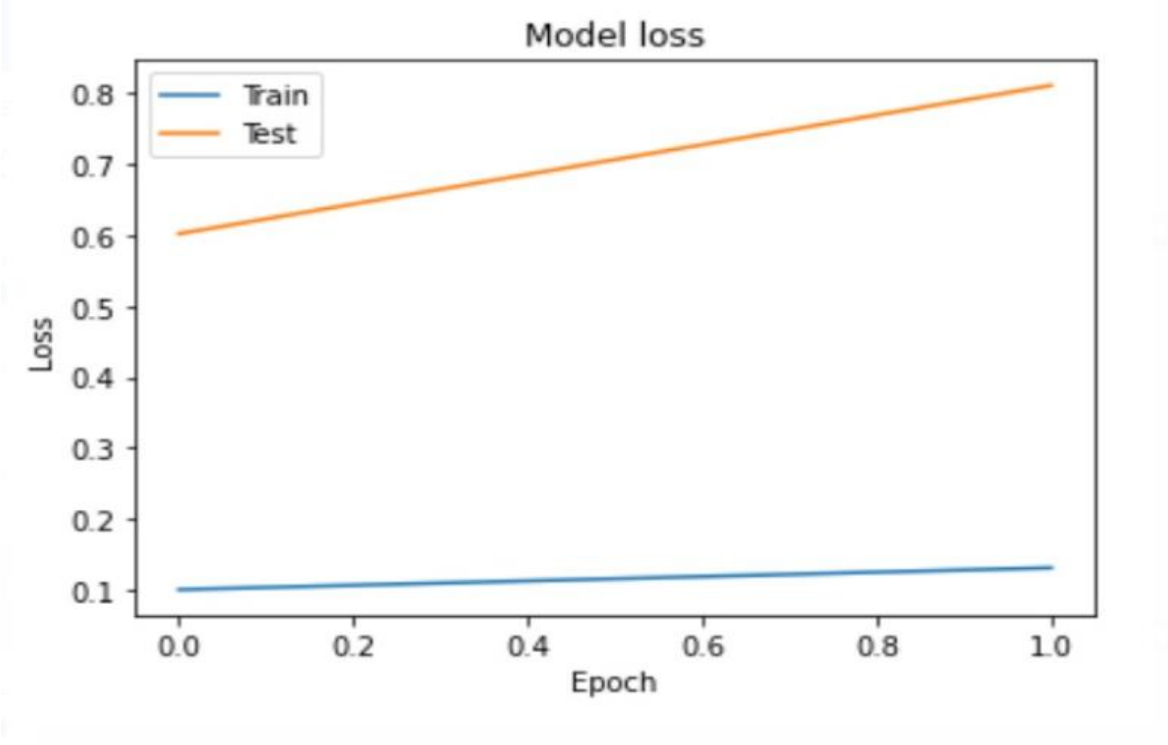


Fig.16.W2V_L Model accuracy

## 4. W2V_GAIN

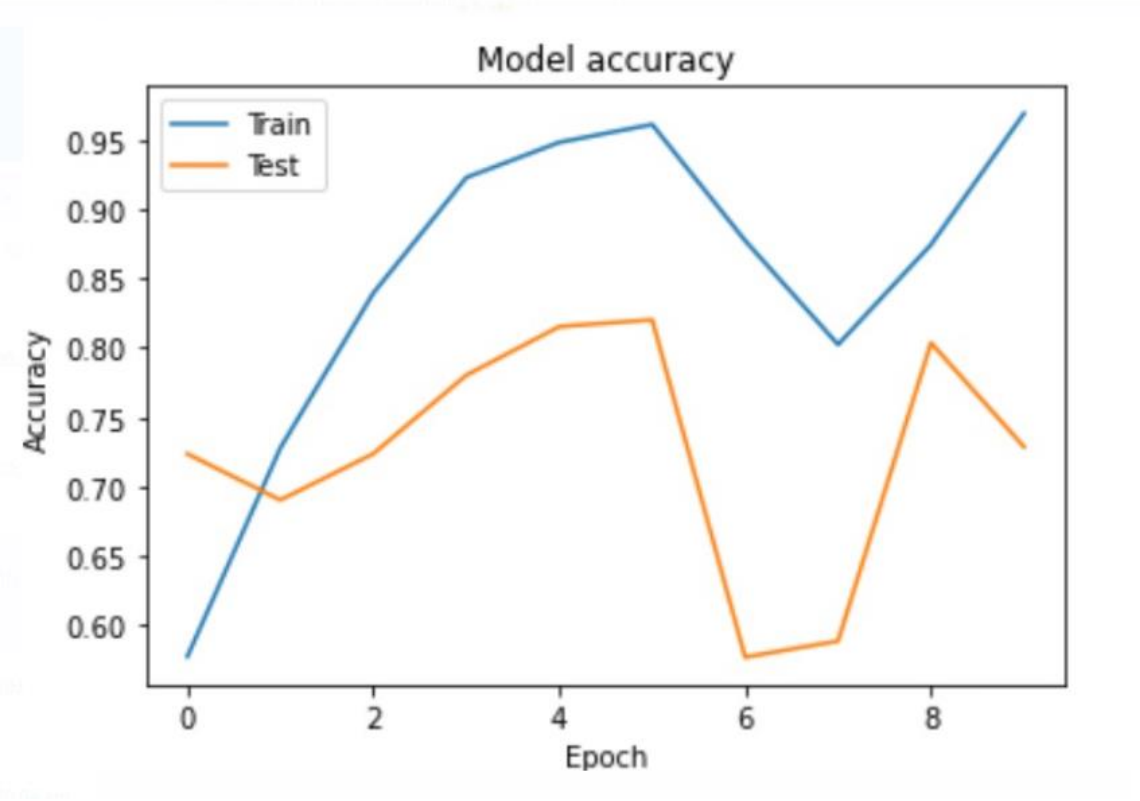


Fig.15.W2V_G Model accuracy

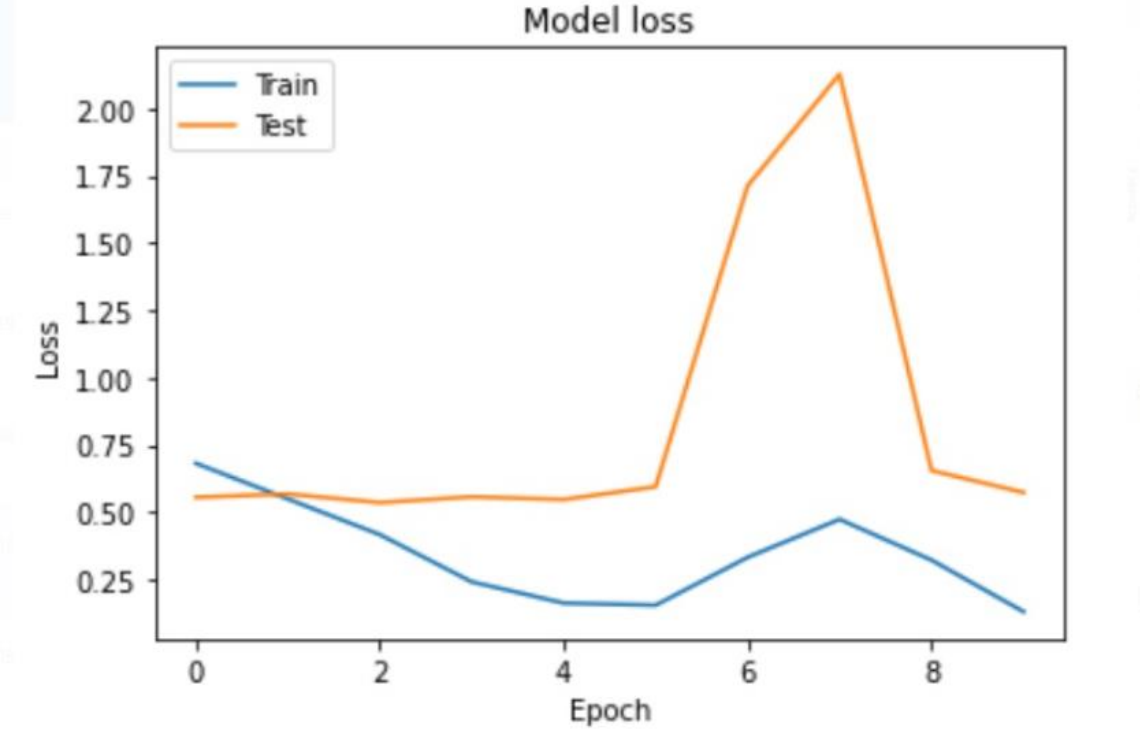


Fig.15.W2V_G  Model accuracy

# 3. W2V with Bi-LSTM

## 1. W2V_NONE

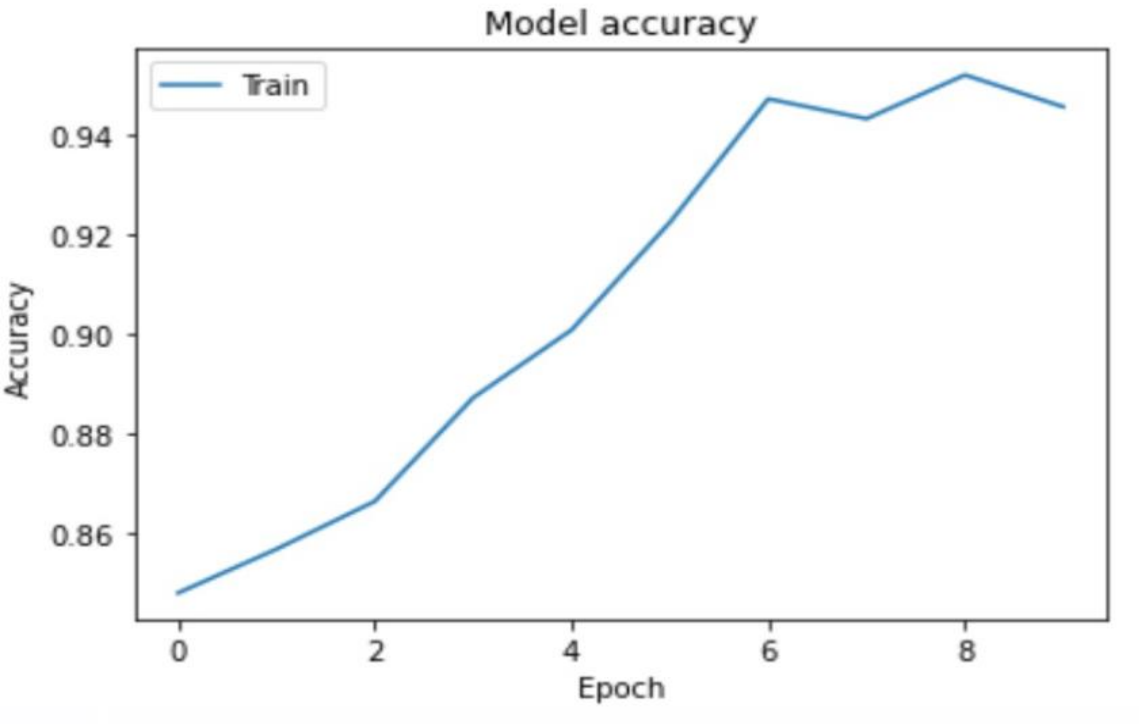


Fig.17.W2V_L Model accuracy

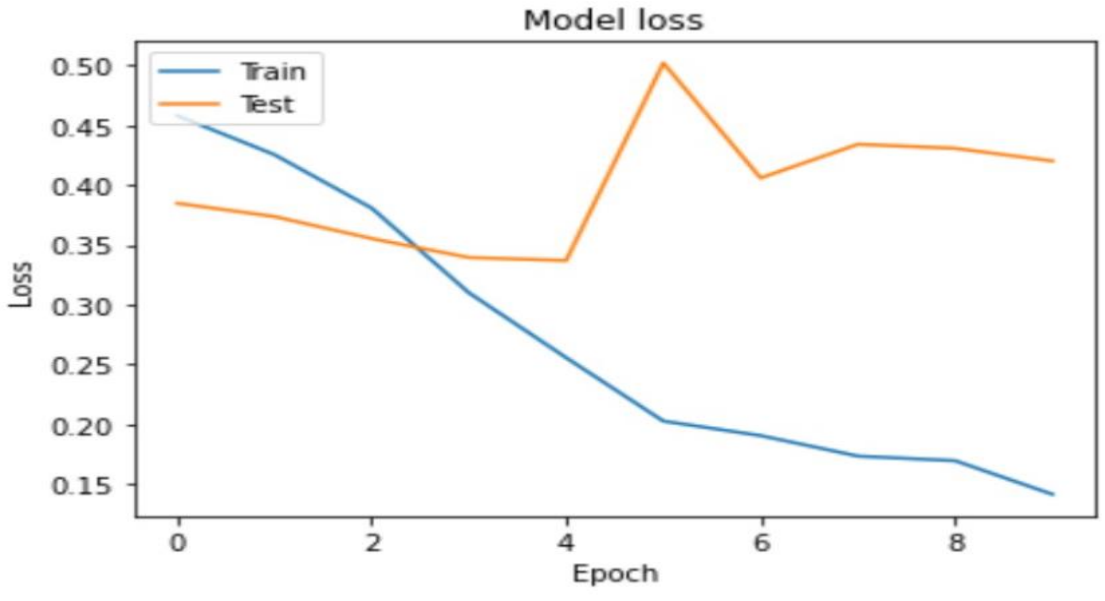


Fig.18.W2V_L Model LOSS

## 2. W2V_GAIN

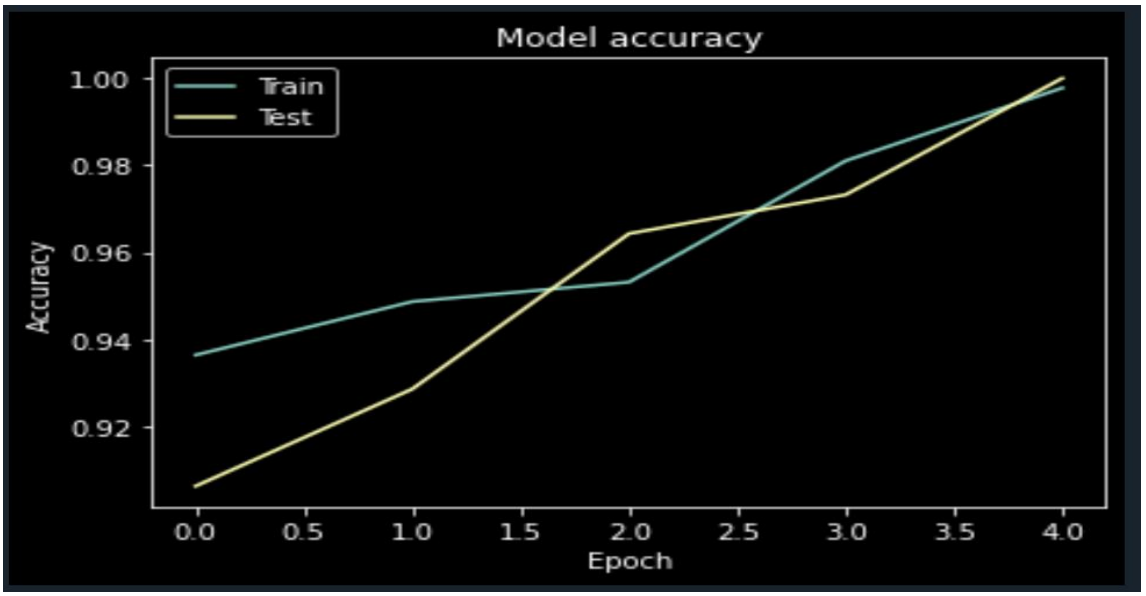


Fig.19.W2V_GAIN Model accuracy

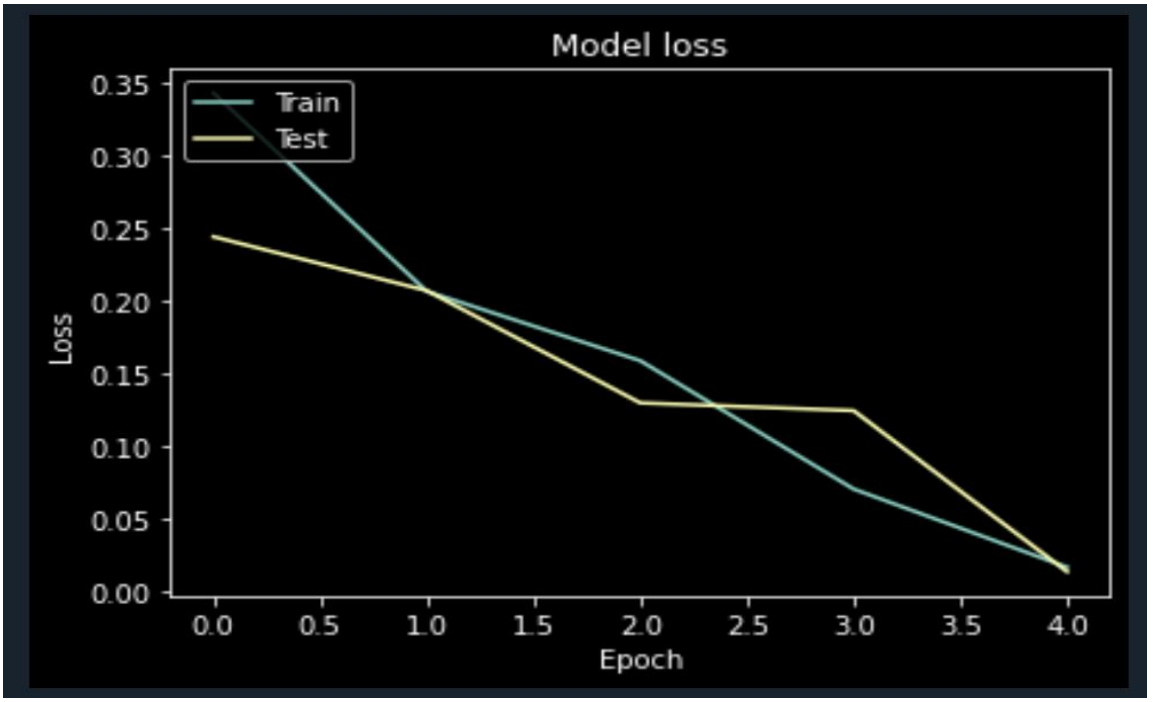


Fig.20.W2V_GAIN Model loss

## 3. W2V_GAINLOSS

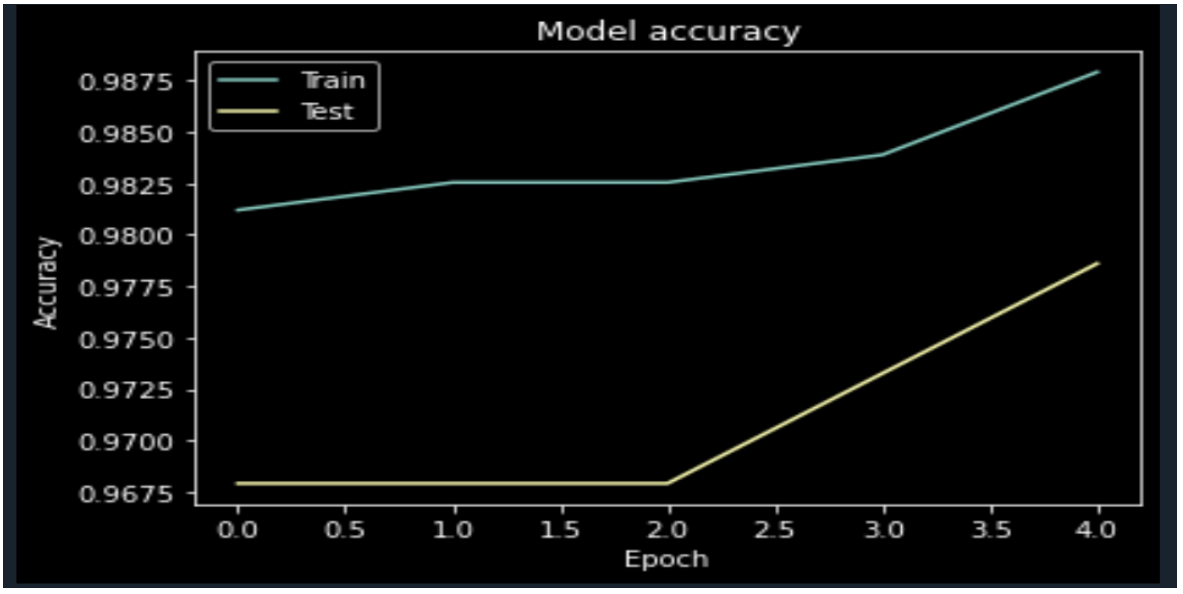


Fig.21.W2V_GL Model accuracy

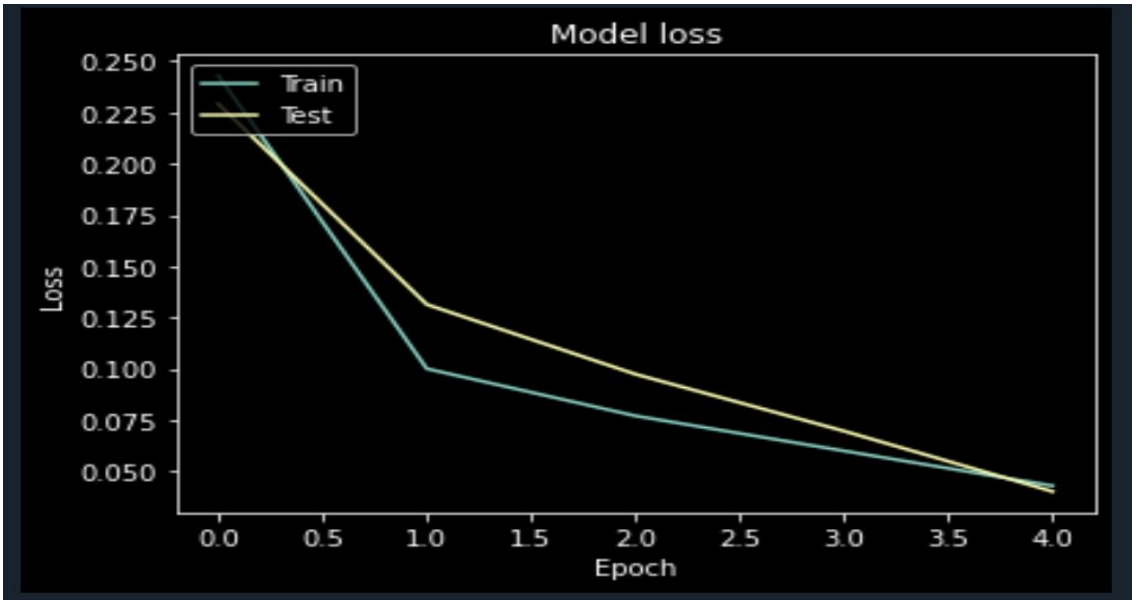


Fig.22.W2V_GL Model loss

## 2. W2V_LOSS

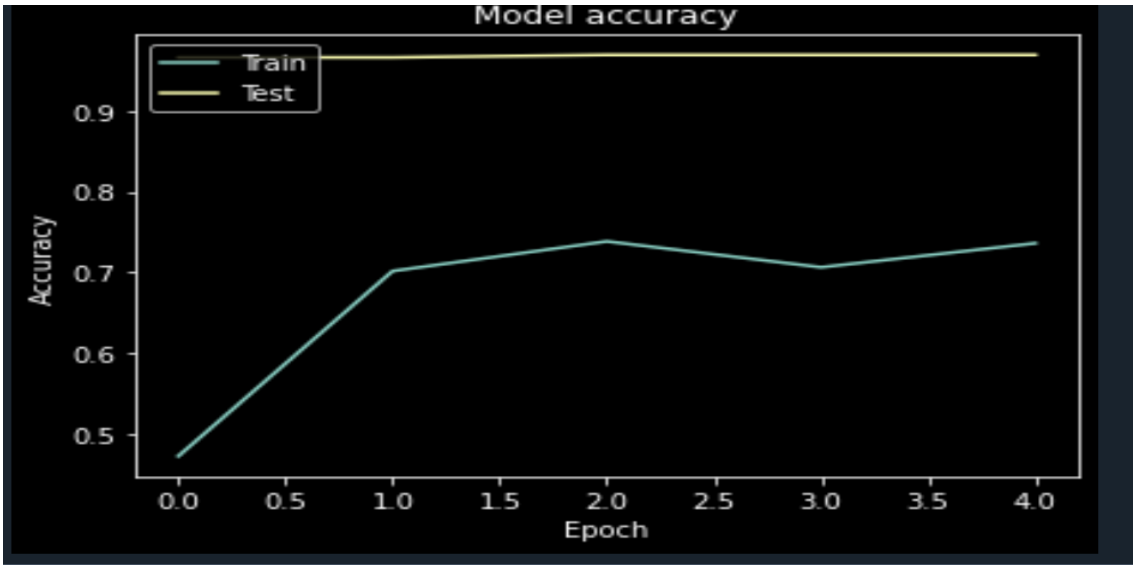


Fig.22.W2V_L Model Accuracy

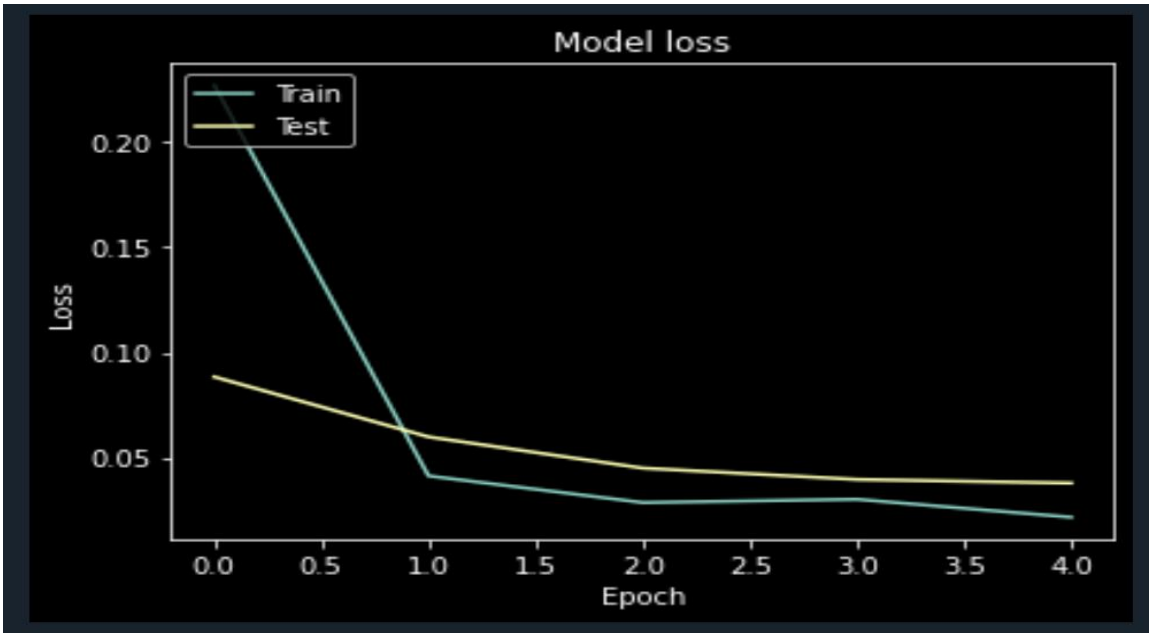


Fig.22.W2V_L Model loss

# CONCLUSION

We present a deep learning model of phishing email detection. Our model implies LSTM, Bi-LSTM and CNN algorithms to model the phishing email detection model with email bodies. In this propose, the combination of Word2Vec models were experimented with three Deep Learning algorithms and compared the result of total twelve models and presented the comparative results. Bi-LSTM taking the more executing time but resulted well than CNN and LSTM. Due to the limited resource we faced the memory problem. LSTM comes the next higher in terms of loss and accuracy. CNN didn't work well as phishing email detection model. These results of models can be improved by feeding much larger dataset and trying hybrid Deep Learning/Machine Learning algorithms as more domain specific Machine Learning and Deep Learning models are coming into real world. Works are to be continued to improve accuracy to detect phishing emails accurately as a countermeasure for phisher who evolve as more intelligent bodies.

# REFERENCES


[1] Rohit Valecha, Pranali Mandaokar .Phishing Email Detection using Persuasion Cues https://ieeexplore.ieee.org/document/9565347

[2] Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey Said Sallouma*, Tarek Gabera,b, Sunil Vaderaa , and Khaled Shaalaanhttps://www.sciencedirect.com/science/article/pii/S1877050921011741

[3] Ayman elaassal, shahryarbaki , avisha das, and rakesh m. verma .An In-Depth Benchmarking and Evaluation of Phishing Detection Research for Security Needs https://ieeexplore.ieee.org/document/8970564

[4] Christopher N. Gutierrez† ,Taegyu Kim† , Raffaele Della Corte‡ , Jeffrey Avery Learning from the Ones that Got Away: Detecting New Forms of Phishing Attacks
†https://ieeexplore.ieee.org/document/8440723

[5] S. Abiramasundari. Spam filtering using Semantic and Rule Based model via supervisedlearning,https://www.researchgate.net/publication/252029329_Spam_Classific ation_Based_on_Supervised_Learning_Using_Machine_Learning_Techniques

[6] Isra'sAbdulnabi Spam email detection using Deep learning Techniques
https://www.sciencedirect.com/science/article/pii/S18770509210074 93

[7] Detection of Online Phishing Email using Dynamic Evolving Neural Network Based on Reinforcement Learning SamiM.Smadi
http://nrl.northumbria.ac.uk/policies.html

[8] Harikrishna N B,Vinaykumar Ravi,Soman K P.A Machine Learning Approach Towards Phishing Email Detection
https://www.researchgate.net/publication/326211065_A_Machine_Learning_Approa ch_Towards_Phishing_Email_Detection_CEN-SecurityIWSPA_2018

[9] Hiransha M, Nidhin A Unnithan, Vinayakumar R, Soman. Deep Learning Based Phishing E-mail Detection CEN-Deepspam ,
https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Deep+Learning+Base d+Ph ishing+E-mail+Detection&btnG=

[10] Minh Nguyen, Toan Nguyen, ThienHuu Nguyen.A Deep Learning Model withHierarchical LSTMs and Supervised Attention for Anti-Phishing
https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=A+Deep+Learning+M odel

+with+Hierarchical+LSTMs+and+Supervised+Attention+for+Anti-Phishing&btnG=

[11] E Rahman , Shofi Ullah .Email Spam Detection using Bidirectional Long ShortTerm Memory with Convolutional Neural Network Sefat https://ieeexplore.ieee.org/abstract/document/9230769

[12] Dilhara, Phishing Email Url detection using hybrids of Long short term memory,https://www.researchgate.net/profile/Shashie-

[13] Ankit Narendra kumarSoni. Spam e-mail detection using advanced deep convolution neural network algorithms. https://jidps.com/wp-content/uploads/2019/05/Spam-e-mail- detection-using-advanced-deep-convolution-neural-network-algorithms.pdf

[14] Ian Fette, Norman Sadeh Learning to Detect Phishing Emails https://www.cs.cmu.edu/~tomasic/doc/2007/FetteSadehTomasicWWW2007.pdf

[15] Comparative study between NB and NN classifiers spam email detection Amit Kumar Sharma, Sudesh Kumar Prajapat https://www.academia.edu/42453336/A_Comparative_Study_between_Na%C3%AFve_Bayes_and_Neural_Network_MLP_Classifier_for_Spam_Email_Detection

[16] A comprehensive survey of AI-enabled phishing attacks detection techniques AbdulBasit1 ·Maham Zafar1 · Xuan Liu2 · Abdul Rehman Javed3 · Zunera Jalil3 https://pubmed.ncbi.nlm.nih.gov/33110340/

[17] Yong fang , Cheng Zhang, Cheng Huang , Liang Liu, and Yue Yang. Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8701426

[18] Barushka and P. Hajek, Spam filtering using integrated distribution- based balancing approach and regularized deep neural networks, *Appl. Intell.*, vol. 48, https://dk.upce.cz/bitstream/handle/10195/72756/Manuscript-OBD.pdf?sequence=1&isAllowed=y

[19] Cach Dang María N. Moreno García Fernando De La Prieta (2020) Sentiment Analysis Based on Deep Learning: A Comparative Study colah. (2015). Understanding LSTM Networks. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[20] Ankit Narendra kumar Soni. Spam e-mail detection using advanced deep convolution neural network algorithms. https://jidps.com/wp-content/uploads/2019/05/Spam-e-mail- detection-using-advanced-deep-convolution-neural-network-algorithms.pdf

[21] Ian Fette, Norman Sadeh Learning to Detect Phishing Emails https://www.cs.cmu.edu/~tomasic/doc/2007/FetteSadehTomasicWWW2007.pdf

[22] Comparitive study between NB and NN classifiers spam email detection Amit KumarSharma, Sudesh Kumar Prajapat https://www.academia.edu/42453336/A_Comparative_Study_between_Na%C3%AFve_Bayes_and_Neural_Network_MLP_Classifier_for_Spam_Email_Detection

[23]A comprehensive survey of AI-enabled phishing attacks detection techniques AbdulBasit1 ·Maham Zafar1 · Xuan Liu2 · Abdul Rehman Javed3 · Zunera Jalil3 https://pubmed.ncbi.nlm.nih.gov/33110340/

[24] Yong fang , Cheng Zhang, Cheng Huang , Liang Liu, and Yue Yang. Phishing EmailDetection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8701426

[25] Sriram Srinivasan, Vinayakumar Ravi, MamounAlazab, Simran Ketha, Ala' M. Al- Zoubi, Soman KottiPadannayil. Spam Emails Detection Based onDistribute WordEmbedding with Deep Learning. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Spam+Emails+Detection+ Based+on+Distributed+Word+Embedding+with+Deep+Learning&btnG=

[26] Barushka and P. Hajek, Spam filtering using integrated distribution- based balancing approach and regularized deep neural networks, *Appl.Intell.*, vol. 48, https://dk.upce.cz/bitstream/handle/10195/72756/Manuscript-OBD.pdf?sequence=1&isAllowed=y

[27] Cach Dang María N. Moreno García Fernando De La Prieta (2020) Sentiment Analysis Based on Deep Learning: A Comparative Study colah. (2015). Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Phishing is an act of social engineering to obtain information from an unsuspecting victim. It involves an attacker who generally masquerades as a legitimate institution to trick users into disclosing sensitive information that can later be used in fraudulent activities. More than 100,000 Internet users around the world are subjected to phishing attacks daily. Phishing emails have become more complicated and harder to detect in recent years. Hackers are using more complex methods of attack to foil their victims, as humans are at the front line of defense in some cases against phishing emails; they are the key point of contact for an attacker to attack.

In our proposed system we modeled the Persuasion cues for phishing email detection using word embedding techniques, with the help of deep learning algorithms. It examines the effectiveness of persuasion cues for phishing email detection. Persuasion cues are signals within phishing emails through which phisher attempt to persuade and