Identification of Fake Tweets using Deep Learning Techniques

Submitted by

Sara Irfan
19i-1218

Supervised by

Dr. Muhammad Asif Naeem
Masters of Science (Data Science)

A thesis submitted in partial fulfillment of the requirements for the degree of
Masters of Science (Data Science)
at National University of Computer & Emerging Sciences

Department of Computer Science
National University of Computer & Emerging Sciences

Islamabad, Pakistan.

December 2021

Plagiarism Undertaking

I take full responsibility of the research work conducted during the Masters Thesis titled *Identification of Fake Tweets using Deep Learning Techniques* . I solemnly declare that the research work presented in the thesis is done solely by me with no significant help from any other person; however, small help wherever taken is duly acknowledged. I have also written the complete thesis by myself. Moreover, I have not presented this thesis (or substantially similar research work) or any part of the thesis previously to any other degree awarding institution within Pakistan or abroad.

I understand that the management of National University of Computer and Emerging Sciences has a zero tolerance policy towards plagiarism. Therefore, I as an author of the above-mentioned thesis, solemnly declare that no portion of my thesis has been plagiarized and any material used in the thesis from other sources is properly referenced. Moreover, the thesis does not contain any literal citing of more than 70 words (total) even by giving a reference unless I have the written permission of the publisher to do so. Furthermore, the work presented in the thesis is my own original work and I have positively cited the related work of the other researchers by clearly differentiating my work from their relevant work.

I further understand that if I am found guilty of any form of plagiarism in my thesis work even after my graduation, the University reserves the right to revoke my Masters degree. Moreover, the University will also have the right to publish my name on its website that keeps a record of the students who plagiarized in their thesis work.

Sara Irfan

Date: ____________________

Author's Declaration

I, Sara Irfan, hereby state that my Masters thesis titled *Identification of Fake Tweets using Deep Learning Techniques* is my own work and it has not been previously submitted by me for taking partial or full credit for the award of any degree at this University or anywhere else in the world. If my statement is found to be incorrect, at any time even after my graduation, the University has the right to revoke my Masters degree.

Sara Irfan

Date:

Certificate of Approval

It is certified that the research work presented in this thesis, entitled "Identification of Fake Tweets using Deep Learning Techniques " was conducted by Sara Irfan under the supervision of Dr. Muhammad Asif Naeem.

No part of this thesis has been submitted anywhere else for any other degree.

This thesis is submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of Masters of Sciencein Data Science

at the

National University of Computer and Emerging Sciences, Islamabad, Pakistan

December' 2021

Candidate Name: Sara Irfan Signature: ____________________

Examination Committee:

1. Name: Dr. Waseem Shahzad Signature: ____________________
   Assistant Professor, FAST-NU Islamabad.
2. Name: Dr. Shoaib Mehmood Signature: ____________________
   Assistant Professor, FAST-NU Islamabad

____

Graduate Program Coordinator, National University of Computer and Emerging Sciences, Islamabad, Pakistan.

____

Head of the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan.

Abstract

Social platforms exchange information by people living in various places. Twitter data is mostly used for the largest micro-blogging platforms, research, political, and business platforms. The malicious information is spreading among the users and to overcome this issue, a large number of tweets have been carried out. The main focus of the research is to gather the tweets related to different universities of Pakistan and identify the fake or non-fake tweets because there were a lot of tweets against different universities of Pakistan during the pandemic situation of Covid-19. The analysis has been performed on real-time data for detecting fake tweets by deep learning techniques. Automated classification of its detection is a challenging task. We explored variant textual properties that can be used to distinguish fake tweets and we worked on Twitter identifying Twitter users. Previously used algorithms like Synthetic Minority Oversampling Technique as SMOTE, SVM-NN, CNN, Naive Bayes, and BERT model have been used in finding fake accounts. The data have been scrapped from Twitter using a hybrid approach taking into account the TWINT library and TWEEPY so that the entire information of a person can be attained which helps identify whether the tweets are fake or not. The collected data-set was pre-processed with the seed features. The features have been used with comments of a person on the tweet, sources, average replies, average retweet, average tweets per day with the unique hashtags. Then data has been encoded and decoded to exclude the emotions and other useless symbols. The sentiment analysis has been performed with VADER without using target labels and generated subjectivity and polarity with a type of sentiment. These generated sentiment types have been used in BERT and Fast-Text embeddings, SVM and Naive Bayes have also been implemented. The BERT-HAN model has been implemented with improved 95% accuracy, The results have shown that the BERT-HAN has performed much better than other results.

Keywords: Deep Learning, EDA, Twint Library, Tweepy Library, BERT-HAN state of art

Acknowledgements

I would first like to thank my thesis advisor Dr. Muhammad Asif Naeem . The door to Dr. Asif Naeem office was always open whenever I had a problem in any task and his response on Email was also fast. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it. I must express my very profound gratitude to my Husband and to my Parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Figures

Chapter 1

Introduction

Online social networking has made an extreme change in our lives. Making new friends, relatives, and knowing their updates has become easier for us. Social networking involves sharing pictures and their views on a common topic. The daily activities are carried out such as posting messages, receiving items, and sharing different content. People use online social networks to keep in contact with each other and run their e-business. E-business is minimizing expenses and making purchases faster, simpler, and less time-consuming. For example, Amazon creates a business relationship with variant sizes of businesses [1]. New and un-established businesses can start with personal accounts and then upgrade to a business account when their sale increases. People create fake accounts and order their goods to the wrong address. It results in wasting others' time. Due to the exponential involvement, there is an increase in variant problems like online impersonation, fake profiles, and hate speeches. The social network has become a target for scammers. [2] There is also a possibility of malicious attempt to disrupt a computer network and malicious threat creates a problem for firewalls, that can be difficult to configure correctly and makes the system slower. It also involves the sending of non-required emails. For example, a group of people may control an opinion, when the other people in the group are not aware of fake or real conversation. There are various ways of creating fake tweets. They can be generated by humans or half-bot. For creating a profile, the user has to enter the data related to that profile. There are mandatory and optional fields. Such entered information can be fake or real.

1.1 Social Issues from Fake Data

There are a lot of other politicians who are spreading fake news to lower down their position. Rumor is an unverified claim about any event, transmitted from individual to an individual in society. It might imply an occurrence, article, and any social issue of open public concern. It might end up being a socially dangerous phenomenon in any human culture. The main goal is to identify fake tweets or users. It helps in extinguishing the root of fake words identifications that can help determine more fake tweets or users. The purpose of fake tweets is to change the action of a group, hide malicious activities, and spread the malware. [3] The malware is the creation of wrong connections to steal the authentication and credentials. It also includes the identification of the tweet's profile itself and using non-verbal standards. The standard attributes alone are found by the number of followers and friends that are available through accounts on social media platforms. The framework is provided in which automatic detection of fake profiles, tweets, and users is efficient. It uses variant technologies and models. The variant social networking platforms have the emergence of unwanted digital bots. They quickly execute those tasks that cannot be performed by humans easily in less amount of time. The massive population of online social networking causes variant problems in terms of privacy and security. These problems need to be tackled and the improvement of identification is necessary for this step. For example, there is an option of "report as spam" on Twitter. There are variant classifications of spam into influential bots, fake follower bots, and social bots. It contains software that sends tweets to the users.[4] They can also be used to promote a business, increase business relationships. On the other hand, it also increases hate speeches.

1.2 Data Privacy

Data privacy is one of the critical concerns of social network users. An efficient model needs to be built to classify the fake tweets. Fake news, hate speech, and misinformation is spreading rapidly through social media. For example, President Trump received backlash over the spread of wrong information and fake comments on Twitter. Twitter could easily make the case to remove President Trump by violating the terms and services of their agreements. It's unable to distinguish truth from fact or news from propaganda. Some people are spreading fake news to spread hate. The Deputy Commissioner of Islamabad has not issued the letter related to holidays during Covid-19 but it had a large number of tweets regarding this letter against the Deputy Commissioner of Islamabad. It gave his wrong impact initially then he cleared.

There is a credibility inference problem in which the real tweets will have high credibility and unauthentic ones will have lower credibility. To address this problem, there would be a great impact on results. It includes a fast-text library, a decision tree, Bert transformations, and BERT-HAN embeddings. FastText is a library for learning word embedding and text classification created by Facebook's AI Research lab. The model allows one to create unsupervised learning or supervised learning algorithms for obtaining vector representations for words. There are different models used for fake tweets identification.

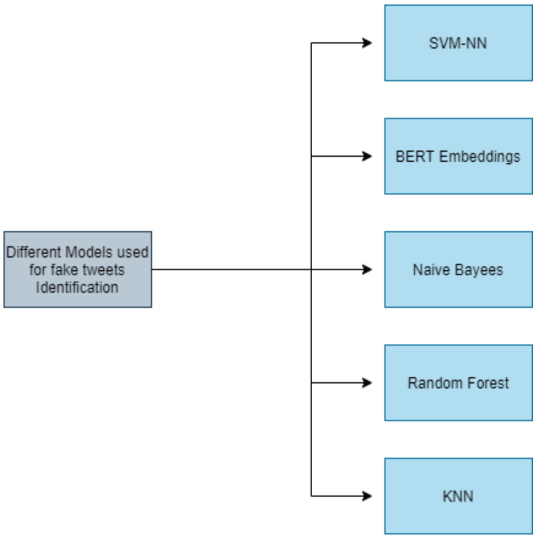


Figure 1.1: Different Models for Fake Tweet Identification

1.3 Factors involved in Data

The rare words have been found by vector representation. These words could be converted into character n-grams and distributed with shared words. For example, the diseases cannot be a common word in medical terms for a model trained on a news data set. It can grant the vector representations for the words available in the dictionary of out of vocabulary words. There are variant models for vector representation but word2vec and glove both fail to provide any vector representations for words not in the dictionary.

For example, for a word like manonfantastictourto, which might not be available in a corpus, the belief that distinctive human characteristics and capacities are determined by genes and that a person's value is based on genotype rather than individual merits. Formal assertion of genism in education or the workplace is regarded as unconstitutional. It returns one of the following two solutions. a) a zero-vector b) a random vector with low magnitude. But Fast-Text can assemble vectors superior to random by breaking the above word into chunks and using the vectors for those chunks to create a final vector for the word. The final vector might be closer to the vectors of fantastic and tourto. Decision Trees are a type of Supervised Machine Learning where the data is continuously split according to a definite parameter. The tree can be described by two entities, namely decision nodes, and leaves. The leaves are the decisions or the outcomes. And the decision nodes are where the data is split. There is a guide to fake tweets detection which includes data-oriented, context-oriented, supervised, unsupervised and feature-oriented.

1.4 Contextual Embeddings in Data

The fake data with tweets contains two different categories, echo chamber effect and fake accounts used.[5] The primary objective of identifying fake tweets is to become a medium for fake activity which contains a social analysis of fake tweets and different trolls. It has been found by statistical inference that a lot of Twitter accounts are controlled by fake tweets against our Politicians, Universities, and other domains. There are different tasks in identifying to create a set of features and allows models to check whether the tweet is fake or not. There are different approaches to identification. The style content is important in identifying the language. The vocabulary of a language and words relations as distinguished from its domain. The frequency of words, word counts, and most unique words are embedded in the lexical features. The part of speech tags and bag of words are embedded in syntactic features. The content needs to be checked whether it is correct or not so it is embedded in knowledge-based technique.

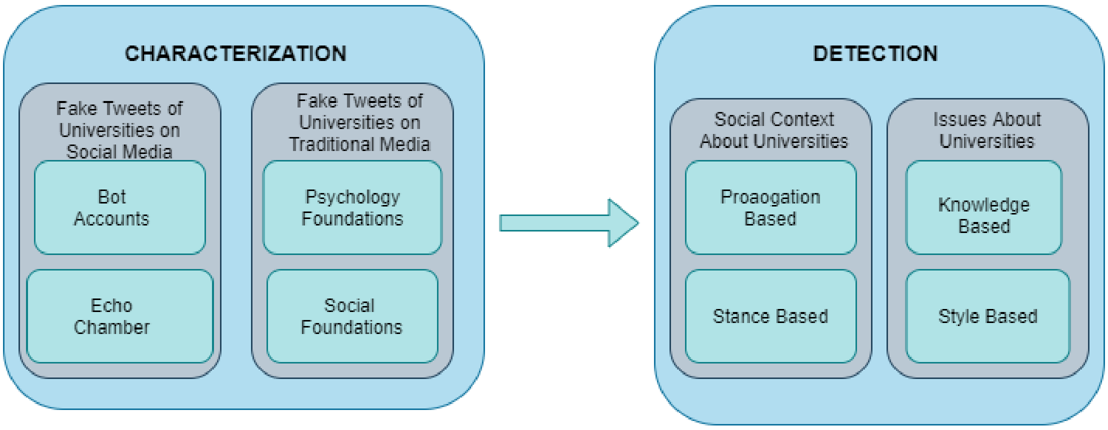


Figure 1.2: Characterization of Models

1.5 Motivation

The main motivation of the proposed work is analyzed by calculations on Consumer News and Business Channel statistics which showed that Twitter currently has 319 million monthly active users, which translates to nearly 48 million bot accounts, using USC's high-end estimate.[6] It's very easy to spread propaganda through online social networks. People are facing issues in real life due to fake Twitter accounts. An article published in NewYork Times on Jan. 8, 2021, that Oakland Calif said that it had per- permanently banned President Trump from its service due to the risk of further incitement of violence, effectively cutting him off from his favorite megaphone for reaching the public and capping a series of actions by mainstream sites to limit his online reach.

It results in spreading disinformation to enter the mainstream so accessing fake tweets needs to be found. There are variant fake accounts and fake tweets on Twitter. Students are spreading fake news about specific universities that result in a bad image of the university. We worked on accessing the Department of Computer Science Faculty of Computing, the fake tweets instead of fake accounts to check how reliable are posted tweets are. Our domain is to fetch those accounts who are tweeting wrong about the university. There are variant models doc2vec, word2vec, and Fast-Text. [7] The main focus of the research aim needs to be improved so that the fake tweets can be identified in an improved way.

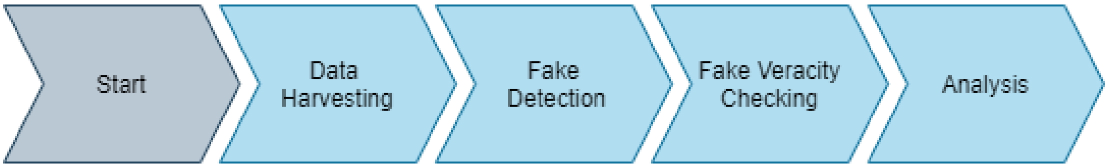


Figure 1.3: Research Aim

1.6 Research Problem

There are different challenges in extracting the data from Twitter and its identification.

• Students may have tweeted about **current affairs** or personal tweets mixed with tweets about online exams.

•Identifying the difference between **online exams tweets** and physical exams tweets is hard for a great analysis.

• The text of the tweet is short, a challenging task because **twitter data is noisy** and we can't get an overview of the tweet easily without the background knowledge. For example, universities can be referred by my students from different backgrounds.[8] Therefore, knowledge-based works very well on such conditions.

• The main issue is the **limit set by Twitter** for extracting tweets from Twitter. For example, there were two thousand, 2000 tweets of "Iqra-UniversityOfficial", but we scraped through twint and Twitter, we just got 700 tweets about "Iqra-UniversityOfficial".

• The **rule-based system** is required for determining the common words and identifying their fine-tune heuristics

1.7 Research Contributions

The thesis contribution is as follows;

•**Feature Improvement:**

Improved features of identifying fake tweets in which the comment of a user on tweet required to get modified because we need to examine whether the user who commented is fake or not? Feature selection and feature reductions play an important role in any research and its results.

•**Performance Optimization:**

We are getting improved accuracy through BERT because our features are improved with the presence of pooled output with 768 neurons. This is sent to the linear output layer with 1 neuron for predicting the results. [9] We have also chosen the transformer optimizer of AdamW and the loss function of Binary-Cross Entropy.

•**Performance Optimization of BERT-HAN:**

We are getting improved accuracy of BERT-HAN state of art model containing different layers and Sigmoid function to improve the performance of data-set with improved dependencies of cross-sentence.

•**Performance analysis** Through all the models, we did performance analysis and our results are better and improved.

1.8 Thesis Scope

The thesis scope is containing the entities involved in the variant chapter. The thesis scope is as follows;

1.8.1 Chapter 1

"Introduction" is containing the main description of social issues that arises from fake data, data privacy, factors involved in twitter data, contextual embeddings in Twitter, motivation, main research problem of the thesis, and a summary of the chapter.

1.8.2 Chapter 2

"Literature Review" is containing the literature review of 18 different papers, strengths, weaknesses, and threats of papers.

1.8.3 Chapter 3 "Problem Statement"

It is containing the problem definition with the research challenges involved in the thesis and the trend analysis that is being followed in related papers.

1.8.4 Chapter 4 "Proposed Solution and Methodology"

It is containing the description of Data Collection, Named entity recognition, entity linking, Data-set details, Data pre-processing, String encoding and decoding of data, Data cleaning, Data caching, Feature engineering and implementation details of Sentiment analysis using VADER, language modeling containing the bag of words, Term frequency-inverse document frequency, Modeling phases of Scikit learn and Sklearn pipeline, Exploratory data analysis, learning algorithms with the BERT Embeddings, implementation of BERT, Pre-training, Fine-Tuning, factors involved in fine-tuning, Python packages required in BERT Embeddings, parameters required in fine-tuning, Cross entropy in fine-tuning, Fast-text and BERT Embeddings, pip-tools, anaconda spyder, Data scrapping, Learning models, Exploratory data analysis and summary of the chapter.

1.8.5 Chapter 5 "BERT-HAN Architecture"

This chapter is containing the Hierarchical attention network with an attention layer and state of art algorithm, Architecture of HAN, text classification with HAN, word-level embeddings, word encoder of state of art algorithms, GRU-based sequence encoders, word attention, sentence level, BERT-HAN architecture, description of every layer involved in BERT-HAN and summary of the chapter.

1.8.6 "Experiments and Evaluations"

This chapter is containing the approaches, images of results with a great description, and summary of the chapter.

1.8.7 Chapter 7 "Conclusion and Future Work"

This chapter is containing the description of the entire summary of the thesis with the contribution, limitations, and future work. The thesis scope is containing the entities involved in variant chapter. The thesis scope is as follows;

1.9 Summary

To conclude, due to the exponential involvement, there is an increase in variant problems like online impersonation, fake profiles, and hate speeches. The data privacy gets disturbed. There are a lot of factors involved in the identification of fake data with the data extraction, pre-processing, cleaning, encoding of the string with the training and testing of the data set. The Twitter data is noisy and challenging to handle, there are different methods discussed to handle such data with no ambiguity.

Chapter 2

Literature Review

Fake account detection has increased and there are variant technologies to detect fake tweets. The following are the related works of different papers;

Saima et al. declared that once a scammer is detected, it becomes easy to block their IP address[10]. They have collected data from Twitter API's in which the data is publicly available on Twitter by accessing the developer account.They identified variant sets of words and other features which can result that a user being spam. Their scope was developing a twitter-based spam detection model which will classify the fake tweets or not using various features. They focused on the motive of spammers and scammers stealing important information from users. Their structure includes labeled spam tweets, labeled non-spam tweets. The data is being pre-processed on these tweets. The extracted features are then demonstrated after applying the model to whether it is spam or not. Their categories of research were the Naive Bayes model, Random forest model, Support vector machine, and K-nearest neighbor. Their features are closest to my research in which they included the number of followers and number of tweets.

L.Singh et al. proposed variant approaches to tackle fake news using complex techniques. They used two data sets Kaggle and LIAR in their experiments. [11] They used four-vector space representations in their experiments. These are Doc2Vec, Word2Vec, TF-IDF, and one-hot encoding. They worked on models that recognize the patterns and help in determining better output. Deep learning models that include Recurrent neural networks, Convolutions neural networks, and artificial neural networks have been proposed. [12]There is no constraint for the number of hidden layers in an artificial neural network.

They showed by experiments that the input vector's dimensions and several layers are directly proportional to each other. They get the accuracies of combinations of vector techniques and analyze the results. In comparison, CNN performed well with

TF-IDF on the Kaggle dataset. To conclude, TF-IDF is depicted that an efficient and simple algorithm for matching words in a query to documents that are relevant to the query. The research done by many of the scholars till now has proven that TF- IDF returns documents that are highly relevant to a particular query. Their proposed LIAR dataset did not perform well on vector space combinations. They did not have many similar words in them. Furthermore, Bi- LSTM encoded words were easy to attain in context to their ordering.

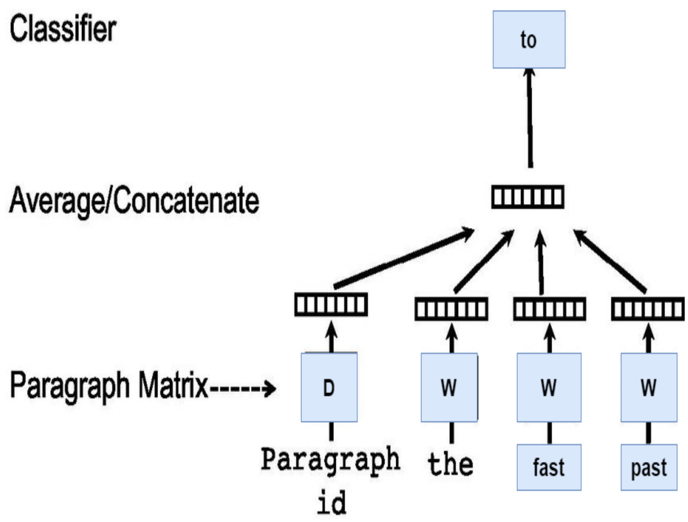


Figure 2.1: Working of Doc2Vec

Ahmed et al. proposed the framework that detects and classifies fake tweets and messages from Twitter using machine learning approaches. [15] They contained 5800 tweets on five rumor stories. They used RNN and several problems have been aroused. They were categorized into vanishing gradient and exploding gradient and resolved using dynamic weight. They also included the back-propagation. Their main approach was LSTMs, in which they depicted preserving the memory from the last phase and include this in the prediction task to attain the accuracy and final results for the identification of fake tweets. They added the hybrid method, in which the performance of the model improved and gave much better output for fake news detection. Their main scope was to detect the veracity of posts on Twitter. Their final results showed that the plain vanilla LSTM model achieved 82% best accuracy.

D. Bhowmik et al. proposed that the main domain was to demonstrate the fake profiles. [16] They have used the Support Vector Machine and classified fake and genuine profiles with a reasonable degree of accuracy. They required a platform that needs binary classification to be deployed on profiles. They limited the data for training.They faced a high variance problem and analyzed the results of high variance prob-

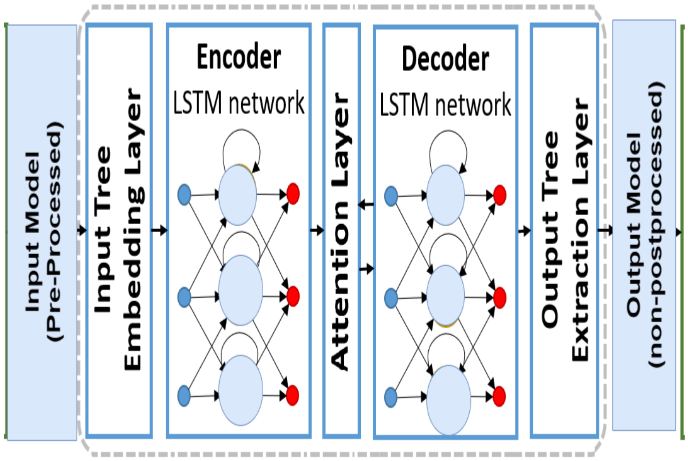


Figure 2.2: Working of LSTM

lems. They selected the profile to be tested, extracted the attributes required, passed it through a classifier and trained the classifier using the feedback.

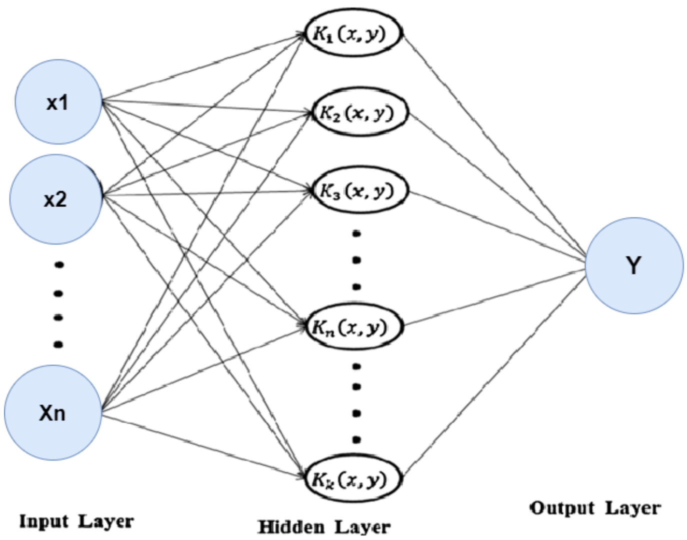


Figure 2.3: Working of SVM

Dr. B Deevena Raju et al. proposed machine learning models for detecting fake tweets in online social networks. [13] Their system utilized three variant classification algorithms to determine that provided dataset is fake or not. They have used supervised machine learning algorithms that include Support Vector Machine, in which they analyzed data for classification. It outputs an optimal hyper-plane that attains new examples. The proposed mathematical model by neural networks and random forest includes that the higher the accuracy results, the higher the number of trees. The data is pre-processed using various python libraries and it is closest to my research because we have also demonstrated my work in python using the fast-text library and several other libraries.

Fake News Detection using Machine Learning Ensemble methods[14], they explored the issue of classifying fake news articles using ensemble techniques. They collected the data from the world wide web and attained the news from variant domains. Their main focus was to identify the patterns in text that differentiate true news. Their learning models were tuned in such a way to increase the efficiency using the LIWC tool. The ensemble learners scored better on all performance metrics. They applied Linear SVM with 5-fold cross-validation for a meaningful comparison. The proposed CNN for automatic detection of fake news and contains only short statements. Furthermore, they have used BI-lSTm with variant feature sets. Their re- results demonstrated that the random forest algorithm achieved 95% accuracy and Bi-LSTM achieved 62% accuracy.

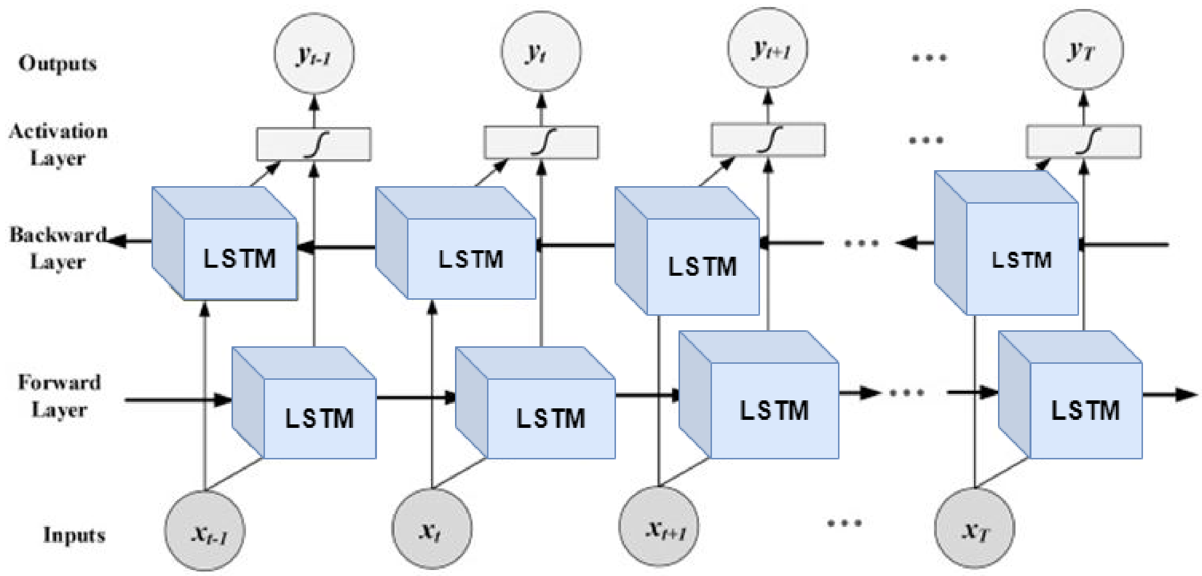


Figure 2.4: Working of Bi-LSTM

Krishnan G et al. explored by the experiments that users are less than the number of accounts present in social media. [17] They proposed the idea of using a gradient boosting algorithm with a decision tree containing three variant attributes. Those attributes are engagement rate, spam comments, and artificial activity. These inputs are utilized to form decision trees that are used in the gradient boosting algorithm.

Its the main benefit is providing the output instead of giving an error on some missing inputs. They got their data from a web scraper with the number of followings, number of posts, and number of followers. The level of post received on social media is measured by engagement rate. They are measured by using its formula. They also worked on calculating the frequency of commenting and liking. If a nominal number of shares, comments, and likes are found, it would be considered a fake account. Decision trees are taking the value which contains more fake accounts. This is close to my research.

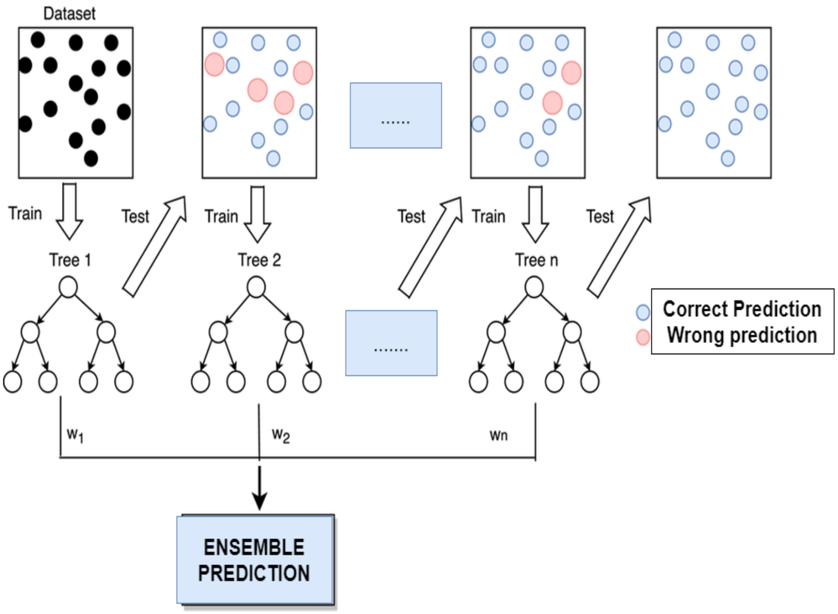


Figure 2.5: Working of Gradient Boosting Algorithm

R. Rathuri et al. proposed the architecture for identifying fake accounts in social media. They implemented machine learning SVM and CNN to attain great results. [18] Three important factors were identifying the account details by utilizing NLP, network identification. They contained twitter tweets, tagging, and several posts as features. Initially, they attained the account details of registration, they checked the network through which they were accessing. If Mac address and IP address are the same then they preferred to ask for authentication proof of the account, otherwise content needs to be checked. They gathered the repetition of words; words count and harmful pair of words. Furthermore, they got 96% accuracy from SVM and 94% accuracy from CNN.

Pereira et al. implemented a highly effective method of identifying fake news spread on Twitter using artificial Intelligence. [19] Their scope was to pre-process the data with the Keras model. This model was able to remove stop words and data was in- sorted into a specific tokenizer object. The data was, trained using machine learn-

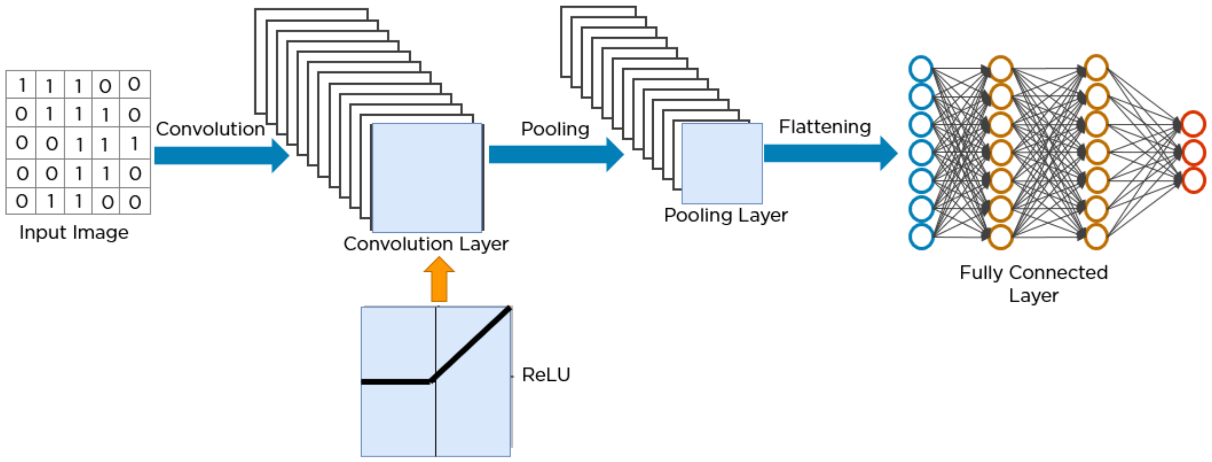


Figure 2.6: Working of Convolutional Neural Network

ing with the identification of patterns and prediction of falsified information. Deep learning develops an ensemble algorithm by configuring to decide on its own. Furthermore, the NLP pipeline was deconstructed into eight components including sentence segmentation. Word tokenization, tagging parts of speech, text lemmatization, stop words identification, dependency parsing, and noun phrases. The analysis has been done on generated figures. To conclude, LSTM had an accuracy of 94.64%, so it is considered the best model in their proposed work.

Chongstitvatana et al. proposed a novelty of work in which tweet vectors are utilized in a pairwise setting. Initially, input is fused into the network at variant layers to achieve optimal accuracy. They used the data of Nikiforos and contained 2363 tweets in English. They extracted 23 features and collected fake tweets were 936 Twitter accounts. [20] They categorized the features into Linguistic features and network account features. Their network contains an Embedding layer which is provided by the Kera's Library. Its input dimensions agree with the vocabulary of each input tweet text. They proposed a pairwise ranking approach with fake content. Their experiments contained several parameters including learning rate, optimizer, and loss function. They used 10-fold cross-validation to evaluate the predictive models by dividing the original sample to evaluate it. To conclude, they achieved 92% accuracy with Naive Bayes, 92.1% with Random Forest, and 92% with CNN.

Tong Zhang et al. proposed a machine learning method to identify misinformation from Twitter data. They first analyzed the misinformation by variant contents. The news content analysis contains the understanding of the context and semantics of words. [21] It also contains an emotional tone from the context of information. The source of information is being used by analysis without content. They presented the profile of distributors and other numerical data as attributes. They also presented the

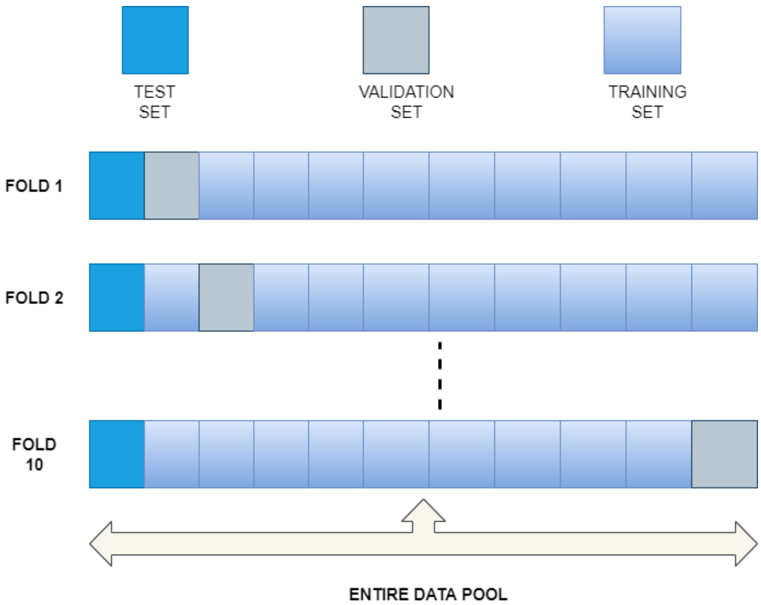


Figure 2.7: Working of 10 Fold Cross Validation

automatic detection of misinformation. In addition, they have used SVM for the representation of input data that is mapped into individual categories and can be utilized by classifying new data.

The data was extracted from Twitter API containing 948,373 tweets. The data was related to general issues in society. They have used Naive Bayes, Neural Network, and Support vector machine. The results showed that the support vector machine had an accuracy of 93.15% accuracy. They identified the keywords and searched the misinformation through those specific keywords. The limitation of their research is creating the data over a short period. To end this issue, more diverse news must be selected. It is also necessary to attain a large amount of data.

Saranya et.al. proposed that business, politics, and culture have a great impact on social media. [22] They proposed the positive and negative effects of fake data in our social lives. People are spreading fake news about different political and financial benefits. There is a great need to attain the fake information and they did the statistical analysis of Twitter account using data mining and different sources of cross verification.

The authors mainly worked on RNN in which they determined the weights and worked on two types including the vanishing gradient and exploding gradient. They have also used backpropagation. Their main limitation was a dependency on input data. The matrix-based approach should be used instead of the min-batch approach.

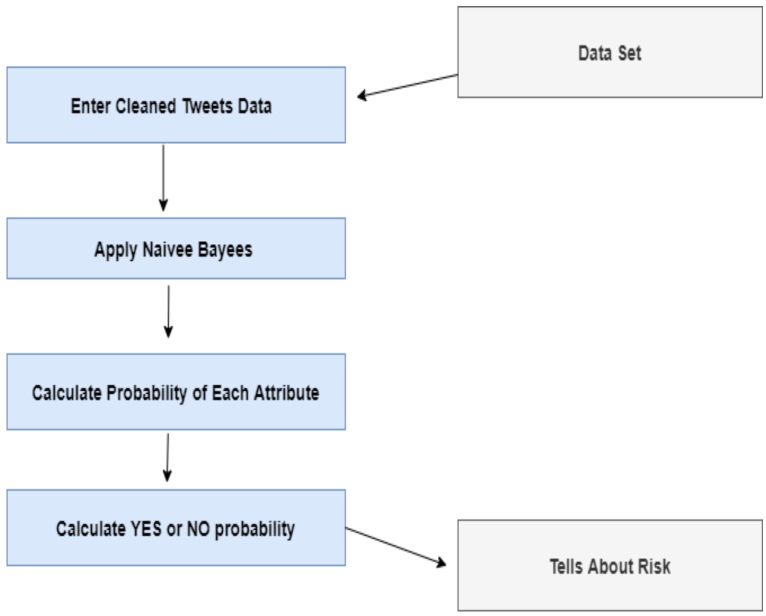


Figure 2.8: Working of Naive Bayees

Lenna et.al. proposed storing the data into the cloud. There is misinformation on social media platforms and it gains negative impacts. The authors have also used Big data analytic in cloud computing to attain the validity of tweets in the Covid19 pandemic. [23] They have used TextBlob and Tweepy to scrap the data and perform the sentiment analysis.

To conclude, it is notified that the implementation of cloud storage to build on a larger scale with the inclusion of different scientific discussions and elections. The verification and testing have been done to contain the best results and their final accuracy was 70.2%

Rohit et.al proposed that sequential neural networks have been used so that the social context level information can be encoded directionally. The ability to capture the long-term dependencies and capture semantic, bidirectional training approaches can be done. [24] The classification has been improved. They also implemented the deep learning approach in which BERT-based transformers have been used for pre-training and fine-tuning with a combination of different filters and sizes of kernels in Convolutional neural networks. They have attained an accuracy of 88% but there was a limitation in their results. To conclude, it was analyzed that CNN with BERT created an encoding issue because there was a lack of ability to the initial data and there is a huge requirement of training data in CNN.

Debanja et.al demonstrated that demo-graphical systems containing local languages with the datasets of two different languages have been used with bERT based model.There

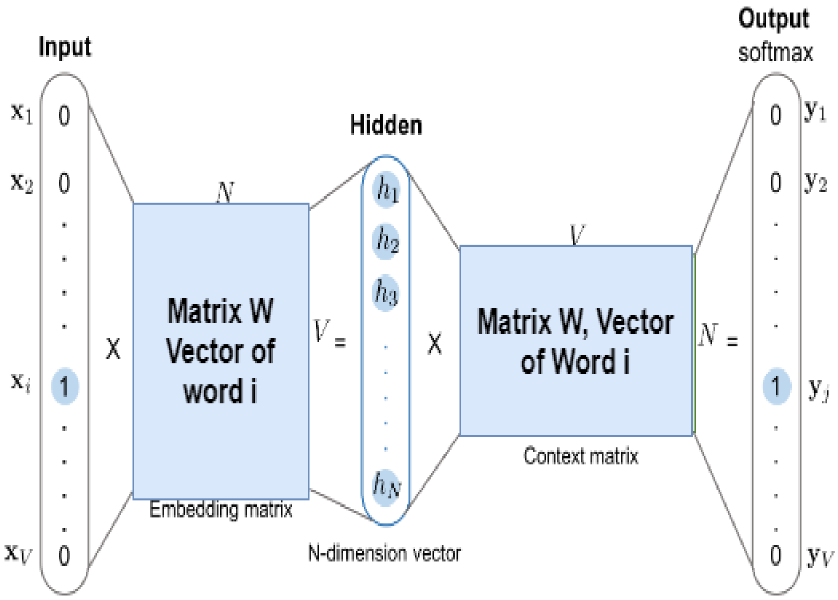


Figure 2.9: Working of Embeddings Matrix

were different features involved in its extraction. They have used Bert based model in which fine-tuning has been done in these two languages. [25] They got an 85% F- score with state of art results. Furthermore, specific use cases have been performed in which no labeled data have been used to classify the data or very few labeled data.

Nick et.al. have proposed that open source tools have been used with the Customer value patterns(CVP) team and predicted 90% on detecting the articles or journals. They demonstrated that BERT is an architecture in machine learning that takes input in the form of a whole phrase. It also allows the learn the embedding based on real-time context. [26] Furthermore, the authors just researched on BERT model and left the implementation for future work.

Junaed et.al. suggested different machine learning models. They have used three different datasets and put a benchmarking with the different machine learning approaches. [27] They also worked on pre-trained models to detect whether the data is real or fake. After analysis, they have found using benchmarking that the BERT model performed best with the fake news detection as it analyze the context from left to right and right to left. They also analyzed the BERT model on different articles' lengths and topics.

Gioqin et.al. worked on disaster management and filer out the noisy information in the beginning so that better results can be carried out with data. They disliked the filtration on basis of keywords. They worked on deep learning techniques using Twitter data. They compared the bidirectional LSTM with embeddings of twitter. [28]

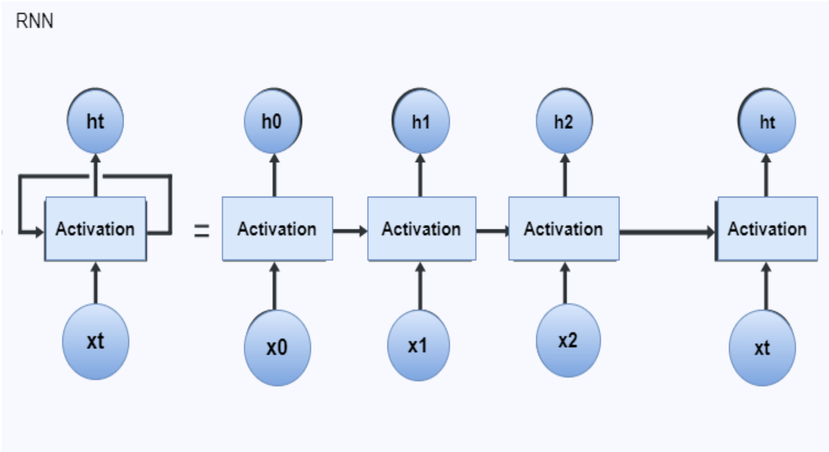


Figure 2.10: Working of Recurrent Neural Networks

They have used GLOVE embeddings (unsupervised learning algorithms) for attaining the vector representation of words. The Bert-based LSTM gave the best results with an average of 3.7% F-1 score. The results have been affected by subjectivity and limitation on ambiguities.

Anne et.al. worked on health-related data in which domain adaption and state of art approaches have been used. They have trained the embeddings with hyper-parameter tuning. [29] They have used the labeled existing data. The team TRMLeiden work has been used in this paper. Furthermore, the transfer learning methods using state-of-art have been classified, extracted, and pre-processed the drug effects. The code of their paper is available publicly.

Thanh et.al. worked on the BERT model containing BERT base architecture. They did training using the RoBERTa procedure of training in which RoBERTbase and XLM-rbase baselines have been used to attain the best results than the state of art models. They have used the MIT License with the release of BERTTweet. [30] Furthermore, They have used the 160/80 =2 times baselines and demonstrated that BERTTweet gave good results than RoBERTBase and XLM-Rbase. The pre-training has been done on English tweets with great results. They haven't used any other language. The dataset was not large.

Vijay et.al. proposed that views have been expressed in the real-time data. The meaningful information has been attained by authors using deep learning techniques. The authors have used two TweetBERT models and pre-training has been done on a large number of datasets containing a million tweets. Their results showed the hypothesis that trains the different models on a Twitter corpus. [31] They attained an accuracy of 78% with low context. Their main issue was backpropagation and the softmax function was not giving the best results which affected the entire results.

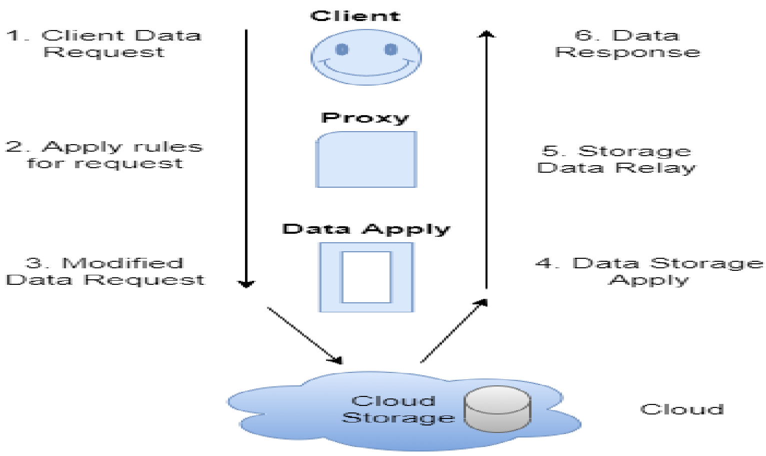


Figure 2.11: Working of Cloud Storage System

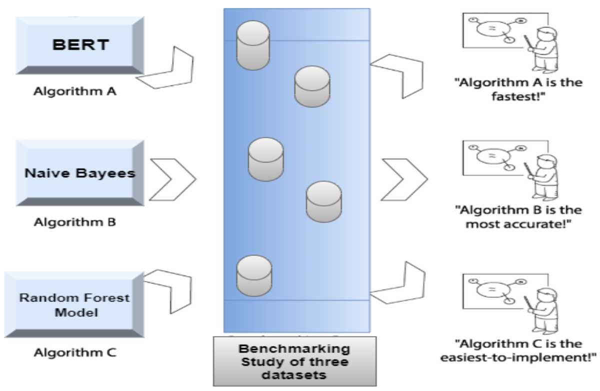


Figure 2.12: Working of Bench-marking Strategy in Machine Learning

Kaliyar et al. proposed in Fake-BERT: Fake news detection in social media with a BERT-based deep learning approach [32], this paper contains the bidirectional training approach. But they have done work on fake news and their accuracy needs to be improved with their context. It contains high priority for modeling the subsequent information of fake news and tweets. It can improve the classification performance to capture long-distance dependencies in tweets.

2.1 Research Gap

BERT is a deep learning approach that works on combining variant parallel blocks of single-layer CNN. [33] It has different filters with the BERT model. There are various features used in related work and new features need to be added to discriminate the work from others. FastText embeddings have been used for character n-grams and provide variant representations for Out-of-Vocabulary words. Its score and period need to be improved. The feature selection and feature reduction are added in the BERT layers. The score and weight of features need to be highlighted to discriminate the work and better results. Contrarily the proposed work has been done by using Bert on Twitter data which is more challenging to deal with because of the nature of the noisy and short text.[34] The BERT-HAN architecture is also used to improve the accuracy and generalized results.

2.2 Summary

The literature review of different papers is an essential part of identifying what is the latest trend involved in its identification. The authors have used the Naive Bayes model, Random forest model, Support vector machine, K-nearest neighbor, Doc2Vec, Word2Vec, TF-IDF, CNN, Bi-LSTM, SVM, SVM with CNN, LSTM, Glove embeddings, BERT Embeddings, and Benchmarking strategy. Furthermore, some authors have used the tweepy library for scrapping the tweets which require a developer account with an access token and secret keys from Twitter after getting approval of developer account and some authors have used the TWINT library of python which just requires a platform of python to run such as Spyder or Jupyter lab. The related papers contained the lack of metric distance, less number of features in some papers, used short sentences with CNN and results were affected, content rechecking was not modified, and lack of improved scores and weights of features in some papers.

Chapter 3

Problem Definition

3.1 Problem Statement

Identifying fake tweets is an optimization problem on Twitter, several techniques have been proposed in the literature to address the problem with different models as doc2vec, word2vec, Random forest model, Naive Bayes, Decision tree, SVM. Some students are spreading fake news about specific universities, which results in a bad image of the university. [35] However, the work is to be done for accessing the fake tweets and our domain is to access the fake tweets and accounts by using the BERT model, pre-training deep bidirectional representations from unlabeled text, and Fast-Text library with the consideration of limitations in the literature work to perform better on the identification of fake tweets.

3.2 Research Objectives

BERT is a deep learning approach that works on combining variant parallel blocks of single-layer CNN. It has different filters with the BERT model. There are various features used in related work and new features need to be added to discriminate the work from others. FastText embeddings have been used for character n-grams and provide variant representations for Out-of-Vocabulary words. [36] Its score and period need to be improved. The feature selection and feature reduction are added in the BERT layers. The score and weight of features need to be highlighted to discriminate the work and better results. Contrarily the proposed work has been done by using Bert on Twitter data which is more challenging to deal with because of the nature of the noisy and short text. The BERT embeddings have been proposed in which the tweets are tokenized using Bert tokenizer and sent to the Bert architecture to get the

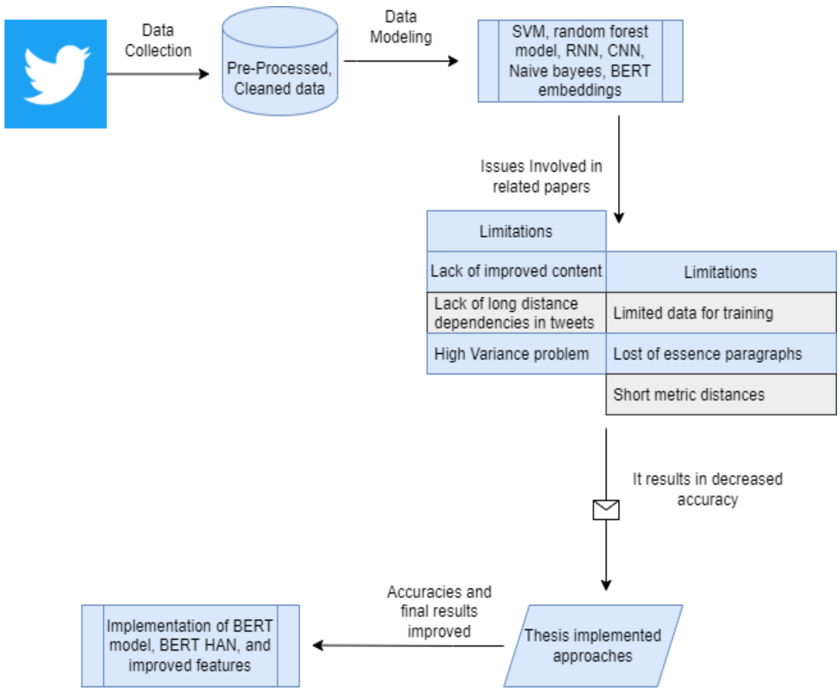


Figure 3.1:

two outputs, Sequence output and Pooled output. [37] We have used Pooled output in a model which contains output neurons of 768 This output Pooled output from Bert is sent to Output Linear layer which contains 1 neuron for prediction of results. [38] We have also chosen transformer optimizer of AdamW and loss function of Binary-Cross Entropy and a scheduler to tweak the learning rate. We have also used BERT-HAN architecture to increase the accuracy.

3.3 Research Questions

- How can we identify **fake tweets** using Deep learning models?
  - Can we include the **user profile features** to better identify fake tweets?
  - What is the various **effect of features** on the classifier?
  - How the **data-set can be improved** to attain the best results?

3.4 Summary

To summarize, the research gap plays an important role in research. We have analyzed eighteen papers and there are some issues in related papers that need to be addressed to attain an improved result of the Twitter data-set. The feature extraction has been done with unique features and existing features. We have used the feature of a user who commented on our attained tweet and an average number of tweets, tweets per day, retweets per day, and account life. Furthermore, these unique features have been used in BERT architecture, BERT-HAN architecture, and other machine learning algorithms to determine accuracy. The accuracy has been compared and BERT-HAN performed very well with increased accuracy.

Chapter 4

Proposed Solution and Methodology

4.1 Proposed Solution

The proposed solution of this research is to identify the fake or non-fake tweets to terminate the gaps identified from the related paper. There were issues in identification and the features have been improved according to the Twitter user. A unique state of art model BERT-HAN has been used containing different layers with Sigmoid function to attain improved accuracy. Furthermore, the solution has been provided mainly to enhance the features and attain the best results.

4.2 Methodology

The viewpoint of research considers a regulated set of variant principles regarding the real-world data analysis of a specified domain, all Universities of Pakistan. The deductive approach is applied here which starts with a definite development of theory on the review of past articles about the same or related topic and the researcher tries to process the model by pre-processing of collected data and further steps will be followed for execution flow with the validated results.

The different strategies will be applied by surveys and systematic related work. [39] Secondary data has been collected from Twitter by scrapping. The proposed approach mainly focuses on the technique of spreading fake information about others and results in hate. The research gap has depicted that the authors worked more on engagement rate but other factors were not evaluated clearly and the accuracy needs to be improved.

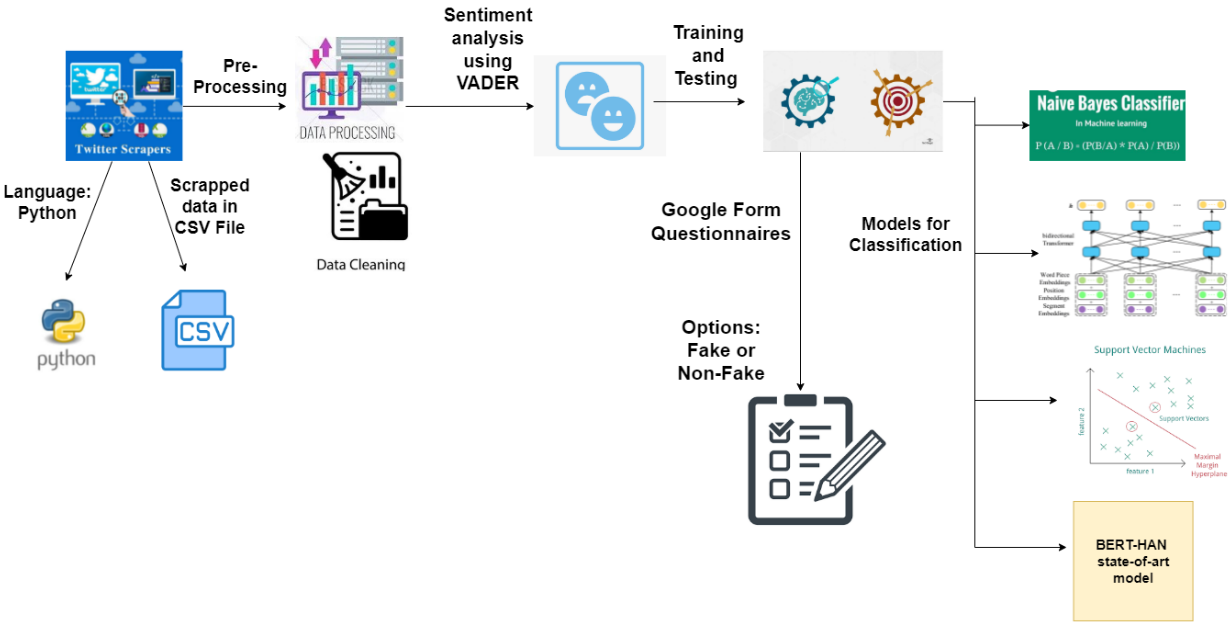


Figure 4.1: Flow of Methodology

4.2.1 Data Collection

There are different ways of scrapping. There was a use of GetOldTweets3 by Twitter scrappers but that is not useful now and twitter has terminated its endpoint. There are no further updates regarding this package. There is paid software available on the web for scrapping and Octaparse is a paid software that can attain the data from the browser without using code and extracts the specified content from the webpage. It's very time-consuming because the webpage needs to get the scroll and it's not free of cost. We have used a python library named Twint for scrapping the data from Twitter that included the created at, date, tweets, retweets, source, and URLs but the number of followers, number of followings, number of statuses count and verified ware not accessible from a library named TWINT so Python library named TWEEPY have been proposed by using names of users and API credentials of twitter given by developing a Twitter developer account. [40] The hybrid approach has been proposed by using these two libraries so that good information can be generated for the identification of fake or real tweets. The tweets have been scraped by using hashtags or searching the name of a string that can be processed to attain the information.

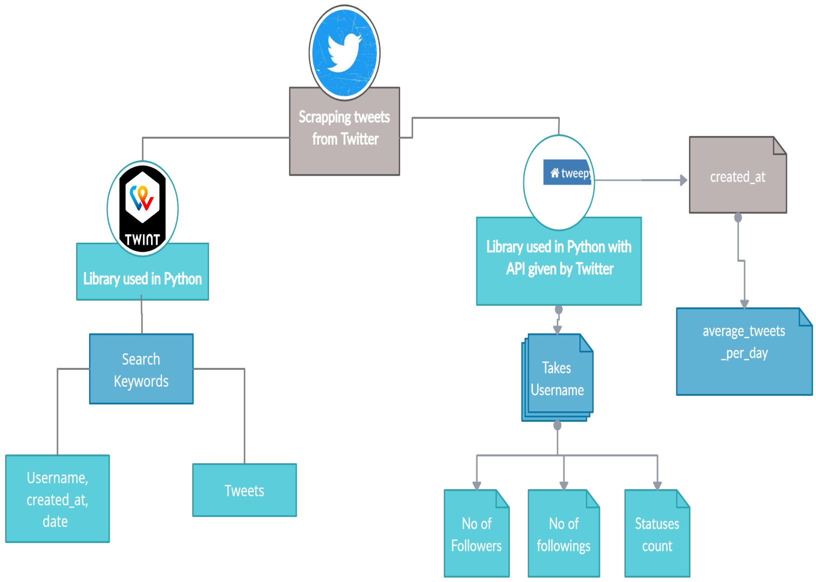


Figure 4.2: Scraping of Tweets

4.2.2 Data Scrapping and Learning Models

Extracting correct data and information played a vital role in project management. The comparison of entities with the bases of existing knowledge or its data has identified entities to determine whether the tweets or account is fake or not.

The hybrid approach has been proposed by using TWINT and TWEEPY, libraries so that good information can be generated for the identification of fake or real tweets.[42] The tweets have been scraped by using hashtags or searching the name of a string that can be processed to attain the information.

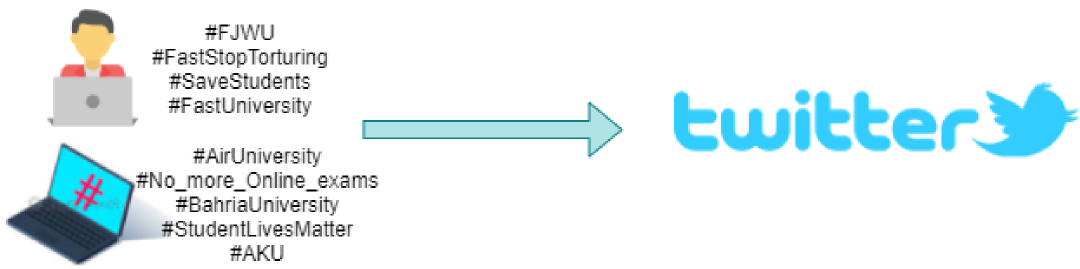


Figure 4.3: Scraping by Hash-tag or Search keyword using TWINT

The scrapped tweets have been encoded and decoded because after generating the data from Twitter, there was a problem of string encode and decode. It needs to be fixed because of smileys and some other symbols otherwise data would not be processed. Pre-processing has been proposed by different methods. The Encoder and Decoder problem suffered from a similar problem if the inputs are not connected, then the output remain as zero. The Decoder will attain that as high bit 00 with logic 0.

```python
import twint
import nest_asyncio
nest_asyncio.apply()

c = twint.Config()
c.Search = '#FastNuces'
c.Since = "2002-03-16 00:00:01"
c.Until = '2021-09-30 23:59:59'
c.Store_json = True
c.Store_object = True
c.Profile_full = True
c.Output = "FastNuces.json"
twint.run.Search(c)
```

Figure 4.4: Scrapping Tweets in PYTHON

Generally, the real-world data is inconsistent with the discrepancies in the names or its codes. The data contain errors or outliers with a lack of certain attributes and incomplete data set with less amount of information involved. [43] The data has been pre-processed in which data cleaning, data integration, data reduction, and data transformation can be done. If we have a raw dataset in our hands, data mining is an important process and it has done by transferring the raw data into an understandable format. Before sending data to the model, pre-processing is the main step otherwise modeling would contain noisy words and errors then our accuracy would be badly affected. The libraries were imported with reading the data through pandas.

4.2.3 Named entity recognition

The classification of entities in an unstructured text into pre-defined categories have been extracted by named-entity recognition. [44] It is an essential task because it helps to identify the students and their tweets related to a specific topic. It worked on checking fake veracity with NLP packages. There are different packages but some packages have decrease the relation mapping score for Twitter data. There is a named entity recognizer, Google Cloud NLP which can be accessed through Google Cloud API and contains the topmost deep learning models, and then it has given many unclassified outputs so the transformer models depicted good results.

4.2.4 Entity Linking

Entity linking contains the named entity disambiguation and named entity recognition. The entity linking and pre-processing are the most important parts of data extraction.[45] The data is structured, informal, unstructured, and edited content. Therefore, there is a great need for entity linking, pre-processing, and data extraction. We have used predefined hashtags and strings used in our domain. Our main domain is fetching the tweets of different universities of Pakistan so we have used many strings to fetch the data. The presence of pre-defined hashtags helped me in further improvement of data instead of fetching tweets of an individual or a single string.

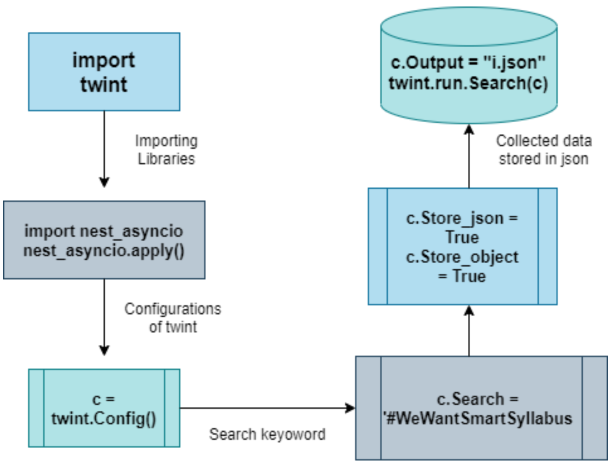


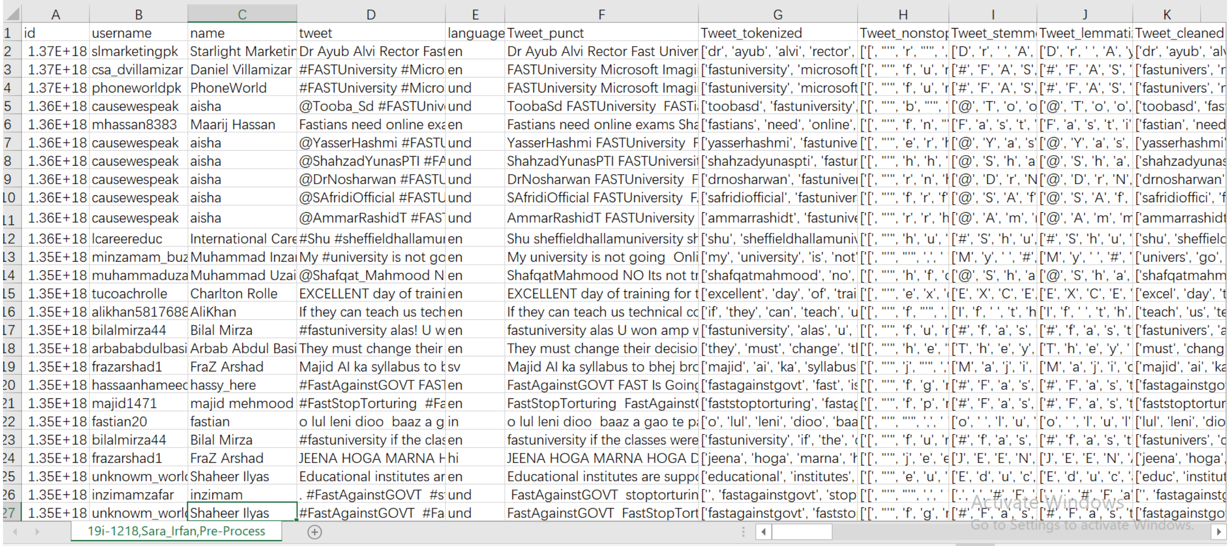
Figure 4.5: Steps of Collected Data through Twint

4.2.5 Twitter Data-set

There are a total of 15,000 tweets about different universities of Pakistan, pre-processing has been done on these tweets for identifying the fake tweets and analyzed for further methodological approach. [46] Firstly, data have been gathered, then the analysis is being done on different steps involving Deep Learning Techniques, it requires the processing of data and then working on data related to Data Science techniques and further steps as shown below.

4.2.6 Data Pre-Processing

The pre-processing has been done with the extraction of punctuation of tweets, tokenization of tweets, removal of non-stop words, tweet stemming, tweet lemmatization, and cleaning of tweets. The data contain irrelevant and missing parts and to terminate this part, cleaning of data is done which contains noisy data and missing data. [47] Tokenization is the process of splitting text into meaningful pieces. Such pieces are called tokens. Stemming is a rule-based terminating the fixes ("s", "ly", "ing" etc) from a word. Lemmatization is grouping with the analysis of a single word's lemma from variant inflected forms of a word.



	A	B	C	D	E	F	G	H	I	J	K
1	id	username	name	tweet	language	Tweet_punct	Tweet_tokenized	Tweet_nonstop	Tweet_stemm	Tweet_lemmati	Tweet_cleaned
2	1.37E+18	slmarketingpk	Starlight Marketin	Dr Ayub Alvi Rector Fast	en	Dr Ayub Alvi Rector Fast Univer	['dr', 'ayub', 'alvi', 'rector',	['[', "", 'r', "", '	['D', 'r', ' ', 'A',	['D', 'r', ' ', 'A', 'y	['dr', 'ayub', 'alv
3	1.37E+18	csa_dvillamizar	Daniel Villamizar	#FASTUniversity #Micro	en	FASTUniversity Microsoft Imagi	['fastuniversity', 'microsoft	['[', "", 'f', 'u', 'r	['#', 'F', 'A', 'S',	['#', 'F', 'A', 'S', '	['fastunivers', 'r
4	1.37E+18	phoneworldpk	PhoneWorld	#FASTUniversity #Micro	und	FASTUniversity Microsoft Imagi	['fastuniversity', 'microsoft	['[', "", 'f', 'u', 'r	['#', 'F', 'A', 'S',	['#', 'F', 'A', 'S', '	['fastunivers', 'r
5	1.36E+18	causewespeak	aisha	@Tooba_Sd #FASTUniv	und	ToobaSd FASTUniversity FASTi	['toobasd', 'fastuniversity',	['[', "", 'b', "", '	['@', 'T', 'o', 'o	['@', 'T', 'o', 'o',	['toobasd', 'fas
6	1.36E+18	mhassan8383	Maarij Hassan	Fastians need online exa	en	Fastians need online exams Sha	['fastians', 'need', 'online',	['[', "", 'f', 'n', "	['F', 'a', 's', 't', '	['F', 'a', 's', 't', 'i	['fastian', 'need
7	1.36E+18	causewespeak	aisha	@YasserHashmi #FASTU	und	YasserHashmi FASTUniversity F	['yasserhashmi', 'fastunive	['[', "", 'e', 'r', 'h	['@', 'Y', 'a', 's	['@', 'Y', 'a', 's',	['yasserhashmi'
8	1.36E+18	causewespeak	aisha	@ShahzadYunasPTI #FA	und	ShahzadYunasPTI FASTUniversi	['shahzadyunaspti', 'fastur	['[', "", 'h', 'h', '	['@', 'S', 'h', 'a	['@', 'S', 'h', 'a',	['shahzadyunas
9	1.36E+18	causewespeak	aisha	@DrNosharwan #FASTU	und	DrNosharwan FASTUniversity F	['drnosharwan', 'fastunive	['[', "", 'r', 'n', 'h	['@', 'D', 'r', 'N	['@', 'D', 'r', 'N',	['drnosharwan'
10	1.36E+18	causewespeak	aisha	@SAfridiOfficial #FASTU	und	SAfridiOfficial FASTUniversity F	['safridiofficial', 'fastuniver	['[', "", 'f', 'r', 'f	['@', 'S', 'A', 'f	['@', 'S', 'A', 'f',	['safridioffici', 'f
11	1.36E+18	causewespeak	aisha	@AmmarRashidT #FAS	und	AmmarRashidT FASTUniversity	['ammarrashidt', 'fastunive	['[', "", 'r', 'r', 'h	['@', 'A', 'm', '	['@', 'A', 'm', 'm	['ammarrashidt
12	1.36E+18	lcareereduc	International Care	#Shu #sheffieldhallamu	en	Shu sheffieldhallamuniversity sh	['shu', 'sheffieldhallamuni	['[', "", 'h', 'u', '	['#', 'S', 'h', 'u',	['#', 'S', 'h', 'u', '	['shu', 'sheffield
13	1.35E+18	minzamam_buz	Muhammad Inza	My #university is not go	en	My university is not going Onli	['my', 'university', 'is', 'not'	['[', "", "", ',', '	['M', 'y', ' ', '#',	['M', 'y', ' ', '#', '	['univers', 'go',
14	1.35E+18	muhammaduza	Muhammad Uzai	@Shafqat_Mahmood N	en	ShafqatMahmood NO Its not tr	['shafqatmahmood', 'no',	['[', "", 'h', 'f', '	['@', 'S', 'h', 'a	['@', 'S', 'h', 'a',	['shafqatmahm
15	1.35E+18	tucoachrolle	Charlton Rolle	EXCELLENT day of traini	en	EXCELLENT day of training for t	['excellent', 'day', 'of', 'trai	['[', "", 'e', 'x', '	['E', 'X', 'C', 'E',	['E', 'X', 'C', 'E', '	['excel', 'day', 't
16	1.35E+18	alikhan5817688	AliKhan	If they can teach us tech	en	If they can teach us technical cc	['if', 'they', 'can', 'teach', 'u	['[', "", 'f', "", '	['I', 'f', ' ', 't', 'h	['I', 'f', ' ', 't', 'h',	['teach', 'us', 'te
17	1.35E+18	bilalmirza44	Bilal Mirza	#fastuniversity alas! U w	en	fastuniversity alas U won amp v	['fastuniversity', 'alas', 'u',	['[', "", 'f', 'u', 'r	['#', 'f', 'a', 's',	['#', 'f', 'a', 's', 't	['fastunivers', 'a
18	1.35E+18	arbababdulbasi	Arbab Abdul Basi	They must change their	en	They must change their decisio	['they', 'must', 'change', 'tl	['[', "", 'h', 'e', "	['T', 'h', 'e', 'y',	['T', 'h', 'e', 'y', '	['must', 'chang
19	1.35E+18	frazarshad1	FraZ Arshad	Majid AI ka syllabus to b	sv	Majid AI ka syllabus to bhej brc	['majid', 'ai', 'ka', 'syllabus	['[', "", 'j', "", '	['M', 'a', 'j', 'i',	['M', 'a', 'j', 'i', 'c	['majid', 'ai', 'ka
20	1.35E+18	hassaanhameed	hassy_here	#FastAgainstGOVT FAST	en	FastAgainstGOVT FAST Is Going	['fastagainstgovt', 'fast', 'is	['[', "", 'f', 'g', 'r	['#', 'F', 'a', 's',	['#', 'F', 'a', 's', 't	['fastagainstgo
21	1.35E+18	majid1471	majid mehmood	#FastStopTorturing #Fa	en	FastStopTorturing FastAgainst(	['faststoptorturing', 'fastag	['[', "", 'f', 'p', 'r	['#', 'F', 'a', 's',	['#', 'F', 'a', 's', 't	['faststoptortur
22	1.35E+18	fastian20	fastian	o lul leni dioo baaz a g	in	o lul leni dioo baaz a gao te pa	['o', 'lul', 'leni', 'dioo', 'baa	['[', "", "", ',', '	['o', ' ', 'l', 'u', '	['o', ' ', 'l', 'u', 'l	['lul', 'leni', 'dio
23	1.35E+18	bilalmirza44	Bilal Mirza	#fastuniversity if the clas	en	fastuniversity if the classes were	['fastuniversity', 'if', 'the', 'c	['[', "", 'f', 'u', 'r	['#', 'f', 'a', 's',	['#', 'f', 'a', 's', 't	['fastunivers', 'c
24	1.35E+18	frazarshad1	FraZ Arshad	JEENA HOGA MARNA H	hi	JEENA HOGA MARNA HOGA D	['jeena', 'hoga', 'marna', 'h	['[', "", 'j', 'e', 'e	['J', 'E', 'E', 'N',	['J', 'E', 'E', 'N', '	['jeena', 'hoga',
25	1.35E+18	unknowm_worl	Shaheer Ilyas	Educational institutes ar	en	Educational institutes are supp	['educational', 'institutes',	['[', "", 'e', 'u', '	['E', 'd', 'u', 'c',	['E', 'd', 'u', 'c', '	['educ', 'institut
26	1.35E+18	inzimamzafar	inzimam	. #FastAgainstGOVT #s	und	FastAgainstGOVT stoptorturin	['', 'fastagainstgovt', 'stop	['[', "", "", ',', '	['.', ' ', '#', 'F', '	['.', ' ', '#', 'F', 'a	['', 'fastagainstg
27	1.35E+18	unknowm_worl	Shaheer Ilyas	#FastAgainstGOVT #Fa	und	FastAgainstGOVT FastStopTort	['fastagainstgovt', 'faststo	['[', "", 'f', 'g', 'r	['#', 'F', 'a', 's',	['#', 'F', 'a', 's', 't	['fastagainstgo

19i-1218,Sara_Irfan,Pre-Process

Figure 4.6: Pre-Processing of Tweets

Due to the non-availability of labels in an unsupervised learning clustering will be done. It can only show the grouping with results of several groups. The groups have been divided into two or more groups with the hard code and define the metric of grouping. [48] There are numerous ways to mimic those groups which are closer to fake or not. For example, if metadata is being used of followers, that likely shows that it is a fake account because of less number of followers. It would not give an accurate result.

In this way, the accuracy of models have been decreased and it is the absolute ground truth. If the accuracy of a model is less but it would not be much useful. The clustering problem is to identify the potential fake tweets in Which users are consistently re-tweeting the same account? Reasonable results are required with the help of supervised learning. A systematic study has been done to get the minimum amount of label tweets that are needed to get a good classifier model.

4.2.7 Encoding and Decoding of Data

The scrapped tweets have been and encoded and decoded because after generating the data from Twitter, there was a problem of string encode and decode. It needs to be fixed because of smileys and some other symbols otherwise data would not be cleaned properly.

Pre-processing has been proposed by different methods. [49] It needs to convert from one encoding scheme to another required encoding scheme and takes the encoding string to decode that string with returning the original string. The encoding gets specified based on decoding. For example, the Unicode error gets generated by the word 'difficult' and 'ignore' just ignoring the error. It can also be used in storing credentials in the back and some other applications are also involved to hide the original text or generate the original text.

	username	name	place	tweet	language	user_nam
18	shabana9	Shabana		@UCPoffi	en	shabana9
18	uetlahore	Maimona	@Activista	#UETians_	und	uetlahore
18	uetlahore	Maimona	@Activista	#UETians_	und	uetlahore
18	zabi_am	��m z��b��		����	ur	zabi_am
18	uetlahore	Maimona	@Activista	#UETians_	und	uetlahore
18	m_bilal_ra	M Bilal Rajpoot		We want j	en	m_bilal_ra
18	m_bilal_ra	M Bilal Rajpoot		UCP GUAR	ca	m_bilal_ra
18	aroojbash	Arooj Basharat		@Muham	und	aroojbash
18	aroojbash	Arooj Basharat		@iaizaz_k	tl	aroojbash
17	bilalah307	BILAL AHMAD		900 Choot	hi	bilalah307
18	justicefrst	Justice For Students		Congratul	en	justicefrst
18	ashmani1	Abdul Manan		Shame sh	en	ashmani1
17	boyxtrove	��������		#Students	und	boyxtrove
18	oye_bhatt	������� �		I don't kno	en	oye_bhatt
17	ikramraja_	Ikram Raja		����	ur	ikramraja_
)9	m_a_atiiii	Wolverine ����		Why you�	en	m_a_atiiii
18	muhamm	Muhammad Mobeer		#UCPians\	und	muhamm
)9	m_a_atiiii	Wolverine ����		@ImranKh	en	m_a_atiiii

Figure 4.7: Issue of Encoding Decoding after scrapping tweets

4.2.8 Data Cleaning

The cleaning of tweets was required because of incorrect, duplicate, or incomplete data within my scraped data. [50] It is clear from the observation if the data is collected from different persons, sources, groups then data contains mislabeled or noisy data and results in unreliable data. The process varies from one data set to another data-set. The duplicate data occurs during the collection of data.

The incorrect capitalization and naming conversations occurred as structural errors. The outliers also occur which means that deviating from the rest of the objects and

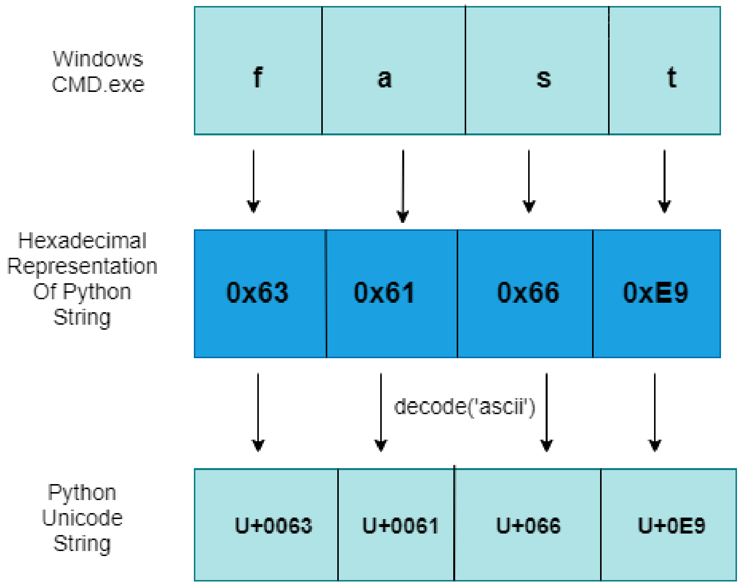


Figure 4.8: Steps of Encoding and Decoding

caused by an execution error that needs to be eliminated. There are some training methods or algorithms, that do not accept missing values and there are a couple of ways to deal with them. [51] We have cleaned our data in python and some questions need to be fulfilled in our observations; **Is there any trend to form the next work after data cleaning? Is it following rules and regulations?**

4.2.9 Data Caching

There was no change once data was cleaned and it took much time as well. [52] The issue have been addressed to get an improved data from model training so we converted our data-set into the bytes and then written into another CSV file. Furthermore, our CSV file is loaded back as an object to attain data whenever required.

4.3 Feature Engineering

We have applied feature engineering on my tweets because some features are contained from our existing data set. The selected features are as follows;

1) There are various users, they tweeted on multiple days and the frequency of tweets need to be calculated so that the behavior of tweets can be analyzed in the trend and named as ”number of days user tweeted”

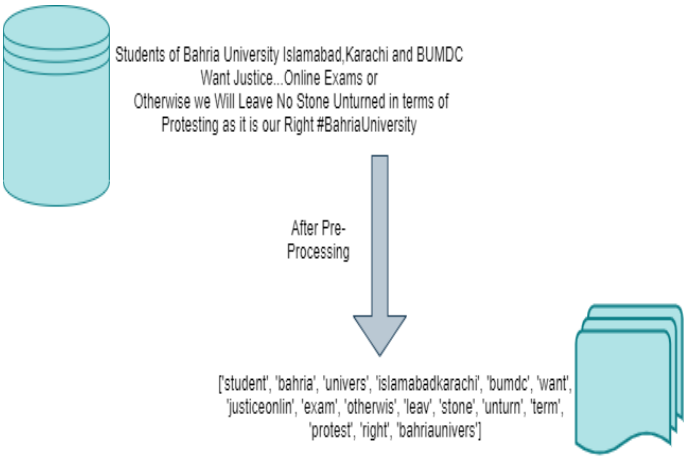


Figure 4.9: Sample of Cleaned tweet

```
def clean_text(text):
    text_lc = "".join([word.lower() for word in text if word not in string.punctuation]) # remove puntuation
    text_rc = re.sub('[0-9]+', '', text_lc)
    tokens = re.split('\W+', text_rc)    # tokenization
    text = [ps.stem(word) for word in tokens if word not in stopword]  # remove stopwords and stemming
    return text
df['Tweet_cleaned'] = df['tweet'].apply(lambda x: clean_text(str(x)))
```

Figure 4.10: Code for cleaning tweets

2) The most popular users contain high average likes, so we considered ”average likes”.

3) The calculation of ”average replies” results in analyzing the posts of users.

4) The involvement information of the author is required and to include that feature, we calculated ”average retweet”.

5)The fake account identification can be considered by calculating the total number of tweets against that user, we considered ”total tweet” as a feature in this case.

6)The tweets are increasing day by day and ”average tweets per day” needs to be considered to get identification if fake or real tweet with the time-frequency of a user.

7)The unique hashtags need to be considered in the identification of fake tweets, that is why we considered ”hashtag uniqueness” as a feature to identify the uniqueness of tweets.

8)A unique feature that attains the person commenting on an account after the post of an account, to check whether that comment-generated account is also fake or not?

9) Sentiment Polarities from tweets

- Sentiment polarity of every tweet.
- Calculation of polarity of negative and neutral tweets.[53]
- Features used for our model training, testing and depicts complete behaviour of users.
- It will improve the accuracy of model.

10) Context Distance from Tweet Embedding

The contextual embeddings set every word's representation on its content and attain the uses of words with the context across languages. The existing contextual embeddings are used in this step for pre-training of a cross-lingual polyglot. The analyses of models with their compression are done in their work.

- Firstly python data frame is required in which tweets of every user will be separated.

  df = pd.DataFrame(tweets) print(df)
- We have used the sentence embedding of Universal Encoders by using BERT Transformers with the conversion of great detail of tweets into the vectors.[54]
- The context of the tweet has been represented by the resultant vector taken from the mean calculation of the entire tweet.
- The standard deviation has been calculated of users' tweets so that the similarity of tweets can be encountered easily and results in contextual information of a single user.
- Furthermore, the cosine distance has been calculated with the user's initial tweets and other tweets so that analysis can be increased.

4.4 Sentiment Analysis using VADER

The sentiment analysis is a methodology of analyzing text that detects subjectivity and polarity within some text, an entire document, and a paragraph. [55] It has different methods with or without labels. We have used VADER that is a Valence aware dictionary for analyzing the sentiment analysis of a text. It generates the polarity of both positive and negative with the strength of emotion. NLTK package is being used for its analysis. It also maps the variant lexical features to attain the sentiment score and sentiment type have been generated with all metrics. This sentiment type have been changed to different numbers such as 0 for Negative, 1 for Positive, and 2 for Neutral type.

The analyses of text are required in every context of identification of text and sentiment analyses detect the polarity of a text including the negative or positive sentiment type in an entire document containing paragraph, or any clause. The computational subjectivity of a text has been declared in sentiment analysis using VADER. [56] For example, a sentence ”The Fast University is good, but the quality of content needs to be improved”. This sentence is depicting two polarities. VADER is such a kind of sentiment reasoning in which the intensity of emotion involves the positive and negative polarity. It has done for unlabeled data-set that contain the package of NLTK and mapping of lexical features with sentiment scores. For example, ’hate’, ’torture’, ’bad’, ’like’ all words convey a negative sentiment and attains capitalization such as ”PATHETIC”

```
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import pandas as pd

df = pd.read_csv("C:/Users/Ali/.spyder-py3/twint/AllUniversities.csv")
df.head()

sid = SentimentIntensityAnalyzer()
df['scores'] = df['tweet'].apply(lambda tweet: sid.polarity_scores(tweet))
df.head()

df['compound'] = df['scores'].apply(lambda score_dict: score_dict['compound'])
df['sentiment_type']=''
df.loc[df.compound>0,'sentiment_type']='POSITIVE'
df.loc[df.compound==0,'sentiment_type']='NEUTRAL'
df.loc[df.compound<0,'sentiment_type']='NEGATIVE'

df.sentiment_type.value_counts().plot(kind='bar',title="sentiment analysis")
df.to_csv("Vaderonlatest15sepdecoded.csv",index=False)
df['sentiment_type'].value_counts()
result1 = pd.read_csv("VaderData.csv")
print(result1)
df['sentiment_type'].value_counts()
```

Figure 4.11: Code of Sentiment Analysis using VADER in Python

Polarity classification

The opinions are preferred in the polarity classification and the subjective of text is not important.

Document-level classification scope

The entire opinion is discussed in this step containing the paragraph and document. [57] It helped us in calculating the sentiment score with the polarity and subjectivity.

Steps for Sentiment Analysis

Working on complete sentences is required, that is why we have used sentiment analysis using VADER and full information can be gathered. There are numerous ways to gather the context of data. For example, two-word combinations can also be considered but it would not easily provide complete information that what this text is about?

For this model, we can use a variety of datasets like scraped data of different universities to get what students spread about universities.VADER's SentimentIntensityAnalyzer() takes in a string and returns a dictionary of scores in each of four categories:

- negative
- neutral
- positive
- compound

To commence, the columns are added in my original data frame with the dictionary of polarity score, that has been stored in our CSV data and new negative /positive labels have been attained from the compound score. The accuracy test has also been included in my data set using VADER. The ratio of positive, negative, and neutral tweets was required with the entire information about the data-set can be maximized. The values greater than zeroes are considered positive tweets and the values less than zero are considered negative tweets.

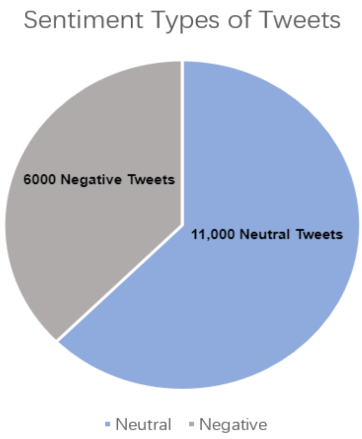


Figure 4.12: PieChart of Sentiment Types

4.5 Language Modelling

The language models show the text as numerical vectors and it is good for machine learning or deep learning models. The predictive models and count-based models are discussed in my language modeling. [58] The count-based method computed the numerical calculations about our word occurrences and the predictive models result in the interpretation of probabilities. It resulted in statistical calculations with a great understanding of a text.

Bag of words

The bag of words contains all the unique words involved in any clause, paragraph, or text. We attained all the unique words in the bag of words. It is used for uni-gram models. [59] For example, in my corpus

(1) I love FAST university too

(2) Hamza likes FAST University too.

In the above sentences, the context of two tweets is considered and resulted in discrimination of classes. N-grams are sequences of characters and result in the probabilities of n-1 words.

TFIDF

Term-Frequency Inverse Document frequency contains term and inverse documents. It can be calculated using the formula;

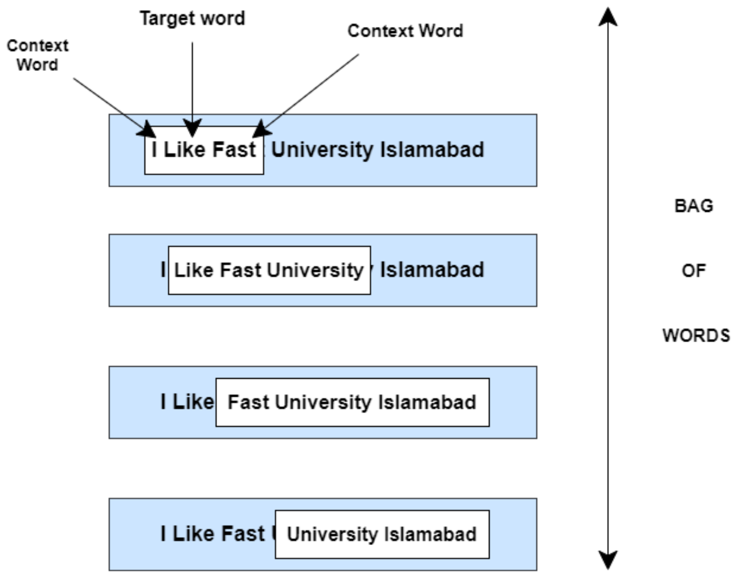


Figure 4.13: Bag Of Words

Tfidf = tf x idf

The inverse document frequency attains the importance of a word in our entire training set and the term frequency reflects the word importance in the whole document. It is a smoothing technique because it improved the distribution of our tweets but there is an absence of morphological aspects of tweets.

Word embeddings

The prediction of context involves word embeddings. The words contain similar meanings with the occurrence of the same context. The similarities of words are captured in the word2vec. [61] The high teams have utilized the word embeddings on Twitter. The Bag of Words has shown better results than word embeddings.

4.6 Modeling Phase

This section covers the implementation of machine learning models that used the data from the Annotation phase.

Scikit Learn

The state-of-art machine learning algorithms contain different modules and Scikit learn is an open-source Python module that is used for clustering and classification algorithms. It supported SVM, Naive Bayes, and Decision trees in my work. It contains the Numpy and Scipy libraries. It is written in Python. The machine learning algorithms contain the training and testing part for model classification. [62] It contains two functions fit and transform.

- Fit Method: Learn the parameters of model in it.
- Transform Method: The model has been applied to unseen data in the transform method

The features are also involved in it. We have used the textual feature from Scikit learn and we have transformed the textual data into the vectors for different algorithms with the pre-processing in the beginning. The tokenization has been done in CountVectorizer for analyzing the matrix of every token. We have used the n-grams, unigrams, bigrams, and trigrams of words and trigram gave me the best results. The TFIDF has taken the word counts from CountVectorizer with the conversion of the tweet into vector-based.

Sklearn Pipeline

The Sklearn merge the different estimators into one in the static sequence of steps. It was used for classification, selection of features, and normalization. The call function is required that fits with the corpus data and analyzes the test data for predictions. [63] Furthermore, it was inserted into the pipeline. The feature union has also been analyzed in my working and an Item-Selector class has also been utilized so that the estimator of Base and Transformer-Mixin have been applied. The fit function and transform function give the assembled data.

Exploratory Data Analysis

The approach has been planned by EDA (Exploratory Data Analysis). The planning investigations were required and have been implemented in this step. The numerous investigations are checking the assumptions, testing inferential and statistical hypotheses with anomaly detection. They have been shown by the graphical analysis by an import named matplotlib as a package in Python. [64] The statistical action on the data-set requires the initial working. The short text has been generated from tweets. The data-driven decisions have been analyzed and implemented by taking into account the EDA with revealing the hidden patterns and internal functionalities of data.

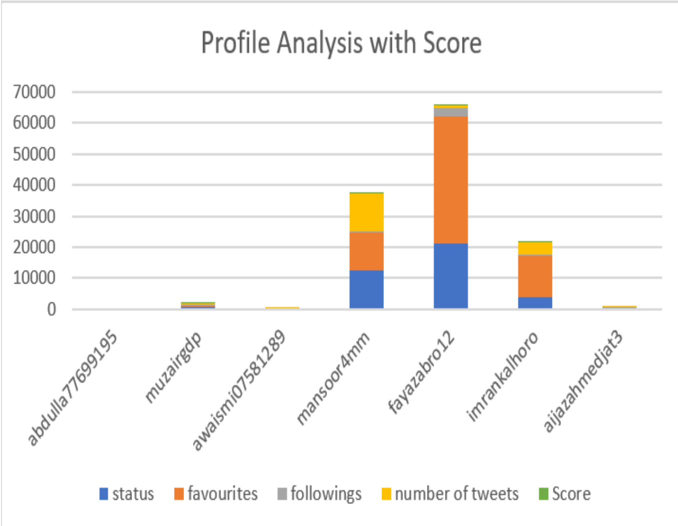


Figure 4.14: Profile Comparison

Implication of Different Metrics

The data has been converted from bool to int so that different metrics can be evaluated involving linear algebra because the mathematical functions have been used in this evaluation. [65] The popularity metric between friends count and followers count have been performed by showing the results in an image.

Popularity metrics is a specialized data system that provides comprehensive social information based on likes and recommendations and that leads to social interest and participation in online news. Twitter metrics fall into a few categories. There are Tweet-level metrics that solely look at the performance of Tweets you post. These include metrics like Top Tweet and Link Clicks. Then there are profile-level metrics that analyze your account as a whole. These include metrics like Profile Clicks and Demographics.

4.7 Learning Algorithms

There are supervised and unsupervised learning methods. The Supervised machine learning algorithms labels with the I/O. The desired output and target output should contain fewer differences. It included SVM and Decision, tree models. [66] There is an absence of labels in the unsupervised machine learning algorithm. We have used the clustering in an unsupervised learning algorithm but the results were not good because K-means clustering has done the groupings of data and the data has been distributed in the groups. The algorithm learns by an error and different trials. The main approach has been done in a supervised learning algorithm and results in the veracity of fake tweets identification.

4.8 Naive Bayes

There are a lot of differences between different machine learning and deep learning models. The Naive Bayes treat the features as independent and SVM attains the interactions between them. SVM contains a non-linear kernel that contains the poly and Gaussian in its training. Naive Bayes is commonly used for different types of classification problems. It contains the text classification in python. The Naive theorem is required for its implementation. [67] The reason for calling it naive because it contains independent prediction with great results containing the estimation of variables involving probability. The calculation of posterior probability with the simple feature. It contains hurdles on a calculation of multi-features and problem of zero probability arises. we have implemented Naive Bayes with its accuracy, precision, and recall. The result of accuracy is 91.6%, precision is 92% and recall is 9

4.9 SVM

Support Vector Machine is a learning algorithm that is used for classification and regression. The data points are plotted in n-dimensional space. Its classification is performed by a hyper-plane that can discriminate between two different classes. The magnitude of the equation is present in SVM which depicts the distance of hyperplane and observation away. When the magnitude of our data points is low, we are not certain about the class assignment but when the magnitude is high, it becomes preferable to observe class assignment.[68] The margin is also present in SVM which calculated the minimum distance. The distance of the hyperplane to the class is called the Margin of SVM. The maximum margin has been calculated in SVM because of attaining high magnitude and it is called the Maximum margin classifier used in the classification with the n-dimensional data points named as features. The accuracy of SVM is 93.8%, the precision is 94.2% and the recall is 97.83%.

4.10 BERT Embeddings

4.10.1 Transformer

The complete detail of transformer step by step is as follows;

- The novel model that has been effectively used for NLP tasks is called a transformer.
- The research paper named "Attention is all you need" concludes the information about the sequence to sequence methodology and transformer architectural flow.[69]
- Sequence to Sequence: A neural network architecture that works on a given words sequence in a sentence. They also worked on transforming the sequence of the words into another language.
- LSTM: Long short-term memory models are those in which the sequence of words can attain meaning while keeping in work.
- It also contained the input sequence in encoder with mapping that into top dimensional space and the decoder of sequence to sequence contains the abstract vector.
- The recurrent neural networks do not imply any Sequence to sequence and concluded that without RNN architecture, the results of transformers have been improved.
- There is pre-training of Deep Bidirectional transformers in which the context have been analyzed from left to right or right to left with deeply contextualized information.

4.10.2 Implementation of BERT

BERT contains deep information about the dataset. The pre-tuning of the model contains the unlabeled data on variant pre-training tasks. The pre-trained parameters are utilized in the fine-tuning. [70] In addition, fine-tuning has been done with the labeled data from the long-term tasks. Each task contained separate models for parameter tuning. The nature of BERT is that it is unified. It is based on source implementation with the tensor2tensor library. The number of layers has been represented with L, the

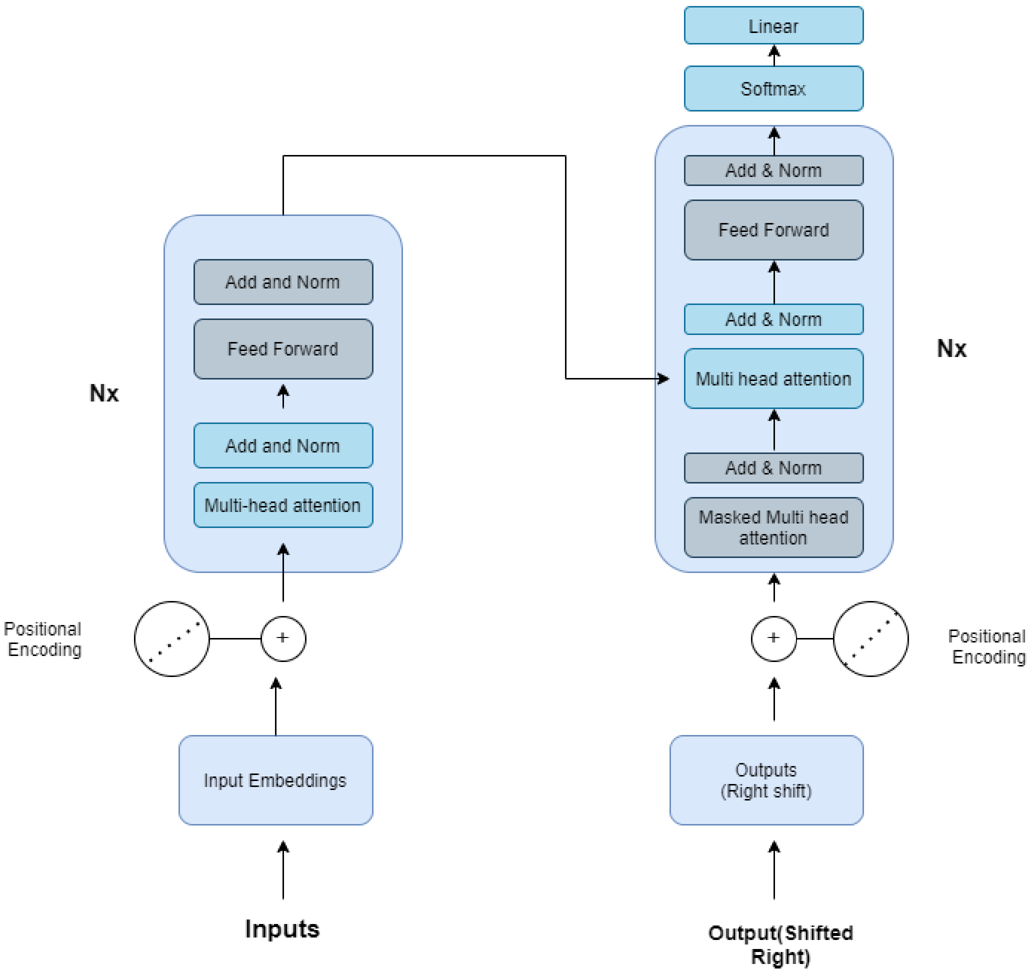


Figure 4.15: Working of Transformer

self-attention heads have been represented with A and the hidden layer has H. The BERT-LARGE is two times of BERT-BASE and all parameters are improved in it.

BERT BASE (L=12, H=768, A=12, Total Parameters=110M)

BERT LARGE (L=24, H=1024,A=16, Total Parameters=340M)

The generative pre-trained transformer takes the self-attention rules and their constraints. It contains the token context on its left side only. On the other hand, the BERT transformers take in the bi-directional transformer with the working of context from left to right and right to left.

4.10.3 Pre-Training

The pre-training has been done on different platforms. It evolves the Books-Corpus containing 800 Million words and Wikipedia containing 2500 Million words. The extraction of long sentences for pre-training is difficult to use in the document-level corpus. [71]We have a pre-trained BERT model with two unsupervised tasks, Masked LM and Next Sentence Prediction.

Task 1: Masked Working

There are different deep bidirectional methods and more improved than the right to left context embeddings and vice versa. There are standard conditioning models too in which there is only right to left and left to the right training. Deep bidirectional models allow predicting the required word in the context of variant layers. The training of a deep bidirectional model can be done by masking out some percentages of initial tokens randomly with generating the target masked tokens.

The hidden vectors communicate to the mask tokens. There is a mathematical function named Soft-max, it works on a vector of probabilities and numbers. Its output is analyzed as the probability of in- integration for each class. [72] The final hidden vectors are inserted in an output layer of softmax. We have masked 20 percent of word-piece tokens randomly in each sequence. The mismatch has been created at this step because the mask token remains unavailable during fine-tuning and it leads to an unimproved working. The actual MASK token and masked words have not replaced in any way. For example, 20 percent of data have been chosen for training, the i-th token can be replaced as follows;

- The Mask token containing 80 percent of the time.
- The 10 percent of static i-th token of the time.
- The 10 percent of time by a randomly generated token.
- The original token can be easily predicted at this step by the cross entropy loss.

Task 2: Next Sentence Prediction (NSP)

The relationship between two sentences plays an important role in concluding the target sentences but it cannot be done by language modeling. The Natural Language Inference and downstream of Question/Answers can be attained by an understanding two sentences. The next tweet prediction has been utilized by pre-training of next sentence prediction as next tweet prediction task. It contains two steps as follows;

- 50 percent of time, B =¿ IsNext =¿ Actual Next Sentence
- 50 percent of time, B =¿ NotNext=¿ Random sentence from corpus

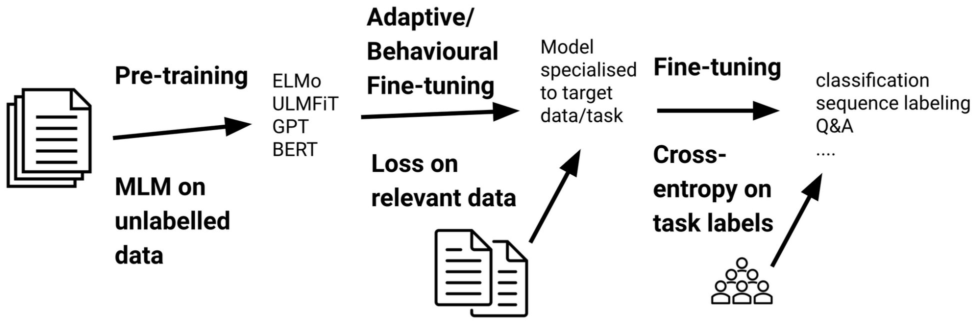


Figure 4.16: Working of BERT-Embeddings

4.10.4 Fine Tuning

Transformers permit BERT embeddings to model the streaming tasks and it have been attained by swapping the inputs and outputs. It contained the pairs of text or a singleton set. Before implications of deep bidirectional cross attention, a common pattern is required for text pairs.

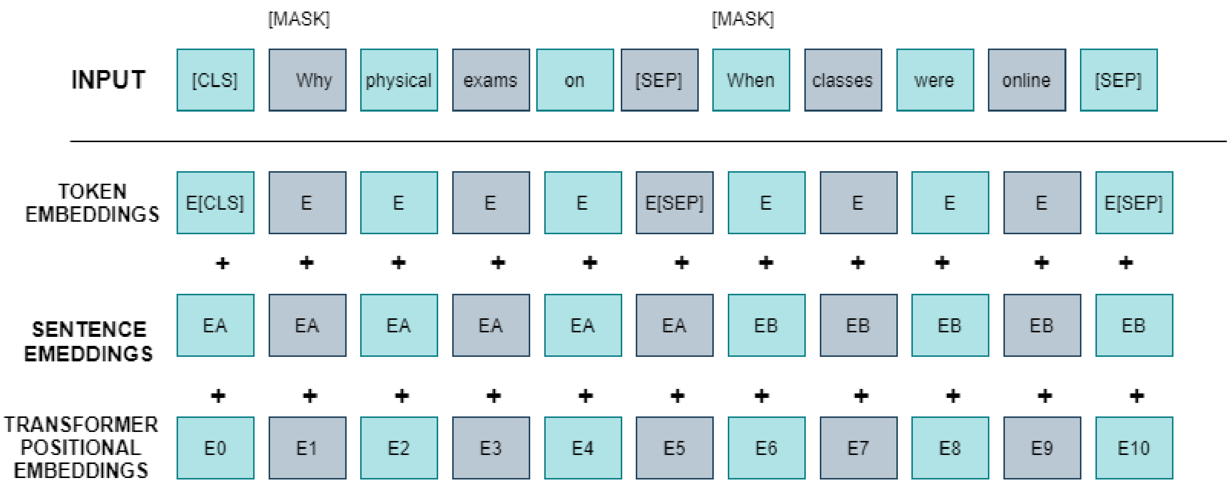


Figure 4.17: Fine Tuning of BERT Model

Two different stages are involved in the fine-tuning of the BERT model. The bidirectional cross attention mechanism have been achieved by taking the context of ”Attention is all you need”. The pairing of self-attention and concatenated text has been analyzed as encoding the strings or text procedures. The specified Input and outputs are plugged into the BERT Embeddings with fine-tuning every parameter from the initial to the final stage. The mapping of these stages contains the following;

- Hypothesis-premise pairs
- Pairs of sentences in its paraphrasing
- The sequence tagging of sentences
- The text classification of degenerating text
- Pairs in Question/ Answers

The token level tasks containing the question answering or tagging of sequence are an important part of fine-tuning in BERT and its representations are sustained into the final output layer. The classification of fine-tuning contains the CLS representation containing the sentiment analysis. The pre-training is more expensive than fine-tuning. These scores have been used for working on BERT Embedding for the classification of tweets.

BERT stands for Bidirectional Encoder Representations from Transformers. It is mainly designed to attain the pre-training deep bidirectional representations from the unlabeled text. It generated the result from both left and right context in such a way that the pre-trained BERT model can be fine-tuned with variant layers. There are different output layers and it generates state of art models with different NLP tasks. The major contribution of the proposed work is identifying the fake tweets using the BERT model which is a bidirectional model that attain the context of words from left to right and vice versa.

The probabilities can be found as follows;

4.10.5 Factors evolve in Fine-tuning

The BERT Embeddings require different parameters. The kaggle import is important in its work because it allows users to indulge and work in competitions of pre- predictive modeling. It has been used to take access to training accelerators. It is good to connect and collaborate to build the best machine learning models. The files have been uploaded on Colaboratary research with different directories of kaggle and json. The zip-file is required from colaboratary for its analysis. There are numerous errors and warnings need to be imported. The main package is the transformers because it plays a major role in BERT Embeddings. The following are the imports used in BERT Embeddings;

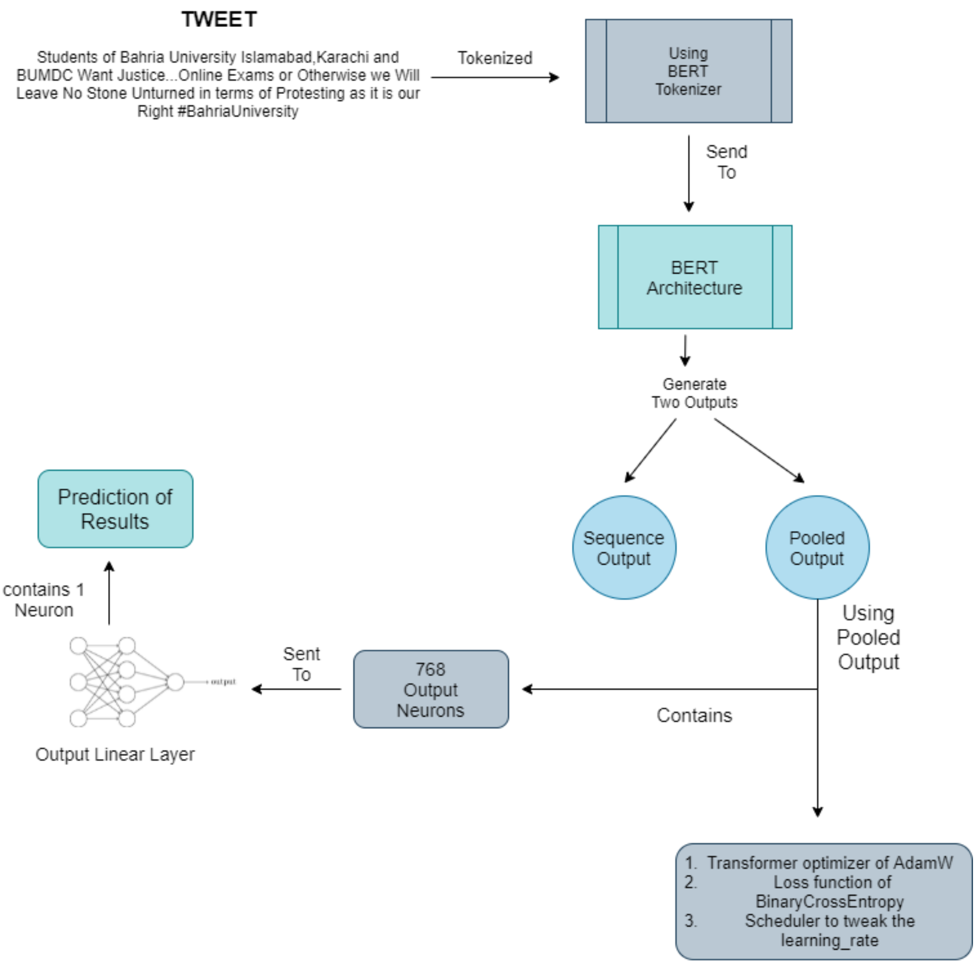


Figure 4.18: Working of BERT

4.10.6 Python Packages

import transformers
from sklearn.model selection import train test split
from torch.utils.data import DataLoader
from transformers import AdamW
from transformers import
get linear schedule with warmup

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}.$$

Figure 4.19: Probabilities for BERT Embeddings

4.10.7 Parameters required in fine-tuning

There are different configurations in its working, some are default and some values have to be granted. The maximum length of 50 have been given.
TRAIN BATCH SIZE = 8
VALID BATCH SIZE = 4
EPOCHS = 1
ACCUMULATION = 2
BERT PATH = 'bert base uncased'
TRAIN NUM WORKERS = 2
VALID NUM WORKERS = 2
LEARNING RATE = 3e-5
WARM UP STEPS = 0

4.10.8 Fine-Tuning Cross Entropy

The model path was required and training have been done with labels processing. The Tokenizer has been used as Bert Tokenizer. It converts plain text into a numerical values sequence and that block can process the text in different languages to the WordPiece method. The class of BERT data-set is required in this step because of initializing some variables. Three different IDs have been proposed; **input, attention, and token types ids**. The Bert-Base un-cases class with a linear output of 768 neurons and 1 hidden layer have been used. The loss function also plays an important role in its working. The neural networks has been trained with the requirement of the loss function to measure the error of the model and it is known as the optimization process. Mean squared error and cross-entropy generates in loss function for training the neural network.

They proposed a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train deep bidirectional representations in an unlabeled text. Such pre-training exists

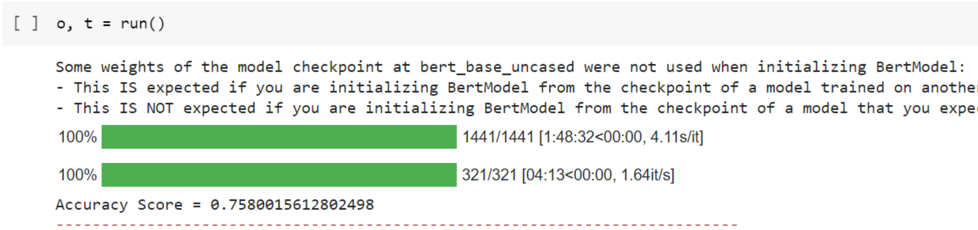


Figure 4.20: Accuracy Result of BERT Model

in both right and left contexts in all layers. Additionally, it has been fine-tuned with the addition of only one output layer. BERT is in the use of transformer which includes two separate mechanisms: reading the text input with encoding and producing a prediction for the task with decoding. We have created state-of-the-art models for a wide range of tasks, such as question answering and language inference, without architecture modifications. We have also proposed their modifications on different natural language processing tasks and obtained the outcome of new state-of-the-art including GLUE score, MultiNLI accuracy, SQuAD v1.1, and SQuAD. We worked to transfer learning with several language models and illustrated that pre-training is an essential part of many language understanding systems. Our output enabled the low-resource tasks to gain from deep unidirectional architectures. We showed our main contribution on generalizing the findings to deep bidirectional architectures and permitted the same pre-trained model to well tackle a broad set of NLP functions. We applied the feature-based approach by removing the activation's from one or more layers in the absence of fine-tuning parameters of BERT. This dependent learned representation for text where words that have the same meaning have a similar representation is used as input to an irregular initialized two-layer dimensional BiLSTM before the classification layer.

4.10.9 Opinion/Weaknesses/Strengths

Weaknesses: There is a need to train separate LTR and RTL models and represent each token as the concatenation of the two models because it was expensive as a single bidirectional model. **Strengths:**Its main strength was the best performing method that chained the token representations from the top four hidden layers of the pre-trained Transformer, which is only behind fine-tuning the entire model and examined that BERT is effective for feature engineering and improved in identification of fake or non-fake tweets.

4.11 Fast-Text and BERT Embeddings

The Fast-Text is a lightweight library that permits users to attain knowledge of text classifiers. The models have been further reduced according to the device specifications. The term abusive, hate and toxic speeches are not distinctive and they are named as toxic words. Therefore, we have proposed the toxic speech of my tweets using different deep learning techniques and word embedding representations. The analysis has been done on multi-class classification as below;

- Firstly, word embeddings using multi-class classification
- Applying classifier of Deep Neural Networks.
- Implementation of fine-tuning the masked LM and next sentence prediction

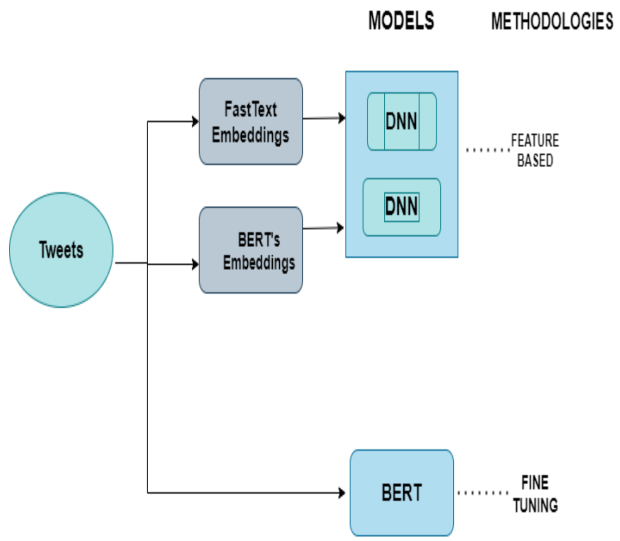


Figure 4.21: Fast-Text and BERT

4.12 Exploratory Data Analysis

The exploratory data analysis has been performed after attaining different metrics and evaluations. To contain the improved understanding of the quality of data, high-quality data is required so that a good model can be generated. It is clear from observations that data can never be clean because there may remain some minor errors in data.

There are possible patterns and supervised learning drivers so that a great amount of cleaned data can be generated. Different methods and techniques have been used to identify whether the tweet is fake or not. Its identification is a great challenge because the data-set of tweets is noisy and difficult to interpret.

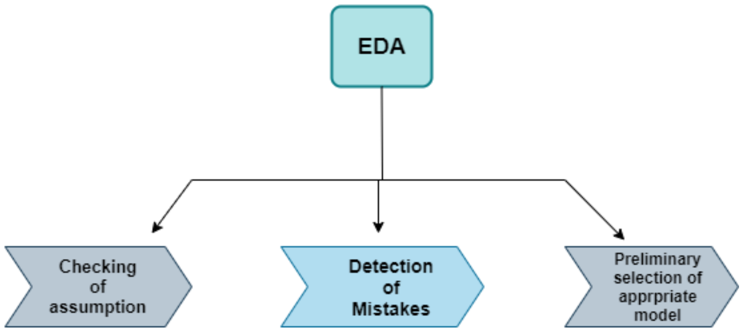


Figure 4.22: Exploratory Data Analysis

4.13 BERT-HAN Architecture

The Attention layer has been applied on the sentence level and word level of BERT-HAN architecture that contains different layers of user posts and user behaviour which helped in identifying **fake tweets**. The uniform distribution is also present with an init initialization. There is a support of true mask in attention networks because fixed inputs are required in its network.[73] Furthermore, the maximum input length is equal to the input length of masks. The mask contains a "1" padding around its input. The mask and input sequence works together with a network. The requirement of the mask is necessary because it relays the maximum length of input or padded input?

The length of the input shape is 3 because of the 3d tensor flow from existing layers. The input of context annotations is produced with the sum of important weights. The masking required a tensor in binary form so there is an expansion of the current variable at this point. The state of art algorithm contained the description of existing knowledge through the results extracted from related published work. [74] The good results of state of art always give an accurate performance. The prediction of failures played an important role in machine learning. The sensory data is obtained from different engineering systems. The Attention layer has been applied on sentence level and word level. The uniform distribution is also present with an init initialization. There is a support of true mask in attention networks because fixed inputs are required in its network.[73]

4.13.1 Hierarchical Attention Network

Initially, the document representation has been represented by the hierarchical structure of different sentences and then combining them in a representation of document for great detail at the start of work. The input has been transferred from the sentence of the document. It contains the sentence vector. The HAN architecture has been worked on word-level initially then to the sentence level. The relevant contexts have been returned by the encoder. The weights of vectors have been attained by an attention mechanism.

4.13.2 BERT-HAN Architecture

The structured tokens are required in word level with *WIT* that contains the word. The encoding mechanism containing the Gated Recurrent network(GRU) contains the vectorized tokens. [75] Recurrent neural networks contain the prediction of words on previous words in it. The recurrent neural networks have been in GRU and it's a forget gate with LSTM but the output gate is not present in it. It contains a final output of HAN in which biases and training weights are initialized randomly and learning embeddings have been used. Hierarchical-Attention-Network has been used with BERT model containing Bi-Directional LSTM containing PyTorch and document classification has been done in this way. The sentence embeddings have been done on traditional Bi- LSTM with the implementation of the BERT model. The 95% accuracy has been achieved with the implementation of BERT-HAN.

4.14 Validation

Quantitative research is being used with the reliability of results and validation. The validity of tweets is required in qualitative research too because a human can judge more accurately than machines. To contain the results of validation, some tweets have been considered and a google form of survey questions have been made to identify whether the tweet is fake or not. There are two options fake or non-fake. The following results showed that there is 97% accuracy in the predictive and questionnaire's result. There are a total of 15 tweets and label 1 showed about the non-fake tweets and label 0 showed about the fake tweets. The average of their results has been calculated in Microsoft Excel 2010 by using the formulas, the standard deviation of results of each tweet has also been calculated. Therefore, these results showed that validation has given good and improved results.

	A	B	C	E	F	G	H	I	J	K	L	M	N	O	P
1	Name	Tweet 1 Bahria U	Tweet 2 @S	Tweet 4 It	Tweet 5 Sc	Tweet 6 If	Tweet 7 U	Tweet 8 Tı	Tweet 9 @	Tweet 10 I	Tweet 11 '	Tweet 12 I	Tweet 13 '	Tweet 14 !	Tweet 15 P
2	Maryam	0	1	0	1	0	0	0	0	0	0	1	0	0	0
3	Farhan Sheikh	0	1	0	1	0	0	0	0	0	0	1	0	0	0
4	AYESHA MALIK	0	1	0	1	0	0	0	0	0	0	1	0	0	0
5	Shafaq	0	1	0	1	0	0	0	0	0	0	0	0	0	0
6	Punjabian	0	1	0	1	0	0	0	0	0	0	1	0	0	0
7	Noor	0	1	0	1	0	0	0	0	0	0	1	0	0	0
8	Wasifsheikh	0	1	0	1	0	0	0	0	0	0	1	0	0	0
9	Sumera	0	0	1	0	0	1	1	0	0	0	1	1	1	0
10	faiza naveed	0	1	0	1	0	0	0	0	0	0	1	0	0	0
11	Bisma	0	1	0	1	0	0	0	0	0	1	1	0	0	0
12	Ansa binte Imran	0	1	1	1	0	0	0	0	0	0	1	0	0	0
13	Sara ai	0	1	1	1	0	0	0	0	0	0	1	0	0	0
14	Standard Deviation	0	0.2886751	0.452267	0.288675	0	0.288675	0.288675	0	0	0.916667	0.288675	0.288675	0.288675	0
15	Survey Results	0	0.9166667	0.25	0.916667	0.099129	0.083333	0.083333	0	0	0.083333	0.916667	0.083333	0.083333	0
16	Predicted Results from Model	0	1	0	1	0	0	0	0	0	0	1	0	0	0

Figure 4.23: Predicted Results and Survey Results

4.15 Tools

Pip

Pip is a package that is used to manage and install the different packages written in Python. It contains variant dependencies in the project. They are listed in one file named requirements.txt. It is a built-in command which has been inserted in the command line so that all packages can be installed using that requirements.txt.

Anaconda Spyder

We have used Spyder IDE for python code and it includes the debugging, testing, and different features involved with the variable explorer and console in which commands can be run individually or in groups as a whole.

4.16 Summary

To conclude, there are different methodologies required in tweets identification. The data has been collected and scrapped through two different libraries Twint and Tweepy. The data has been scraped using hashtags and strings. The named entity recognition has been used with entity linking. The scrapped data was containing the symbols used in the tweets. We have encoded and decoded them in Python. Then the extracted data-set has been pre-processed with stemming, lemmatization, removal of non-stop words, and tweet cleaning. The feature engineering has been done with the existing features and the data-set has been analyzed by Sentiment analysis. It has been done using VADER that generates both positive and negative with the strength of emotion. It has generated different sentiment types with polarity and subjectivity. The language modeling with the bag of words has been performed in extracting the context word and target word. The TF-IDF has been performed with the word embeddings. The modeling phase contained the scikit learn and sklearn. The learning algorithm and implementation of BERT embeddings have been performed with pre-training and fine-tuning. The fine-tuning cross-entropy has also been used to measure the error of the model. The 75% accuracy has been attained and fast-text and BERT Embeddings have also been attained in PYTHON and compared the results and in BERT-HAN, the Recurrent neural networks contain the prediction of words on prevarious words in it. The documentation is going through backward and forward phases in which the Bi-GRU is used for the context of sentences. The attention layer is fed into the fully-connected layer and it's then emerged into a concatenation layer with the combination and results of all the previous layers. The Sigmoid activation function analyzes whether values should be passed or not.

Chapter 5

BERT-HAN Architecture

5.1 Hierarchical Attention Network

5.1.1 Attention Layer

The Attention layer has been applied on a sentence level and word level. The uniform distribution is also present with an init initialization. There is a support of true mask in attention networks because fixed inputs are required in its network.[73]

Furthermore, the maximum input length is equal to the input length of masks. The mask contains a "1" padding around its input. The mask and input sequence works together with a network The requirement of a mask is necessary because it relays the maximum length of input or padded input? The length of the input shape is 3 because of the 3d tensor flow from existing layers. The input of context annotations is produced with the sum of important weights. The masking requires a tensor in binary form so there is an expansion of the current variable at this point.

5.1.2 State of Art Algorithm

The state of art algorithm contains the description of existing knowledge through the results extracted from related published work. [74] The good results of state of art always give an accurate performance. The prediction of failures played an important role in machine learning. The sensory data has been obtained from different engineering systems. It was based on traditional systems in which feature engineering has been proposed by a specific algorithm. The Convolutional neural networks and bi-directional LSTms with great results. The different datasets have been used for the training dataset. The analysis shows that convolutional neural networks perform better than deep learning approaches.

The Attention layer has been applied on sentence level and word level. The uniform distribution is also present with an init initialization. There is a support of true mask in attention networks because fixed inputs are required in its network.[73]

5.1.3 Hierarchical Attention Networks

The hierarchical attention network has been proposed for the classification of the document. There are two different characteristics involved in it.

- The hierarchical structure is present for structuring the documents.
- The word and sentence levels have been proposed on two-level attention mechanisms. It attains less or more important information in the representation of a document.

Our experiments are conducted on two different large-size data sets containing tweets of political parties of Pakistan and different universities of Pakistan to identify whether the tweet is fake or real. [75] This architecture also contains the margin with the attention layers and its visualization. It also collects the qualitative sentences and words information. The HAN architecture contains the following parts;

- A word sequence encoder
- A word-level attention layer
- A sentence encoder
- A sentence-level attention layer

5.1.4 Architecture of HAN

Initially, the document representation has been represented by the hierarchical structure of different sentences and then combining them in a representation of document for great detail at the start of work. The encoding was required in the sentence representation in which the word of a sentence can be encoded.

In addition to that, the attention mechanism has been applied in a vector form containing sentences. The input is being transferred from the sentence of the document. It contains the sentence vector. The HAN architecture has been worked on word-level initially then to the sentence level. The relevant contexts have been returned by an encoder. The weights of vectors have been attained by an attention mechanism.

5.1.5 Text Classification with HAN

The text classification of HAN includes an attention mechanism that contains the sequence in sentences and different words of those sentences in a document. The resulting weights have been attained from existing words in a dataset. There is a key point that document containing every sentence and each word in a specified sentence are not equally important to consider the main message of a document. The context should be cleared and text classification with HAN takes it an important factor.

For Example

The sentence containing the meaning of the word "pretty" can be used in a positive and negative context. Its meaning can be altered in a sentence.

- She is so pretty.
- That hotel provides pretty bad food.

5.1.6 Word Level

The structured tokens are required in word level with *WIT* that contains the word. The data pre-processing is required in this section. There is a string type data containing plain text and an embedding layer are also used here. It contains multidimensional vectors in each token. The word representation has been done by *xit* in a vector space. The Glove and Word2vec embedding algorithms have been used with pre-training of word embeddings in which the model can be accelerated.

5.1.7 Word Encoder

The encoding mechanism containing the Gated Recurrent network(GRU) contains the vectorized tokens. [75] Recurrent neural networks contain the prediction of words on previous words in it. The hidden state is present in Gated Recurrent Network and information can be easily transferred. The information has been kept, forget or extract from relevant context. There are per- word annotation sin word encoder of HAN. The annotations of words can be attained with Bi-GRU and it results in getting the context from left to right and vice versa.

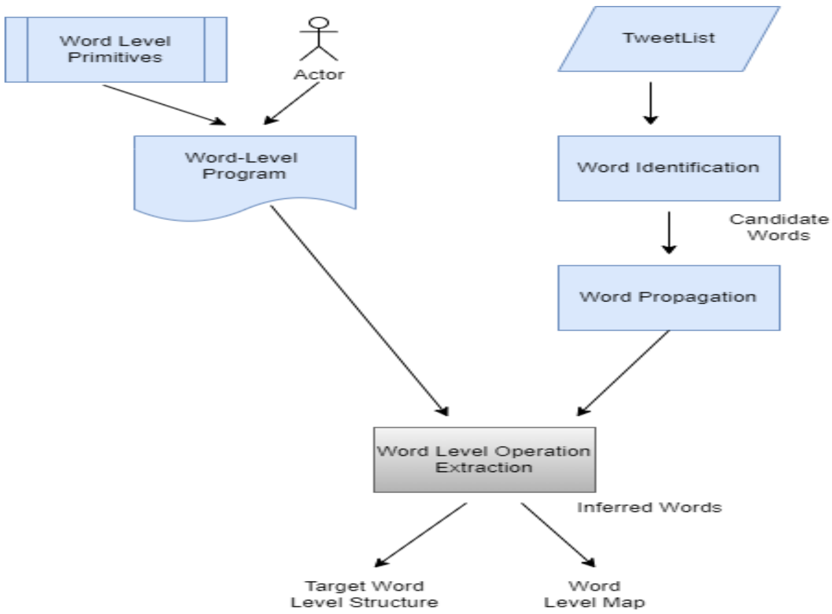


Figure 5.1: Word Level of State of art Algorithms

GRU-based sequence encoder

The recurrent neural networks have been in GRU and it's a forget gate with LSTM but the output gate is not present in it. It contains modeling of speech signals and different models for music. It gives good performance on the small amount of data and sequences are being tracked here with an absence of separate memory cells. There are the following types of gates in it;

- The reset gate rt.
- An update gate zt.

5.1.8 Word Attention

There is an annotation *hit* that is contained in an attention layer which begins with uni-layer perceptron containing Multilayer. The training of word attention involves biases and variant weights. It shows that there is no delay of the network with a function of *tanh*. The word attention mechanism involves the mapping of zero to near-zero and marks correct the initial given values between -1 and 1. There is a context vector used for training *uw* which gives the product of new annotations and itself. The sentence vector is the concatenation of previous context annotations with the sum of weights by *si*.

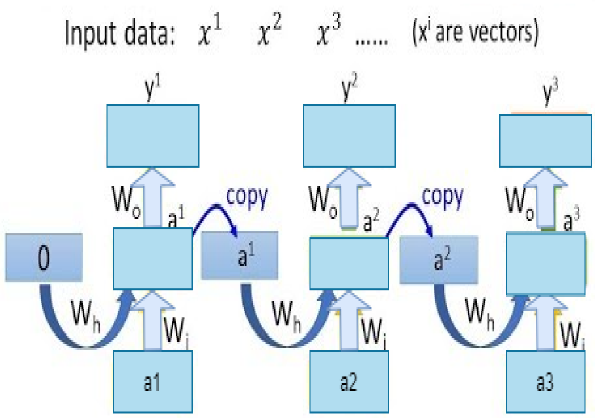


Figure 5.2: Word Level of Gated Recurrent Neural Networks

5.1.9 Sentence Level

The network follows the same method as on the word level and an embedding layer is absent. The input from the word level gives the sentence vectors then these vectors are used in the sentence level.

Sentence Encoder

The documentation have used backward and forward phases in which the Bi- GRU, used for the context of sentences.[76]

Sentence Attention

It contains a final output of HAN in which biases and training weights are initialized randomly and learning embeddings have been used.

Model of HAN

- The layers gets combined on different levels of sentence and word.
- Each sentence contained a time distribution on all word level layers.
- The dimensionality contains 100 dimensions with backward and forward running.

- The over-fitting can be prevented by terminating the number of different layers containing neuron in which high variance has been achieved at 0.5

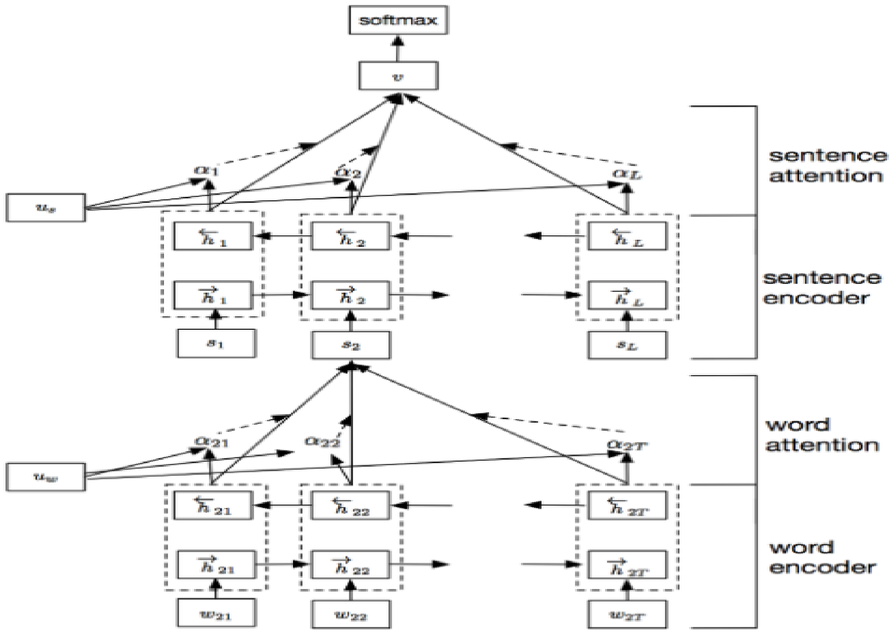


Figure 5.3: HAN Architecture

The self-attention mechanism has been used with the results of state of the art, and HANS. There have been used two classification tasks of Political parties of Pakistan and different universities of Pakistan. HCANS have also been used and results demonstrated that HCANs perform better than HAN. It also gives the fastest training. The RNN based networks have been replaced by reducing the training and there would be no effect on accuracy. There is an efficient implementation of data using the word encoder, word level, and sentence level.

5.2 BERT Hierarchical Attention Network

Hierarchical-Attention-Network has been used with BERT model containing Bi-Directional LSTM containing PyTorch and document classification has been done in this way. The sentence embeddings have been done on traditional Bi-LSTM with the implementation of the bERT model. The sentence embeddings feds into BiLSTM with an attention

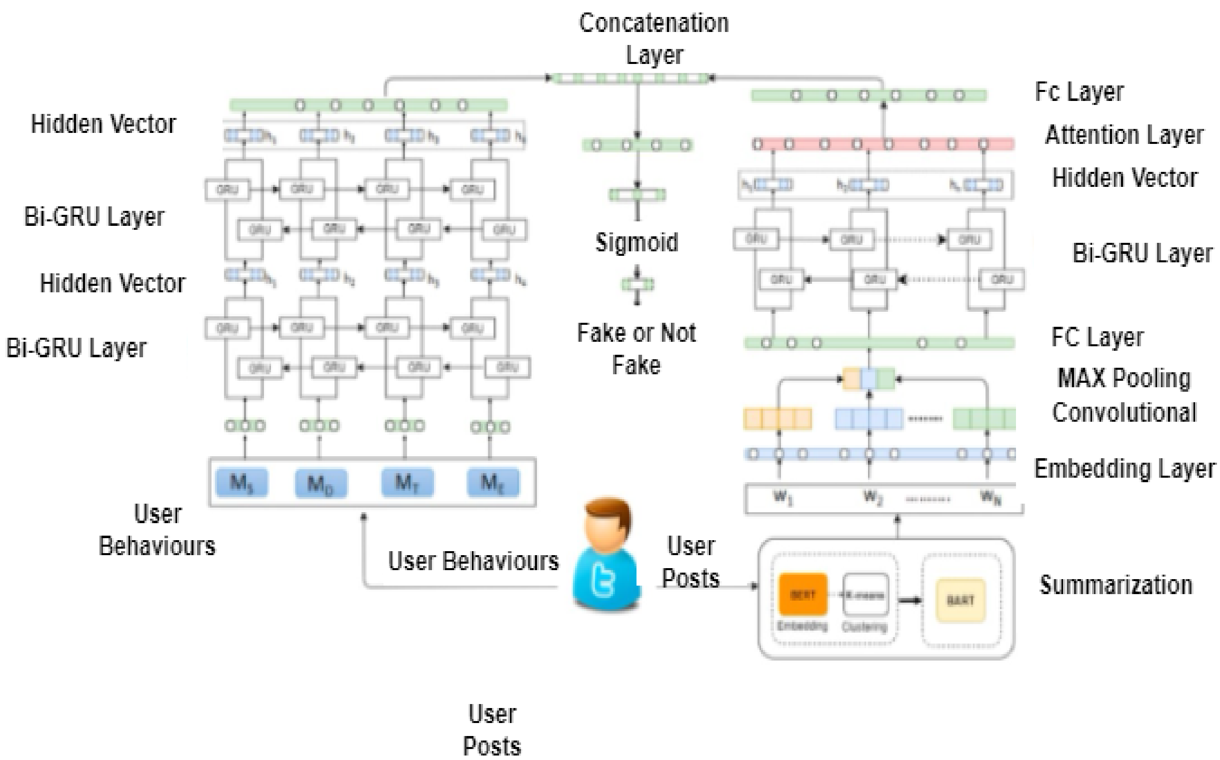


Figure 5.4: BERT-HAN Architecture containing different Layers

mechanism to contain a multilayer perceptron with a softmax activation function and it is in the output layer of the neural network.

There is an implementation of different layers with the following factors involved in it.

5.2.1 User Behaviour of Cleaned and Labeled Tweets

5.2.2 BiGRU Layer

There are two gated recurrent units involved in it. The first is taking the input in the backward direction and the second is taking input in a forward direction. It contains only one input and attains other forget output. Initially, Bi-GRU layers are present in parallel and indulge in hidden vectors.

5.2.3 Hidden Vector

The hidden vector contains the properties of security involved in it. There are no full details about Hidden vector encryption but great details about definitions related to predictions. The binary alphabets are involved in it with the () function and constant greater than zero. The hidden vectors are again fed into Bi-GRU layers.

5.2.4 Concatenation Layer

Then at the last phase of user behaviors involved, the hidden vectors are fed into a concatenation layer in which the inputs are being taken and combined the inputs in a given dimension. There are two types of layers connect and disconnect layers. The input names are being used in these types.

5.2.5 User Posts

5.2.6 BERT Embeddings with User Posts

The embeddings are representations of points in high dimensions. The word embeddings are dense vector representations of invariant words. The user posts of Twitter are fed into the combination of BERT embeddings and k-means clustering. It would generate a BERT with Hierarchical attention networks involved in it. The neural networks are easily utilized by the word embedding model. The mathematical operations are involved with the semantic and syntactic importance of words.

5.2.7 K-Means Clustering

The clustering divides the dataset into subgroups and each category of data contains one group. These are non-overlapping groups. It also examines the inter and intra-clusters. There is a collection of data points and arithmetic mean of groups. The cluster centroid contains the arithmetic mean and a sum of squared distances of those data points are also present.

"More similar the data points, less variation within the clusters"

The number of clusters needs to be initialized in the beginning. The cluster centroids containing the arithmetic mean of data points are initialized with the shuffling of a dataset and no replacement is involved in it. Then assign the data points to the arithmetic mean of a specified group. The BERT Embedding and k-means clustering has given the summarised form of BERT and then it is fed into the embedding layer.

The clustering divides the dataset into subgroups and each category of data contains one group. These are non-overlapping groups. It also examine the inter and intra clusters. There is a collection of data points and arithmetic mean of groups. The cluster centroid contains the arithmetic mean and sum of squared distances of those data points are also present.

5.2.8 Embedding Layer

It is the layer of the network in which the hidden layer is present. The input sequences are also present in embedding layer with the implementation of Keras model. For example if a document containing 500 words, then the embedding layer would contain 500 words. To conclude, the embedding layer is then fed into Convolutional neural networks.

5.2.9 Convolutional Neural Networks

It contained the bases and weights of tweets contained from an embedding layer and pre-processing have also been done in this step. It also classifies the labeled classes with the images involved in it related to my dataset. [77] Therefore the output of CNN represented the classes and labeling.

5.2.10 Max Pooling

The max-pooling takes the size of the kernel to attain a maximum overview with the stride of the window which contains the kernel size also. The parameters are also involved in it and contain the window in which the elements are involved with the zero padding. The CNN layers are combined and generate the max-pooling which is then fed into fully connected layers. It contains the following library;

"torch.nn.MaxUnpool2d "

5.2.11 Fully Connected Layer

There are lower layers in which they remain at the beginning of hierarchy and input reduces in size. The inputs result in a different features representation. Each starting inputs are connected to all neurons present in it. It can easily be seen after Convolutional neural networks and objectives are defined at this step. This fully connected layer is then fed into the Bi-GRU layer. THe Bi-GRU layer is then fed into a hidden vector. Furthermore, the hidden vectors are fed into an attention layer.

5.2.12 Attention Layer

The softmax function contains the probabilities of taking the input and vector z of real numbers. The attention layer gives the output using the softmax function. The attention the mechanism permits the translators to get hold of core sentences and generate the context from exact wording.

5.2.13 Sigmoid Function

The attention layer is fed into the fully connected layer and then emerged into a concatenation layer with the combination and results of all the previous layers. The sigmoid activation function analyzes whether values should be passed or not. The concrete-nation layer then combines the results of a layer from user behavior and posts into the sigmoid function.

5.3 Summary

To summarize, the Attention layer is being applied on a sentence level and word level. The uniform distribution is also present with an init initialization. The text classification of HAN includes an attention mechanism that contained the sequence in sentences and different words of those sentences in a document. The structured tokens are required in word level with *WIT* that contains the word. The data pre-processing has been required and the encoding mechanism containing the Gated Recurrent network(GRU) with the vectorized tokens. Recurrent neural networks contain the prediction of words on previous words in it. The documentation is going through backward and forward phases in which the Bi-GRU is used for the context of sentences. The attention layer is fed into the fully connected layer and then emerged into a concatenation layer with the combination and results of all the previous layers. The sigmoid activation function analyzes whether values should be passed or not.

Chapter 6

Experiments and Evaluation

The evaluations and different results have been shown in great detail. There are step-by-step results according to the work being done on our thesis. Our primary goal of this proposed work is to identify the tweets predicting the fake identity using Deep Learning Techniques. The most important thing is to evaluate the results and their performance. The results are as follows;

6.1 Results

6.1.1 Component Performance

There are some rule-based systems for identifying fake tweets in which the different deep learning classifiers have been used for veracity checking of fake tweets and fake account detection. There are some measures for veracity checking and giving the best results. The F1 score, Precision, and Recall are evaluated. We have used a python library named Twint for scrapping the data from Twitter that included the created at, date, tweets, retweets, source, and URLs but the number of followers, number of followings, number of statuses count and verified ware not accessible from a library named TWINT so Python library named TWEEPY have been proposed by using names of users and API credentials of twitter given by developing a Twitter developer account. The hybrid approach has been proposed by using these two libraries so that good information can be generated for the identification of fake or real tweets. The tweets have been scraped by using hashtags or searching the name of a string that can be processed to attain the information.

The data has been scraped by using the library of python named twint. In addition, labeling is required and there are unsupervised and supervised labeling that is being

	A	B	C	D	E	F
1	user_name	created_at	status	favourites	followings	verified
2	abdulla77699195	1/24/2021 11:34	3	6	0	FALSE
3	muzairgdp	10/17/2020 10:52	644	655	91	FALSE
4	awaismi07581289	6/26/2019 18:57	64	156	31	FALSE
5	ceeiba	10/7/2011 10:48	2669	100	1924	FALSE
6	legend_samii	5/31/2020 2:37	26841	43146	3387	FALSE
7	laadli_baroch	1/15/2021 21:35	17646	20656	1165	FALSE
8	legend_samii	5/31/2020 2:37	26841	43146	3387	FALSE
9	legend_samii	5/31/2020 2:37	26841	43146	3387	FALSE
10	laadli_baroch	1/15/2021 21:35	17646	20656	1165	FALSE
11	legend_samii	5/31/2020 2:37	26841	43146	3387	FALSE
12	laadli_baroch	1/15/2021 21:35	17646	20656	1165	FALSE
13	abdulla78421485	2/10/2019 16:01	8034	23424	1498	FALSE
14	legend_samii	5/31/2020 2:37	26841	43146	3387	FALSE
15	laadli_baroch	1/15/2021 21:35	17646	20656	1165	FALSE
16	mansoor4mm	4/10/2014 8:13	12706	12172	406	FALSE
17	zameerlangah	8/17/2014 5:05	797	790	61	FALSE
18	fayazabro12	10/6/2013 13:59	21285	40921	2388	FALSE
19	imrankalhoro	10/16/2010 9:19	4048	13256	311	FALSE

Figure 6.1: Scrapping through TWEEPY library

used on different data-sets in deep learning. Due to the non-availability of labels in unsupervised learning, clustering will be done. It can only show the grouping with the results of several groups.

The F1 score is the mean value of precision and recall. The general metrics include the sentiment expressed with the number of followings, followers, and status count. It contains the metrics against every user except those whose accounts are not found. The overall accuracy of sentiment analysis has also been calculated 95%

The word cloud of tweets contained the frequency of all words present in the data-set and have shown in an image.

The frequency of the data-set have been calculated by tokenizing the string into separate words. The module of collections. the counter has been used to count elements and it results in the corpus of word counts. The text converted into lower case, HTML tags have been removed with the punctuation's removal. The split function has been used for attaining the individual tokens. The frequency have shown much better in an image to show the most frequent words.

Working on complete sentences was required so that is why we have used sentiment

	A	B	C	D	E	F	G	H	I	J	K
1	id	username	name	tweet	language	Tweet_punct	Tweet_tokenized	Tweet_nonstop	Tweet_stemm	Tweet_lemmati	Tweet_cleaned
2	1.37E+18	slmarketingpk	Starlight Marketin	Dr Ayub Alvi Rector Fast	en	Dr Ayub Alvi Rector Fast Univer	['dr', 'ayub', 'alvi', 'rector',	['[', "'", 'r', "'", ',	['D', 'r', ' ', 'A',	['D', 'r', ' ', 'A', 'y	['dr', 'ayub', 'alv
3	1.37E+18	csa_dvillamizar	Daniel Villamizar	#FASTUniversity #Micro	en	FASTUniversity Microsoft Imagi	['fastuniversity', 'microsoft	['[', "'", 'f', 'u', 'r	['#', 'F', 'A', 'S',	['#', 'F', 'A', 'S', '	['fastunivers', 'r
4	1.37E+18	phoneworldpk	PhoneWorld	#FASTUniversity #Micro	und	FASTUniversity Microsoft Imagi	['fastuniversity', 'microsoft	['[', "'", 'f', 'u', 'r	['#', 'F', 'A', 'S',	['#', 'F', 'A', 'S', '	['fastunivers', 'r
5	1.36E+18	causewespeak	aisha	@Tooba_Sd #FASTUniv	und	ToobaSd FASTUniversity FASTi	['toobasd', 'fastuniversity',	['[', "'", 'b', "'", '	['@', 'T', 'o', 'o	['@', 'T', 'o', 'o',	['toobasd', 'fas
6	1.36E+18	mhassan8383	Maarij Hassan	Fastians need online exa	en	Fastians need online exams Sha	['fastians', 'need', 'online',	['[', "'", 'f', 'n', "'	['F', 'a', 's', 't', '	['F', 'a', 's', 't', 'i'	['fastian', 'need
7	1.36E+18	causewespeak	aisha	@YasserHashmi #FASTU	und	YasserHashmi FASTUniversity F	['yasserhashmi', 'fastunive	['[', "'", 'e', 'r', 'h	['@', 'Y', 'a', 's'	['@', 'Y', 'a', 's',	['yasserhashmi'
8	1.36E+18	causewespeak	aisha	@ShahzadYunasPTI #FA	und	ShahzadYunasPTI FASTUniversi	['shahzadyunaspti', 'fastur	['[', "'", 'h', 'h', '	['@', 'S', 'h', 'a	['@', 'S', 'h', 'a',	['shahzadyunas
9	1.36E+18	causewespeak	aisha	@DrNosharwan #FASTU	und	DrNosharwan FASTUniversity F	['drnosharwan', 'fastunive	['[', "'", 'r', 'n', 'h	['@', 'D', 'r', 'N	['@', 'D', 'r', 'N',	['drnosharwan'
10	1.36E+18	causewespeak	aisha	@SAfridiOfficial #FASTU	und	SAfridiOfficial FASTUniversity F.	['safridiofficial', 'fastuniver	['[', "'", 'f', 'r', 'f'	['@', 'S', 'A', 'f'	['@', 'S', 'A', 'f',	['safridioffici', 'f
11	1.36E+18	causewespeak	aisha	@AmmarRashidT #FAS	und	AmmarRashidT FASTUniversity	['ammarrashidt', 'fastunive	['[', "'", 'r', 'r', 'h	['@', 'A', 'm', 'i	['@', 'A', 'm', 'm	['ammarrashidt
12	1.36E+18	lcareereduc	International Care	#Shu #sheffieldhallamu	en	Shu sheffieldhallamuniversity sh	['shu', 'sheffieldhallamuni	['[', "'", 'h', 'u', '	['#', 'S', 'h', 'u',	['#', 'S', 'h', 'u', '	['shu', 'sheffield
13	1.35E+18	minzamam_buz	Muhammad Inza	My #university is not go	en	My university is not going Onli	['my', 'university', 'is', 'not'	['[', "'", "'", ',', '	['M', 'y', ' ', '#',	['M', 'y', ' ', '#', '	['univers', 'go',
14	1.35E+18	muhammaduza	Muhammad Uzai	@Shafqat_Mahmood N	en	ShafqatMahmood NO Its not tr	['shafqatmahmood', 'no',	['[', "'", 'h', 'f', 'c	['@', 'S', 'h', 'a	['@', 'S', 'h', 'a',	['shafqatmahm
15	1.35E+18	tucoachrolle	Charlton Rolle	EXCELLENT day of traini	en	EXCELLENT day of training for t	['excellent', 'day', 'of', 'trai	['[', "'", 'e', 'x', 'c	['E', 'X', 'C', 'E',	['E', 'X', 'C', 'E', '	['excel', 'day', 't
16	1.35E+18	alikhan5817688	AliKhan	If they can teach us tech	en	If they can teach us technical cc	['if', 'they', 'can', 'teach', 'u	['[', "'", 'f', "'", ',	['I', 'f', ' ', 't', 'h	['I', 'f', ' ', 't', 'h',	['teach', 'us', 'te
17	1.35E+18	bilalmirza44	Bilal Mirza	#fastuniversity alas! U w	en	fastuniversity alas U won amp v	['fastuniversity', 'alas', 'u',	['[', "'", 'f', 'u', 'r	['#', 'f', 'a', 's',	['#', 'f', 'a', 's', 't	['fastunivers', 'a
18	1.35E+18	arbababdulbasi	Arbab Abdul Basi	They must change their	en	They must change their decisio	['they', 'must', 'change', 'tl	['[', "'", 'h', 'e', "'	['T', 'h', 'e', 'y',	['T', 'h', 'e', 'y', '	['must', 'chang
19	1.35E+18	frazarshad1	FraZ Arshad	Majid AI ka syllabus to b	sv	Majid AI ka syllabus to bhej brc	['majid', 'ai', 'ka', 'syllabus	['[', "'", 'j', "'", ',	['M', 'a', 'j', 'i',	['M', 'a', 'j', 'i', 'c	['majid', 'ai', 'ka
20	1.35E+18	hassaanhameec	hassy_here	#FastAgainstGOVT FAST	en	FastAgainstGOVT FAST Is Going	['fastagainstgovt', 'fast', 'is	['[', "'", 'f', 'g', 'r	['#', 'F', 'a', 's',	['#', 'F', 'a', 's', 't	['fastagainstgo
21	1.35E+18	majid1471	majid mehmood	#FastStopTorturing #Fa	en	FastStopTorturing FastAgainst(	['faststoptorturing', 'fastag	['[', "'", 'f', 'p', 'r	['#', 'F', 'a', 's',	['#', 'F', 'a', 's', 't	['faststoptortur
22	1.35E+18	fastian20	fastian	o lul leni dioo baaz a g	in	o lul leni dioo baaz a gao te pa	['o', 'lul', 'leni', 'dioo', 'baa	['[', "'", "'", ',', '	['o', ' ', 'l', 'u', '	['o', ' ', 'l', 'u', 'l'	['lul', 'leni', 'dio
23	1.35E+18	bilalmirza44	Bilal Mirza	#fastuniversity if the clas	en	fastuniversity if the classes were	['fastuniversity', 'if', 'the', 'c	['[', "'", 'f', 'u', 'r	['#', 'f', 'a', 's',	['#', 'f', 'a', 's', 't	['fastunivers', 'c
24	1.35E+18	frazarshad1	FraZ Arshad	JEENA HOGA MARNA H	hi	JEENA HOGA MARNA HOGA D	['jeena', 'hoga', 'marna', 'h	['[', "'", 'j', 'e', 'e	['J', 'E', 'E', 'N',	['J', 'E', 'E', 'N', ',	['jeena', 'hoga',
25	1.35E+18	unknowm_worl	Shaheer Ilyas	Educational institutes ar	en	Educational institutes are supp	['educational', 'institutes',	['[', "'", 'e', 'u', '	['E', 'd', 'u', 'c',	['E', 'd', 'u', 'c', '	['educ', 'institut
26	1.35E+18	inzimamzafar	inzimam	. #FastAgainstGOVT #s	und	FastAgainstGOVT stoptorturin	['', 'fastagainstgovt', 'stop	['[', "'", "'", ',', '	['.', ' ', '#', 'F', '	['.', ' ', '#', 'F', 'a'	['', 'fastagainstg
27	1.35E+18	unknowm_worl	Shaheer Ilyas	#FastAgainstGOVT #Fa	und	FastAgainstGOVT FastStopTort	['fastagainstgovt', 'faststo	['[', "'", 'f', 'g', 'r	['#', 'F', 'a', 's',	['#', 'F', 'a', 's', 't	['fastagainstgo

19i-1218,Sara_Irfan,Pre-Process

Figure 6.2: Pre-Processing of Tweets

$$
\begin{aligned}
precision &= \frac{TP}{TP + FP} \\
recall &= \frac{TP}{TP + FN} \\
F1 &= \frac{2 \times precision \times recall}{precision + recall}
\end{aligned}
$$

Figure 6.3: Formulas used for Component Performance

analysis using VADER and full information has been gathered. There are numerous ways to gather the context of data. For example, two-word combinations can also be considered but it would not easily provide complete information that what this text is about? For this model, we can use a variety of data sets like scraped data of different universities to get what students spread about universities.

There are a lot of differences between different machine learning and deep learning models. The Naive Bayes treat the features as independent and SVM attains the interactions between them. SVM contains a non-linear kernel that contains the poly and Gaussian in its training. The accuracy has been calculated for our data set.

BERT contains deep information about the dataset. The pre-tuning of the model contains the unlabeled data on variant pre-training tasks. The pre-trained parameters are utilized in the fine-tuning. [70] In addition, fine-tuning has been done with

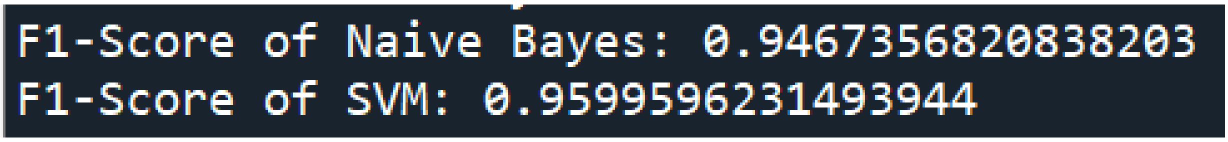


Figure 6.4: F1-Score

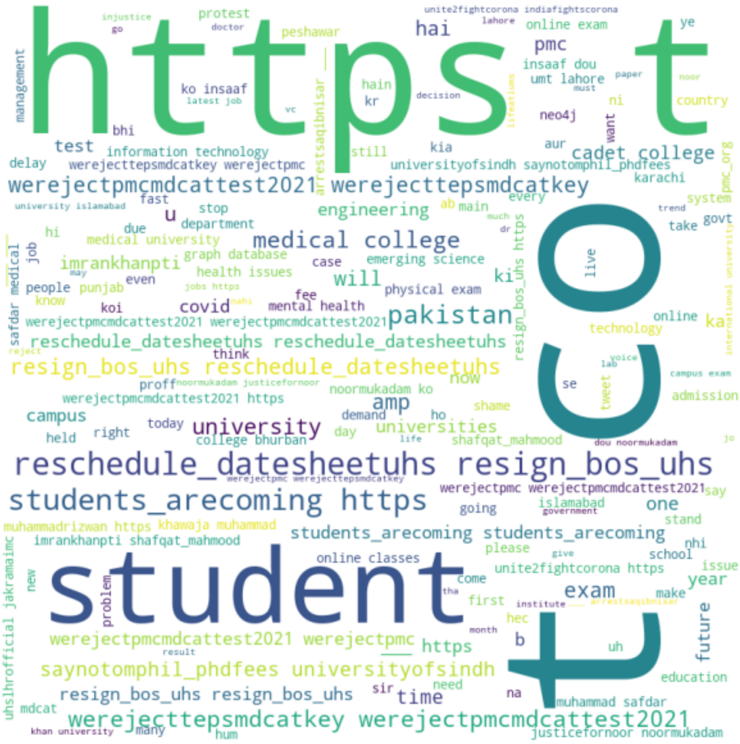
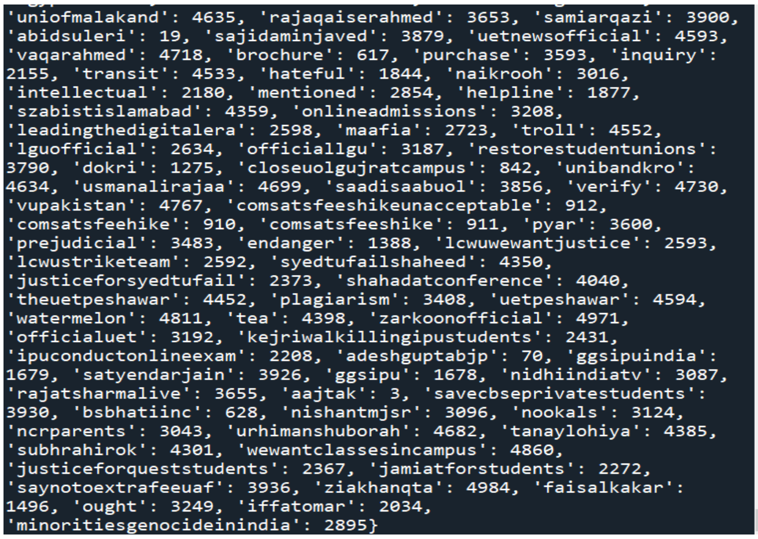


Figure 6.5: Word Cloud of Tweets

the labeled data from the long-term tasks. Each task contains separate models for parameter tuning. The nature of BERT is that it is unified. It is based on source simple-implementation with the tensor2tensor library. The number of layers has been represented with L, the self-attention heads have been represented with A, and the hidden layer as H.Therefore, the predictions have been generated with sentiment analysis and inserted into a CSV file.

The BERT Embeddings have been worked to transfer learning with several language models and illustrated that pre-training is an essential part of many language understanding systems. We enabled the low-resource tasks to gain from deep unidirectional architectures. They showed their main contribution on generalizing their findings to deep bidirectional architectures and permitted the same pre-trained model



```
'uniofmalakand': 4635, 'rajaqaiserahmed': 3653, 'samiarqazi': 3900,
'abidsuleri': 19, 'sajidaminjaved': 3879, 'uetnewsofficial': 4593,
'vaqarahmed': 4718, 'brochure': 617, 'purchase': 3593, 'inquiry':
2155, 'transit': 4533, 'hateful': 1844, 'naikrooh': 3016,
'intellectual': 2180, 'mentioned': 2854, 'helpline': 1877,
'szabistislamabad': 4359, 'onlineadmissions': 3208,
'leadingthedigitalera': 2598, 'maafia': 2723, 'troll': 4552,
'lguofficial': 2634, 'officiallgu': 3187, 'restorestudentunions':
3790, 'dokri': 1275, 'closeuolgujratcampus': 842, 'unibandkro':
4634, 'usmanalirajaa': 4699, 'saadisaabuol': 3856, 'verify': 4730,
'vupakistan': 4767, 'comsatsfeeshikeunacceptable': 912,
'comsatsfeehike': 910, 'comsatsfeeshike': 911, 'pyar': 3600,
'prejudicial': 3483, 'endanger': 1388, 'lcwuwewantjustice': 2593,
'lcwustriketeam': 2592, 'syedtufailshaheed': 4350,
'justiceforsyedtufail': 2373, 'shahadatconference': 4040,
'theuetpeshawar': 4452, 'plagiarism': 3408, 'uetpeshawar': 4594,
'watermelon': 4811, 'tea': 4398, 'zarkoonofficial': 4971,
'officialuet': 3192, 'kejriwalkillingipustudents': 2431,
'ipuconductonlineexam': 2208, 'adeshguptabjp': 70, 'ggsipuindia':
1679, 'satyendarjain': 3926, 'ggsipu': 1678, 'nidhiindiatv': 3087,
'rajatsharmalive': 3655, 'aajtak': 3, 'savecbseprivatestudents':
3930, 'bsbhatiinc': 628, 'nishantmjsr': 3096, 'nookals': 3124,
'ncrparents': 3043, 'urhimanshuborah': 4682, 'tanaylohiya': 4385,
'subhrahirok': 4301, 'wewantclassesincampus': 4860,
'justiceforqueststudents': 2367, 'jamiatforstudents': 2272,
'saynotoextrafeeuaf': 3936, 'ziakhanqta': 4984, 'faisalkakar':
1496, 'ought': 3249, 'iffatomar': 2034,
'minoritiesgenocideinindia': 2895}
```


Figure 6.6: Word-Count of Tweets

to well tackle a broad set of NLP functions.

We applied the feature-based approach by removing the activation's from one or more layers in the absence of fine-tuning parameters of BERT. This dependent learned representation for text where words that have the same meaning have a similar representation is used as input to an irregular initialized two-layer dimensional BiLSTM before the classification layer. The accuracy result of BERT embeddings has been generated by GOOGLE COLAB research.

6.1.2 Working of BERT-HAN in Python

Initially, the Attention layer has been applied on sentence level and word level. The uniform distribution is also present with an init initialization. The text classification of HAN includes an attention mechanism that contains the sequence in sentences and different words of those sentences in a document. The structured tokens are required in word level with *WIT* that contains the word. The data pre-processing is required in this section. The encoding mechanism containing the Gated Recurrent network(GRU) contains the vectorized tokens.

There is a sequence class which contained long short term memory model with the BERT model and fully connected layers with hidden layer. It also contained a sigmoid function for its evaluation. The model contained the input size, hidden size and sequence length with the fully connected layers and generated final output with RUN function.

scores	compound	sentiment_type
{'neg': 0.197, 'neu': 0.762, 'pos': 0.041, 'compound': -0.5	-0.5994	NEGATIVE
{'neg': 0.105, 'neu': 0.822, 'pos': 0.074, 'compound': -0.2	-0.2732	NEGATIVE
{'neg': 0.105, 'neu': 0.895, 'pos': 0.0, 'compound': -0.25}	-0.25	NEGATIVE
{'neg': 0.219, 'neu': 0.781, 'pos': 0.0, 'compound': -0.846	-0.8462	NEGATIVE
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL
{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}	0	NEUTRAL

Figure 6.7: Sentiment Analysis of Tweets

```
SVM Accuracy Score ->  93.80530973451327
Precision of SVM: 0.9422060766182299
Recall of SVM: 0.9783950617283951
Naive Bayes Accuracy Score ->  91.69703279541905
Precision of Naive Bayes: 0.9225512528473804
Recall of Naive Bayes: 0.9722222222222222
```

Figure 6.8: Accuracy Result of SVM and Naive Bayes

The attention layer is fed into the fully connected layer and its then emerged into a concatenation layer with the combination and results of all the previous layers. The sigmoid activation function analyzes that values should be passed or not.

Feature selection and feature reduction are the most important factors in selecting features. Transformation of data from high dimensional space to low dimensional space has been done by excluding some features and selecting important features that are average tweets per day, users comment on tweet, number of days user tweeted, and the total number of tweets against that user. It has provided the classifiers to be accurate and improved with different models.

Sr	pred	actual
0	TRUE	1
1	TRUE	0
2	TRUE	1
3	TRUE	1
4	TRUE	1
5	TRUE	1
6	TRUE	0
7	TRUE	0
8	TRUE	1
9	TRUE	1
10	TRUE	1
11	TRUE	0
12	TRUE	1
13	TRUE	0
14	TRUE	1
15	TRUE	1
16	TRUE	1
17	TRUE	1

df

Figure 6.9: Results of BERT Embeddings CSV file

```
[ ] o, t = run()

Some weights of the model checkpoint at bert_base_uncased were not used when initializing BertModel:
- This IS expected if you are initializing BertModel from the checkpoint of a model trained on anothe
- This IS NOT expected if you are initializing BertModel from the checkpoint of a model that you expe
100% 1441/1441 [1:48:32<00:00, 4.11s/it]
100% 321/321 [04:13<00:00, 1.64it/s]
Accuracy Score = 0.7580015612802498
-------------------------------------------------------------------------------
```

Figure 6.10: Accuracy result of BERT Embeddings

6.2 Comparison of Results of different Classifier

The different models have been used to evaluate the accuracy with precision and recall. It is clearly shown from the results that BERT-HAN gave the best results out of other classifiers. The BERT-HAN contains different layers with the sigmoid function. The fully connected, attention, Bi-GRU, embedding, and hidden vectors played a vital role in increasing the accuracy. The number of epochs increased in such a way that great accuracy can be achieved. Furthermore, it has shown that SVM gave results less than BERT-HAN but more than Naive Bayes results. This is a popular deep learning algorithm that is mostly implemented in Python. It contains the accurate predictions with support vectors and hyper-plane.

```
class Bert(nn.Module):
  def __init__(self, seq_len, hidden_size, num_layers=1):
    super(Bert, self).__init__()
    self.bert = transformers.BertModel.from_pretrained(config.bert_path, return_dict=False)
    self.embedding = nn.Linear(768, 1024)
    self.conv = nn.Conv1d(seq_len, seq_len, 3, 1, 1)
    self.maxpool = nn.MaxPool1d(3, 1, 1)
    self.fc = nn.Linear(1024, 1024)
    self.lstm1 = nn.LSTM(1024, hidden_size, num_layers, batch_first=True, bidirectional=True)
    self.final_fc = nn.Linear(hidden_size*2, hidden_size*2)

  def forward(self, ids, masks, token_type_ids):
    sequence_output, pooled_output = self.bert(
      ids,
      attention_mask=masks,
      token_type_ids=token_type_ids
    ) # seq_out shape is (N, seq_len, 768)

    out_embedding_layer = self.embedding(sequence_output)  # shape is (N, seq_len, 1024)
    conv = self.conv(out_embedding_layer)  # shape is (N, seq_len, 1024)
    maxpool = self.maxpool(conv)  # shape is (N, seq_len, 1024)
    fully_connected = self.fc(maxpool)  # shape is (N, seq_len, 1024)
    bilstm, _ = self.lstm1(fully_connected)  # shape is (N, seq_len, hidden_size*2)
    final_output = self.final_fc(bilstm)  # shape is (N, seq_len, hidden_size*2)
    final_output = final_output.reshape(final_output.shape[0], -1)

    return final_output
```

Figure 6.11: BERT class with Forward function and parameters of Layers

```
user_behaviours = torch.randn(64, 32, 128)
ids = torch.randint(0, 10000, (64, 32))
masks = torch.ones((64, 32), dtype=torch.int64)
token_type_ids = torch.zeros((64, 32), dtype=torch.int64)

seq_len = user_behaviours.shape[1]
input_size = user_behaviours.shape[2]
hidden_size = 256

model = Sequence(input_size, hidden_size, seq_len)
out = model(user_behaviours, ids, masks, token_type_ids)
print(out.shape)
```

Figure 6.12: Final Output

```
[ ]  bert_preprocess = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3")
     bert_encoder = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4")
```

Figure 6.13: Bert-Based Uncased with TensorFlow

```
[ ]  model.evaluate(X_test, y_test)

     41/41 [==============================] - 34s 749ms/step - loss: 0.1634 - accuracy: 0.9594 - precision: 0.9762 - recall: 0.9701
     [0.16343721747398376,
      0.9594067335128784,
      0.9761658310890198,
      0.970133900642395]
```

Figure 6.14: Accuracy, Precision and Recall of BERT-HAN

6.3 Human Based Validation

There are different ways of validation. It measures what is required to calculate and predict the truthfulness of outputs. Quantitative research is being used with the reliability of results and validation. The validity of tweets is required in qualitative research too because a human can judge more accurately than machines. To contain the results of validation, some tweets have been considered and a google form of survey questions have been made to identify whether the tweet is fake or not. There are two options fake or non-fake.

According to the results from survey questionnaires, we identified with statistical calculations that human-based validation depicted the results closer to the predicted results from the model. For example, tweet 1 is identified as fake by survey response and predicted results from the model. Other tweets are also closer to these results. Furthermore, it is shown that 95% accuracy is much improved with BERT-HAN, 93% accuracy by SVM model, 91% accuracy by Naive Bayes, and 75% accuracy of BERT model.

Line graphs analyzed the trends and data variables very clearly and contain predictions about the results. This line chart is showing the relationship between dependent and independent variables in one picture.

6.4 Summary

To summarize, the results are an important asset of research to analyze the work being done. The different approaches have been performed with component performance

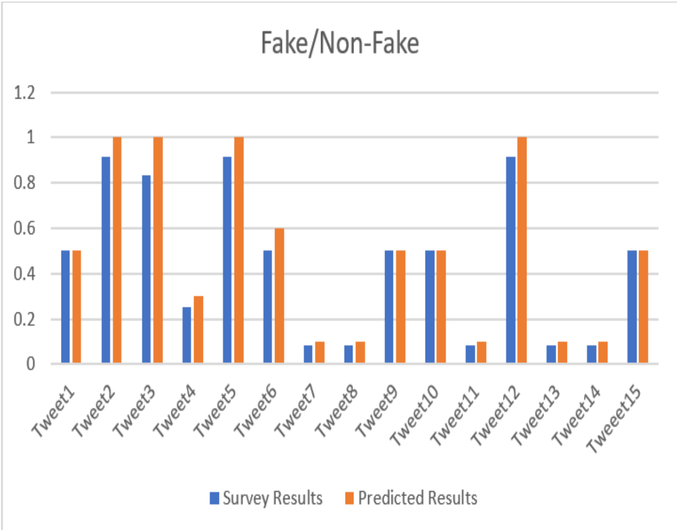


Figure 6.15: Bar Chart with Survey and Predicted Results

containing F1 score, precision, and recall. The scrapping results have been shown with a pre-processing image of the CSV file. The number of followers, following, and statuses counts have been achieved in PYTHON using the tweepy library. The results of the CSV file have been shown in this chapter. The feature selection and feature reduction have also been shown with the Sentiment analysis using VADER. An image of the CSV file containing scores of sentiment analysis is being attached. The BERT embeddings result in the CSV file and accuracy is also shown. The code flow of BERT-HAN architecture has been proposed with the final results of BERT-HAN containing different layers involved with the Sigmoid function that results in the predicted probabilities.

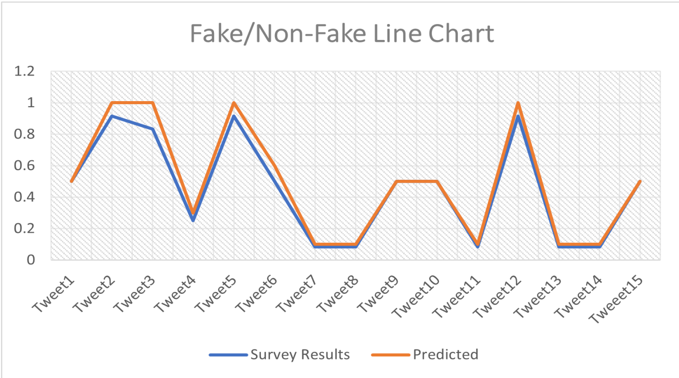


Figure 6.16: Line Chart with Survey and Predicted Results

CLASSIFIER	ACCURACY	PRECISION	RECALL
BERT-HAN	**95.9%**	97.62%	97.01%
SVM	93.8%	94.2%	97.83%
NAÏVE BAYES	91%	92%	97.1%
BERT	75%	79%	80%

Figure 6.17: Comparison of Different Classifiers

Chapter 7

Conclusion and Future Work

7.1 Conclusion

To conclude, it is depicted that the fake tweets and bot accounts are very common to spread hate and impersonation among others. To detect such tweets, we have implemented the improved steps to identify whether the tweet is fake or non-fake. The data has been extrcated from twitter using TWEEPY and TWINT, two libraries of PYTHON. The data has been pre-processed and cleaned to contain a good view and results. The data have been analyzed by setiment analyzer using Twitter and attained the results of 95% accuracy using a library named VADER in python. The labels have been achieved through it and then used for classification of models. Four different models have been used; SVM, Naive Bayes, BERT, and BERT-HAN model. The accuracies have been achieved and resulted showed that BERT-HAN containing different layers and vectors involved in BERT-HAN gave the best results of 95% accuracy. These predicted results of 15 tweets have been taken and the questionnaires has been made on Google Form. The responses of 15 people have been achieved. The accuracy has been calculated with the mean, standard deviation and comparsion of predicted and survey results. The both results were closest. Therefore, it is clear that human based validation gives good performance than machines. The final results have been achieved and shown in and images.

7.2 Thesis contribution

Following are the important contributions of our thesis;

- **Feature Improvement:**

Improved features of identifying fake tweets in which the comment of a user on tweet have been attained to get modified because there was a need to examine whether the user who commented is fake or not? Feature selection and feature reductions played an important role in our research and its results.

•Performance Optimization:

We got improved accuracy through BERT because our features are improved with the presence of pooled output with 768 neurons. It was sent to a linear output layer with 1 neuron for predicting the results. [9] We had also chosen transformer optimizer of AdamW and loss function of Binary-Cross Entropy.

•Performance Optimization of BERT-HAN:

We achieved improved accuracy of BERT-HAN state of art model containing different layers and Sigmoid function to improve the performance of data-set with improved dependencies of cross-sentence.

•Performance analysis

Through all the models, we did performance analysis and our results are better and improved.

7.3 Limitations

The limitations are as follows;

- The limitation of our data-set is the nominal data-set and it needs to be more structural.
- The more data is proposed, the more will be accurate.
- We have selected the main domain of different universities and the extracted tweets were not more.
- The inclusion of ontology and detailed semantic information is required in sentiment analysis.
- The correlation of sentiment with satisfaction is required. Improved accuracy is also required.

7.4 Future work

The future work for **Identification of Fake Tweets using Deep Learning Techniques** are as follows;

- For future work, more features are required to add to identify the increased efficiency.
- The Twitter data for larger sets of users are also required. An investigation on Pre-processing and fine-tuning is required.
- There will be more categories of data-set and comparisons will be made to get improved results so that the raw results and embeddings will be obtained.
- The characterization of language models is also required to improve.
- The proposed approach will be compared with more data sets, other social network sites, and variant language modeling.
- There will be working on more training data and there will be a comparison of execution and training time with the existing approaches to improve the performances.

References

[1] S. Krishnan and M. Chen, "Identifying tweets with fake news," *IEEE International Conference on Information Reuse and Integration (IRI)*, no. 1, pp. 460–464, 2018.

[2] Orteza, Amalian, Aroslaw, P. E., Rilepsky, on, Hai, Ergei, T. K., and Uritsyn, "Periodic nonlinear fourier transform for fiber-optic communications , part i : Theory and numerical methods," 2016.

[3] A. Albahr and M. A. Albahar, "An empirical comparison of fake news detection using different machine learning algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 11, 2020.

[4] W. Cui, N. Yousaf, J. Bhosle, A. Minchom, M. Ahmed, F. McDonald, I. Locke, R. Lee, M. E. R. O'Brien, and S. Popat, "1748p real-world outcomes in thoracic cancer patients (pts) with severe acute respiratory syndrome coronavirus 2 (covid-19): Single uk institution experience," *Annals of Oncology*, vol. 31, pp. S1020 – S1021, 2020.

[5] O. Ajao, D. Bhowmik, and S. Zargari, "Fake news identification on twitter with hybrid cnn and rnn models," in *Proceedings of the 9th International Conference on Social Media and Society*, (New York, NY, USA), p. 226–230, Association for Computing Machinery, 2018.

[6] S. P. Maniraj and H. Krishnan, "Fake account detection using machine learning and data science," *International Journal of Innovative Technology and Exploring Engineering*, 2019.

[7] O. Kadam and N. Surse, "Detection of fake social network account," 2021.

[8] E. Cueva, G. Ee, A. Iyer, A. S. Pereira, A. S. Roseman, and D. Martinez, "Detecting fake news on twitter using machine learning models," 2020.

[9] R. K. Kaliyar, A. Goswami, and P. Narang, "Fakebert: Fake news detection in social media with a bert-based deep learning approach," *Multimedia Tools and Applications*, pp. 1 – 24, 2021.

[10] P. Chongstitvatana and S. Aphiwongsophon, "Identifying misinformation on twitter with a support vector machine," *CA: A Cancer Journal for Clinicians*, vol. 70, 2020.

[11] S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, and R. Ghani, "Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining," *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[12] G. A. Ruz, P. A. Henríquez, and A. Mascareño, "Sentiment analysis of twitter data during critical events through bayesian networks classifiers," *Future Gener. Comput. Syst.*, vol. 106, pp. 92–104, 2020.

[13] R. Z. Candil, "Exploiting temporal context in speech technologies using lstm recurrent neural networks," 2018.

[14] J. J. Padilla, A. Tolk, and S. Y. Diallo, "Ms methodological challenges," EAIA and MatH '13, (San Diego, CA, USA), p. 9, Society for Computer Simulation International, 2013.

[15] B. T. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *ICWSM*, 2010.

[16] L. Luceri, T. Braun, and S. Giordano, "Analyzing and inferring human real-life behavior through online social networks with social influence deep learning," *Applied Network Science*, vol. 4, pp. 1–25, 2019.

[17] J. Li, H. Xu, X. He, J. Deng, and X. Sun, "Tweet modeling with lstm recurrent neural networks for hashtag recommendation," pp. 1570–1577, 2016.

[18] S. G. Zadeh and M. Schmid, "Bias in cross-entropy-based training of deep survival networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 3126–3137, 2021.

[19] L. Mai and H. B. Le, "Joint sentence and aspect-level sentiment analysis of product comments," *Annals of Operations Research*, pp. 1–21, 2021.

[20] D. Cheng, J. Li, L. Liu, K. Yu, T. D. Lee, J. L. S. of Information Technology, M. Sciences, U. of Electronic Science, I. Engineering, and H. U. of Technology, "Towards unique and unbiased causal effect estimation from data with hidden variables," *arXiv: Methodology*, 2020.

[21] M. R. Huq, A. Ali, and A. Rahman, "Sentiment analysis on twitter data using knn and svm," *International Journal of Advanced Computer Science and Applications*, vol. 8, 2017.

[22] S. MohiadeenAbdulKadhar and T. Revathi, "An efficient monarchic reconfiguration protocol with deadlock freedom on interconnection networks," *International Journal of Computer Applications*, vol. 43, pp. 1–6, 2012.

[23] D. C. Lee, "Enhanced ip services for cisco networks," 1999.

[24] D. Huang, Y. Huang, S. Khanna, P. Dwivedi, N. Slopen, K. M. Green, X. He, R. C. Puett, and Q. Nguyen, "Twitter-derived social neighborhood characteristics and individual-level cardiometabolic outcomes: Cross-sectional study in a nationally representative sample (preprint)," 2020.

[25] A. M. Wazarkar, "Python: A quintessential approach towards data science," *International Journal for Research in Applied Science and Engineering Technology*, 2021.

[26] S. García, J. Luengo, and F. Herrera, "Data preprocessing in data mining," in *Intelligent Systems Reference Library*, 2015.

[27] L. Berti-Équille, "Reinforcement learning for data cleaning and data preparation," 2019.

[28] A. Rietzler, S. Stabinger, P. Opitz, and S. Engl, "Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification," in *LREC*, 2020.

[29] X. Zhou, N. Pappas, and N. A. Smith, "Multilevel text alignment with cross-document attention," in *EMNLP*, 2020.

[30] "Proceedings of 2021 ieee international workshop on metrology for aerospace ieee metroaerospace," *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 1–1, 2021.

[31] T. Weiss, A. M. Fulterer, and A. Knotzer, "Energy flexibility of domestic thermal loads – a building typology approach of the residential building stock in austria," *Advances in Building Energy Research*, vol. 13, pp. 122 – 137, 2019.

[32] E. A. Fox, Z. Xie, and M. Klein, "Introduction to the web archiving and digital libraries 2015 workshop issue," *Bull. IEEE Tech. Comm. Digit. Libr.*, vol. 11, 2015.

[33] L. Gotsev and E. Shoikova, "An analysis of scientific production in big data knowledge domain on google books, youtube and ieee explore® digital library," *Proceedings of the 2020 4th International Conference on Cloud and Big Data Computing*, 2020.

[34] A. Deronja, K. Houser, A. Bapary, J. Barsch, R. Cunico, D. Fontana, C. Holt, M. Jensen, Y. Liao, R. Midence, M. Nagpal, M. Patel, I. Tualla, and S. Ward, "Exploring ieee c37.246-2017 guide for protection systems of transmission-to-generation interconnections," *2019 72nd Conference for Protective Relay Engineers (CPRE)*, pp. 1–7, 2019.

[35] F. Alam, "Computational models for analyzing affective behaviors and personality from speech and text," 2016.

[36] "Artificial intelligence : powering the deep-learning machines of tomorrow deep learning neural networks demand sophisticated power solutions," 2017.

[37] M. Sabin, S. Peltsverger, and C. Tang, "Updating the acm/ieee 2008 curriculum in information technology (abstract only)," *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015.

[38] V. G. Svenda, A. M. Stankovic, A. T. Sarić, and M. K. Transtrum, "Probabilistic network observability of a hybrid power system with communication irregularities," *2019 North American Power Symposium (NAPS)*, pp. 1–6, 2019.

[39] J. Á. G. García, R. Marante, and M. N. Ruiz, "Gan hemt class e 2 resonant topologies for uhf dc / dc power conversion,"

[40] D. Zimbra, A. Abbasi, D. D. Zeng, and H. Chen, "The state-of-the-art in twitter sentiment analysis," *ACM Transactions on Management Information Systems (TMIS)*, vol. 9, pp. 1 – 29, 2018.

[41] "Autonomous hospital management system using bluetooth," 2021.

[42] L. J. Garcia, G. Merino, S. C. Domenech, E. A. D. Merino, and A. L. Pinto, "Projeto centrado no ser humano: um panorâma bibliométrico — human centred design: a bibliometric overview based on science direct com base na science direct," 2016.

[43] O. Shvets, K. Murtazin, and G. Piho, "Providing feedback for students in e-learning systems: a literature review, based on ieee explore digital library," *2020 IEEE Global Engineering Education Conference (EDUCON)*, pp. 284–289, 2020.

[44] S. J. Julier, R. W. Lindeman, and C. Sandor, "Guest editor's introduction to the special section on the ieee international symposium on mixed and augmented reality 2014," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, pp. 1321–1322, 2015.

[45] R. R. Mandical, N. Mamatha, N. Shivakumar, R. R. Monica, and A. Krishna, "Identification of fake news using machine learning," *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1–6, 2020.

[46] S. Manzoor, J. Singla, and Nikita, "Fake news detection using machine learning approaches: A systematic review," pp. 230–234, 04 2019.

[47] J. Shaikh and R. B. Patil, "Fake news detection using machine learning," *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, pp. 1–5, 2020.

[48] S. A. Alkhodair, S. H. H. Ding, B. C. M. Fung, and J. Liu, "Detecting breaking news rumors of emerging topics in social media," *Inf. Process. Manag.*, vol. 57, p. 102018, 2020.

[49] H. Gupta, M. S. Jamal, S. Madisetty, and M. S. Desarkar, "A framework for real-time spam detection in twitter," *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, pp. 380–383, 2018.

[50]

[51] A. Frame and G. Brachotte, "Engineering victory and defeat: the role of social bots on twitter during the french presidentialelections," 2018.

[52] I. Ahmad, M. W. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complex.*, vol. 2020, pp. 8885861:1–8885861:11, 2020.

[53] S. D. P. Reddy, "Fake profile identification using machine learning," 2019.

[54] m. kesharwani, s. kumari, and V. Niranjan, "Detecting fake social media account using deep neural networking," 07 2021.

[55] R. Raturi, "Machine learning implementation for identifying fake accounts in social network," 08 2018.

[56] M. Li, T. Zhang, Y. Chen, and A. Smola, "Efficient mini-batch training for stochastic optimization," *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[57] A. Munshi, M. Arvindhan, and K. Thirunavukkarasu, "Random forest application of twitter data sentiment analysis in online social network prediction," *Emerging Technologies for Healthcare*, 2021.

[58] F. Kronthaler and S. Zöllner, "R and rstudio," *Data Analysis with RStudio*, 2020.

[59] A. Bisht, A. Singh, H. S. Bhadauria, J. Virmani, and Kriti, "Detection of hate speech and offensive language in twitter data using lstm model," 2020.

[60] "Proceedings of the international workshop 'integration of international expertise in the development of a mental health surveillance system in germany'," *BMC Proceedings*, vol. 14, 2020.

[61] "Synopsis of ieee std c95.1™-2019 "ieee standard for safety levels with respect to human exposure to electric, magnetic, and electromagnetic fields, 0 hz to 300 ghz","

[62] J. Marques-Silva, T. Gerspacher, M. C. Cooper, A. Ignatiev, and N. Narodytska, "Explaining naive bayes and other linear classifiers with polynomial time and delay," *ArXiv*, vol. abs/2008.05803, 2020.

[63] R. Srebrovic and J. Yonamine, "Leveraging the bert algorithm for patents with tensorflow and bigquery,"

[64] W. Antoun, F. Baly, and H. M. Hajj, "Arabert: Transformer-based model for arabic language understanding," *ArXiv*, vol. abs/2003.00104, 2020.

[65] D. Hwang, S. Yang, Y. Kwon, K.-H. Lee, G. Lee, H. Jo, S. Yoon, and S. Ryu, "Comprehensive study on molecular supervised learning with graph neural networks," *Journal of chemical information and modeling*, 2020.

[66] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[67] K. Demertzis, L. S. Iliadis, and E. Pimenidis, "Large-scale geospatial data analysis: Geographic object-based scene classification in remote sensing images by gis and deep residual learning," in *EANN*, 2020.

[68] S. Seema, S. Goutham, S. Vasudev, R. R. Putane, and X. Li, "Surveillance video analysis using deep learning techniques for traffic and crowd management," 2019.

[69] C. B. Thacker and R. M. Makwana, "Ensemble of multi feature layers in cnn for facial expression recognition using deep learning," 2019.

[70] M. Dorado-Moreno, N. Navarin, P. A. Gutiérrez, L. Prieto, A. Sperduti, S. Salcedo-Sanz, and C. Hervás-Martínez, "Multi-task learning for the prediction of wind power ramp events with deep neural networks," *Neural networks : the official journal of the International Neural Network Society*, vol. 123, pp. 401–411, 2020.

[71] A. AgnesArchanaD, Bavithra, R. Manojkumar, M. Yogeshkumar, and T. Kavitha, "Road crack detection on drone images using convolutional neural network(cnn) deep learning algorithm," 2020.

[72] P. Kanade, F. David, and S. Kanade, "Convolutional neural networks(cnn) based eye-gaze tracking system using machine learning algorithm," 2021.

[73] N. Ihsan, Zulman, and S. F. Utami, "The effect of polymeric exercises for front speed kick," 2020.

[74] Y. Liu, X. Yu, Y. Wu, and S. Song, "Forecasting variation trends of stocks via multiscale feature fusion and long short-term memory learning," *Sci. Program.*, vol. 2021, pp. 5113151:1–5113151:9, 2021.

[75] X. Li, H. Xiong, H. An, C.-Z. Xu, and D. Dou, "Rifle: Backpropagation in depth for deep transfer learning through re-initializing the fully-connected layer," in *ICML*, 2020.

[76] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Q. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *NAACL*, 2018.

[77] N. Boudad, S. Ezzahid, R. Faizi, and R. O. H. Thami, "Exploring the use of word embedding and deep learning in arabic sentiment analysis," 2019.

[78] S. S. Irish, M. Sherin, R. Surya, Y. Vidhya, and V. Ramamoorthy, "Opinion poll: Big data implementation of unstructured data analytics of social network reviews using sentiment analysis svm," *International Journal of Advanced engineering, Management and Science*, vol. 3, pp. 288–291, 2017.

[79] A. P. Rodrigues and N. N. Chiplunkar, "Real-time twitter data analysis using hadoop ecosystem," *Cogent Engineering*, vol. 5, 2018.

[80] A. Park and D. S. West, "Understanding socio-cultural factors related to obesity: Sentiment analysis on related tweets," *Online Journal of Public Health Informatics*, vol. 11, 2019.

[81] S. Iasulaitis and I. Vicari, "The salience of traditional moral values: Bolsonaro's electoral competition strategy on twitter," *International Journal of Social Science Studies*, 2021.

[82] D. Valdez, M. ten Thij, K. C. Bathina, L. A. Rutter, and J. Bollen, "Social media insights into us mental health during the covid-19 pandemic: Longitudinal analysis of twitter data," *Journal of Medical Internet Research*, vol. 22, 2020.