WELFAKE: WORD EMBEDDING OVER LINGUISTIC FEATURES FOR FAKE NEWS DETECTION

Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree Of

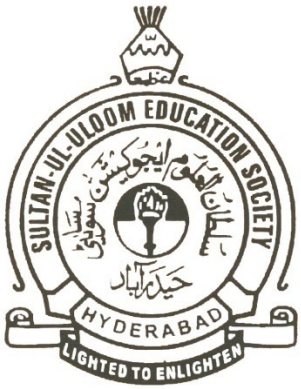
BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

MOHAMMED FAYAZ MOQUEEM (1604-18-733-038)
MOHAMMED HASSAN SHAJI KHAN (1604-18-733-040)
SYED HABEEB UDDIN (1604-18-733-041)



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY
(Affiliated to Osmania University)
Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills, Hyderabad-34

2022

Date: 26 / 05 / 2022

CERTIFICATE

This is to certify that the project dissertation titled "**WELFAKE: Word Embedding Over Linguistic Features for Fake News Detection***"* being submitted by

1. *Mohammed Fayaz Moqueem (1604-18-733-038)*
2. *Mohammed Hassan Shaji Khan (1604-18-733-040)*
3. *Syed Habeeb Uddin (1604-18-733-041)*

in Partial Fulfillment of the requirements for the award of the degree Of BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2021-22 is the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signatures:

Internal Project Guide

(Dr Syed Shabbeer Ahmad)

(Professor and Associate Head CSED)

Head of the Department CSED

(Dr. Ahmed Abdul Moiz Qyser)

External Examiner

DECLARATION

This is to certify that the work reported in the major project entitled "**WELFAKE: Word Embedding Over Linguistic Features for Fake News Detection**" is a record of the bonafide work done by us in the Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Osmania University. The results embodied in this report are based on the project work done entirely by us and not copied from any other source.

1. *Mohammed Fayaz Moqueem* *(1604-18-733-038)*
2. *Mohammed Hassan Shaji Khan* *(1604-18-733-040)*
3. *Syed Habeeb Uddin* *(1604-18-733-041)*

ACKNOWLEDGEMENT

My heart is filled with gratitude to the Almighty for empowering me with courage, wisdom and strength to complete this project successfully. I give him all the glory, honor and praise.

We thank our Parents for having sacrificed a lot in their lives to impart the best education to me and making me a promising professional for tomorrow.

We would like to express my sincere gratitude and reverence to my project supervisor **Dr. Syed Shabbeer Ahmad** for his valuable suggestions, interest and commendable guidance throughout the course of this project.

We are happy to express gratitude to **Prof. Dr. Ahmed Abdul Moiz Qyser,** Head of the Computer Science and Engineering Department, for his valuable and intellectual suggestions apart from educate guidance constant encouragement.

With a great pleasure and privilege, I extend my gratitude to **Prof. Dr. Syed Shabbeer Ahmed,** Associated Head of Computer Science and Engineering Department, project in-charge, who offered valuable suggestions at every step.

We are pleased to acknowledge our thanks to all those who devoted themselves directly or indirectly to make this project work a total success.

MOHAMMED FAYAZ MOQUEEM

MOHAMMED HASSAN SHAJI KHAN

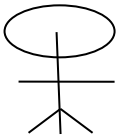
SYED HABEEB UDDIN

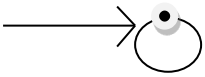
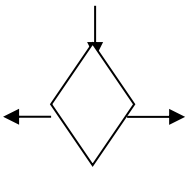
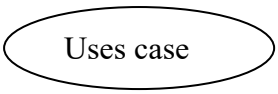
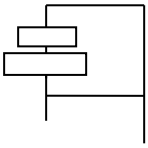
ABSTRACT

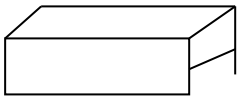
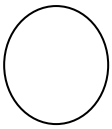
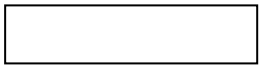
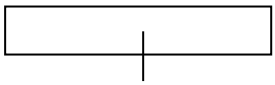
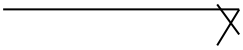
Social media is a popular medium for the dissemination of real-time news all over the world. Easy and quick information proliferation is one of the reasons for its popularity. An extensive number of users with different age groups, gender, and societal beliefs are engaged in social media websites. Despite these favorable aspects, a significant disadvantage comes in the form of fake news, as people usually read and share information without caring about its genuineness. Therefore, it is imperative to research methods for the authentication of news. To address this issue, this article proposes a two-phase benchmark model named WELFake based on word embedding (WE) over linguistic features for fake news detection using machine learning classification. The first phase pre-processes the data set and validates the veracity of news content by using linguistic features. The second phase merges the linguistic feature sets with WE and applies voting classification. To validate its approach, this article also carefully designs a novel WELFake data set with approximately 72 000 articles, which incorporates different data sets to generate an unbiased classification output. Experimental results show that the WELFake model categorizes the news in real and fake with a 96.73% which improves the overall accuracy by 1.31% compared to bidirectional encoder representations from transformer (BERT) and 4.25% compared to convolutional neural network (CNN) models. Our frequency based and focused analyzing writing patterns model outperforms predictive-based related works implemented using the Word2vec WE method by up to 1.73%.

LIST OF FIGURES

LIST OF SYMBOLS

S.NO	NAME	NOTATION	DESCRIPTION
1.	Class	+ *public* -*private* # *protected* *Class Name* *-attribute* *-attribute* *+operation* *+operation* *+operation*	Represents a collection of similar entities grouped together.
2.	Association	Class A NAME Class B Class A Class B	Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classis.
5.	Aggregation	Class A Class A Class B	Interaction between the system and external environment
5.	*Relation* (uses)	*Uses*	Used for additional process communication.
6.	Relation	extends	Extends relationship is used when one use case is

	(extends)		similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Usecase		Interact ion between the system and external environment.
14.	Component		Represents physical modules which is a collection of components.

15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message	Message 	Represents the message exchanged.

LIST OF ABBREVATION

S.NO	ABBREVATION	EXPANSION
1.	DB	DataBase
2.	JVM	Java Virtual Machine
3.	JSP	Java Server Page
4.	CB	Collective Behavior
5.	RSSS	Ramp secret sharing scheme
6.	JRE	Java Runtime Environment

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 GENERAL

Deep Learning:

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the back propagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

Machine-learning technology powers many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones. Machine-learning systems are used to identify objects in images, transcribe speech into text, match news items, posts or products with users' interests, and select relevant results of search. Increasingly, these applications make use of a class of techniques called deep learning. Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing a pattern-recognition or machine-learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning subsystem, often a classifier, could detect or classify patterns in the input. Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. Deep-learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be

learned. For classification tasks, higher layers of representation amplify aspects of the input that are important for discrimination and suppress irrelevant variations. An image, for example, comes in the form of an array of pixel values, and the learned features in the first layer of representation typically represent the presence or absence of edges at particular orientations and locations in the image. The second layer typically detects motifs by spotting particular arrangements of edges, regardless of small variations in the edge positions. The third layer may assemble motifs into larger combinations that correspond to parts of familiar objects, and subsequent layers would detect objects as combinations of these parts. The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure. Deep learning is making major advances in solving problems that have resisted the best attempts of the artificial intelligence community for many years. It has turned out to be very good at discovering intricate structures in high-dimensional data and is therefore applicable to many domains of science, business and government. In addition to beating records in image recognition and speech recognition, it has beaten other machine-learning techniques at predicting the activity of potential drug molecules, analysing particle accelerator data, reconstructing brain circuits, and predicting the effects of mutations in non-coding DNA on gene expression and disease. Perhaps more surprisingly, deep learning has produced extremely promising results for various tasks in natural language understanding, particularly topic classification, sentiment analysis, question answering and language translation. We think that deep learning will have many more successes in the near future because it requires very little engineering by hand, so it can easily take advantage of increases in the amount of available computation and data. New learning algorithms and architectures that are currently being developed for deep neural networks will only accelerate this progress.

Nowadays people around the world are getting much involved on online social networks regardless of age, community, or sex [1]. Communicating using social networks is simple, fast, and attractive to share and transfer information. Currently, social network sites like Facebook trailed by Twitter are the market pioneers, facilitating over 1.3 billion clients with a dynamic monthly variation of 300 million users in average [2]. Their collaborations generate Terabytes of information every second [3], [4]. Online social networks are attractive because of the simple and convenient way to access and circulate information with other people. However, the fast scattering of data at a high rate with minimal effort enables the widespread of false information, such as fake news, which are harmful to society and people.

Fake news are low-quality information with purposefully false data, propagated by individuals or bots that deliberately manipulate message for tattle or political plans. Schudson and Zelizer [5] claimed that the term “fake news” originated in previous centuries together with the mass media itself.

Nevertheless, this term attracted increased attention after the U.S. presidential elections of 2016, when the propagation of fake news on social media pulled the attention of a larger number of online users than traditional newsreaders. In the last five months before the elections, approximately 7.5 million tweets contained a link to exceptionally one-sided or false news websites. An interesting and worrying aspect is that false and unsubstantiated news from doubtful sources attracts more audiences than credible information [6]. Relevant work on this topic concluded that fake news spread quicker, penetrate further, and have a deeper impact than true news [7]. There are numerous cases where people accept and spread news without checking their correctness certified by sources. By doing this, they become part of a group that deliberately or unintentionally propagates fake news. The intention behind the proliferation of fake news may be manipulation of public views for financial or political benefit, or simply fun. The negative consequences of this phenomenon are, therefore, undeniable, ranging from wrong decision-making to episodes of bullying and violence. Fig. 1(a) and (b) shows two common examples of fake news over social networks.

False information categories are fake news, satire, misinformation, rumor, hoax, disinformation, propaganda, and opinion spam [8]. These categories are not mutually exclusive, but many researchers used them with different storylines. Although there exist a few websites to check the authenticity of the news like PolitiFact [9], The Washington Post Fact Checker [10], FactCheck [11], Snopes [12], TruthOrFiction [13], FullFact [14], HoaxSlayer [15], Vishvas News [16], Factly Media & Research [17] yet, these websites are unable to spontaneously react to any fake news event [18].

Source: https://www.truthorfiction.com/covid-19-hipaa-face-mask-exemption-passes/

Fake news examples. (a) Decontextualized news. (b) False news.

As online social networks are major sources of information that can mislead individuals or communities [19], there is a serious need for solutions to verify the authenticity of the content. Many researchers consistently try to develop machine learning (ML) models with different sets of features targeted toward automating the fake news detection process [20], [21] using visual [22] or text-based linguistic approaches.

1.2 SCOPE OF THE PROJECT

We proposed a novel WELFake model for fake news detection in two steps. 1) collection of various linguistic features from state-of-the- art methods and identification of a subset that performs well on the larger WELFake data set, and 2) ensemble learning on WE features using CNN Model Architecture.

1.3 OBJECTIVE

This article proposes a two-phase benchmark model named WELFake based on word embedding (WE) over linguistic features for fake news detection using machine learning classification. The first phase preprocesses the data set and validates the veracity of news content by using linguistic features. The second phase merges the linguistic feature sets with CNN Model Architecture. To validate its approach, this article also carefully designs a novel WELFake data set with few articles, which incorporates different data sets to generate an unbiased classification output. Experimental results show that the WELFake model categorizes the news in real and fake and achieved an accuracy of 93%.

1.4 PROBLEM STATEMENT

However, the following four questions remain unanswered.

1) Which linguistic features are most significant in classifying the news data into real and fake?

2) Which word embedding (WE) technique with linguistic features predicts fake news better than other ML methods like convolutional neural networks (CNNs) or bidirectional encoder representations from transformers (BERTs)?

3) Which classification method is the most appropriate for fake news detection on available data sets?

4) Does ensemble voting classifier improve the fake news detection results?

To answer these questions, we propose a new method called WELFake exclusively focused on text data in three stages.

1) Fake news prediction using linguistic feature sets (LFS);

2) WE over LFS for improved fake news detection over a WELFake data set.

3) Comparative analysis of the linguistic features-based results with state-of-the-art CNN and BERT methods.

The WELFake model does not require additional metadata information related to the user or media [24] for the classification of real and fake news. Instead, it aims for a reformation of the state-of-the-art techniques in the detection of fake news over social media websites by using a combined LFS and WE technique. We highlight three contributions of our WELFake model.

1.5 EXISTING SYSTEM:

- Burgoon et al. used 16 linguistic features categorized in four classes, which achieved an accuracy of 60.72% using a DT algorithm with 15-fold cross-validation.
- Vicario et al. used different features like text (e.g., number of characters, words, sentences, question marks, and negations), user-specific, and message specific (e.g., number of replies, likes) to identify hoaxes and fake news on social media using linear regression, logistic regression, support vector machine (SVM), K-nearest neighbor (KNN), and NNs. The validation on an Italian Facebook data set with new features achieved an accuracy of 91% on the linear regression classification algorithm.
- Pérez-Rosas et al. used major linguistic features (e.g., n-grams, punctuation, psycho-linguistic, readability, and syntax) and achieved an accuracy of 76% on two novel data sets covering seven domains.

1.5.1 EXISTING SYSTEM DISADVANTAGES

- The fast scattering of data at a high rate with minimal effort enables the widespread of false information, such as fake news, which are harmful to society and people.
- One major challenge that is associated with OSNs is verification of messages exchanged as well as the authenticity of users. Some messages that are spread through these social networks may create horrible situations regarding peace and harmony in society. Such messages, currently coined as fake news, can also be life-threatening.
- The existing system too is a sparse matrix. This means its not storage efficient & calculations are inefficient to run on top
- Larger the vocabulary size, larger the matrix size (not scalable to large vocabulary)
- Not all word associations can be understood using this technique.

1.5.2 LITERATURE SURVEY

TITLE **:** A situational analytic method for user behavior pattern in multimedia social networks

AUTHOR **:** Z. Zhang, R. Sun, X. Wang, and C. Zhao

YEAR **:** 2019

DESCRIPTION

The past decade has witnessed the emergence and progress of multimedia social networks (MSNs), which have explosively and tremendously increased to penetrate every corner of our lives, leisure and work. Moreover, mobile Internet and mobile terminals enable users to access to MSNs at anytime, anywhere, on behalf of any identity, including role and group. Therefore, the interaction behaviors between users and MSNs are becoming more comprehensive and complicated. This paper primarily extended and enriched the situation analytics framework for the specific social domain, named as SocialSitu, and further proposed a novel algorithm for users' intention serialization analysis based on classic Generalized Sequential Pattern (GSP). We leveraged the huge volume of user behaviors records to explore the frequent sequence mode that is necessary to predict user intention. Our experiment selected two general kinds of intentions: playing and sharing of multimedia, which are the most common in MSNs, based on the intention serialization algorithm under different minimum support threshold (Min_Support). By using the users' microscopic behaviors analysis on intentions, we found that the optimal behavior patterns of each user under the Min_Support, and a user's behavior patterns are different due to his/her identity variations in a large volume of sessions data.

TITLE : A large-scale study of the Twitter follower network to characterize the spread of prescription drug abuse tweets

AUTHOR : R. Sequeira, A. Gayen, N. Ganguly, S. K. Dandapat, and J. Chandra

YEAR : 2019

DESCRIPTION

In this article, we perform a large-scale study of the Twitter follower network, involving around 0.42 million users who justify DA, to characterize the spreading of DA tweets across the network. Our observations reveal the existence of a very large giant component involving 99% of these users with dense local connectivity that facilitates the spreading of such messages. We further identify active cascades over the network and observe that the cascades of DA tweets get spread over a long distance through the engagement of several closely connected groups of users. Moreover, our observations also reveal a collective phenomenon, involving a large set of active fringe nodes (with a small number of follower and following) along with a small set of well-connected nonfringe nodes that work together toward such spread, thus potentially complicating the process of arresting such cascades. Furthermore, we discovered that the engagement of the users with respect to certain drugs, such as Vicodin, Percocet, and OxyContin, that were observed to be most mentioned in Twitter is instantaneous. On the other hand, for drugs, such as Lortab, that found lesser mentions, the engagement probability becomes high with increasing exposure to such tweets, thereby indicating that drug abusers engaged on Twitter remain vulnerable to adopting newer drugs, aggravating the problem further.

TITLE : Human-centric cyber social computing model for hotevent detection and propagation

AUTHOR : L.-L. Shi et al.

YEAR : 2019

DESCRIPTION

Microblogging networks have gained popularity in recent years as a platform enabling expressions of human emotions, through which users can conveniently produce contents on public events, breaking news, and/or products. Subsequently, microblogging networks generate massive amounts of data that carry opinions and mass sentiment on various topics. Herein, microblogging is regarded as a useful platform for detecting and propagating new hot events. It is also a useful channel for identifying high-quality posts, popular topics, key interests, and high-influence users. The existence of noisy data in the traditional social media data streams enforces to focus on human-centric computing. This paper proposes a human-centric social computing (HCSC) model for hot-event detection and propagation in microblogging networks. In the proposed HCSC model, all posts and users are preprocessed through hypertext induced topic search (HITS) for determining high-quality subsets of the users, topics, and posts. Then, a latent Dirichlet allocation (LDA)-based multiprototype user topic detection method is used for identifying users with high influence in the network. Furthermore, an influence maximization is used for final determination of influential users based on the user subsets. Finally, the users mined by influence maximization process are generated as the influential user sets for specific topics. Experimental results prove the superiority of our HCSC model against similar models of hot-event detection and information propagation.

TITLE : Study and detection of fake news: P2C2-based machine learning approach

AUTHOR : P. K. Verma and P. Agrawal.

YEAR : 2020

DESCRIPTION

News is the most important and sensitive piece of information which affects the society nowadays. In the current scenario, there are two ways to propagate news all over the world; first one is the traditional way, i.e., newspaper and second is electronic media like social media websites. Electronic media is the most popular medium these days because it helps to propagate news to huge audience in few seconds. Besides these benefits of electronic media, it has one disadvantage also, i.e., "spreading the Fake News". Fake news is the most common problem these days. Even big companies like Twitter, Facebook, etc. are facing fake news problems. Several researchers are working in these big companies to solve this problem. Fake news can be defined as the news story that is not true. In some specific words, we can say that news is fake if any news agency declares a piece of news deliberately written as false and it is also verifiably as false. This paper focuses on some key characteristics of fake news and how it is affecting the society nowadays. It also includes various key viewpoints which are useful to categorize whether the news is fake or not. At last, this paper discussed some key challenges and future directions that help in increasing accuracy in detection of fake news on the basis of P2C2 (Propagation, Pattern, Comprehension & Credibility) approach having two phases: Detection and Verification. This paper helps readers in two ways (i) Newcomer can easily get the basic knowledge and impact of fake news; (ii) They can get knowledge of different perspectives of fake news which are helpful in the detection process.

TITLE : An analysis of the internal organization of Facebook groups

AUTHOR : A. De Salve, P. Mori, B. Guidi, and L. Ricci

YEAR : 2019

DESCRIPTION

With the rapid development and growth of online social networks (OSNs), researchers have been pushed forward to improve the knowledge of these complex networks by analyzing several aspects, such as the types of social media, the structural properties of the network, or the interaction patterns among users. In particular, a relevant effort has been devoted to the study and identification of cohesive groups of users in OSNs (also referred as communities) because they are the basic building block of each OSN. While several research works on groups in OSNs have mainly focused on identifying the types of groups and the contents created by their members, the analysis of internal organizations of such groups remains unexplored due to the lack of real data sets containing information about such groups, about their members, and the interactions among them. In this article, we compensate for this shortcoming by studying the main properties of groups defined in OSNs, taking as reference use cases 40 real Facebook groups of different categories that account for a total of about 500.000 users. In particular, we exploit interaction patterns among users and social network analysis to uncover interesting aspects related to the internal organization of groups. Experimental results reveal that the majority of the collected groups exhibit an internal structure where members can be clustered in four subgroups according to the level of tie strength of the relations they have. Furthermore, clusters identified on Facebook groups can provide relevant information about the importance of users within such groups.

TITLE : Predicting ayurveda-based constituent balancing in human body using machine learning methods.

AUTHOR : V. Madaan and A. Goyal.

YEAR : 2020

DESCRIPTION

Human Body constitution (prakriti) defines what is in harmony with human nature and what will cause to move out of balance and experience illness. Tridosha defines the three basic energies or principles that determine the function of our body on the physical and emotional levels. The three energies are known as VATT, PITT and KAPH. Each individual has a unique balance of all three of these energies. Some people will be predominant in one, while others will be a mixture of two or more. Ayurveda-dosha studies have been used for a long time, but the quantitative reliability measurement of these diagnostic methods still lags behind. A careful and appropriate analysis leads to an effective treatment. To collect a meaningful data set, a questionnaire with 28 different characteristics is validated by Ayurveda experts. Authors calculate Cronbach alpha of VATT-Dosha, PITT-Dosha and KAPH-Dosha as 0.94, 0.98 and 0.98, respectively to check the reliability of the questionnaire. Authors analyzed questionnaires of 807 healthy persons aged 20-60 years and found 62.1% men and 37.9% women. The class imbalance problem is resolved with oversampling and the equally distributed data set of randomly selected 405 persons is used for the actual experiment. Using computer algorithms, we randomly divide the data set (8:2) into a training set of 324 persons and a test data set of 81 persons. Model is trained using traditional machine learning techniques for classification analysis as Artificial Neural Network (ANN), K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Naive Bayes (NB) and Decision Tree (DT). System is also implemented using ensemble of several machine learning methods for constitution recognition. Evaluation measures of classification such as root mean square error (RMSE), precision, recall, F-score, and accuracy is calculated and analyzed. On analyzing the results authors find that the data is best trained and tested with CatBoost, which is tuned with hyper parameters and achieves 0.96 precision, 0.95 recall, 0.95 F-score and 0.95 accuracy rate. The experimental result shows that the proposed model based on ensemble learning methods clearly surpasses conventional methods. The results conclude that advances in boosting algorithms could give machine learning a leading future.

1.6 PROPOSED SYSTEM

- We proposed a novel WELFake model for fake news detection in two steps. 1) collection of various linguistic features from state-of-the- art methods and identification of a subset that performs well on the larger WELFake data set, and 2) ensemble learning on WE features using CNN Model Architecture.
- We applied an adversarial approach to evaluate the model generalization and effectiveness by training and testing on separate data sets. Experimental results on the WELFake data set revealed that our model achieved a fake news classification accuracy of up to 93%.

1.6.1 PROPOSED SYSTEM ADVANTAGES

- The WELFake model does not require additional metadata information related to the user or media for the classification of real and fake news. Instead, it aims for a reformation of the state-of-the-art techniques in the detection of fake news over social media websites.
- The proposed system model applied on a larger data set with over many news achieved a higher accuracy of 93% compared to the related methods.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL:

We proposed a novel WELFake model for fake news detection in two steps.

1) collection of various linguistic features from state-of-the- art methods and identification of a subset that performs well on the larger WELFake data set, and

2) ensemble learning on WE features using various ML methods.

2.2 METHODOLOGIES

2.2.1 MODULES NAME:

MODULES:

- Data Collection
- Dataset
- Importing the necessary libraries
- Word Embedding
- Splitting the dataset
- Building the model
- Analyze and Prediction
- Apply the model and plot the graphs for accuracy and loss
- Accuracy on test set
- Saving the Trained Model

MODULES DESCSRIPTION:

Data Collection:

In the first module, we developed the system to get the input dataset for the training and testing purpose. The project name is Word Embedding over Linguistic Features for Fake News Detection. We give the data set in model folder.

Dataset:

The dataset consists of 10018 individual data. There are 4 columns in the dataset, which are described below

1. Id: unique
2. title: new title name
3. text: review
4. label: fake or real

Importing the necessary libraries:

We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy, matplotlib and tensorflow.

Word Embedding:

Word embedding represents the density of the word vector, unlike what we have done with the Countvectorizer. It is a different way to preprocess the data. This embedding can map semantically similar words. It does not consider the text as a human language but maps the structure of sets of words used in the corpus. They aim to map words into a geometric space which is called an embedding space. Keras provides a couple of methods for text preprocessing and sequence preprocessing. We can use them to make our data a better fit for the TextCNN model

Splitting the dataset:

Split the dataset into train and test. 80% train data and 20% test data.

Building the model:

We use a Convolutional Neural Network (CNN) as they have proven to be successful at document classification problems. A conservative CNN configuration is used with 32 filters (parallel fields for processing words) and a kernel size of 8 with a rectified linear ('relu') activation function. This is followed by a pooling layer that reduces the output of the convolutional layer by half.

Next, the 2D output from the CNN part of the model is flattened to one long 2D vector to represent the 'features' extracted by the CNN. The back-end of the model is a standard Multilayer Perceptron layers to interpret the CNN features. The output layer uses a sigmoid activation function to output a value between 0 and 1 for the negative and positive sentiment in the review.

We can see that the Embedding layer expects documents with a length of 547 words as input and encodes each word in the document as a 11element vector.

We use a binary cross entropy loss function because the problem we are learning is a binary classification problem. The efficient Adam implementation of stochastic gradient descent is used and we keep track of accuracy in addition to loss during training. The model is trained for 18 epochs, or 8 passes through the training data.

The network configuration and training schedule were found with a little trial and error, but are by no means optimal for this problem.

Analyze and Prediction:

In the actual dataset, we chose only 1 feature:

1. text: comment
2. Labels: Labels

   Fake or real.

Apply the model and plot the graphs for accuracy and loss:

We will compile the model and apply it using fit function. The batch size will be 32. Then we will plot the graphs for accuracy and loss. We got average validation accuracy of 99.6% and average training accuracy of 93.3%.

Accuracy on test set:

We got an accuracy of 93.7% on test set

Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like `pickle`. Make sure you have `pickle` installed in your environment.

Next, let's import the module and dump the model into `.pkl` file.

2.3 TECHNIQUE USED OR ALGORITHM USED
PROPOSED TECHNIQUE:

WELFake Model

- In this the WELFake model for fake news detection divided into four phases, Data set preparation involves the collection and preprocessing of the data in a proper format, as a fundamental task of any ML model.
- Feature engineering involves linguistic feature extraction and selection.
- WE identifies the most appropriate technique connected with the LFS.
- Fake news detection tunes the model parameters and applies a hard voting classifier for better accuracy.

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

We sequentially explored different hyperparameter value combinations from the given possible value ranges and tuned them until we obtained a state-of-the-art accuracy of at least 96%. We evaluated the performance of each ML model on different training and testing data distributions as explained and found out that a 70%–30% data distribution gives better accuracy for all six ML methods.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shouls what the system do and not how it should be implemented.

- Processor - Pentium –IV
- Speed - 1.1 GHz
- Ram - 256 MB
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Back End : Python
- Operating System : Windows 7
- IDE : Spider3

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, and outputs. We sequentially explored different hyperparameter value combinations from the given possible value ranges and tuned them until we obtained a state-of-the-art accuracy of at least 96%. We also analyzed the performance of different ML models in terms of accuracy, precision, recall, and F1-score, and found out that SVM produced the most accurate result

3.5 NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows

- **Usability**

The system is designed with completely automated process hence there is no or less user intervention.

- **Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

- **Performance**

This system is developing in the high-level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

- **Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having Python, built into the system.

- **Implementation**

The system is implemented in web environment using struts framework. The apache tomcat is used as the web server and windows xp professional is used as the platform. Interface the user interface is based on Struts provides HTML Tag.

CHAPTER 4

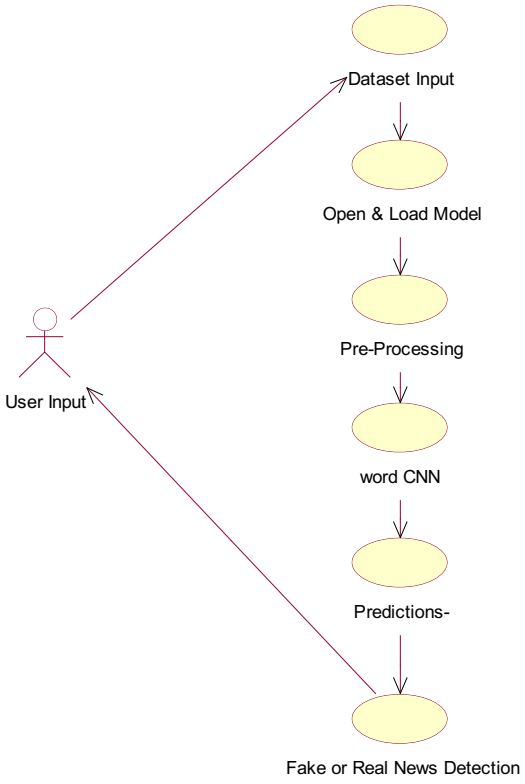
DESIGN ENGINEERING

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

4.2 UML DIAGRAMS

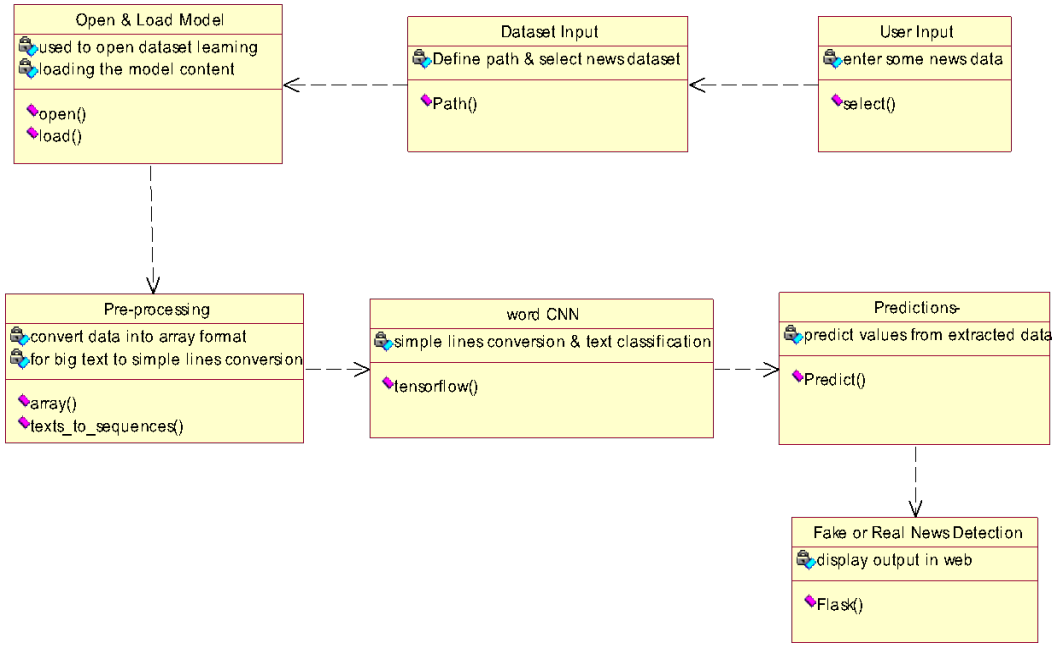
4.2.1 USE CASE DIAGRAM



EXPLANATION:

A use case diagram in the Unified Modeling Language (UML) user has a login. After login it has a user dataset. It can upload a data. It have a preprocessing a data. It will apply a algorithm and it has a predictions of values and then it display output.

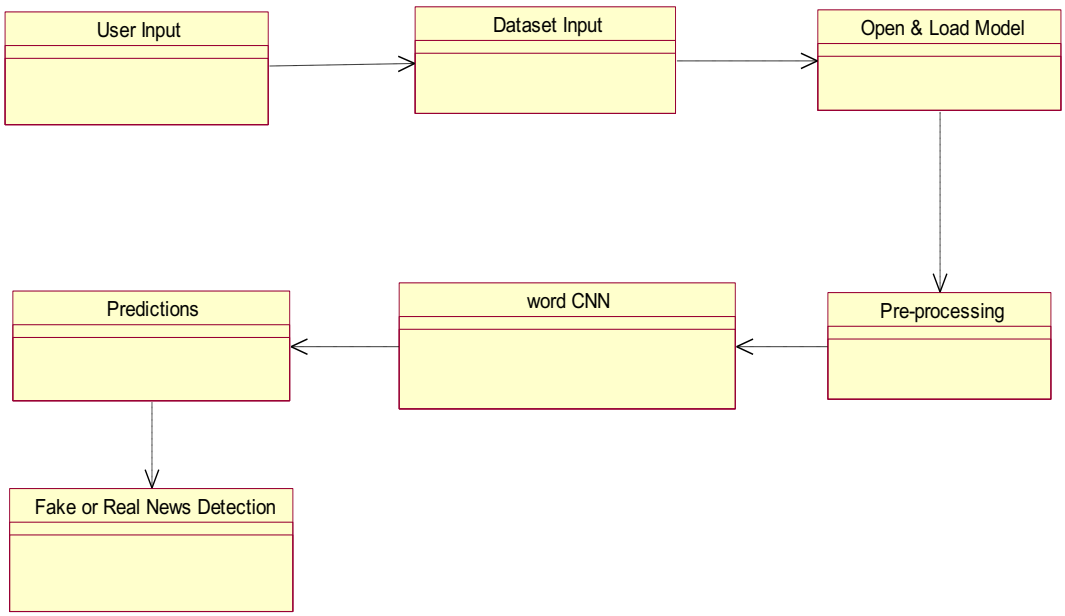
4.2.2 CLASS DIAGRAM



EXPLANATION

In software engineering, a class diagram in the Unified Modelling Language (UML) User has a login with a user id and password. User dataset has a fake job data. It has a pre-processing can remove a unwanted data. It has apply a algorithm CNN, decision tree and other classifier. It has a prediction of data. It detect a fake news.

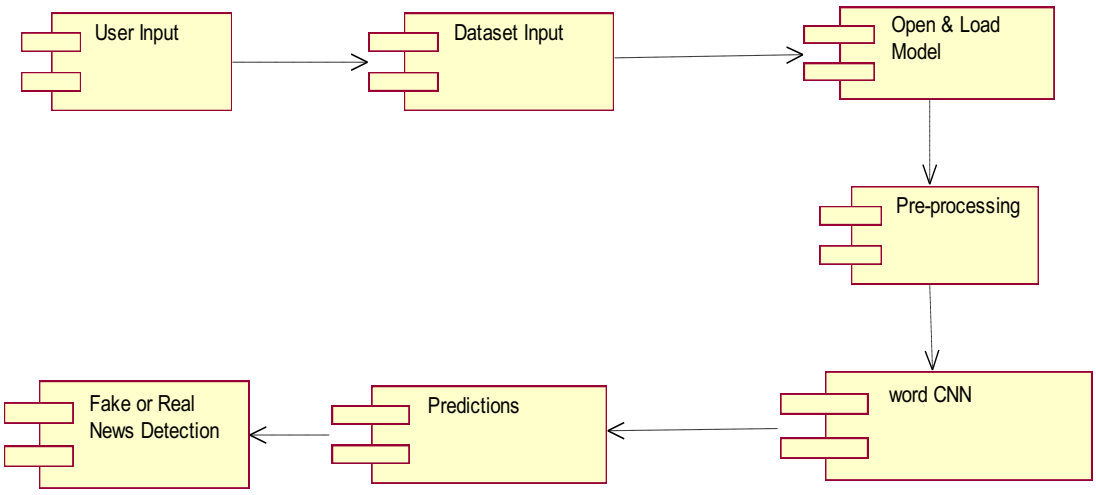
4.2.3 OBJECT DIAGRAM



EXPLANATION:

In the above digram tells about the flow of objects between the classes. It also a User input data to link with a dataset.It also have a preprocessing data to the algorithm.It have a predictions of the values.It also recognize a fake news and it display a output.

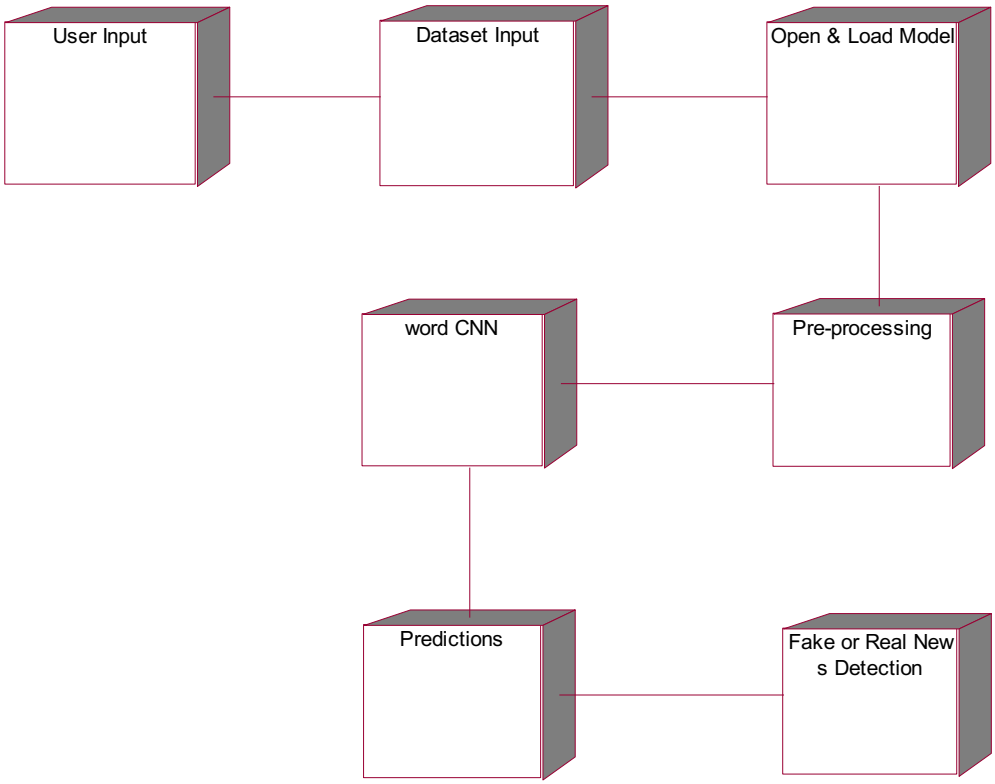
4.2.4 COMPONENT DIAGRAM



EXPLANATION

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

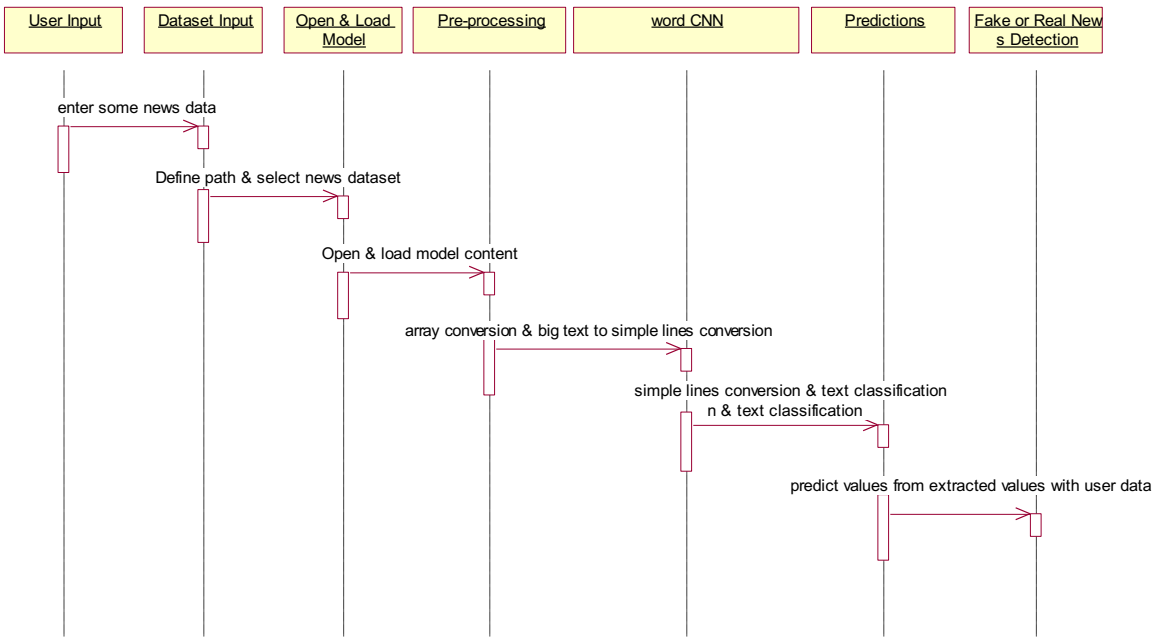
4.2.5 DEPLOYMENT DIAGRAM



EXPLANATION:

Deployment diagrams is a kind of structure diagram used in the user has a link with a user dataset it was also have a pre-processing a data and it has a training and testing data from the dataset. It has a predictions to predict a values and it has a detect a fake news data.

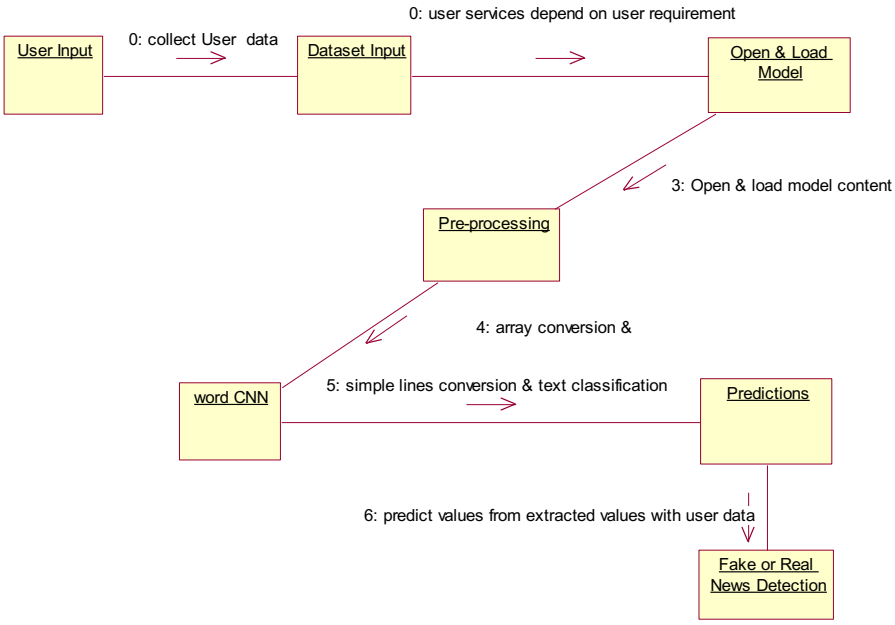
4.2.6 SEQUENCE DIAGRAM



EXPLANATION:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a user has link with dataset it sends a message to dataset. From the dataset it has a recognized a fake news. It also have a pre-processing has removes unnecessary data. Classifier has a algorithm performed in CNN, decision trees. It has a prediction of a fake news.

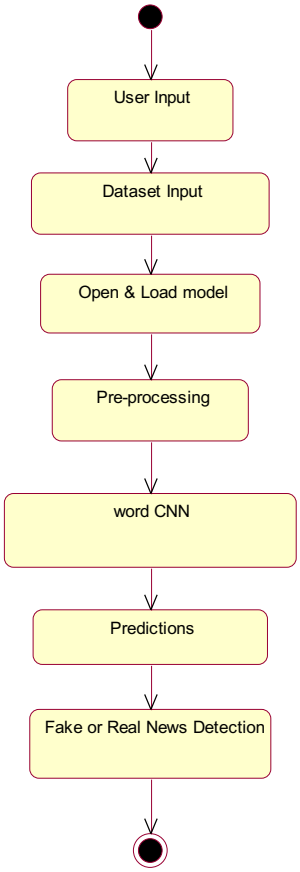
4.2.7 COLLABORATION DIAGRAM



EXPLANATION:

A collaboration diagram, also called a communication diagram user has a login with a data. Dataset has it detects a fake news data. Classifier it has a apply algorithms in a CNN, decision trees and links to the predictions. Pre-processing it also have a removes a unwanted data from predictions to predict values.

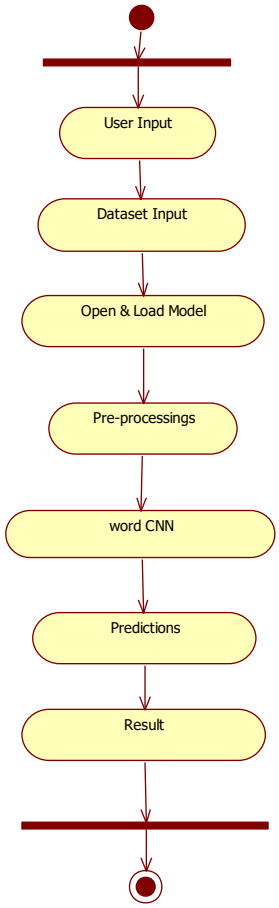
4.2.8 STATE DIAGRAM



EXPLANATION:

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system it describes a It has a user has to login. It was a user dataset. It also has a pre-processing a data. It has a testing and training a data to the dataset. It applies a algorithm in CNN, decision trees and other. it has also had a prediction to detect a fake news.

4.2.9 ACTIVITY DIAGRAM

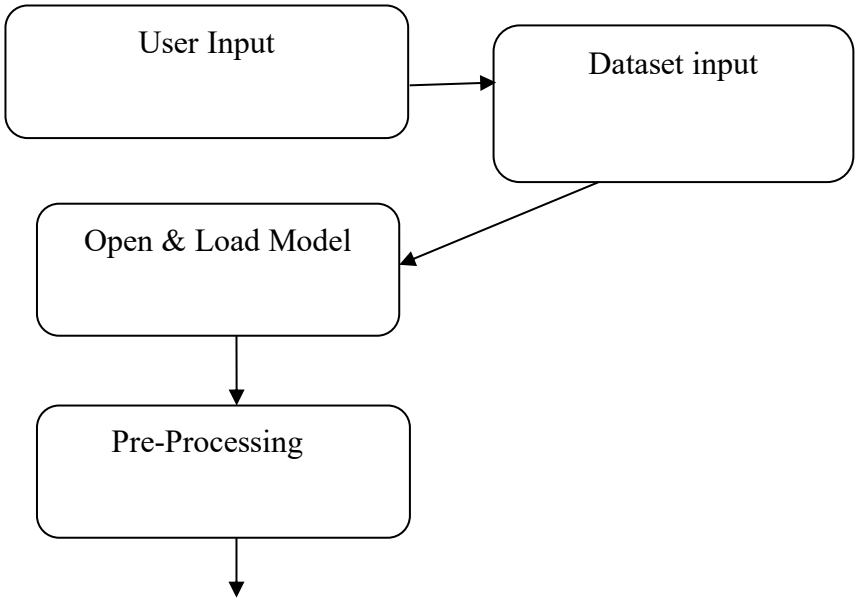


EXPLANATION:

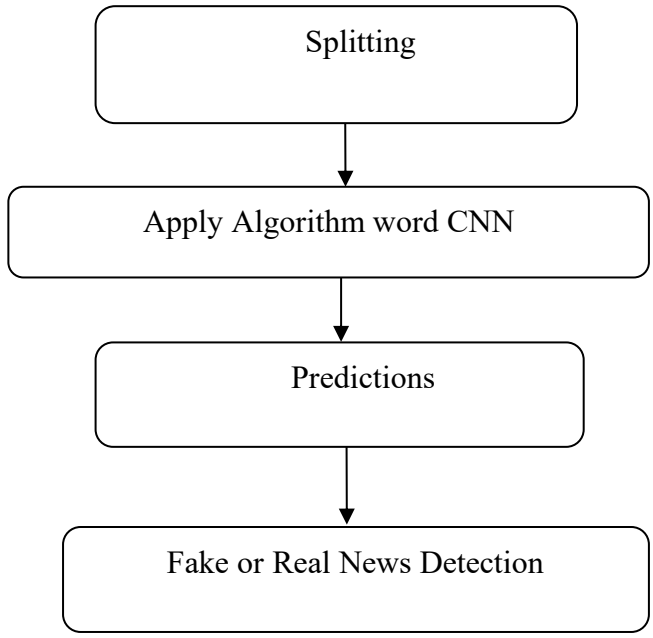
Activity diagrams are graphical shows a step-by-step manner operation. It has a user has to login. It was a user dataset. It also has a pre-processing a data. It has a testing and training a data to the dataset. It applies a algorithm in CNN, decision trees and other. it has also had a prediction to detect a fake news.

4.2.10 DATA FLOW DIAGRAM:

Level-0:



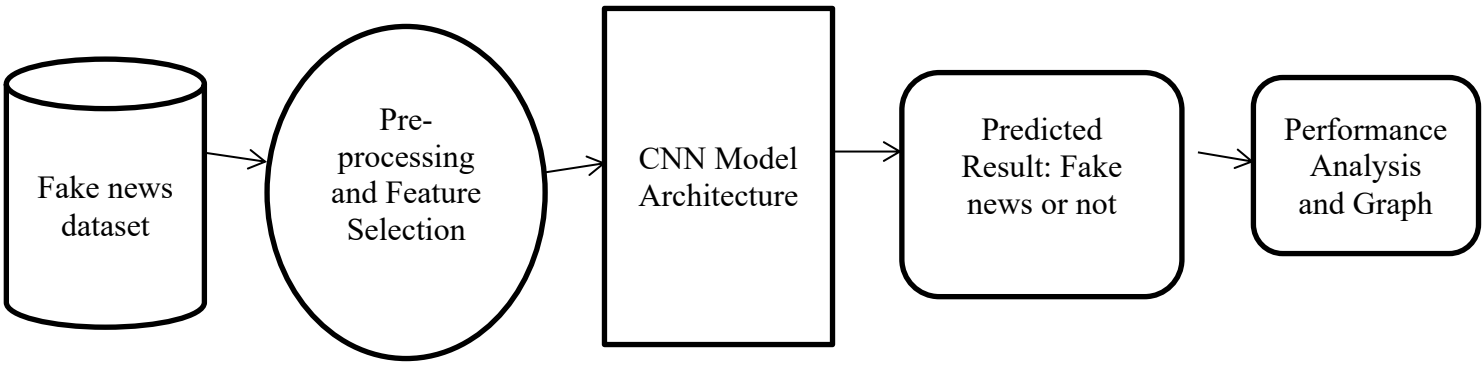
Level-1:



EXPLANATION:

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

4.3 SYSTEM ARCHITECTURE



CHAPTER 5
DEVELOPMENT TOOLS

Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Importance of Python

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Features of Python

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.

- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Libraries used in python:

- numpy - mainly useful for its N-dimensional array objects.
- pandas - Python data analysis library, including structures such as dataframes.
- matplotlib - 2D plotting library producing publication quality figures.
- scikit-learn - the machine learning algorithms used for data analysis and data mining tasks.

Figure : NumPy, Pandas, Matplotlib, Scikit-learn

CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

CODING:

```
from flask import Flask, render_template, request, url_for, Markup, jsonify

import pickle

import pandas as pd

import numpy as np

import pandas as pd

import numpy as np

import sys

import os

import glob

import re

import numpy as np

import tensorflow as tf

import tensorflow as tf



from tensorflow.compat.v1 import ConfigProto

from tensorflow.compat.v1 import InteractiveSession
```

```
config = ConfigProto()

config.gpu_options.per_process_gpu_memory_fraction = 0.2

config.gpu_options.allow_growth = True

session = InteractiveSession(config=config)

# Keras

from tensorflow.keras.applications.resnet50 import preprocess_input

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

import numpy as np

import pandas as pd

import pickle

from keras.preprocessing.text import Tokenizer

from keras.preprocessing.sequence import pad_sequences

import keras.models

from keras.models import model_from_json



# Flask utils

from flask import Flask, redirect, url_for, request, render_template

from werkzeug.utils import secure_filename

from keras.preprocessing.text import Tokenizer
```

```
from keras.preprocessing import sequence

from keras.models import Model, Input, Sequential, load_model

import pickle

import h5py

# create Flask application

app = Flask(__name__)

# read object TfidfVectorizer and model from disk

MODEL_PATH ='cnn.h5'

model = load_model(MODEL_PATH)

with open('tokenizer.pickle', 'rb') as handle:

    tokenizer = pickle.load(handle)

@app.route('/')

@app.route('/first')

def first():

                                return render_template('first.html')

@app.route('/login')

def login():

                                return render_template('login.html')

@app.route('/upload')

def upload():
```

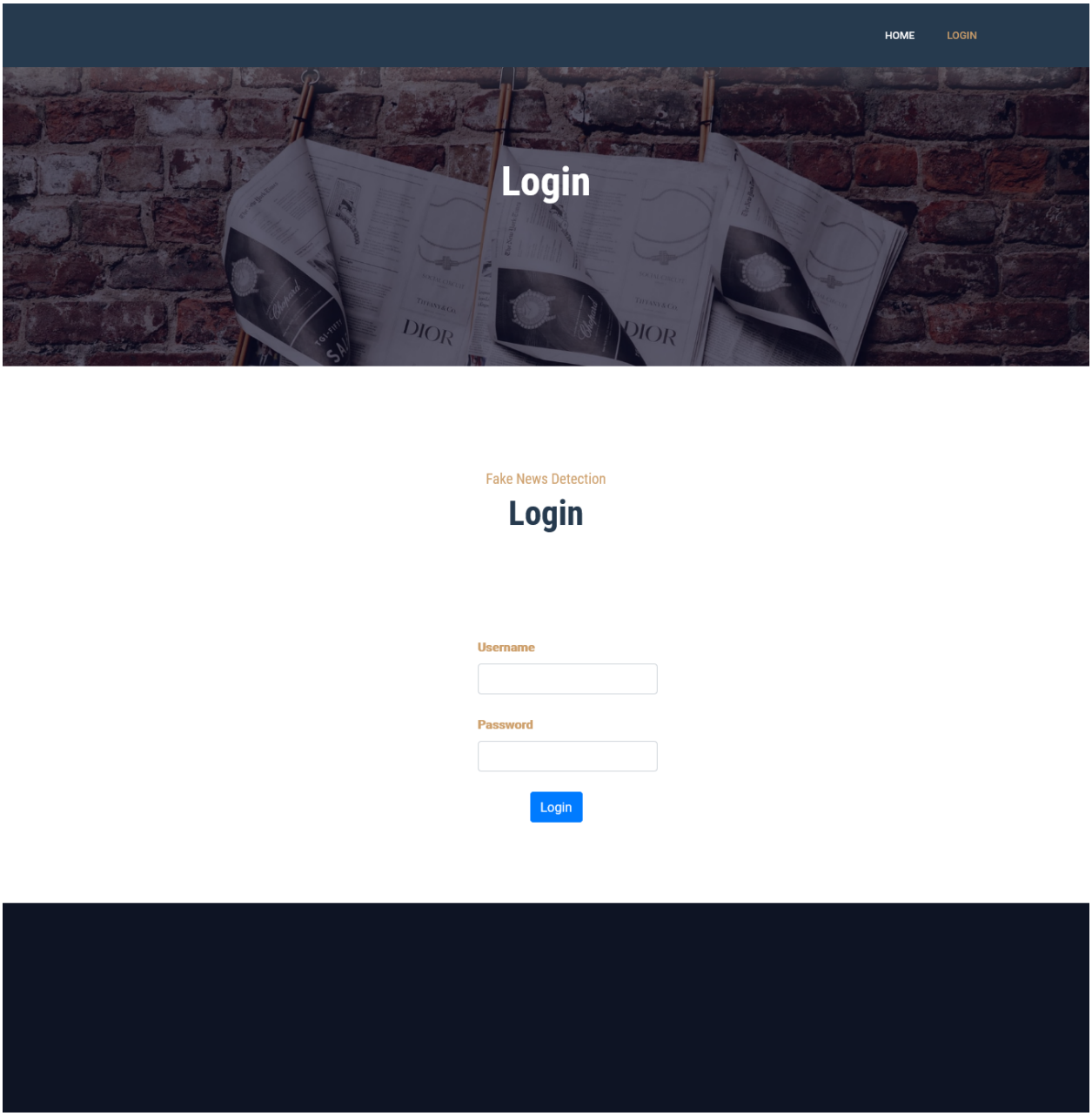
```
                return render_template('upload.html')

@app.route('/preview',methods=["POST"])

def preview():

    if request.method == 'POST':

        dataset = request.files['datasetfile']

        df = pd.read_csv(dataset,encoding = 'unicode_escape')

        df.set_index('Id', inplace=True)

        return render_template("preview.html",df_view = df)

@app.route('/home')

def home():

    return render_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

    error = None

    if request.method == 'POST':

        # message

        msg = request.form['message']

        msg = pd.DataFrame(index=[0], data=msg, columns=['data'])

        # transform data
```

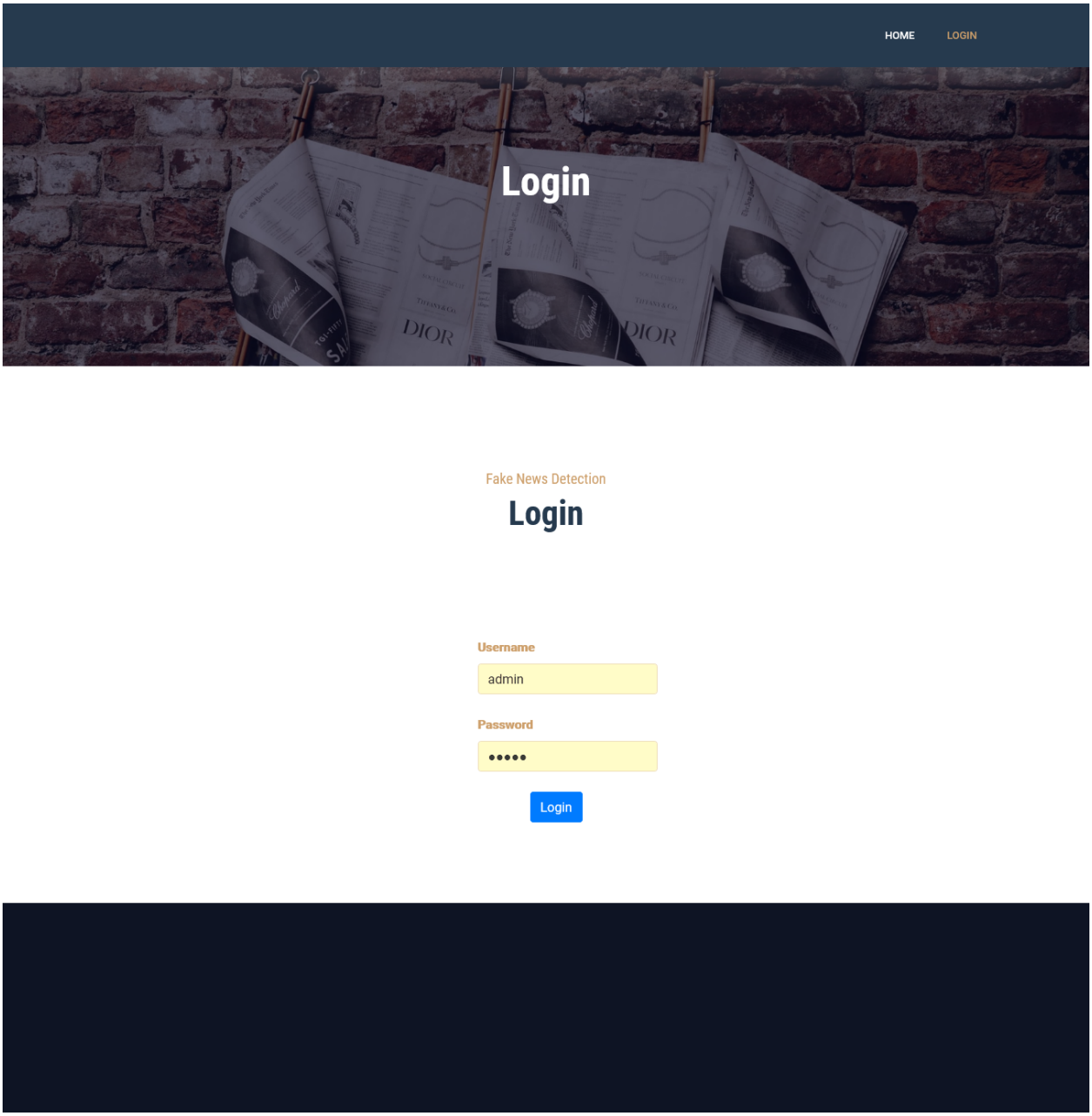
```
        new_text                                                                    =
sequence.pad_sequences((tokenizer.texts_to_sequences(msg['data'].astype('U'))),
maxlen=547)

            # model

        result = model.predict(new_text,batch_size=1,verbose=2)

        if result >0.5:

            result = 'Fake'

        else:

            result = 'Real'

        return render_template('index.html', prediction_value=result)

    else:

        error = "Invalid message"

        return render_template('index.html', error=error)

@app.route('/chart')

def chart():

                                return render_template('chart.html')

if __name__ == "__main__":

    app.run(debug=True)
```

CHAPTER 7

SNAPSHOTS

7.1 SNAPSHOTS


HOME
LOGIN
Login
DIOR
DIOR
Fake News Detection
Login
Username
Password
Login


HOME
LOGIN
Login
Fake News Detection
Login
Username
admin
Password
Login

Fake News Detection

Upload

Browse... upload.csv

Upload

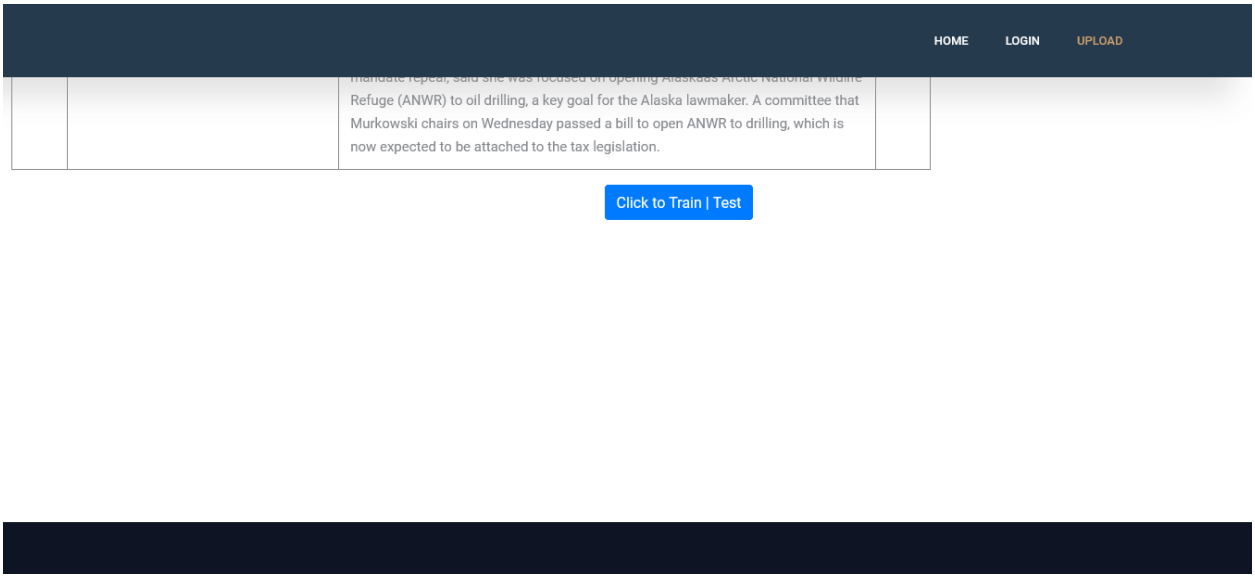
Fake News Detection

HOME
LOGIN
UPLOAD

Fake News Detection

Preview

	title	text	label
Id			
0	LAW ENFORCEMENT ON HIGH ALERT Following Threats Against Cops And Whites On 9-11By #BlackLivesMatter And #FYF911 Terrorists [VIDEO]	No comment is expected from Barack Obama Members of the #FYF911 or #FukYoFlag and #BlackLivesMatter movements called for the lynching and hanging of white people and cops. They encouraged others on a radio show Tuesday night to turn the tide and kill white people and cops to send a message about the killing	1


HOME
LOGIN
UPLOAD
mandate repeal, said she was focused on opening Alaska's Arctic National Wildlife Refuge (ANWR) to oil drilling, a key goal for the Alaska lawmaker. A committee that Murkowski chairs on Wednesday passed a bill to open ANWR to drilling, which is now expected to be attached to the tax legislation.
Click to Train | Test

Fake News Detection

Prediction

Enter your news

submit

News is:

Fake News Detection

Prediction

BRUSSELS (Reuters) - British Prime Minister Theresa May s offer of settled status for EU residents is flawed and will leave them with fewer rights after Brexit, the European Parliament s Brexit coordinator said on Tuesday. A family of five could face a bill of 360 pounds to acquire the new status, Guy Verhofstadt told May s Brexit Secretary David Davis in a letter seen by Reuters a very significant amount for a family on low income . Listing three other concerns for the EU legislature, which must approve any treaty on the March

submit

News is:

Fake News Detection

Prediction

Enter your news

submit

News is:Real

Fake News Detection

Prediction

21st Century Wire says Amid the tossing and turning of media hit pieces and partisan mud slinging in advance of the US presidential vote in November, very little focus is given on the actual record and policies of Hillary Clinton as Secretary of State.With this week s damning revelations regarding Gulf Arab monarchs buying access into the US State Dept via the Clinton Foundation, a clearer picture is now emerging about how the sponsorship of religious extremism, as well as geopolitical instability in the Middle East and beyond links

submit

News is:

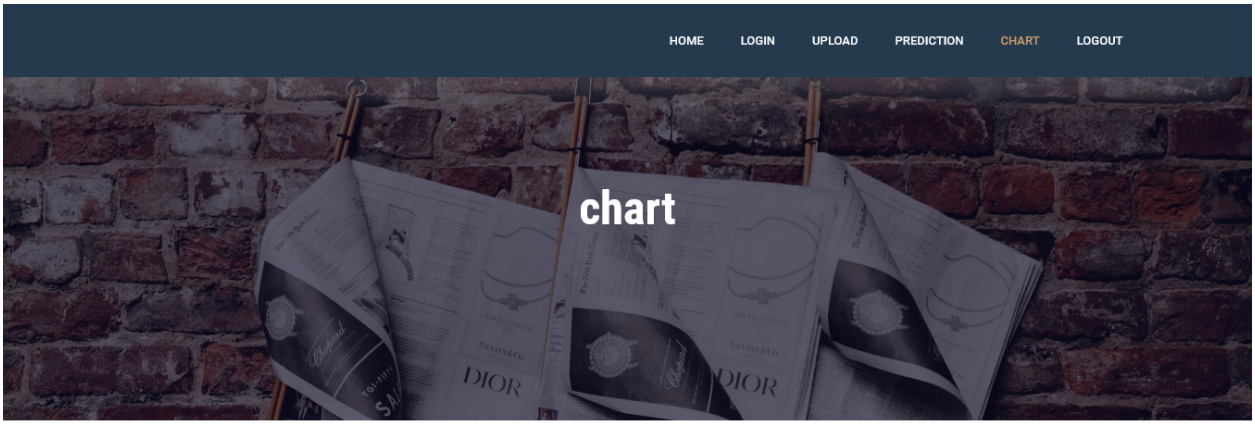
Fake News Detection

Prediction

Enter your news

submit

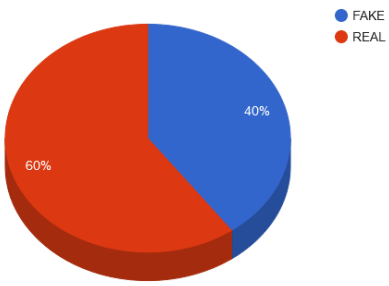
News is:Fake



Fake News Detection

chart

Fake News Detection (pie chart analysis)



CHAPTER 8
SOFTWARE TESTING

8.1 GENERAL

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3TYPES OF TESTS

8.3.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3.2 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

8.3.3 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 PERFORMANCE TEST

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

ACCEPTANCE TESTING FOR DATA SYNCHRONIZATION:

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

8.2.7 BUILD THE TEST PLAN

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 9
APPLICATION

9.1 FUTURE ENHANCEMENT

We plan to extend our work in the future with other factors like knowledge graphs and user credibility for further verification of the output generated by the WELFake model.

CHAPTER 10
CONCLUSION AND REFERENCES

10.1 CONCLUSION

We presented a new model called WELFake for text fake news detection. For this purpose, we prepared a larger data set called WELFake with over news articles combining four open-source data sets (i.e., Kaggle, McIntire, Reuters, and BuzzFeed) to reduce their individual limitation and bias. Afterward, we analyzed inguistic features from state-of-the- art works and increase the standard classifiers' accuracy. We applied two WE-based methods (i.e., TF-IDF, CV) over these linguistic features using CNN Model Architecture. We finally applied the result of this CNN Model Architecture to the next level with the best model results of TF-IDF and CV over LFS and obtained the final classification. Experimental results show that the WELFake model produces a high 93% accuracy on the WELFake data set. We also analyzed the performance of CNN Model Architecture in terms of accuracy, precision, recall, and F1-score, and found out that CNN Model Architecture produced the most accurate results.

10.2 REFERENCES

[1] W. Jiang, J. Wu, F. Li, G. Wang, and H. Zheng, “Trust evaluation in online social networks using generalized network flow,” IEEE Trans. Comput., vol. 65, no. 3, pp. 952–963, Mar. 2016.

[2] M. Alrubaian, M. Al-Qurishi, A. Alamri, M. Al-Rakhami, M. M. Hassan, and G. Fortino, “Credibility in online social networks: A survey,” IEEE Access, vol. 7, pp. 2828–2855, 2019.

[3] S. Ranganath, S. Wang, X. Hu, J. Tang, and H. Liu, “Facilitating time critical information seeking in social media,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 10, pp. 2197–2209, Oct. 2017.

[4] Z. Zhang, R. Sun, X. Wang, and C. Zhao, “A situational analytic method for user behavior pattern in multimedia social networks,” IEEE Trans. Big Data, vol. 5, no. 4, pp. 520–528, Dec. 2019.

[5] M. Schudson and B. Zelizer, “Fake news in context,” in Understanding and Addressing the Disinformation Ecosystem. Philadelphia, PA, USA: Annenberg School for Communication, Apr. 2017, pp. 1–4.

[6] S. Zaryan, “Truth and trust: How audiences are making sense of fake news,” M.S. thesis, Media Commun. Studies, Lund Univ. Publications Student Papers, Stockholm, Sweden, Jun. 2017. [Online]. Available: https://lup.lub.lu.se/student-papers/search/publication/8906886

[7] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” Science, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018.

[8] R. Sequeira, A. Gayen, N. Ganguly, S. K. Dandapat, and J. Chandra, “A large-scale study of the Twitter follower network to characterize the spread of prescription drug abuse tweets,” IEEE Trans. Comput. Social Syst., vol. 6, no. 6, pp. 1232–1244, Dec. 2019.

[9] Politifact News Dataset. Accessed: Mar. 31, 2020. [Online]. Available: http://www.politifact.com/

[10] The Washingtonpost Fact Checker. Accessed: Mar. 31, 2020. [Online]. Available: https://www.washingtonpost.com/news/fact-checker

[11] Fact Check. Accessed: Mar. 31, 2020. [Online]. Available: https://www.factcheck.org/

[12] Snopes. Accessed: Mar. 31, 2020. [Online]. Available: https://www.snopes.com/

[13] Truthorfiction. Accessed: Mar. 31, 2020. [Online]. Available: https://www.truthorfiction.com/

[14] Fullfact. Accessed: Mar. 31, 2020. [Online]. Available: https://fullfact.org/

[15] Hoax Slayer. Accessed: Mar. 31, 2020. [Online]. Available: http://hoaxslayer.com/
[16] Viswas News. Accessed: Mar. 31, 2020. [Online]. Available: http://www. vishvasnews.com/
[17] Factly. Accessed: Mar. 31, 2020. [Online]. Available: https://factly.in/
[18] L.-L. Shi et al., "Human-centric cyber social computing model for hotevent detection and propagation," IEEE Trans. Comput. Social Syst., vol. 6, no. 5, pp. 1042–1050, Oct. 2019.
[19] M. Glenski, T. Weninger, and S. Volkova, "Propagation from deceptive news sources who shares, how much, how evenly, and how quickly?" IEEE Trans. Comput. Social Syst., vol. 5, no. 4, pp. 1071–1082, Dec. 2018.
[20] E. Lancaster, T. Chakraborty, and V. S. Subrahmanian, "MALT P : Parallel prediction of malicious tweets," IEEE Trans. Comput. Social Syst., vol. 5, no. 4, pp. 1096–1108, Dec. 2018.
[21] P. K. Verma and P. Agrawal, "Study and detection of fake news: P2C2-based machine learning approach," in Proc. Int. Conf. Data Manage., Anal. Innov., vol. 1175. Singapore: Springer, Sep. 2020, pp. 261–278.
[22] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," IEEE Trans. Multimedia, vol. 19, no. 3, pp. 598–608, Mar. 2017.
[23] B. Ratner, "The correlation coefficient: Its values range between +1/− 1, or do they?" J. Targeting, Meas. Anal. Marketing, vol. 17, no. 2, pp. 139–142, Jun. 2009.
[24] A. De Salve, P. Mori, B. Guidi, and L. Ricci, "An analysis of the internal organization of Facebook groups," IEEE Trans. Comput. Social Syst., vol. 6, no. 6, pp. 1245–1256, Dec. 2019.
[25] P. K. Verma, P. Agrawal, and R. Prodan, WELFake Dataset for Fake News Detection in Text Data (Version: 0.1) [Data Set]. Genéve, Switzerland: Zenodo, 2021.
[26] V. Madaan and A. Goyal, "Predicting ayurveda-based constituent balancing in human body using machine learning methods," IEEE Access, vol. 8, pp. 65060–65070, 2020.
[27] M. Li, G. Clinton, Y. Miao, and F. Gao, "Short text classification via knowledge powered attention with similarity matrix based CNN," 2020, arXiv:2002.03350. [Online]. Available: http://arxiv.org/abs/2002.03350
[28] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification," in Proc. China Nat. Conf. Chin. Comput. Linguistics, vol. 11856. Cham, Switzerland: Springer, Feb. 2020, pp. 194–206. [Online]. Available: https://arxiv.org/abs/1905.05583.

[29] K. Dzmitry Bahdanau, Y. Cho, and Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. 3rd Int. Conf. Learn. Represent., 2015, pp. 1–15. [Online]. Available: https://arxiv.org/abs/1409.0473.

[30] Y. Chen, N. J. Conroy, and V. L. Rubin, “Misleading online content: Recognizing clickbait as ‘false news,”’ in Proc. ACM Workshop Multimodal Deception Detection, Nov. 2015, pp. 15–19.

[31] P. Bourgonje, J. Moreno Schneider, and G. Rehm, “From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles,” in Proc. EMNLP Workshop, Natural Lang. Process. Meets Journalism, 2017, pp. 84–89.

[32] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, “Truth of varying shades: Analyzing language in fake news and political factchecking,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 2931–2937.

[33] M. Alrubaian, M. Al-Qurishi, M. Mehedi Hassan, and A. Alamri, “A credibility analysis system for assessing information on Twitter,” IEEE Trans. Depend. Sec. Comput., vol. 15, no. 4, pp. 661–674, Aug. 2018.

[34] C. Castillo, M. Mendoza, and B. Poblete, “Information credibility on Twitter,” in Proc. 20th Int. Conf. World Wide Web (WWW), 2011, pp. 675–684.

[35] D. Benjamin, D. Horne, and S. Adali, “This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news,” in Proc. 2nd Int. Workshop News Public Opinion, Mar. 2017, pp. 1–9.

[36] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, “FNDNet—A deep convolutional neural network for fake news detection,” Cognit. Syst. Res., vol. 61, pp. 32–44, Jun. 2020.

[37] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, “Behind the cues: A benchmarking study for fake news detection,” Expert Syst. Appl., vol. 128, pp. 201–213, Aug. 2019.

[38] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, “dEFEND: Explainable fake news detection,” in Proc. KDD 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 395–405.

[39] R. Zellers et al., “Defending against neural fake news,” 2019, arXiv:1905.12616. [Online]. Available: http://arxiv.org/abs/1905.12616

[40] J. K. Burgoon, J. Blair, T. Qin, and J. Nunamaker, “Detecting deception through linguistic analysis,” in Proc. 1st NSF/NIJ Conf. Intell. Secur. Inform., Berlin, Germany: Springer, May 2003, pp. 91–101.

[41] M. D. Vicario, W. Quattrociocchi, A. Scala, and F. Zollo, “Polarization and fake news: Early warning of potential misinformation targets,” ACM Trans. Web, vol. 13, no. 2, pp. 1–22, Apr. 2019.

[42] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, “Automatic detection of fake news,” in Proc. 27th Int. Conf. Comput. Linguistics, Santa Fe, NM, USA: Association for Computational Linguistics, Aug. 2018, pp. 3391–3401.

[43] C. Buntain and J. Golbeck, “Automatically identifying fake news in popular Twitter threads,” in Proc. IEEE Int. Conf. Smart Cloud (Smart- Cloud), New York, NY, USA, USA, Nov. 2017, pp. 208–215.

[44] T. Mitra and E. Gilbert, “Credbank: A large-scale social media corpus with associated credibility annotations,” in Proc. 9th Int. AAAI Conf. Web Social Media. Apr. 2015, pp. 258–267.

[45] A. Zubiaga, G. W. S. Hoi, M. Liakata, and R. Procter, “PHEME dataset of rumours and non-rumours,” Univ. Warwick, Coventry, U.K., Oct. 2016. [Online]. Available: https://figshare.com/articles/dataset/PHEME_dataset_of_rumours_and_non-rumours/4010619

[46] M. L. Newman, J. W. Pennebaker, D. S. Berry, and J. M. Richards, “Lying words: Predicting deception from linguistic styles,” Personality Social Psychol. Bull., vol. 29, no. 5, pp. 665–675, May 2003.

[47] L. Zhou, J. K. Burgoon, J. F. Nunamaker, and D. Twitchell, “Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications,” Group Decis. Negotiation, vol. 13, no. 1, pp. 81–106, Jan. 2004.

[48] H. Ahmed, I. Traore, and S. Saad, “Detection of online fake news using N-gram analysis and machine learning techniques,” in Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, vol. 10618. Cham, Switzerland: Springer, Oct. 2017, pp. 127–138.

[49] K. Shu, S. Wang, and H. Liu, “Exploiting Tri-relationship for fake news detection,” Dec. 2018, arXiv:1712.07709v1. [Online]. Available: https://arxiv.org/abs/1712.07709v1.

[50] C. Burfoot and T. Baldwin, “Automatic satire detection: Are you having a laugh?” in Proc. ACL-IJCNLP Conf. Short Papers ACL-IJCNLP, 2009, pp. 161–164.

[51] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, “A simple but tough-to-beat baseline for the fake news challenge stance detection task,” 2017, arXiv:1707.03264. [Online]. Available: http://arxiv.org/abs/1707.03264.

[52] W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Short Papers), vol. 2, 2017, pp. 422–426.

[53] Mcintire Fake News Dataset. Accessed: Apr. 15, 2020. [Online]. Available: https://github.com/lutzhamel/fake-news.

[54] Fake News Kaggle Dataset. Accessed: Apr. 15, 2020. [Online]. Available: https://www.kaggle.com/c/fake-news/data?select=train.csv.

[55] Benjamin Political News Dataset. Accessed: May 15, 2020. [Online]. Available: https://github.com/rpitrust/fakenewsdata1.

[56] Burfoot Satire News Dataset. Accessed: May 15, 2020. [Online]. Available: http://www.csse.unimelb.edu.au/research/lt/ resources/satire.

[57] Buzzfeed News Dataset. Accessed: May 15, 2020. [Online]. Available: https://github.com/BuzzFeedNews/2016-10-facebook-fact-check/ tree/master/data.

[58] Credbank Dataset. Accessed: May 15, 2020. [Online]. Available: http://compsocial.github.io/CREDBANK-data.

[59] Fake News Challenge Dataset. Accessed: May 15, 2020. [Online]. Available: https://github.com/FakeNewsChallenge/fnc-1.

[60] Fakenewsnet Dataset. Accessed: May 15, 2020. [Online]. Available: https://github.com/KaiDMML/FakeNewsNet.

[61] Liar Dataset. Accessed: May 15, 2020. [Online]. Available: https://www. cs.ucsb.edu/~william/data/liar_dataset.zip.

[62] T. M. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill, 1997.

[63] G. Shmueli, N. R. Patel, and Bruce, Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel With XLMiner. Hoboken, NJ, USA: Wiley, 2007.