FAKE NEWS DETECTION IN SOCIAL MEDIA

By E. Shashi Kiran Reddy and Jayanthi Challa

Abstract:

The widespread dissemination of fake news on social media and various other platforms poses a significant concern due to its potential to cause social and national harm with severe consequences. Numerous research efforts have been dedicated to detecting fake news. This paper conducts an analysis of research related to fake news detection and explores traditional machine learning models to select the most suitable one. The aim is to develop a product model using a supervised machine learning algorithm capable of classifying fake news as true or false, utilizing tools such as Python's scikit-learn and NLP for textual analysis. This process involves feature extraction and vectorization. We propose using the Python scikit-learn library for text data tokenization and feature extraction, as it provides useful tools like CountVectorizer and TfidfVectorizer. Subsequently, we will employ feature selection methods to experiment and select the most appropriate features to achieve the highest precision, as indicated by confusion matrix results.

Introduction:

The project "Fake News Analysis in Social Media using IBM Watson" utilizes IBM Watson's powerful tools and technologies to combat the spread of fake news on social media platforms. By leveraging Watson's natural language processing (NLP) capabilities, the project analyzes news articles and social media posts to identify linguistic cues and patterns that indicate the presence of fake news. Machine learning models trained with labeled data help classify news articles as authentic or fake. Additionally, Watson's cognitive computing capabilities are employed to fact-check news content and cross-reference it with trusted sources. Social media analytics tools monitor the spread of news articles, enabling the project to identify trends and influencers associated with fake news dissemination. Interactive applications or chatbots powered by Watson engage with users, providing real-time feedback on the credibility of news articles. The project aims to empower users with information to make informed decisions and contribute to a healthier information ecosystem.

This paper proposes a methodology to create a model that will detect if an article is authentic or fake based on its words, phrases, sources and titles, by applying supervised machine learning algorithms on an annotated (labeled) dataset, that are manually classified and guaranteed. Then, feature selection methods are applied to experiment and choose the best fit features to obtain the highest precision, according to confusion matrix results. We propose to create the model using different classification algorithms. The product model will test the unseen data, the results will be plotted, and accordingly, the product will be a model that detects and classifies fake articles and can be used and integrated with any system for future use.

2. Related Work:

2.1 Social Media and Fake News

Social media encompasses websites and applications dedicated to forums, social networking sites, microblogging, social bookmarking, and wikis [1][2]. Conversely, some researchers attribute fake news to accidental incidents such as educational shocks or unintentional actions, as seen in the case of the Nepal Earthquake [3][4]. In 2020, there was widespread dissemination of fake news concerning health, which posed a significant risk to global health. In early February 2020, the World Health Organization (WHO) issued a warning about the COVID-19 outbreak causing a massive "infodemic," involving a mix of real and fake news, including a significant amount of misinformation.

2.2 Natural Language Processing

The primary purpose of employing Natural Language Processing (NLP) is to enable specialization within a system or algorithm. NLP ratings of algorithmic systems facilitate both speech understanding and speech generation, as well as the ability to detect actions across different languages. [6] proposed an ideal system for extracting actions from English, Italian, and Dutch languages by utilizing various language pipelines, such as Emotion Analyzer and Detection, Named Entity Recognition (NER), Parts of Speech (POS) Taggers, Chunking, and Semantic Role Labeling, which made NLP a significant subject of study [5][6].

Sentiment analysis [7] involves extracting emotions related to a particular subject. It includes identifying specific terms associated with a subject, extracting sentiments, and conducting connection analysis. Dual language resources, such as a Glossary of meaning and Sentiment models database, are used for sentiment analysis to classify words on a scale of -5 to 5 based on their constructive or destructive nature. Parts of speech taggers for European languages have been developed, and efforts are underway to create similar tools for languages like Sanskrit [8], Hindi [9], and Arabic. These taggers efficiently mark and categorize words as nouns, adjectives, verbs, etc. While most parts of speech techniques work effectively in European languages, they may not be as effective in Asian or Arabic languages. For instance, the Sanskrit word "speak" is specifically tagged using the tree-bank method, while Arabic utilizes Vector Machine (SVM) [10] to automatically identify symbols and parts of speech and reveal basic sentence structures in Arabic text [11].

2.3 Data Mining

Data mining techniques are typically categorized into two main methods: supervised and unsupervised. In supervised methods, training data is used to predict underlying patterns or activities. Unsupervised Data Mining, on the other hand, attempts to identify hidden data patterns without the use of labeled training data, such as pairs of input labels and categories. An example of unsupervised data mining is aggregate mines and syndicate bases [12].

2.4 Machine Learning (ML) Classification

Machine Learning (ML) is a set of algorithms that enable software systems to achieve more accurate results without the need for direct reprogramming. Data scientists identify patterns or features that the model needs to analyze and utilize to make predictions. Once training is complete, the algorithm applies the learned patterns to new data. In this paper, six algorithms are used for classifying fake news.

2.5 Decision Tree

The decision tree is a crucial tool used primarily for classification problems, working on a flowchart-like structure. Each internal node of the decision tree represents a condition or "test" on an attribute, and branching occurs based on these test conditions and results. Eventually, the leaf node holds a class label determined after evaluating all attributes. The distance from the root to the leaf represents the classification rule. What's remarkable about decision trees is their ability to work with categorical and dependent variables. They excel in identifying the most important variables and effectively illustrate the relationship between variables. Additionally, they play a significant role in generating new variables and features, which aids in data exploration and efficiently predicting the target variable.

```
Decision Tree Pseudo-code
GenerateDecisionTree(Sample s, features F)
1. If stop _conditions(S,F) = true then a.
   leaf = create_Node()
   b.    Leaf.lable= classify(s)
   c.    Return leaf
2. root = create_Node()
3. root.testcondition = find_bestSplit(s,f)
4. v = { v I v a possible outcome of root.testconditions)
5. for each value v□ V:
6. sv: = {s □root.testcondition(s) = v and s□ S};
7. child = Tree_Growth(Sv ,F) ;
8. Grow child as a descent of roof and label the edge (root→child) as
   v
       Return root
```

Tree based learning algorithms are widely with predictive models using supervised learning methods to establish high accuracy. They are good in mapping non-linear relationships. They solve the classification or regression problems quite well and are also referred to as CART [13][14][15].

2.6 Random Forest

Random Forest are built on the concept of building many decision tree algorithms, after which the decision trees get a separate result. The results, which are predicted by large number of decision tree, are taken up by the random forest. To ensure a variation of the decision trees, the random forest randomly selects a subcategory of properties from each group [16][17]

The applicability of Random forest is best when used on uncorrelated decision trees. If applied on similar trees, the overall result will be more or less similar to a single decision tree. Uncorrelated decision trees can be obtained by bootstrapping and feature randomness.

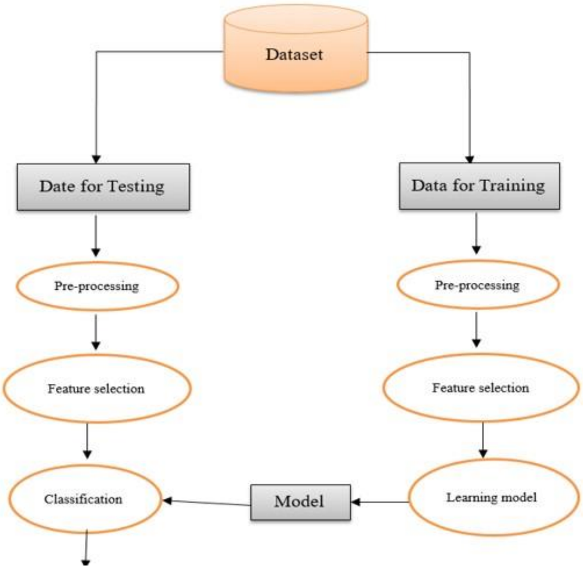
```
Random Forest Pseudo-code
To make n classifiers:
For i = 1 to n do

Sample the training data T randomly with replacement for Ti output

Build a Ti-containing root node, Ni

Call BuildTree
(Ni) end For
BuildTree (N):

If N includes instances of only one class, then returns
else
   Select z% of the possible splitting characteristics at random in N

   Select the feature F with the highest information gain to split on
   Create f child nodes of $N$, $Ni$ ,..., $Nf$ , where $F$ has $f$ possible values ( $F1$, ... , $Ff$ )
For $i$ = 1 to $f$ do
      Set the contents of $Ni$ to $Ti$, where $Ti$ is all instances in $N$ that match $Fi$
      Call Buildtree ($Ni$ )
end for          end if [18]
```

2.8. Naive Bayes

This algorithm operates on Bayes' theorem, assuming independence among predictors, making it applicable to various machine learning problems [18]. In simple terms, Naive Bayes assumes that one feature in a category is unrelated to another. For instance, a fruit will be classified as an apple if it's red, has swirls, and its diameter is close to 3 inches. Naive Bayes assumes that these features, whether dependent on each other or not, all contribute independently to the evidence for the apple class [14].

3. Methodology

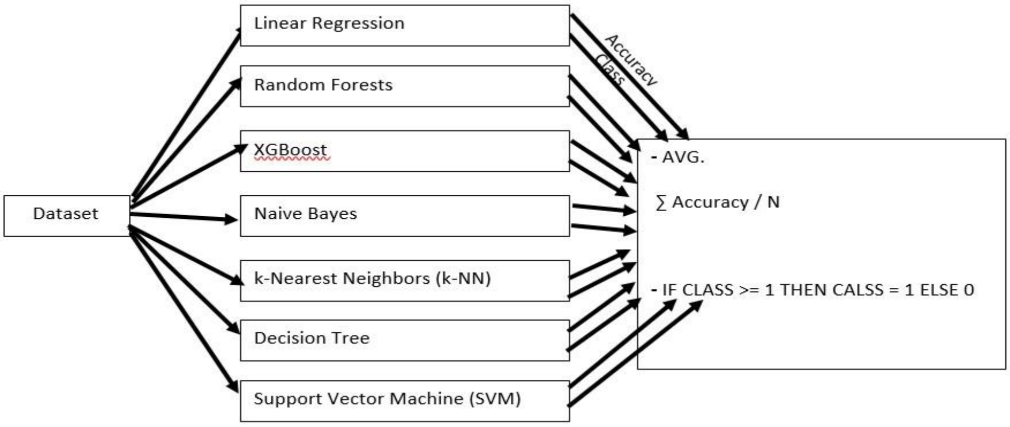
This section presents the methodology used for the classification. Using this model, a tool is implemented for detecting the fake articles. In this method supervised machine learning is used for classifying the dataset. The first step in this classification problem is dataset collection phase, followed by preprocessing, implementing features selection, then perform the training and testing of dataset and finally running the classifiers [35][36][37][38][39]. Figure [1] describes the proposed system methodology. The methodology is based on conducting various experiments on dataset using the algorithms described in the previous section named Random forest, SVM and Naïve Bayes, majority voting and other classifiers. The experiments are conducted individually on each algorithm, and on combination among them for the purpose of best accuracy and precision [40][41][42].



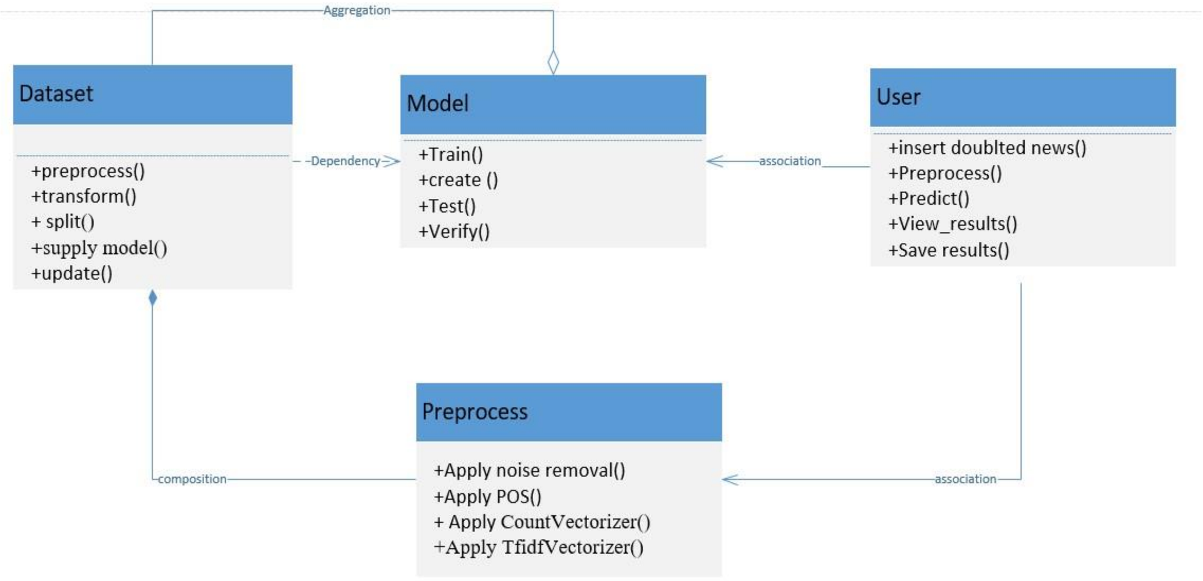
The main goal is to apply a set of classification algorithms to obtain a classification model in order to be used as a scanner for a fake news by details of news detection and embed the model in python application to be used as a discovery for the fake news data [43][44]. Also, appropriate refactorings have been performed on the Python code to produce an optimized code [25][26].

The classification algorithms applied in this model are k-Nearest Neighbors (k-NN), Linear Regression, XGBoost, Naive Bayes, Decision Tree, Random Forests and Support Vector Machine (SVM). All these algorithms get as accurate as possible. Where reliable from the combination of the average of them and compare them.

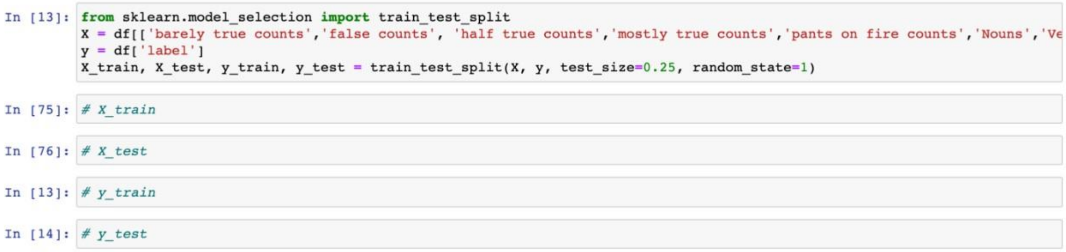
As shown in the figure [2], the dataset is applied to different algorithms in order to detect a fake news. The accuracy of the results obtained are analyzed to conclude the final result.



In the process of model creation, the approach to detecting political fake news is as follows: First step is collection political news dataset, (the Liar dataset is adopted for the model), perform preprocessing through rough noise removal, the next step is to apply the NLTK (Natural Language Toolkit) to perform POS and features are selected. Next perform the dataset splitting apply ML algorithms (Naïve bays and Random forest) then create the proposed classifier model. The Fig 2 shows that after the NLTK is applied, the Dataset gets successfully preprocessed in the system, then a message is generated for applying algorithms on trained portion. The system response with N.B and Random forest are applied, then the model is created with response message. Testing is performed on test dataset, and the results are verified, the next step is to monitor the precision for acceptance. The model is then applied on unseen data selected by user. Full dataset is created with half of the data being fake and half with real articles, thus making the model's reset accuracy 50%. Random selection of 80% data is done from the fake and real dataset to be used in our complete dataset and leave the remaining 20% to be used as a testing set when our model is complete. Text data requires preprocessing before applying classifier on it, so we will clean noise, using Stanford NLP (Natural language processing) for POS (Part of Speech) processing and tokenization of words, then we must encode the resulted data as integers and floating point values to be accepted as an input to ML algorithms. This process will result in feature extraction and vectorization; the research using python scikit-learn library to perform tokenization and feature extraction of text data, because this library contains useful tools like Count Vectorizer and Tiff Vectorizer. Data is viewed in graphical presentation with confusion matrix. Refer figure 3.



The data is divided it into two parts: The first section, which consists of **75%** of the data, is a trained data, where the algorithm detects the real news and false news ,then the data is labeled in the form of 0 and 1 where 0 is for false news and 1 for true news. After that, the rest of the data, which is **25%** of it, will do a test on it, so that it is sure whether the news is nature or forged, and then return it in case it was right or wrong, and according to the percentage of right and wrong, the algorithm percentage will be formed.



```
In [13]: from sklearn.model_selection import train_test_split
         X = df[['barely true counts','false counts', 'half true counts','mostly true counts','pants on fire counts','Nouns','Ve
         y = df['label']
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=1)

In [75]: # X_train

In [76]: # X_test

In [13]: # y_train

In [14]: # y_test
```

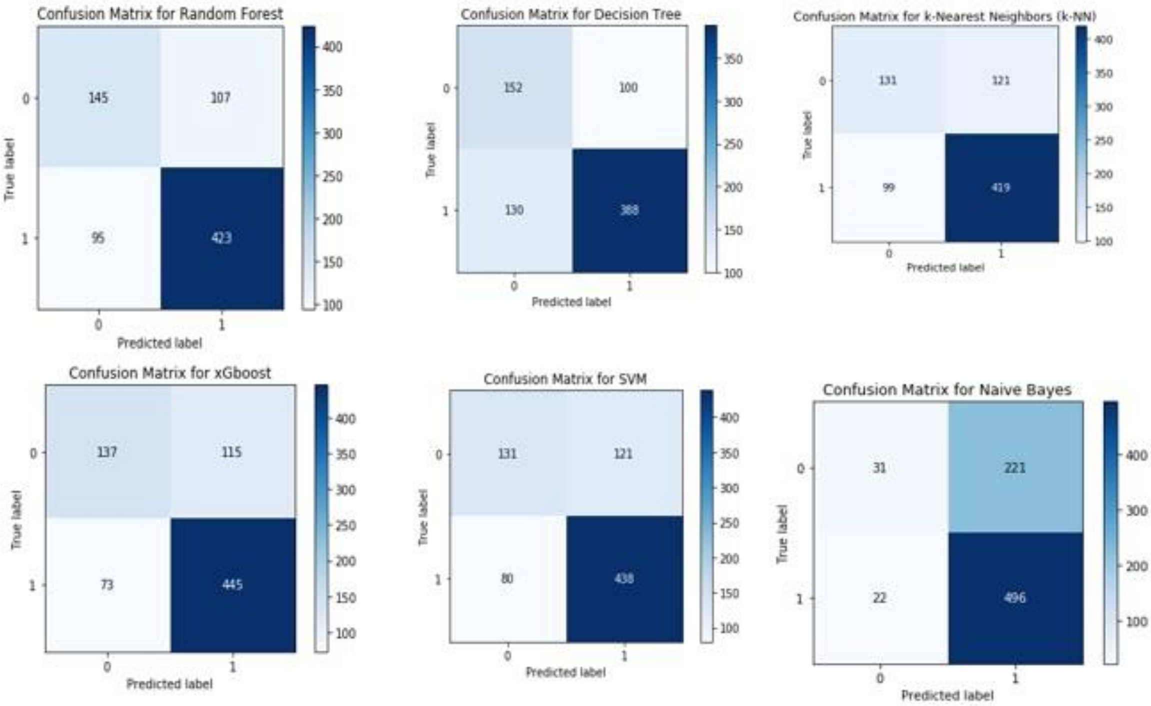
4. Results

The scope of this project is to cover the political news data, of a dataset known as Liar-dataset, it is a New Benchmark Dataset for Fake News Detection and labeled by fake or trust news. We have performed analysis on "Liar" dataset . The results of the analysis of the datasets using the six algorithms have been depicted using the confusion matrix. The six algorithms used for the detection are as:

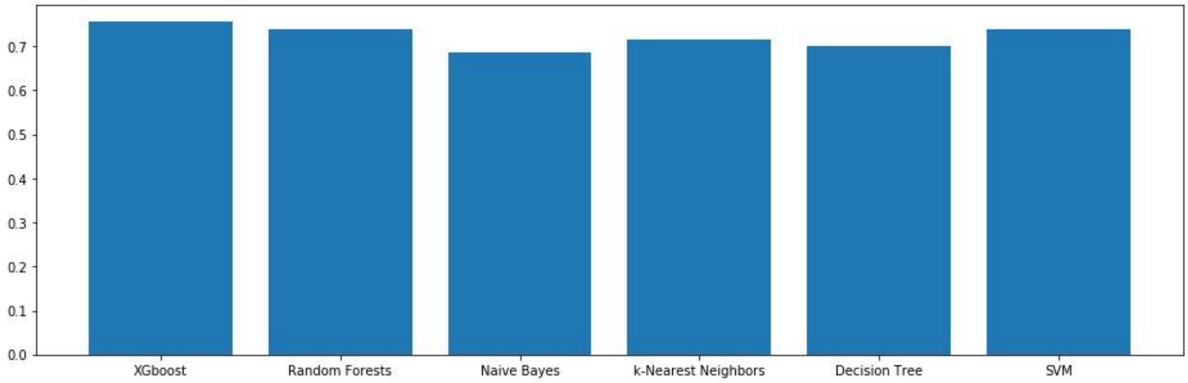
- XGboost.
- Random Forests.
- Naive Bayes.
- K-Nearest Neighbors (KNN).
- Decision Tree.
- SVM

The confusion matrix is automatically obtained by Python code using the cognitive learning library when running the algorithm code in Anaconda platform.

The Confusion Matrix for all the algorithms are depicted below in figure:



This Figure expresses the accuracies of these algorithms. As shown the XGBOOST is depicting the highest accuracy with more than 75%, next is SVM and Random forest with approximately 73% accuracy.



5. Conclusion

The research outlined in this paper focuses on detecting fake news through two stages: characterization and disclosure. In the characterization stage, the paper emphasizes the fundamental concepts and principles of fake news on social media. In the discovery stage, various methods for detecting fake news are reviewed, employing different supervised learning algorithms.

In reference to [20], the fake news detection approaches presented in the paper utilize text analysis models based on speech characteristics and predictive models that differ from current ones. [21] utilizes a Naive Bayes classifier to detect fake news from various sources, achieving an accuracy of 74%. [22] combines ML algorithms, but relies on an unreliable probability threshold with an accuracy ranging from 85% to 91%. [23] uses Naive Bayes to detect fake news from different social media platforms, but the results are inaccurate for untrustworthy sources. [24] obtains data from Kaggle with an average accuracy of 74.5%. [27] uses Naive Bayes algorithms to detect Twitter spam senders, with accuracy ranging from 70% to 71.2%. [28] explores different approaches with an accuracy of 76%. [29] employs three common methods: Naïve Bayes, Neural Network, and Support Vector Machine (SVM). Naïve Bayes achieves an accuracy of 96.08% for detecting fake messages, while Neural Network and SVM reach 99.90%. [30] combines KNN and random forests, resulting in an 8% improvement in fake message detection. [31] examines the performance of 8 supervised machine learning classifiers on the 2012 Dutch elections fake news on Twitter, concluding that the decision tree algorithm performs best with an F score of 88%. [32] presents a counterfeit detection model using N-gram analysis, achieving the highest accuracy of 92% with a unigram and linear SVM.

Based on the research summary and system analysis, it's evident that most papers utilize the Naïve Bayes algorithm, with prediction precision ranging from 70% to 76%. These papers predominantly rely on qualitative analysis, including sentiment analysis, titles, and word frequency repetition [40][41][42]. In our approach, we propose to augment these methodologies with another aspect: POS textual analysis, which is a quantitative approach. This involves adding numeric statistical values as features. We believe that increasing these features and utilizing random forests will further improve precision results. The features we propose to add in our dataset include total words (tokens), total unique words (types), type/token ratio (TTR), number of sentences, average sentence length (ASL), number of characters, average word length (AWL), and parts of speech such as nouns, prepositions, adjectives, etc.

6. References

[1]. Economic and Social Research Council. Using Social Mmedia. Available at: https://esrc.ukri.org/research/impact-toolkit/social-media/using-social-media

[2]. Gil, P. Available at: https://www.lifewire.com/what-exactly-is-twitter-2483331. 2019, April 22.

[3]. E. C. Tandoc Jr et al. “Defining fake news a typology of scholarly definitions”. *Digital Journalism* , 1–17. 2017.

[4]. J. Radianti et al. “An Overview of Public Concerns During the Recovery Period after a Major Earthquake: Nepal Twitter Analysis.” *HICSS '16 Proceedings of the 2016 49th Hawaii International Conference on System Sciences* (HICSS) (pp. 136-145). Washington, DC, USA : IEEE. 2016.

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

[6]. Jeonghee Yi et al. “Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. ”In Data Mining, 2003. ICDM 2003. *Third IEEE International Conference* (pp. 427-434). http://citeseerx.ist.psu.edu. 200).2003

[7]. Tapaswi et al. “Treebank based deep grammar acquisition and Part-Of-Speech Tagging for Sanskrit m sentences.” *Software Engineering (CONSEG), on Software Engineering* (CONSEG), (pp. 1-4). IEEE. 2012

[8]. Ranjan et al. “Part of speech tagging and local word grouping techniques for natural language parsing in Hindi”. *In Proceedings of the 1st International Conference on Natural Language Processing* (ICON 2003). Semanticscholar. 2003

[9]. MonaDiab et al. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. *Proceedings of HLT-NAACL 2004: Short Papers* (pp. 149–152). Boston, Massachusetts, USA: Association for Computational Linguistics. 2004

[10]. Rouse, M. https://searchenterpriseai.techtarget.com/definition/machine-learning-ML May 2018

[11]. Sumeet Dua, Xian Du. “Data Mining and Machine Learning in Cybersecurity”. *New York: Auerbach Publications.*19 April 2016.

[12]. RAY, S. https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/ 2017, September

[13]. Huang, T.-Q. (n.d.) https://www.researchgate.net/figure/Pseudo-code-of-information-gain-basedrecursive-feature-elimination-procedure-with-SVM_fig2_228366941 2018

[14]. Researchgate.net. Available at: Available at: https://www.researchgate.net/figure/Pseudocode-ofnaive-bayes-algorithm_fig2_325937073. 2018.

[15]. Researchgate.net. Available at: https://www.researchgate.net/figure/Pseudocode-for-KNNclassification_fig7_260397165, 2014.

[16]. Rampersad G, Althiyabi T 2020 “Fake news: Acceptance by demographics and culture on social media” *J. Inf. Technol. Politics* 2020, 17, 1–11.

[17]. NaphapornSirikulviriya; SukreeSinthupinyo. “Integration of Rules from a Random Forest.” *International Conference on Information and Electronics Engineering* (p. 194 : 198). Singapore: semanticscholar.org. 2011.

[18]. Jasmin Kevric et el. “An effective combining classifier approach using tree algorithms for network intrusion detection.” *Neural Computing and Applications* , 1051–1058. 2017.

[19]. ShivamB.Parikh and PradeepK.Atrey. “Media-RichFake News Detection: A Survey.” *IEEE Conference on Multimedia Information*. Miami, FL: IEEE. 2018.

[20]. MykhailoGranik and VolodymyrMesyura. “Fake news detection using naive Bayes classifier.” *First Ukraine Conference on Electrical and Computer Engineering* (UKRCON). Ukraine : IEEE. 2017.

[21]. Gilda, S. “Evaluating machine learning algorithms for fake news detection.” *15th Student Conference on Research and Development* (SCOReD) (pp. 110-115). IEEE. 2017.

[22]. Akshay Jain and AmeyKasbe. “Fake News Detection.” *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science* (SCEECS). Bhopal, India: IEEE. 2018.

[23]. Yumeng Qin et al. “Predicting Future Rumours.” *Chinese Journal of Electronics* (Volume: 27 , Issue: 3 , 5 2018, 514 - 520.