Identification of Fake News Using Machine Learning

Rahul R Mandical [1]
rahul.mohan28@gmail.com

Mamatha N [1]
mamathan539@gmail.com

Shivakumar N [1]
shivakumarnarayana00@gmail.com

Monica R [1]
monicaraghu.98@gmail.com

Krishna A N, Member IEEE [1]
ankrishna@sjbit.edu.in

[1]. *Department of Computer Science and Engineering*
SJB Institution of Technology
Bangalore, India

Abstract - **Fake news has been a problem ever since the internet boomed. The very network that allows us to know what is happening globally is the perfect breeding ground for malicious and fake news. Combating this fake news is important because the world's view is shaped by information. People not only make important decisions based on information but also form their own opinions. If this information is false it can have devastating consequences. Verifying each news one by one by a human being is completely unfeasible. This paper attempts to expedite the process of identification of fake news by proposing a system that can reliably classify fake news. Machine Learning algorithms such as Naive Bayes, Passive Aggressive Classifier and Deep Neural Networks have being used on eight different datasets acquired from various sources. The paper also includes the analysis and results of each model. The arduous task of detection of fake news can be made trivial with the usage of the right models with the right tools.**

Keywords - **Fake news Detection; Natural Language processing; Machine Learning; Naïve–Bayes; Passive Aggressive Classifier; Deep Neural Network;**

I. INTRODUCTION

Macmillan English Dictionary defines fake news as "a story that is presented as being a genuine item of news but is in fact not true and is intended to deceive people". But the definition isn't as clear cut in the 21st century. The water becomes murky as we start to question if rumors are fake news or if parodies or political humor is fake news. Can an exaggeration of a simple news article be considered fake as they may portray the subject in a different, if not in a negative view? While some of these questions have been covered extensively by **Ammara** *et. al*[1] and **Aswini** *Thota*[2], it is not on the agenda of this paper.

One can ask the question, why even attempt to combat fake news? The average internet user is not always knowledgeable about the news that they view. The chances of them spreading the said news increases exponentially when the news has an emotional attachment to themselves. This is most seen in the topics related to politics or religion. If left ignored, fake news can have disastrous consequences.

It is of paramount importance that we confront and try to eliminate the problem of fake news. There have been many methods that are proposed ranging from NLP analysis to clustering. ***Chaowei***[3] uses a system where trusted news is used to create clusters. Whereas, Dinesh Kumar et *al*[4]. takes a unique approach to classifying the news. It proposes an architecture where the text is extracted and cleaned from an image of news. This text is then fed to Google and the scrapped results are compared to determine if the news is true or fake. While these two papers look at the news, ***Jahankhani*** [5]analyses the website itself. It checks for three levels which include URL, Blacklist and Image screening. Classifying the website itself as fake can nip the bud of the fake news originating from the website. While this classifies the platform, *Costel*[6] proposes a system where the users in twitter can be given a credibility rating after analysis of the users' tweets. While this paper does not directly identify the fake news, it can mark the users who have the highest potential to create and spread fake news. Currently, there exist some systems which one can use to help in the identification of fake news. A plugin known as BS Detector searches through a catalog of web pages that has been flagged as unreliable or fake in their database. Politifact is another US-based website that gives credibility to the claims by US politicians.

While the above systems take various methods and perspectives, this paper is confined to dealing with fake news in both Machine Learning and NLP. We have acquired various datasets from different sources whose descriptions will be in present under the section **[III]**. We have chosen to use Multinomial Naive Bayes, Passive-aggressive classifier and deep neural network on the datasets. In-depth information on the methodology has been provided in the section **[IV]** and models generated have been analyzed and documented in the section **[V]**. The following section

978-1-7281-6828-9/20/$31.00 ©2020 IEEE

provides insight into some of the literature papers that also used machine learning and other similar methods to identify fake news.

II. LITERATURE SURVEY

In this section, we discuss some of the papers who used machine learning to identify and classify the fake news. ***Atik***[7] uses a distinctive technique in detecting fake news by creating an 'Ensemble Voting classifier' . It uses many well-known machine-learning classifiers such as Naïve Bayes, K-NN, SVM and many more to verify the news. Further, cross-validation was used and the top three machine learning algorithms with the best accuracy were used in Ensemble Voting Classifier. This model proposed a recognition structure that can productively predict the output and find the important highlights of the news. This allowed for a results ranging from early to late 90s. Text-mining based methods for the detection of fake news have been evaluated by **Harita Reddy et *al***[8]. This paper provided a hybrid approach that combines word vector representations and stylometric features using ensemble methods like bagging, boosting and voting. After the selection of important features, Random Forest, Naïve Bayes, SVM and many more algorithms were applied. This resulted in accuracies up to 95.49 %. Natural Language processing technique was exploited by **Kushal Agarwalla et *al***[9]**.** to verify the news. NLTK from Python was used with various models including Logistic Regression, SVM, and Naïve Bayes with Lidstone Smoothing. Naive bayes with Lidstone smoothing performed admirably and gave a result of 83% . Perhaps, using only the vector-based methods to extract certain features and to train the classifiers is not an accurate solution as these are fixed to the particular dataset.

Mykhailo Granik et *al*[10]. implemented a basic approach using Naive Bayes classifier for the detection of fake news. The model was built as a software system and validated over a set of Facebook news posts. It describes the similarity between spam messages and fake news articles by concluding that identical approaches can be taken for both fake news detection and spam filtering by producing a result of 75% accuracy**. Akshay Jain et *al*** [11]. proposed a model with two variants which uses Naive Bayes classifier to predict whether a post on Facebook will be labeled as REAL or FAKE. The first model used the title as their source for vocabulary building, using count vectorizer. And the second model used text as their source. The results were compared based on their AUC score and the second model was found to be better with a score of 0.93 and 0.912 with and without n_grams respectively.

Aswini Thota et *al*[2] used Deep Learning architectures to detect fake news. Tf-IDF, GloVe and Word2Vec were used along with the DNN model to precisely predict the stance between the article body and given pair of the headline.This paper was able to produce an overall accuracy of 94.31%. ***Samir***[12] explored different models ranging from Logistic Regression to CNN, RNN, and GRU. This work is mainly concentrated on using pure NLP perspective to identify the presence of fake news by utilizing the linguistic features. Highest precision of 0.97 was obtained using CNN with Max Pooling and Attention. This approach might lose its viability as the fake news gets better at replicating true news. **A.Lakshmanarao et *al***[13] . employed SVM, KNN, Decision tree, and Random forest to build a four models and compared them.It was observed that Random Forest Classification gave the highest score of 90.7% while least was provided by Support Vector Machines at 75.5%. **Shlok Gilda et *al***[14]**.** worked only using Natural Language Processing technique to identify the fake news. Probabilistic context-free grammar (PCFG) and Term frequency-inverse document frequency (TF-IDF) of bigrams were applied with various models like gradient boosting and stochastic gradient descent. Among other models, TFIDF of bi-grams with stochastic gradient descent identified fake news with higher accuracy.

While the previous papers applied machine learning to the detection of fake news, ***Veronica***[15] brings something new to the table in the form of human testing. This is the only paper that has tallied and compared the performance of humans against machines. This paper uses two different datasets namely FakeNewsAMT which consists of general news and Celebrity news which as the name suggests contains news about celebrities. The paper used two annotators to classify if the news were true or fake and they had an agreement rate of 70 percent. It is observed that the annotators beat the automated system when dealing with celebrity news dataset but lost by a margin of 3-4% in FakenewsAMT. Multi-feature extraction was also done and it showed that FakeNewsAMT performs best when relying on stylistic features and Celebrity on LIWC features.

While the previously mentioned papers have used a plethora of machine learning algorithms, This paper confines itself to three most important algorithms namely Naive Bayes, Passive Aggressive Classifier and Deep neural networks. The models are built in 8 different datasets whose description is present in the following section.

III. DATASETS

This section covers the description of seven different datasets that were used in this paper which were acquired from a diverse set of sources. Superset is the dataset obtained by extracting the Satement attribute from other various datasets used in this paper. All the datasets underwent similar pre-processing such as cleaning the dataset of corrupted data or dropping the missing value rows. All datasets were divided into train, dev and test sets for the models. If the number of articles in the dataset was less than 10000, the ratio of the split was taken as 90:5:5. Anything greater than 10000 was split in the ratio 80:10:10. All the datasets contain an almost equal number of true and false articles without being skewed towards one side.

Dataset	Attributes	Size
FND- jru[16]	URL+Headline+Body	4K*3
$Politifact$[17]	Statement+Speaker+URL	10K*3
$Pontes$[18] Route 1	Domain+Content+Title+ Author	93K*4
$Pontes$[18] Route 2	Domain+Content+Title	140K*3
$Claimskg$[19]	Text+Headline+Source+ Keywords	10K*4
$Kaggle$ [20] competition	Title+Author+Text	25K*3
$Liar$[21]	Statement+Sub+Speaker+Job+State +Party+Context	8.4K*7
$Newsfiles$[22]	Statement	18K
Superset	Statement	233K

Table 1. Dataset description

Attributes present in the Table Dataset description are elaborated as follows. Headline and Body refers to the header and main body of the article respectively. URL refers to the uniform resource locator a.k.a the address of the article. Statement and Content are the same and refer to the body of the news article. Similar to Headline, the Title also represents the same. Author is the person who wrote the news article while Domain refers to the category to which the news article belongs to. Source is synonymous with URL except for the format of the address. State and Party refers to what state the news article originates from and which political party it represents.

IV. METHODOLOGY AND IMPLEMENTATION

Machine Learning is one of the most powerful tools that are available right now. In this paper, we have thoroughly used ML to build our models. The task of choosing the classifiers emerged from the suitable properties of algorithms. As Naïve Bayes is popular for its multi-class prediction, it was picked up for its ease and robustness of predicting the class of the text. In fact, one of the problem with other methods is when new samples are collected, model must be retrained to predict the output for new data. This is overcome by using passive aggressive classifier which trains the model incrementally, allowing modifications of the parameters only when needed, while discarding these updates when they don't alter the equilibrium. We focused on the problem based on both conventional and deep learning architecture. Deep neural network was used to increase the efficiency in identification of fake news. Following paragraphs dives deeper into each algorithm.

Naïve Bayes classifier assumes that features are statistically independent of one another. It explicitly models the features as conditionally independent given the class. Because of the independence assumption they are highly scalable and can quickly learn to use high dimensional features with limited training data. Given datapoint $\vec{x}$ of n features, naïve bayes predicts the class C_k for datapoint, according to the Bayes' theorem it can be factored as,

$$\mathrm{p}(C_k|\vec{x}) = \frac{\mathrm{p}(\vec{x}|C_k)\mathrm{p}(C_k)}{p(\vec{x})} = \frac{p(x_1, \ldots., \; x_n|C_k)p(C_k)}{p(x_1, \ldots., \; x_n)}$$

As we see this classifier is best suited for small size dataset, we have passive aggressive classifier implemented due to its specific properties.

Passive aggressive classifier is simple and their performance has been proofed to be superior to many other alternatives. Let's suppose to have a dataset where $\bar{x}$ is the datapoint and y_i is the predicted output. Given a weight vector w, the prediction is simply obtained as: $\tilde{y}_t$ = sign($\overline{w}^T . \bar{x}_t$), algorithm works generically with this update rule:

$$\overline{w}_{t+1} = \overline{w}_t + \frac{\max(0{,}1 - y_t(\overline{w}^T . \; \bar{x}_t))}{||x_t||^2 + \frac{1}{2C}} \; y_t\bar{x}_t$$

Hence this classifier trains the model incrementally. These are the conventional means of algorithms whose accuracy is limited when compared with deep learning architecture.

The flowchart [Fig 1] provides a brief overview of the entire process of building the models. The datasets were first cleaned for any corrupted data and the missing values were dropped. Certain datasets contained extra columns a.k.a attributes that were dropped based on its relevancy. The datasets were then split in to train, dev and test sets. The models were optimized over train and dev and finally tested on the 'test' set. We took mainly two different approaches in converting the textual data of news articles into its numerical form.

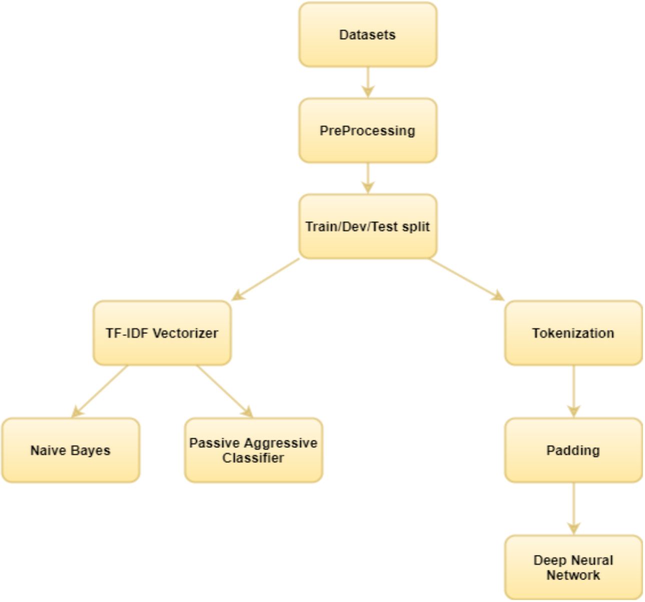

Fig 1. Flowchart

The first path led us in converting using the TF-IDF vectorizer. This vectorization comes under the bag of words model where the words are treated as numbers based on the number of occurrences. TF-IDF stands for term frequency-inverse document frequency, where the value increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. While this vectorization is effective, in its bid to convert to numbers, the semantic meaning of the words is lost.

The vectorized dataset was then fed into the models based on the Naive Bayes Classifier and Passive-Aggressive Classifier. Multinomial NB was chosen over the other variants in this paper. Naive Bayes finds the probability of an event occurring given the probability of another event already occurred. It predicts the relationship probabilities for each class such as the probability that a given record or data point belongs to a particular class. Naive Bayes classifier assumes that all the features are unrelated to each other, the presence or absence of a feature does not influence any of the other features. In this paper, Naive Bayes was implemented with Lidstone smoothing in order to optimize the model and its performance. Grid search was also used to set the hyper parameters such as alpha value.

The other Classifier used in this paper is the Passive-Aggressive Classifier. This is the only linear based model used in the paper. This classifier works on the following rule: "The classifier is passive when correct classification is obtained else the rule becomes very aggressive. It looks for the new weight which is closest to previous and satisfies the L value. “With Passive-aggressive Classifier, we also implemented two different variations, with and without the use of early stopping. Early stopping is when the training of the model is compromised in favor of better validation accuracy. The training is stopped prematurely when the validation accuracy starts to degrade. While this provides a better result, it is at the cost of the model not being completely trained on the dataset.

While Naive Bayes and Passive Aggressive Classifier were fed Tf-Idf vectorized data. We chose to use tokenizers for feeding into Deep Neural Network models. Keras offers a simple API, where one can vectorize text corpus by converting each word into vectors or sequences of integers. It splits the text into a list of tokens where co-efficient for each token could be based on word count. The dictionary of the tokenizer has been prepared using the train set of each dataset. The tokenizer once constructed can be fit on the raw text data. Since each different article contained different numbers of words, we used padding to keep the size uniform.

The below figure represents a Neural network constructed for datasets with three attributes.

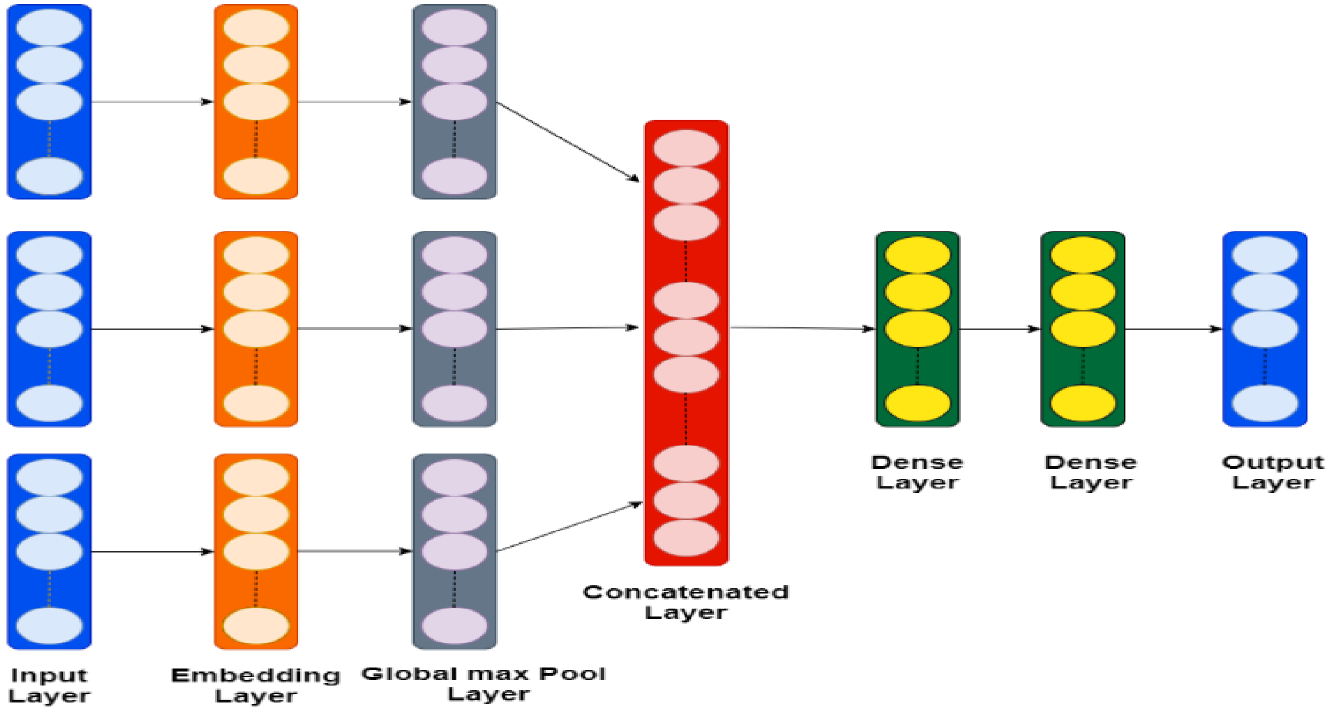

Fig 2. Neural Network architecture

There are mainly 7 layers i.e 1 input, 1 output, and 5 hidden layers. Each attribute is individually tokenized and fed to the network. Each input layer takes the tokenized value from a single attribute of the dataset. Each Input layer is fed to the embedding layer which learns all the embeddings of the word. This layer allows us to take large inputs like sparse vectors representing words. The output of this layer is fed to a global max pool layer which is used to compute the single max value for each of its input channels. It is a great alternative to flattening. All three outputs from max pool layers are concatenated to form a single layer. This layer is fed to a series of Dense layer which connects all the neurons together to form a network. The final output is fed to the output layer where the sigmoid activation function is used so that output is either true or false. The Dense layers contained Relu activation functions. Activity, kernel, and bias regularizers along with Dropout were all used as seen fit.

For datasets with only one attribute, we used sequential DNN and complex networks were built using functional DNN. The optimized models for all algorithms have been documented in the next section.

V. RESULT AND ANALYSIS

After the careful construction of models on multiple datasets. The results have been documented in the following table.

Dataset	**Methodology**	**Results**		
		Stop	**Test**	**Train**
Jru	Naïve Bayes		96%	99%
	Passive aggressive	yes	99%	100%
		No	99%	100%
	DNN		99%	100%
Pontes	Naïve Bayes		96.6%	96.5%
	Passive aggressive	yes	98.5%	99%
		no	98.4%	98.9%
	DNN		97%	98%
	Naïve Bayes		96.9%	96.8%
	Passive aggressive	Yes	100%	100%
		no	98.9%	99.8%
	DNN		98%	99%
ClaimsKG	Naïve Bayes		77.2%	89.4%
	Passive aggressive	yes	73.5%	97.3%
		no	73.7%	99.8%
	DNN		77.9%	99.8%
Kaggle	Naïve Bayes		85.1%	85.6%
	Passive aggressive	Yes	83.8%	89.2%
		no	82.6%	89.8%
	DNN		87%	99%
Liar	Naïve Bayes		71.8%	77.4%
	Passive aggressive	yes	64.7%	86.6%
		no	64.3%	99.9%
	DNN		63.9%	73.6%
Newsfiles	Naïve Bayes		97.8%	99.3%
	Passive aggressive	yes	99%	99%
		no	99%	100%
	DNN		98%	100%
Superset	Naïve Bayes		84.6%	88.7%
	Passive aggressive	yes	87.1%	94.3%
		no	85.5%	97%
	DNN		85%	99%

Table 1. Dataset documentation

It can be seen from the documented table, that certain datasets have performed significantly better than other datasets. It is observed that the JRU dataset with only three attributes has performed quite well with almost 100 percent accuracy. While one could chalk up the success of the dataset to its small size, the Pontes dataset, on the other hand, proves the conjecture false. Pontes dataset was split into two routes based on author attribute, which contained 40% of missing values. Although the dataset is massive, both the routes performed equally well with the accuracies in the high 90s. On further analysis, we can conclude that Route 2 models are better when expanded and generalized. ClaimsKG models performed relatively poor compared to other datasets with accuracies in the early 70s. This performance can be ascribed to the fact that only 10000 articles were present. Perhaps more articles and revaluation of its attributes could provide better results.

The Kaggle competition dataset, on the other hand, has performed relatively well with accuracies in the mid-80s. Liar dataset performed disappointingly even though it contained the highest number of attributes. Naive Bayes stood out on top, surprisingly performing better than DNN but only in this dataset. This can be attributed to robustness of Naïve Bayes algorithms. A lot more news articles from varied sources are required to buttress the models

The final two datasets were approached with a purely NLP perspective. NewsFiles like JRU performed stupendously well in all metrics. Superset with its massive number of articles (233413) showed that an NLP only approach is viable with a respectable mid 80's accuracy.

It can be observed from the table that DNN outperformed both naive Bayes and passive-aggressive classifiers in every dataset except one. The success of DNN is due to the fact Neural networks better represent complex, nonlinear structures which in this case is a perfect fit. Naive Bayes and Passive aggressive classifiers both produced similar if not the same results. While adding early stopping helped, there was no significant improvement to warrant its usage. Comparing our results with that of the reference papers. It is seen that some of our models have performed better than those papers that used similar algorithms.

Bar graphs with comparative results of all datasets have been provided for the reader’s convenience.

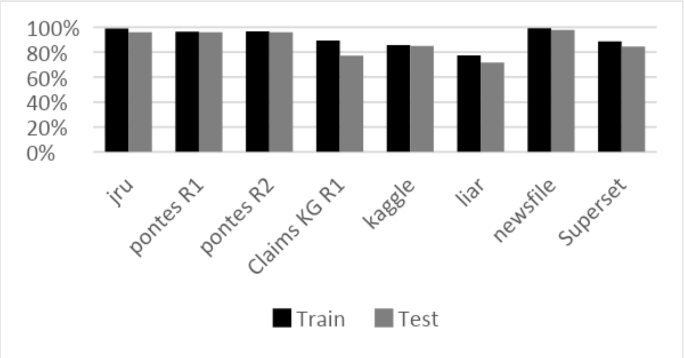

Fig 3. Naïve Bayes comparison results

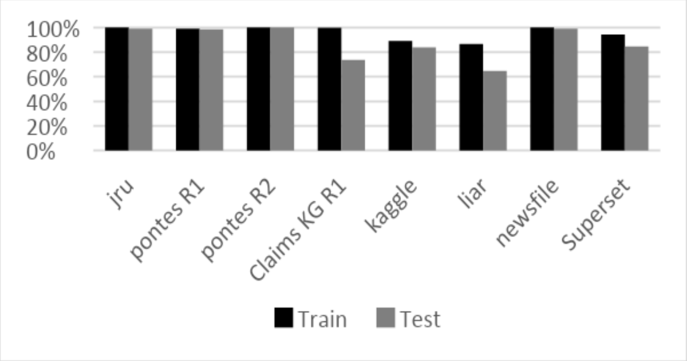

Fig 4. Passive aggressive classifier comparison results

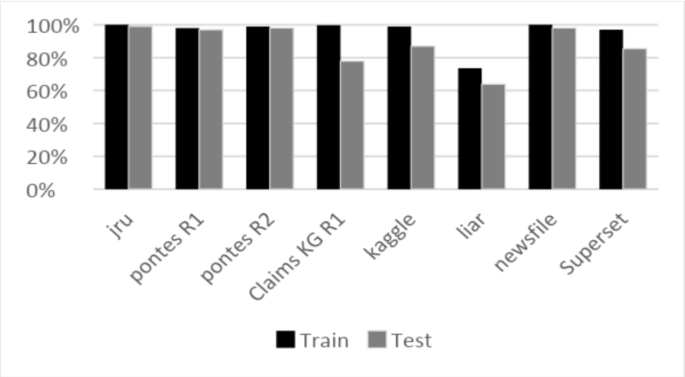

Fig 5. Deep Neural Network comparison results

VI. CONCLUSION AND FUTURE WORK

Machine Learning has opened a new front in the warfare against Fake news ,one must take advantage of this front and exploit it thoroughly. This paper has shown that the front is viable. The usage of Machine learning in identification of fake news is still in its infancy. Every model built or a system proposed is one step closer to a fake news free internet.

While we have only used basic algorithms, there is a high potential for the creation of better models. The usage of CNN and RNN shows great prospects that can be exploited. Pre-trained word embeddings such as word2Vec and GloVe could be used. A system can be created by combining ML models to detect fake news with the system from $Costel^{6]}$ and $Jahankhani^{5]}$. The war against fake news never ends, but the damage can be curbed using the right tool at the right time. In this day and age of internet dominance, Identifying fake news is and always will be an important factor in our lives.

VII. REFERENCES

[1]. Ammara Habib, Muhammad Zubair Asghar, Adil Khan, Anam Habib, Aurangzeb Khan, "**False information detection in online content and its role in decision making: a systematic literature review**", Springer-Verlag GmbH Austria, part of Springer Nature 2019.

[2]. Thota, Aswini; Tilak, Priyanka; Ahluwalia, Simrat; and Lohia, Nibrat (2018) "**Fake News Detection: A Deep Learning Approach**," SMU Data Science Review: Vol. 1 : No. 3 , Article 10.

[3]. Chaowei Zhang, Ashish Gupta, Christian Kauten, Amit V Deokar, Xiao Qin, "**Detecting fake news for reducing misinformation risks using analytics approaches** ", European Journal of Operational Research Elsevier 279 (2019).

[4]. Dinesh Kumar Vishwakarma, Deepika Varshney, Ashima Yadav, "**Detection and veracity analysis of fake news via scrapping and authenticating the web search**", Cognitive Systems Research 58.

[5]. Hossein Jahankhani, Thulasirajh Jayaraveendran, and William Kapuku-Bwabw, "**Improved Awareness on Fake Websites and Detecting Techniques",** ICGS3/e-Democracy 2011, LNICST 99.

[6]. Costel-Sergiu Atodiresei, Alexandru Tanaselea, Adrian Iftene, "**Identifying Fake News and Fake Users on Twitter**", International Conference on Knowledge Based and Intelligent Information and Engineering Systems.

[7]. Atik Mahabub, "**A robust technique of fake news detection using Ensemble Voting Classifier and comparison with other classifiers**".

[8]. Harita Reddy, Namratha Raj, Manali Gala, Annappa Basava, "**Text-mining-based Fake News Detection Using Ensemble Methods",** International Journal of Automation and Computing.

[9]. Kushal Agarwalla, Shubham Nandan, Varun Anil Nair, D.Deva Hema, "**Fake News Detection using Machine Learing and Natural Language Processing",** International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6.

[10]. Mykhailo Granik, Volodymyr Mesyura, "**Fake News Detection using Naïve Bayes Classifier",** 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering.

[11]. Akshay Jain, Amey Kasbe, "**Fake News Detection**", 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Sciences.

[12]. Samir Bajaj," **The Pope Has a New Baby! Fake News Detection Using Deep Learning".**

[13]. A.Lakshmanarao, Y.Swathi, T.Srinivasa Ravi Kiran," **An Efficient Fake News Detection System Using Machine Learning**", International Journal of Innovative Technology and Exploring Engineering, Volume-8, Issue-10, August 2019.

[14]. Shlok Gilda," **Evaluating Machine Learning Algorithms for Fake News Detection",** 2017 IEEE 15th Student Conference on Research and Development.

[15]. Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea,"**Automatic Detection of Fake News".**

[16]. Jru dataset available at kaggle: https://www.kaggle.com/jruvika/fake-news-detection#__sid=js0

[17]. Politifact dataset available at: https://homes.cs.washington.edu/~hrashkin/factcheck.html

[18]. Pontes dataset: https://www.kaggle.com/pontes/fake-news-sample

[19]. ClaimsKG dataset available at : https://data.gesis.org/claimskg/explorer/research

[20]. Kaggle fake news set: https://www.kaggle.com/c/fake-news/dataa

[21]. Liar dataset extracted from github: https://github.com/thiagorainmaker77/liar_dataset

[22]. Newsfiles dataset available at: https://homes.cs.washington.edu/~hrashkin/factcheck.html