An Empirical Comparison of Fake News Detection using different Machine Learning Algorithms

Abdulaziz Albahr[1], Marwan Albahar[2]
College of Applied Medical Sciences, King Saud bin Abdulaziz University for Health Sciences, Alahsa, Saudi Arabia[1]
King Abdullah International Medical Research Center, Alahsa, Saudi Arabia[1]
Umm Al Qura University[2]

***Abstract*—Relying on social networks to follow the news has its pros and cons. Social media websites indeed allow the spread of information among people quickly. However, such websites might be leveraged to circulate low-quality news full of misinformation, i.e., "fake news." The wide distribution of fake news has a considerable negative impact on individuals and society as a whole. Thus, detecting fake news published on the various social media websites has lately become an evolving research area that is drawing great attention. Detecting the widespread fake news over the numerous social media platforms presents new challenges that make the currently deployed algorithms ineffective or not applicable anymore. Basically, fake news is deliberately written on the first place to mislead readers to accept false information as being true, which makes it difficult to detect based on news content solely; consequently, auxiliary information, like user social engagements on social media websites, need to be taken into account to help make a better detection. Using such auxiliary information is challenging because users' social engagements with fake news produce noisy, unstructured, and incomplete Big-Data. Due to the fact that fake news detection on social media is fundamental, this research aims at examining four well-known machine learning algorithms, namely the random forest, the Naïve Bayes, the neural network, and the decision trees, distinctively to validate the efficiency of the classification performance on detecting fake news. We conducted an experiment on a widely used public dataset i.e. LIAR, and the results show that the Naïve Bayes classifier defeats the other algorithms remarkably on this dataset.**

***Keywords*—Fake news; classification; machine learning; performance comparison**

I. Introduction

Many people follow the news through different social media platforms because of their ease of access. For instance, about two-thirds of the Americans follow the news through social media websites [1][2]. Newman et al. [3] reported the increased usage of various digital platforms in Great Britain as the main source of the news feed. Because of circulating the breaking news swiftly, social media platforms are significantly better than traditional media [4]. However, not all posted news items are true. There are many economic, social, and political reasons behind people's manipulation of data and information changing. Therefore, these manipulated data leads to creating news items that are neither totally true nor totally false [5]. This, in turn, leads to misleading information on social media networks that causes several predicaments in society. Such misinformation (also known as "Fake News") has a broad spectrum of types and forms. For example, rumors, fake advertisements, satires, and false political reports are different types of fake news [1]. The spread of fake news becomes more viral than the true news items [6] urged many researchers to concentrate on innovating efficient automated solutions for detecting fake news [7]. Google has announced a new service named "Google News Initiative" aimed at tracking and eliminating fake news [8]. This project will assist users in distinguishing fake news and reports [9]. In fact, the task of detecting fake news is challenging. A fake news detection model aims at identifying purposely misleading news relying on investigating the previously reviewed fake and real news. This brings us to shed light on the availability of large-scale top-quality training data as one of the cornerstones. The fake news detection framework's task can be considered a simple binary classification or a fine-tuned classification in a challenging setting [10]. After 2017, various fake news datasets were introduced. Researchers sought to improve the deployed models' performance using these different datasets such as (ISOT, Kaggle, and LIAR datasets), which are well-known publicly available datasets [11].

In the current research paper, we compare different machine learning classifiers' performance for detecting fake news. The key contributions of this research paper are as follows:

- A detailed performance analysis of four machine learning algorithms using different NLP techniques for detecting fake news.
- Different machine learning-based models are implemented to detect and classify fake news. Each model's performance is measured to categorize various news items correctly, which revealed each model's ability to improve its accuracy of detecting fake news.

This paper is arranged as follows: Section II presents the related works. The objective of this study is clearly highlighted in Section III. In Section IV, we review different classifiers. Section V will explain the data collection process and provide an analysis of the dataset. In Section VI, we present the experimental setup and the evaluation metrics. In Section VII, the examined models' methodology containing the data preparation and handling the missing data problem is discussed in detail. The experimental results of the implemented models are discussed in Section VIII. A discussion of the obtained results and a conclusion of this study are shown in Section XI and X.

II. Related Works

As people tend to consume more news on social media, fake news on social media has emerged as a critical problem that has a negative impact on society and government [25]. An early study on detecting fake news concentrated on detecting rumors on twitter, and these studies were conducted by social scientists [12]. Later, researchers have focused on understanding the structure and characteristics of fake news in order to identify fake news. As a result, numerous approaches for automatic fake news detection have been proposed in the literature. Most of these approaches transform the fake news detection into a binary classification task, where each statement, "i.e., news" is labeled as true or false using various machine learning techniques (e.g., [13][14]) or deep learning based techniques [16]. These approaches require data corpus to correctly detect fake news. Rubin [15] introduced three criteria used to determine the quality of created text corpus for identifying fake news, i.e., all facts included in the dataset must be verified; all facts occur in a specific period (e.g., during US election); the way used to observe the facts must be similar, and the facts must have a different level of impact on society. The text corpus has an advantage that the pre-processing is straightforward and simple. However, it suffers from the following limitation, i.e., the only text analysis will reveal limited clues that are not enough to effectively detect fake news. Therefore, current approaches have integrated information based on the propagation network of news that captures how they spread. Ruchansky [16] introduced a new approach, called CSI that encapsulates three modules: Capture, Score, and Integrate. The capture module captures the temporal patterns of the users with the textual information of the news. Score Module exploits users' profiles to learn their vector representation and computes a score for each user engaged in spreading news. It then combines the output of the previous two modules to classify the news as fake. Singhania et al. [17] propose an approach that is based on deep learning techniques. In their approach, three layers are used to explore the different levels of the text of news separately (i.e., word level, sentence level, and title level). Liu and WU [18] propose an approach that aims to detect fake news at an early stage by exploiting the propagation network of news. In their approach, news are modeled as a multivariate sequence whose elements represent users involved in spreading news. Users are represented as a vector based on features extracted from their profile. Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) are applied to learn vector representation of news's sequence, which feeds into a multi-layer neural network to classify news as fake or not. Wu and Liu [19] propose an approach based on tracing network of news and using the LSTM-RNN model for classification. Instead of identifying helpful features that apply to detect fake news, Vo and Lee [20] identify people, called guardians, who are interested in correcting fake news and propose a recommendation system that recommends URLs of fact checking to guardians to integrate with fake news. Karimi et al. [21] propose an approach that considers fake news detection a multi-class classification task. In their approach, CNN and LSTM are used to automatically extracted feature vectors from each textual source of news and used an interpretable multi-source fusion model to integrate the learned feature vectors into one vector. Then, the Multi-class Discriminative Function component is used to determine the class of the fakeness of news. Aghakhani et al. [22] propose an approach called FakeGAN, which uses GAN algorithms to detect false reviews. Goldani et al. [23] propose a method that uses capsule neural networks with word embedding representation to enhance fake news detection performance. Two widely used datasets, i.e., ISOT and LIAR, evaluate the proposed method's performance. Wang et al. [24] propose an end to end approach that teaches common features representation among events and uses them to detect fake news on new events.

III. Research Objective

There is no doubt that the current political events have led to an increase in fake news circulation. In fact, humans are inconsistent and very poor in detecting fake news. Thus, researchers have exerted their efforts to automate the process of identifying fake news. The most well-known attempts blacklist authors and sources that are unreliable. However, we need to consider more complex cases where reliable authors and sources publish fake news to have a reliable, fully automated detecting solution. Machine learning proves to be useful in detecting language patterns. Hence, this research aims to use different machine learning models to detect language patterns that distinguish between real and fake news.

IV. Background

A. Naive Bayes

The Naïve Bayes classifier is a classifier based on Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)\times P(A)}{P(B)} \qquad (1)$$

Where A and B denote two conditions. The naive Bayes classifier considers each semantic feature as a condition and classifies the samples with the highest occurring probability. Noteworthy, it assumes that the semantic features are independent. Naïve Bayes is considered to be one of the most efficient and effective classification algorithms. It can work on small sample sizes and produces an accurate classification result [25].

B. Random Forest

A decision tree comprises parents with different conditions branch, in which, each node represents a class for classification. The random forest classifier is an ensemble method for classification which construct a multitude of decision trees. We set parameters such as n_estimators, min_samples_split, random_state, max_depth to obtain the best performance. In which, n_estimators represents the number of decision trees in the random forest, min_samples_split represents the minimum amount of samples to split an internal node, and max_depth represents the maximum depth of a decision tree.

C. Decision Trees

A decision tree is a set of decision nodes that start at the root. The benefits of utilizing a decision tree include easy interpretation, efficient handling of outliers, no need for the linear separation of classes, dependent features. Nevertheless, the existence of so many sparse features could lead a decision tree to overfit, and thus it performs poorly.

D. *Artificial Neural Network*

A neural network is made of interconnected processing nodes known as neurons that work together to solve very particular problems. Using deep neural networks is considered one of the most successful methods of machine learning. Lately, the new advents of neural networks and pre-trained word embedding have become the main basis of new rich ideas of NLP tasks. Nevertheless, the current model treats all words as a network of input and does not take into account the function of keywords. Consequently, redesigning the neural network model by combining the advantages of the two methods and increasing the weight of keywords in the network could lead to a remarkable improvement.

V. Dataset

We use a public dataset in [26], which comprises 12.8K human-labeled short statements from PolitiFact through its API. POLITIFACT.COM editor was applied to each statement to evaluate its validity. Six fine-grained labels for news truthfulness are considered into multiple classes, including false, true, pants-fire, mostly-true, half-true and barely-true. The distribution of labels in this dataset is as follows: a range between 2,063 to 2,638 for multiple labels and 1,050 pants-fire labels. Moreover, the dataset comprises of different metadata. These metadata contain valuable information about the speaker, total credit history count of the speaker, state, subject, party, and job. The total credit history count, including the half-true counts, false counts, pants-fire counts, barely-true counts, mostly-true counts. The statistics of the dataset are listed in Table I. Some selection samples from the dataset are presented in Table II.

TABLE I. The Statistics of LIAR Dataset [26]

LIAR Dataset Statistics	
Training set size	10,269
Validation set size	1,284
Testing set size	1,283
Avg. statement length (tokens)	17.9
Top-3 Speaker Affiliations	
Democrats	4,150
Republicans	5,687
None (e.g., FB posts)	2,185

TABLE II. Two Random Excerpts from the LIAR Dataset [26]

Sample 1	**Sample 2**
Statement: The last quarter, it was just announced, our gross domestic product was below zero. Who ever heard of this? It's never below zero.	**Statement:** Under the health care law, everybody will have lower rates, better quality care and better access.
Speaker: Donald Trump	**Speaker**: Nancy Pelosi
Context: presidential announcement speech	**Context**: on Meet the Press
Label: Pants-fire	**Label**: False
Justification: According to Bureau of Economic Analysis and the National Bureau of Economic Research, the growth in the gross domestic product has been below zero 42 times over 68 years. That's a lot more than never. We rate his claim Pants on Fire!	**Justification**: Even the study which Pelosis staff cited as a source of that the statement suggested that some people would pay more for health insurance. Analysis at the state level found the same thing. The general understanding of the word everybody is every person. The predictions do not back that up. We rule this statement False.

VI. Evaluation

A. *Experimental Setup*

The experiments of this paper were conducted on a server having 32 GB RAM, GeForce GTX 1080 GPU of 8 GB GDDR5X memory, and 2560 NVIDIA CUDA cores. We used Keras library for implementing the proposed models.

B. *Evaluation Metrics*

To evaluate the models, training and validation accuracy are reported for the data partitions. Accuracy is calculated based on the following mathematical representation. Apart from accuracy, other performance measures, that is, True Positive Rate (TPR) also known as Recall, Precision (Pre), and F1 measures, are calculated based on equations 2, 3, 4 and 5, respectively.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$TPR = \frac{TP}{TP+FN} \tag{3}$$

$$Pre = \frac{TP}{TP+FP} \tag{4}$$

$$F1 = 2 * \frac{TPRX\ Pre}{TPR+Pre} \tag{5}$$

where FP, TN, TP, and FN denote false positives, true negatives, true positives and false negatives, respectively.

VII. Methodology

The LIAR dataset is a well-known dataset in the realm of detecting fake news. It consists of 12,836 human-labeled short statements. The chosen instances in this dataset are from more natural contexts such as political debates, Facebook, tweets posts, etc. Furthermore, it contains 12787 news items. In each item, the following features are provided:

- News statement
- Barely true counts
- Subject of news
- False counts
- Half true counts
- Speaker name
- Speaker's job title
- Mostly true counts
- Pants on fire
- State information
- Party affiliation
- Venue

A. Data Preparation

We split the categorical (text) features and numerical features into two categories:

- Numerical features are (false counts, mostly true counts, pants on fire, barely true counts, mostly true counts, and half true counts). As we know, we do not need to do pre-processing on the numerical features because these features contain true counts and false counts, so we will use these counts for each news item.
- Categorical features are (Party affiliation, Venue, Subject of news, Speaker name, Speaker's job title, State information, and News statement).

Our primary focus was on feature engineering; if we could add some other features or fine-tune the features, detecting news accuracy can be much efficient. Therefore, we explore all categorical features to extract the best feature that distinguishes true and fake news.

In the party affiliation feature, we extracted the party affiliation's unique parties and then replaced all the different parties into four categories named as republican, democrat, unknown, and others. Consequently, we intend to make feature values closer to the class label of news.

While in other features, we tokenize all the words to work on each word separately. After tokenization, we removed the stop words of English because stop words are not good words that cannot distinguish between true and fake news. Thus, we removed these stop words because they fall into both: true or fake news. Next, we applied stemming on the words/tokens because we wanted to use only the (base/root) form of words. Although stemming does not work well on all the words, our goal was to convert all these features into categories.

There were unique words in Label/class, speaker, and state info features, so we encoded them into unique numbers (half-true as 0, false as 1, mostly-true as 2, barely true as 3, true as 4, and pants-fire as 5). We encoded the other features categories into unique numbers as well.

Finally, in the Statement of news feature, we removed the punctuation from the Statement's sentences. We then removed the repeating characters; we also clean hyperlinks and other special characters from the text of statement news. After cleaning the news statement, we applied unigram feature extraction, bigram features extraction, and trigram features extraction. We observed that the trigram feature yields good results, among others (see Fig. 1).

B. Missing Data

We conduct an investigation to check the missing value because it can affect the overall performance of algorithms. Thus, we found that 3565 speaker job-titles were missing, 2747 state information missing, and 129 venues. Initially, we replaced the missing values with NaN, and after that, we replaced these with unknown words. It is worth mentioning that, based on our observation, using our method to handle the missing value does not make a significant difference or even any difference.

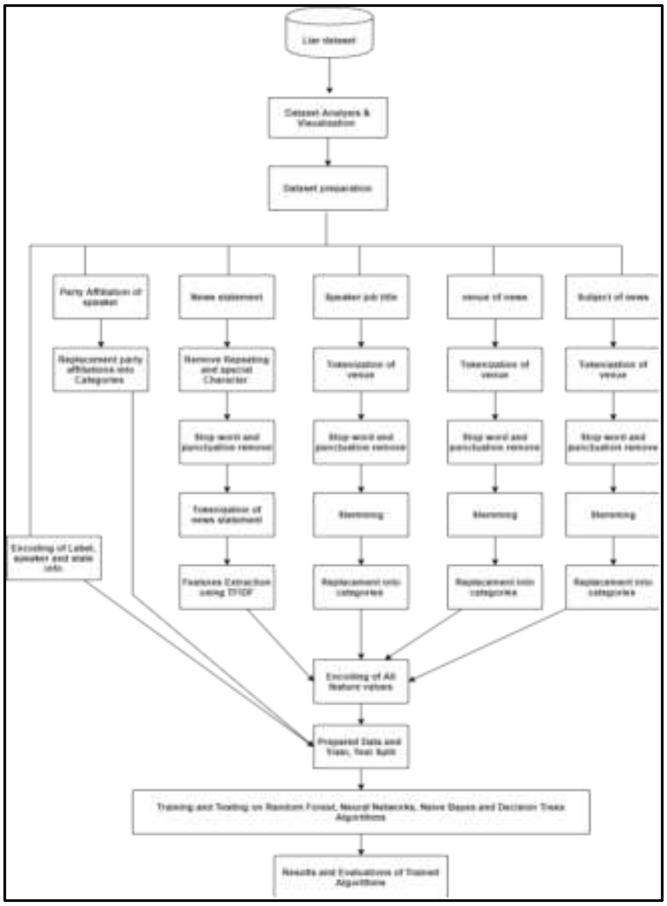

Liar dataset
Dataset Analysis & Visualization
Dataset preparation
Party Affiliation of speaker
News statement
Speaker job title
venue of news
Subject of news
Replacement party affiliations into Categories
Remove Repeating and special Character
Tokenization of venue
Tokenization of venue
Tokenization of venue
Stop word and punctuation removal
Stop word and punctuation remove
Stop word and punctuation remove
Stop word and punctuation remove
Tokenization of news statement
Stemming
Stemming
Stemming
Encoding of Label, speaker and state info
Features Extraction using TFIDF
Replacement into categories
Replacement into categories
Replacement into categories
Encoding of All feature values
Prepared Data and Train, Test Split
Training and Testing on Random Forest, Neural Networks, Naive Bayes and Decision Trees Algorithms
Results and Evaluations of Trained Algorithms

Fig. 1. The High-Level Architecture of the Entire Process.

C. Overfitting and Cross Validation

Overfitting is one of the central problems in machine learning. It arises when the model performs poorly on unseen data while giving excellent results on training data. Cross-validation is a way to overcome such an issue; it aims to test the model's ability to correctly predict new data that was not used in its training. Cross-validation shows the model generalization error and performance on unseen data. K-fold cross-validation is one of the most popular versions. In our experiment, we use k-fold cross-validation to ensure we avoid overfitting.

VIII. Experimental Result

A. Training and Testing

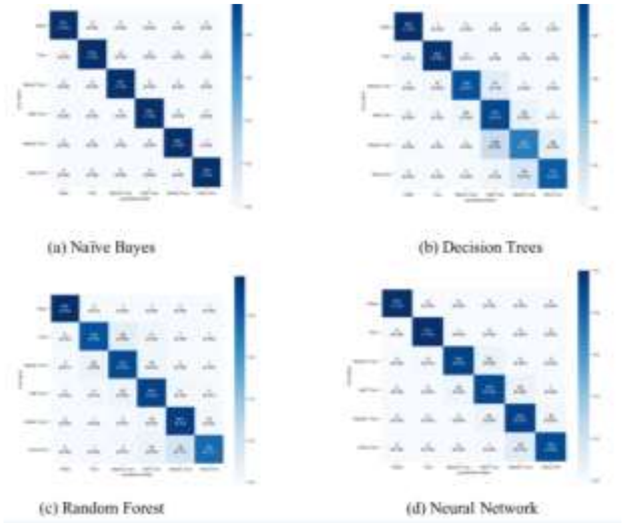
There were three separate files in this dataset. We combined all three files and pre-processed the data. Then, we prepared it for the training and testing sets. We divided the data 70-30% for training and testing sets. Next, we used cross-validation 5 times, with 80-20% split every time for training and testing. It is important to note that we shuffled all the rows with random state 6 to avoid any train models' biases. We employed four different machine learning algorithms, and we used Python 3.6.5 as our programming language for the implementation. The classification models that we implemented are the random forest, Naïve Bayes, neural network, and decision trees algorithms to explore further how well our data fit into the models. These algorithms are suitable for several classifications as they have their own properties. We observed different variations during the training of machine learning algorithms. We tune the parameters of each algorithm and get different results. Still, k-fold validation techniques can be applied. Accordingly, we can use the data in different folds to train the algorithms each time on a different training set. After training, we used the trained models of each algorithm and tested them on the testing set.

B. Result

To evaluate the classification process of each algorithm, standard metrics that measure the overall performance were considered. The number of predictions (whether correct or incorrect) with each class are shown in the confusion matrix (see Fig. 2). Fig. 2 shows the confusion matrix of each classifier on the test set. Naïve Bayes classifies all classes with more accuracy. Other classifiers encounter some difficulties classifying mostly true and pants-fire. For these labels, detecting the correct label is more challenging, and many pants-fire texts are predicted as false.

Nevertheless, it is challenging to distinguish between false and barely-true and between mostly-true and true. To have a thoroughly detailed analysis, we evaluated each model's performance within and across different K-fold (see Tables III to VII). This allowed us to further study the overall performance of each algorithm and see the generalization as well.

Fig. 3 presents the overall comparison of all algorithms. Among all the classifiers we have implemented, Naïve Bayes gives the highest accuracy. Other classifiers, such as random forest, appeared to be vulnerable to overconfidence due to the usage of independent variables to predict outcomes. Noteworthy, it requires that each data point needs to be independent of all other data points. In this dataset, the news statement determines the feature word length.



(a) Naïve Bayes

(b) Decision Trees

(c) Random Forest

(d) Neural Network

Fig. 2. Confusion Matrix of Classification using different Machine Learning Algorithms for LIAR Dataset.

TABLE III. K-Fold-1 Results 80% Training and 20% Testing

Evaluation measures	Random Forest	Naïve Bayes	Neural Network	Decision Trees
Accuracy	0.90%	0.99%	0.97%	0.94%
Precision	0.90%	100%	0.97%	0.93%
Recall	0.89%	100%	0.97%	0.95%
F1-Score	0.89%	100%	0.97%	0.94%

TABLE IV. K-Fold-2 Results 80% Training and 20% Testing

Evaluation measures	Random Forest	Naïve Bayes	Neural Network	Decision Trees
Accuracy	0.91%	0.99%	0.89%	0.91%
Precision	0.91%	100%	0.91%	0.92%
Recall	0.90%	100%	0.90%	0.91%
F1-Score	0.91%	100%	0.90%	0.92%

TABLE V. K-Fold-3 Results 80% Training and 20% Testing

Evaluation measures	Random Forest	Naïve Bayes	Neural Network	Decision Trees
Accuracy	0.93%	0.99%	0.94%	0.96%
Precision	0.92%	100%	0.95%	0.96%
Recall	0.91%	100%	0.95%	0.96%
F1-Score	0.91%	100%	0.95%	0.96%

TABLE VI. K-Fold-4 Results 80% Training and 20% Testing

Evaluation measures	Random Forest	Naïve Bayes	Neural Network	Decision Trees
Accuracy	0.91%	0.99%	0.94%	0.95%
Precision	0.92%	100%	0.93%	0.97%
Recall	0.90%	100%	0.95%	0.94%
F1-Score	0.91%	100%	0.94%	0.95%

TABLE VII. K-FOLD-5 RESULTS 80% TRAINING AND 20% TESTING

Evaluation measures	Random Forest	Naïve Bayes	Neural Network	Decision Trees
Accuracy	0.92%	0.99%	0.86%	0.76%
Precision	0.93%	100%	0.90%	0.81%
Recall	0.91%	100%	0.84%	0.77%
F1-Score	0.92%	100%	0.85%	0.75%

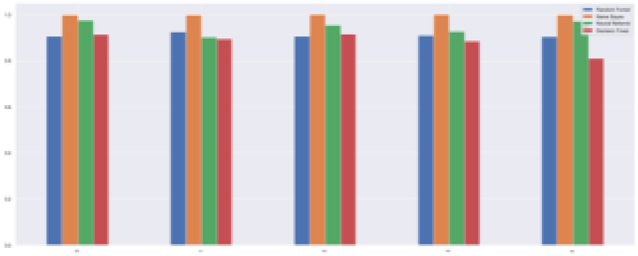


Fig. 3. Overall Accuracy Comparison of different Machine Learning Algorithms for LIAR.

Consequently, the model will tend to overweight the significance of observations when such observations are related. It is worth mentioning that, as we can see from the results, there is no overfitting problem because, in every split, the model performance is consistent. Likewise, we have evaluated the trained models, and we could see the fine details of each algorithm's confusion matrix and that there is no false rate of predictions on class level.

IX. DISCUSSION

We examined the performance of several machine learning algorithms: the random forest, the Naïve Bayes, the neural network, and the decision trees algorithms. In general, our obtained results verify the pros and cons of the compared different machine learning algorithms when they have been used in detecting fake news. In the following few lines, we analyze the results and give an insight into Naïve Bayes performance:

- In this research work, we intend to train the dataset on different algorithms to determine which algorithm performs well. The reason for the Naïve Bayes algorithm's good performance is that it works well on the text, based on Bayes theorem. Naïve Bayes computes conditional probabilities of two events on the basis of text occurrence individually and differentiates each event/class accordingly.

- Naïve Bayes algorithm is better than other algorithms in this dataset. The accuracy is good as we evaluated the trained model with different evaluation measures, and the converging/training time of the Naïve Bayes is excellent (see Table VIII).

TABLE VIII. THE RUN-TIME FOR THE DIFFERENT MACHINE LEARNING ALGORITHMS (IN SECONDS)

	Random Forest	Naïve Bayes	Neural Network	Decision Trees
Run Time	54.3 s	1.3 s	420.2 s	540.1 s

- When applied to this dataset, the computational run-time and accuracy comparisons led us to conclude that the Naïve Bayes is the best method in general.

X. CONCLUSION

Understanding the rationale of specific fake news items infers many details about the different involved factors. Recently, a rapidly increased number of models were proposed in the literature to automatically detect fake news. The two influential factors that significantly impact these models' accuracy are the datasets and a set of explicit classes. Our experiment posits that good models should require a reasonable number of fine-tuning when tested on different datasets. This paper investigates four machine learning classifiers' performance, namely, the random forest, the Naïve Bayes, the neural network, and the decision trees algorithms for identifying fake news. We used a publicly well-known dataset, i.e., LIAR. Based on our results, we observed good performance of the Naïve Bayes algorithm because of the computation of conditional probabilities of two events on the basis of text occurrence individually and the differentiation between each event/class accordingly.

REFERENCES

[1] Zhang, X.; and Ghorbani, A. A. (2019). An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management, 57, 102025.

[2] Dale, R. (2017). NLP in a post-truth world. Natural Language Engineering, 23, 319-324.

[3]]Newman, N.; Dutton, W.; and Blank, G. (2013). Social media in the changing ecology of news: Thefourth and fifth estates in Britain. International journal of internet science,7.

[4] Bondielli, A.; and Marcelloni, F. (2019). A survey on fake news and rumour detection tech-niques.Information Sciences, 497, 38–55.

[5] Granik, M.; and Mesyura, V. (2017). Fake news detection using naive Bayes classifier. Proceedings of 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) (pp. 900–903).

[6] Balmas, M. (2014). When fake news becomes real: Combined exposure to multiple news sourcesand political attitudes of inefficacy, alienation, and cynicism. Communication Research, 41,430–454.

[7] Horne, B. D.; and Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repeti-tive content in text body, more similar to satire than real news. In Eleventh International AAAIConference on Web and Social Media.

[8] Google News Initiative (2018). https://newsinitiative.withgoogle.com/, Retrieved March 25, 2020.

[9] Google Announcement. (2018). Google anounces google news initiative to help quality journalismin digital. Retrieved March 21, 2020, from age.http://canoe.com/technology/google-announces-googlenews-initiative-to-help-quality-journalism-in-digital-age.

[10] Tin, P. T. (2018). A study on deep learning for fake news detection.

[11] Meel, P.; and Vishwakarma, D. K. (2019). Fake News, Rumor, Information Pollution in Social Media and Web: A Contemporary Survey of State-of-the-arts, Challenges and Opportunities. Expert Systems with Applications.

[12] Vosoughi, S.; Roy, D.; Aral, S. (2018). The spread of true and false news online. Science359,1146–1151.

[13] Shu, K.; Sliva, A.; Wang, S.; Tang, J.; and Liu, H. (2017). Fake news detection on social media: a data mining perspective. SIGKDD Explor. 19(1), 22–36.

[14] Zannettou, S.; Sirivianos, M.; Blackburn, J.; and Kourtellis, N. (2018). The web of false information: rumors, fake news, hoaxes, clickbait, and various other shenanigans.

[15] Rubin, V.; Conroy, N.; Chen, Y.; and Cornwell, S. (2016). Fake news or truth? using satiricalcues to detectpotentially misleading news.

Proceedings of the Second Workshop on Computational Approaches to Deception Detection. 7–17.

[16] Ruchansky, N.; Seo, S.; and Liu, Y. (2017). CSI: a hybrid deep model for fake news detection. Proceedings of the 2017 ACM Conference on Information and Knowledge Management, CIKM. Singapore, 797–806.

[17] Singhania, S.; Fernandez, N.; and Rao, S. (2017). 3han: A deep neural network for fake news detection. Proceedings of International Conference on Neural Information Processing. Springer, Cham, 572-581.

[18] Liu, Y.; and Wu, Y. B. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. Proceedings of the Thirty Second AAAI Conference on Artificial Intelligence. New Orleans, USA.

[19] Wu, L.; Liu, H. (2018). Tracing fake news footprints: characterizing social media messages by how they propagate. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM. Marina Del Rey, USA, 637–645.

[20] Vo, N.; and Lee, K. (2018). The rise of guardians: fact checking url recommendation to combat fake news. Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. Ann Arbor, USA, 275–284.

[21] Karimi, H.; Roy, P.; Saba-Sadiya, S.; and Tang, J. (2018). Multi-source multi-class fake news detection. Proceedings of the 27th International Conference on Computational Linguistics, COLING. SantaFe, USA, 1546–1557.

[22] Aghakhani, H.; Machiry, A.; Nilizadeh, S.; Kruegel, C; and Vigna, G. (2018). Detecting deceptive reviews using generative adversarial networks. Proceedings of 2018 IEEE Security and Privacy Workshops (SPW). 89-95.

[23] Goldani, M. H.; Momtazi, S.; and Safabakhsh, R. (2020). Detecting Fake News with Capsule Neural Networks. arXiv preprint arXiv:2002.01030.

[24] Wang, Y.; Ma, F.; Jin, Z.; Yuan, Y.; Xun, G.; Jha, K.; Su, L.; and Gao, J. (2018). Eann: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 849–857.

[25] Mansour, A. M. (2018). Texture Classification using Naïve Bayes Classifier. IJCSNS International Journal of Computer Science and Network Security.

[26] Wang, W. Y. (2017). ” Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 422-426.