Performances of Different Approaches for Fake News Classification: An Analytical Study

Chapter · January 2022

DOI: 10.1007/978-3-030-96040-7_53

4 authors:

Md Abdullah Al Kafi
Daffodil International University

Israt Jahan Tasnova
Daffodil International University

Md. Wadud Islam
Daffodil International University

Sumit Kumar Banshal
Alliance University

Performances of Different Approaches for Fake News Classification: An Analytical Study

Md. Abdullah-Al-Kafi[1,1], Israt Jahan Tasnova[1,2], Md. Wadud Islam[1,3] and Sumit Kumar Banshal[1,4[1-7795-1355]]

[1] Department of Computer Science & Engineering, Daffodil International University, Dhaka -1207, Bangladesh
[1]abdullah15-12152@diu.edu.bd, [2]israt15-12932@diu.edu.bd, [3]wadud15-12547@diu.edu.bd,[4]sumit.cse@diu.edu.bd

Abstract. The penetration of social and online platforms has opened a new substantial domain of Fake news dissemination in the current time. Also, this dynamic form of data opens up new dimensions for researchers to detect Fake news from the ocean of data. Therefore, Fake news detection has attracted both academia and industry indifferently as research or analytical domain in the concurrent time. Due to data availability, the classification tasks have been tested in different sets and types of data. Detecting Fake news evolves as an actual potential domain to explore with more efficient algorithms and parameter-based modified algorithms. In this work, an analytical sketch has been drawn to compare the performances of different classifiers depending on accuracy and time. Seven classifiers of four different types have been implemented and tested namely, Multilayer Perceptron, Sequential Minimal Optimization, Logistic Regression, Decision Tree, J48, Random Forest and Naïve Bayes Classifier. The analytical evaluation process has been designed with three experimental setups, 10-fold cross-validation, 70% split and 80% split. The separate setups show distinctive outcomes across the algorithms. Naïve-Bayes classifier model shows its prominence along with the Random Forest classifier. However, the and Decision Tree-based classifiers perform differently from earlier knowledge. Furthermore, this paper identifies a different aspect of using testing-training splitting in classifier tasks.

Keywords: Fake News Detection, Machine Learning, Classification.

1 Introduction

The term "Fake news" is used to describe false stories spreading on social media. It has been invoked to discredit some news organizations' critical reporting [1]. That means the news that is based on false facts is called Fake news. Fake news became popularized during the 2016 U.S elections, where the top twenty frequently-discussed false election stories generated 8,711,000 shares, reactions, and comments on Facebook. Ironically, more significant than the total of 7,367,000 for the top twenty most-discussed factually correct election stories posted by 19 major news websites [2]. Fake news can be disseminated in society through different mediums. Sometimes it can be spread through people, and sometimes, it can be spread through news mediums. But nowadays, the most prominent medium of spreading Fake News is Social media and online platforms such as Facebook, Twitter, YouTube, and other websites [3].

Fake news is generated to convince its readers to believe in a particularly intended purview. So, Fake News can create mistrust among the people in society. Fake news highlights the erosion of long-standing institutional earthworks against misinformation in the internet age, a global problem. Fake news overlaps with other information disorders like misinformation and disinformation [4]. Misinformation is false or misleading information and disinformation is purposely spread information to deceive people. This misleading approach towards information has transformed Fake news as a political weapon[4] . During the election, voters can be influenced by misleading political statements and claims [5]. Fake news has drawn significant concerns from both industry and academia due to its use in the current era of technology. A massive amount of misleading information is created and displayed on the internet. It hurt the internet activities like online shopping and social marketing [5]. There are countless web pages established to publish fake news and stories. Researchers identified these types of several various pages such as denverguardian.com, wtoe5news.com, ABCnews.com.co, and so on [5]. Due to its speed and potent of spreading misinformation, the Fake news detection topic has gained a great deal of interest from researchers across the globe [3]. A substantial number of academic articles used the term "fake news" between 2003 and 2017 resulted in a typology of types of fake news: news satire, news parody, fabrication, manipulation, advertising, and propaganda[1]. Fake news can be any content that is not truthful and generated to convince its readers to believe in something that is not true. So fake news can create mistrust among the people in society. This news is nearly impossible to verify analytically because of its huge quantity and high dimensions. So, a machine learning approach is better in this situation. In the computer science domain as well, the fake news classification is a well-discussed domain. There are so many papers [6–8] that most of the approaches share the same characteristics. Fake news classification is a Supervised Text classification task. To identify fake news Kareem et al. used seven different supervised learning classifications in their paper and compared results of classification[9] Lui et al. [10] proposed an ensemble framework in their paper to address the fake news classification challenge that was in ACM WSDM Cup 2019. Hakak et al. [11] also proposed an ensemble classification model to detect fake news. In Othman et al. [12] investigate the performance of different classification or clustering methods for a set of large data in their paper. Rubin et al. [13] proposed the SVM classification using five features. LR avoids general-purpose nonlinear optimization algorithms and its works well in text classification [14]. Naive Bayes algorithm is used in text classification because of its simplicity and effectiveness. A Naive Bayesian model is easy to build and it has no complicated iterative parameter estimation which makes it particularly useful for very large datasets [15]. SMO(SVM) is known as the hyperplane, between classes. This hyperplane separates the data into classes. It makes the path easier to get the result of text classification[16].This work aims to assess the performance of different classification algorithms using the WEKA data mining tool for classifying fake news. WEKA or Waikato Environment for Knowledge Analysis is a data mining and machine learning tool to help users make a wide range of sophisticated learning algorithms available through open-source packages [17].

2 Related work

The evolution of social and online platforms enables researchers to harvest various data from these dynamic mediums. Which eventually trigger the usage of classification algorithms to detect Fake news from such broadcasting sources. This robust and gigantic data facilitate researchers to train these classifiers with real-time scenarios to investigate the authenticity of the news. Two classifier approaches, the Support Vector Machine (SVM) and Naïve Bayes were noted as better-performing algorithms. The performance was evaluated based on their accuracy of detecting Fake news correctly in this approach [18]. In another work, Kareem et al. [9] noted the K Nearest Neighbors (KNN) as the best performing classifier with 70% accuracy followed by logistic regression with 69% accuracy on their dataset. They used seven different media fake news classification approaches on 344 news articles by scrapping popular news websites. They labeled the data to train their classifiers in two categories: Fake or True. They used two feature extraction techniques: Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF).

Beyond traditional direct classification approaches, there were several approaches [3,4,8,9] implemented with an ensemble framework. Liu et al. [3] proposed an ensemble framework to address the fake news classification challenge in ACM WSDM Cup 2019. They regarded this problem as the Natural Language Inference (NLI) task and proposed a novel empirical ensemble framework. This framework performed with more than 85% accuracy. In another approach of ensemble classification model for detection of the fake news, a better level of accuracy was achieved [11]. Their proposed model extracts relevant features from the fake news dataset. Then the extracted features were classified using the ensemble model comprising three popular machine learning models, i.e., Decision Tree, Random Forest, and Extra Tree Classifier. They achieved a training and testing accuracy of 99.8% and 44.15%, respectively, on the ISOT dataset. At the same time, the accuracy went up to 100% for the Liar dataset.

In another work, SVM classification-based approach, researchers proposed five feature-based models to identify satire and humor news articles [13]. In this work, 360 satirical news articles were explored from four domains and achieved 90% accuracy in detecting satire and humor. The final findings were reported based on three (Absurdity, Grammar, and Punctuation) features instead of five. Also, several approaches are implemented in different works where a limited number of instances are used or tested with a limited number of algorithms or used a dataset with fewer events or relied on crowdsourcing for validation, etc. [6, 8, 19]. Furthermore, Twitter threads were also characterized to understand their potential to create fictitious information [20]. And surprisingly enough, this microblogging platform was noted as one of the prominent sources to be used as evidence to produce fabricated news [21]. Though there are several approaches have been implemented to detect fake news. But still, we found this domain to be in scattered implementation with different smaller datasets. Therefore, we tried to fill this well-addressed research problem with a more comprehensive approach. We tested four different types (Function-based, Ensemble, Tree, Bayesian) of seven classifiers on the same dataset [22] . Although these papers have addressed different algorithms on different datasets. But a comparative analysis of the performance of the algorithms depending on Time and Accuracy is missing, In this paper, we tried to bridge the gap by introducing a comparative landscape on different algorithms on different experiment setups.

3 Methods

3.1 Data

Fake news detection is a text processing technique where the text has been used to detect its validity. In our dataset, we have text data from [23] which had 2 separate file with 4 columns such as “Title, Text, Subject, Date”. The news data are collected form social media focusing on three subjects i.e., Politics, world news, and others. The dataset has 21417 instances of true news and 23481 instances of fake news in 2 separate files. A new dataset was created after margining Fake.csv and True.csv [23] at random and reducing the dimensions [22]. The dataset used in this work comprises 7819 instances with four fields, namely ‘id’, ‘title’, ‘text’ and ‘label’ with some errors and extra characters. The initially collected dataset has been cleaned and processed in multiple steps to remove unwanted features, texts, symbols, unnecessary punctuations, as well as unreadable sentence structures. The text is further processed by removing invalid sequences/characters in Unicode language tools so that WEKA can work fine on the dataset. Then the CSV file was converted to attribute relation file format manually. Because WEKA ARFF loaders work better than CSV converters and are more reliable. After that, the dataset was ready for applying different WEKA Filters for further cleaning and processing. Several WEKA features have been used to perform these preprocessing of data such as, for stemmer “weka.core.stemmers.IteratedLovinsStemmer”; “weka.core.tokenizers.WordTokenizer” for tokenizing purposes. To change the strings to word vector “StringToWordVector” from weka.filters were used. After applying the filters, a dataset with 381 attributes and 254 instances was created [22]. The data cleaning and preprocessing workflow have been depicted in **Fig.1** for better visualization of the approach.

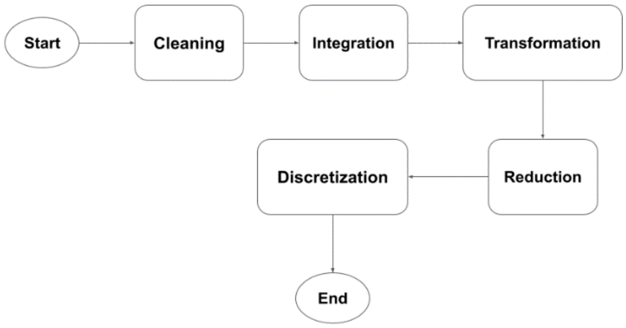

Fig. 1. Data Preprocessing Workflow Diagram

3.2 Experiment

The practical design concerns detecting the Fake news which is enhanced from text classification eventually. Two possible classes exist in the detection process, i.e., 'fake' and 'real'. Therefore, it's a binary classification problem and also non-linear. Algorithms such as Neural Network, Multilayer perceptron, Support Vector machine, Logistic Regression, Tree algorithms such as Decision Tree are ubiquitous in Text and fake news classification [9, 13, 20, 24, 25]. So, in this experiment the aim is to measure the performance in terms of time and correct classification rate of Multilayer Perceptron (MLP), Sequential Minimal Optimization (SMO), Logistic Regression (LR), J48, Random Forest (RF), Decision Tree (DT) and Naïve Bayes (NB) on the same dataset. For binary classification problems, logistic regression works very efficiently [26] whereas, MLP or Neural network is heavily used in Natural Language Processing (NLP) and text synthesis[27]. Though there might be some issues with the Decision tree classifier as it is based on nodes and our dataset is not the same. But still, we want to keep it along with J48. Random forest is an Ensemble classifier and it can easily overfit to noise in the data. Whereas, it tries to control the variance in the dataset. Naïve Bayes classifier is a well-known classifier using probabilities that eventually performs well in text classification tasks using count vectorization [28]. SVM is a very well-established classifier in various studies and also in fake news detection. It was found to show good results in identifying fake news correctly[18]. While implementing the experiment, all the parameters were set to default for achieving a neutral environment for all the algorithms.

As the study focused on the performance of the algorithms depending on time and correct classification, we used three different approaches namely, 10-fold Cross-Validation, 70% split, and 80% split of the dataset. Though, cross-validation is reported to have the problem of overfitting. Algorithms such as the Decision tree do prune and can face overfitting. We also checked the dataset by splitting it into two parts as training and testing. In 70% split, dataset split 70% as training dataset and 30% as testing dataset. We used a 70% split on the dataset to evaluate the performance more effectively. The most common split ratio is 80:20 that data scientists use. In 80% split, dataset split 80% as training dataset and 20% as testing dataset. We used an 80% split on the dataset to compare 70% split and 80% split in terms of time and accuracy. Ten-fold cross-validation, 70% split, and 80% split were implemented to measure the performance with default parameters for each of these seven classifier algorithms.

3.3 Function Classifiers

Logistic Regression (LR). Logistic regression is an algorithm used to predict the categorical dependent variable using a given set of independent variables. The logistic regression model is susceptible to "bad" data [29]. "Bad" data pointing the outlying responses and extreme points in the design space(X) [29]. The model takes the natural logarithm of the odds as a regression function of the predictors. The fundamental equation of the generalized linear model is,

$$g(E(y)) = \alpha + \beta x1 + yx2$$

It predicts the probability of occurrence of an event by fitting data to a logit function[30].

Multilayer Perceptron (MLP). Artificial neural networks are an alternative to many statistical modeling techniques used across different scientific sectors. A multilayer perceptron is eventually a form of artificial neural network. Most of the applications of MLP are related to classification, prediction, pattern recognition [31]. In general, MLP can be depicted as,

$$y = f(x)$$

Where,

y = [n * j]
x = [n * k]
n is the number of training instances
k is the number of input variables
j is the number of output variables

Backpropagation is used to find the weight optimizes the function y = f(x), where the x and y are training matrices.

Sequential Minimal Optimization (SMO). SMO or Sequential Minimal Optimization algorithm effectively trains support vector machines (SVMs) on classification[32]. Flake et al. [8] express the runtime of a single SMO step as,

$(p \cdot W \cdot n + (1 - p) \cdot \cdot n)$

Here,
p = the probability that the second Lagrange multiplier is in the working set
W = the size of the working set
n = the input dimensionality

SMO breaks significant quadratic programming problems into a series of most minor possible quadratic problems.

3.4 Tree Classifiers

Decision Tree (DT). A decision tree is widely used as a learning algorithm called Decision Tree Learning [33]. In a decision tree, each node contains a decision rule. Decision tree split based on the condition. A Dataset is assigned a label to characterize its data point [12]. The Model needs to learn features to take and corresponding correct threshold to optimally split the dataset. It is possible by information theory. When the information-theoretic point of view is pursued, the amount of average mutual information is gained at each tree level [34]. Information gain is calculated by comparing the entropy of the dataset. So, the way to quantify:

$$Entropy = \sum -p_i \log(p_i)$$

Here, p_i =probability of class i. Entropy is measured between 0 and 1. The state that gives minimum entropy is a pure node.

J48. J48 is an approach to discover the hidden relationships among data [35]. J48 has been considered the most efficient machine learning algorithm for predicting any crime dataset [35]. J48 is a decision tree algorithm based on ID3 and C4.5 algorithms [36]. It performs better both in performance and execution time. Kaur et al proposed [37] the modified J48 classifier to increase the accuracy of the data mining procedure. The algorithm is applied by the popular WEKA tool.

3.5 Ensemble Classifier

Random forest (RF). Random Forest classifier produces multiple decision trees. To decrease the correlation between decision trees, random forest considers controlling the term ρσ2 [33]. ρσ2 is the main part of the variance. Lee et al introduced average relative importance

$$I_j^2 = \frac{1}{B}\sum_{b=1}^{B} I_j^2(b)$$

WhereI_j^2(b) is the relative importance for the b-th decision trees[33]. RF classifiers can successfully handle high data dimensionality and multicollinearity, being both fast and insensitive to overfitting. Random Forest is a type of machine learning called bootstrap aggregation or bagging. Combining results from multiple models is called aggregation (majority votes). By bagging Random Forest algorithms gain better accuracy.

3.6 Bayesian classifier

Naïve Bayes (NB). The Naive Bayes algorithm is a simple probability classifier. It calculates a set of probabilities by counting the frequency and combinations of values in a given data set [28]. This classifier learns from training data. In this classifier, the conditional probability of each attribute Ai given the class label C [38, 39]. Naive Bayes is applied on some data set and the confusion matrix is generated for class having possible values. For example, in a news dataset the method follows:

$$\text{pr[E/H]} = N! * \prod_{i=1}^{k} \frac{{p_i}^{n_i}}{n_i!}$$

Here pr[E/H] is the probability of the document/news given its class H. and N is the number of words in the report.n_i is the time of occurrence of the word in the news. pi is the probability of obtaining the word from the news concerning category H.

3.7 Tools and Data preprocessing

Tools. For this experiment used tools are: Machine Configuration:

1.Cpu: I3 1005 (2core,4 thread)
2.Storage: SSD 256GB (R=465MB, W=375MB)
3.Ram: 8gb DDR4 2600mhz
3.No discrete GPU
4.Windows 10 pro 64bit
6.Weka 3.9
7.Python 3.9

4 Result

As discussed in the previous section, to investigate the performance of classification approaches, MLP, SMO, LR, J48, RF, DT, and NB algorithms were selected. WEKA uses ARFF file format which is “Attribute Relation File Format” where all the features are considered attributes and data are viewed as an instance. The experiment is performed in three parts for evaluating each of the algorithms. Those are related to performance at 10-fold cross-validation, 70% split, and 80% split. Hereafter, these three approaches are addressed as ex1, ex2 and ex3, respectively. The experimental results have been elaborated in several tables and figures for better visualization. The performances of these algorithms have been compared using five different measures: Accuracy, Incorrectness, Time taken, Kappa statistic and finally comparing Accuracy. The accuracy has been compared in three possible combinations: cross-validation vis-à-vis 70% split, cross-validation vis-à-vis 80% split and 70% split vis-à-vis 80% split.

Table 1. Accuracy of the Algorithms

Algorithm	Cross-Validation	70% split	80% split
MLP	78.7402	72.3684	72.5490
SMO	73.6220	73.6842	68.6275
LR	77.5591	80.2632	76.4706
J48	64.9606	67.1053	64.7059
RF	82.2835	78.9474	82.3529
DT	67.3228	67.1053	68.6275
NB	81.8898	85.5263	78.4314

The first experimental result has been represented in terms of the correctness of each of these algorithms in three approaches, i.e., 10-fold Cross-Validation, 70% Split and 80% Split. In **Table 1.** & **Fig.2**, all the classifiers can be seen visualized with their accuracy (in percentage) for all these three approaches. Among all the implementations, NB classifiers were prominent with more than 85% accuracy for the ex2 approach. This performance is followed by RF having more than 82% accuracy for both ex1 and ex3. However, the performance of RF in ex2 is relatively low but not much behind with more than a 78% accuracy level. Among all the other algorithms, LR only crosses the 80% accuracy level in the ex2.

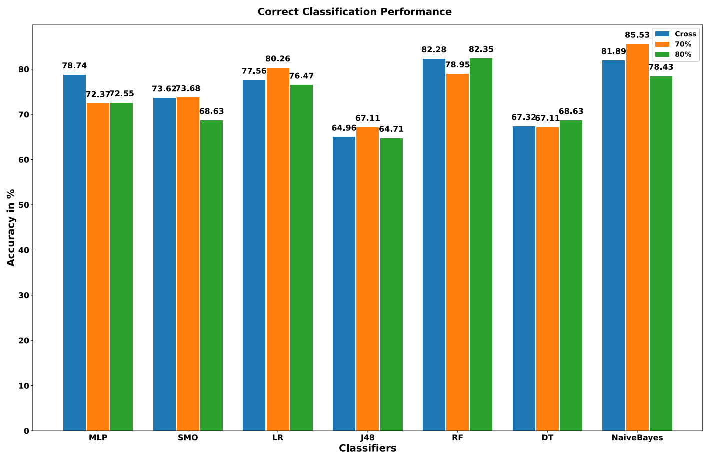

Fig. 2. Accuracy Levels of All Classifiers (in Different Experimental Setups)

The effective performance of NB is also noted in previous works [9] [24] though the accuracy levels were found lower with 75% and 63%, respectively. In earlier work [9], LR and SVM performed with lower accuracy levels around 60%, but we found both these classifiers performed way ahead in this experimental setup. With more than 80% in the case of LR (in ex2 arrangement), SMO performed way ahead of 70% accuracy in both ex1 and ex2 design. Though the performance of SMO in ex3 is relatively lower with 68% accuracy but still better than previously noted [9]. In the case of DT, the current experimental setup found to perform poorly comparing previous implementation [25]. Where, 70% accuracy level was achieved using DT with N-gram analysis in contrast we found less than 70% accuracy in all three experimental setups. Still the lagging is not much as the highest performance of DT is noted more than 68% in ex3 configuration. For better visualization, we have also plotted the percentages of wrongly classified news by all these algorithms in different experimental setups in **Fig.3**. From this representation, it is more apparent that J48 performed with significantly lower accuracy in all three experimental setups. DT follows this in our implementation and obviously NB performed best for ex2 whereas for ex1 & ex3, RF performed best with the lowest wrong classification percentages respectively.

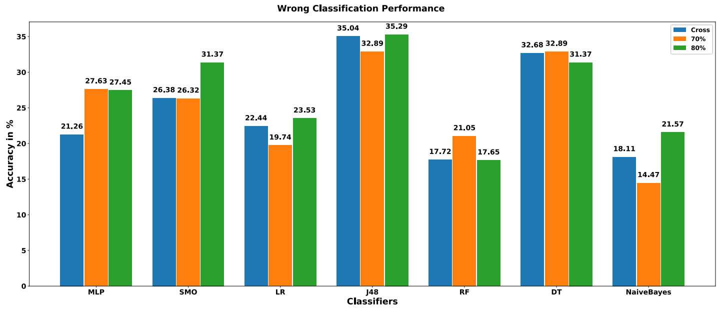

Fig. 3. Statistics of Wrongly Classified Data (in Percentage).

Kappa statistics are used for evaluating the accuracy of classifiers. Kappa is robust and can be used in both nominal and ordinal data. The original intent of Cohen's Kappa was to measure the degree of agreement or disagreement of two or more people observing the same phenomenon [40, 41]. From **Fig.4** we can see that the Random Forest has the highest kappa value. And Naive Bayes has the second-highest value. If we compare **Fig.4** with Landis and Koch [42] interpretation metrices, we can see that most of the MLP is at a "moderate" level as the Kappa value is between 0.41-0.60. SMO belongs to Moderate level for cross validation and 70% split but for ex3, performance reduces and the value is between "0.21-0.40" can be interpreted as a fair level. LR is at moderate level with a value between "0.52-0.602". J48 and DT have the lowest level of "Fair" in all the cases and the performance is inferior concerning other algorithms. RF and NB algorithms have the highest level of significance, which is "Substantial" with values between 0.61-0.75. In the case of RF in ex2 and NB in ex3 the performance reduces which is "moderate". So, kappa Statistics also shows that Naïve Bayes is the top-performing algorithm for this dataset followed by RF and LR.

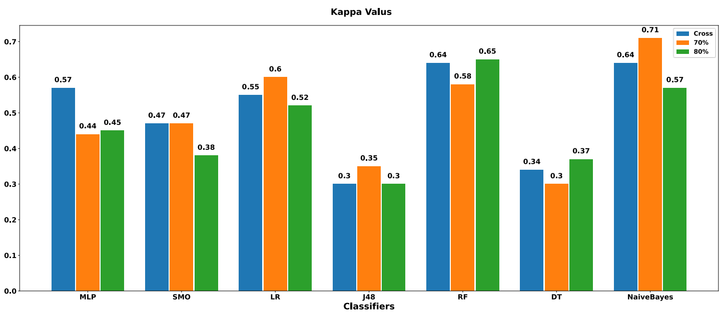

Fig. 4. Kappa Statistic for all the Classifiers in Different Experimental Setups

Fig. 5 represents the time taken by algorithms on logarithmic scale. From the **Fig.5** we can see that for all the three experimental arrangements, the highest time is taken by MLP, and the Lowest time is taken by Naive Bayes. MLP took the highest time at ex3 which is 113.83 seconds and Naive Bayes took the highest time at ex1 which is 0.04s. SMO took slightly more time than Naive Bayes in all three experimental setups. In ex1, SMO took 0.1s which is the highest time among all these three experimental designs. Decision Tree is in third place with a consistent time. It took nearly the same time in all 3 experiments. the highest time taken by Decision Tree is for ex3. Logistic regression is at the fourth position with a highest time of 0.21s for ex2. J48 is in the fifth position with the highest time of 0.36s among three experiments. Random forest is in the sixth position with nearly consistent speed. It took 0.45s in both ex2 and ex3 and with 0.47s for ex1 placed in the seventh position.

If we consider time with the accuracy, we can see that Naive Bayes took the lowest time and gave the highest accuracy. All the algorithms took more time in Cross-validation among the three experiments except for Logistic Regression. It took significantly less time and the accuracy was 77.5591%. Considering the Time taken by this algorithm Naïve Bayes is still on the first place because it took only 0.04s, 0.01s, 0.01s in three tests. But the Random Forest algorithm took 0.4s,0.45s,0.45s in three tests. Logistic Regression took less time than Random Forest. It took 0.17s, 0.21s, 0.2s in three tests. But MLP took the highest time of 104.5s,103.69s,113.83s in three tests. From **Fig. 5** and **Fig. 2** it can be seen that Naive Bayes is clearly on the top position considering time and accuracy Logistic regression is in the 2nd place and Random Forest is in the third position.

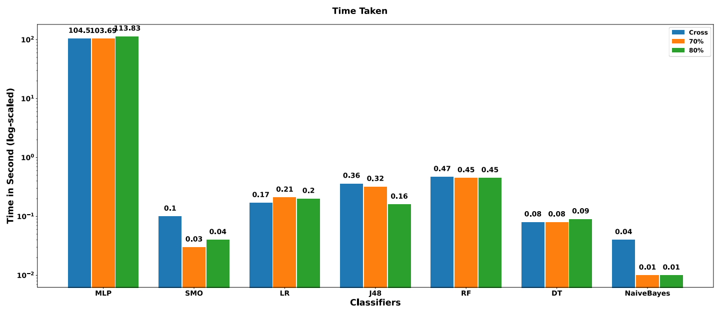

Fig. 5. Time graph

In final segment of experimental result representation, the difference of accuracy levels for all these seven classifier approaches have been computed and listed in **Table 2**. The difference has been computed between two experimental setups which make three combinations. In the first pair of 10-fold cross-validation (ex1) vs. 70%split (ex2), MLP performs better for ex1 comparing ex2 followed by RF which is also performed better in ex1. Other than these two all the negative differences indicate that ex2 is better setup for these algorithms than ex1. Though for SMO & DT it can't be said evidently as the difference is too less to indicate any significance between these setups. Therefore, it can be concluded. These two algorithms are working indifferently irrespective of the experimental design.

Table 2. Comparing Performances of Algorithm in terms of Accuracy Difference

Algorithm	Cross-Validation vis-à-vis 70% split	Cross-Validation vis-à-vis 80% split	70% split vis-à-vis 80% split
MLP	6.37	6.19	-0.18
SMO	-0.06	4.99	5.06
LR	-2.7	1.09	3.79
J48	-2.14	0.25	2.4
RF	3.34	-0.07	-3.41
DT	0.22	-1.3	-1.52
NB	-3.64	3.46	7.09

In contrast, ex3 seems to be a less proper setup for almost all the algorithms except DT and RF though the insignificant difference in the case of RF fails to highlight any suitability between ex1 and ex3. Other than these two, for all the algorithms ex1 is found to be more suitable except J48 where the difference is insignificant. For the 80% split, the lower performance has been noted comparing 70% split in four of the algorithms (SMO, LR, J48, NB) with a significant margin. In the MLP also, ex3 is favorable set up with very little margin comparing ex2. These significant margins favoring ex2 (70% splits) show that the commonly used 80% split setup is not much suitable with these algorithms for our dataset. This is a kind of new enlightenment where; it can be said the commonly used 80% split setup should not be considered alone as an excellent setup to evaluate classifiers. As it is a well-known fact that, machine learning algorithms are performed differently on different datasets. Similarly, the split setup gives an altered picture for the different dataset.

5 Conclusion

This study aims to compare the performance of seven algorithms using three different experimental setups for detecting fake news. The study is based on two key aspects of these algorithms, accuracy levels and time consumed to classify. Furthermore, it compares these algorithms in different experimental based on their experimental setups. Also, the significance has been measured Kappa statistics. The top-performing algorithms are Naïve Bayes, followed by the Random Forest and Logistic Regression. But in terms of time Naïve Bayes and Logistic Regression outperformed the Random Forest algorithm. Also, in the experimental setup, 70% split setup is more suitable than an 80% split setup for most of the algorithms. K-fold Cross Validation (here K=10) is an ideal for Perceptron based approach. It is well known that Deep Learning algorithms show great accuracy but comes with huge time and resource overhead. Perceptron based approach in our experiment also shows the same. On the other hand, Algorithms such as Sequential Minimal Optimization, Logistic Regression, Decision Tree, J48, Random Forest, and Naïve Bayes Classifiers are easier to implement and can show better results in some cases and in this paper we compared algorithms such as these on different experiments to compare the performance on fake news dataset.

References

1. Jr., E.C.T., Lim, Z.W., Ling, R.: Defining “Fake News.” https://doi.org/10.1080/21670811.2017.1360143. 6, 137–153 (2017). https://doi.org/10.1080/21670811.2017.1360143.
2. ZhouXinyi, ZafaraniReza: A Survey of Fake News. ACM Comput. Surv. 53, (2020). https://doi.org/10.1145/3395046.
3. ShuKai, SlivaAmy, WangSuhang, TangJiliang, LiuHuan: Fake News Detection on Social Media. ACM SIGKDD Explor. Newsl. 19, 22–36 (2017). https://doi.org/10.1145/3137597.3137600.
4. Lazer, D.M.J., Baum, M.A., Benkler, Y., Berinsky, A.J., Greenhill, K.M., Menczer, F., Metzger, M.J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S.A., Sunstein, C.R., Thorson, E.A., Watts, D.J., Zittrain, J.L.: The science of fake news. Science (80-.). 359, 1094–1096 (2018). https://doi.org/10.1126/SCIENCE.AAO2998.
5. Zhang, X., Ghorbani, A.A.: An overview of online fake news: Characterization, detection, and discussion. Inf. Process. Manag. 57, 102025 (2020). https://doi.org/10.1016/J.IPM.2019.03.004.
6. Gupta, A., Kumaraguru, P.: Credibility ranking of tweets during high impact events. ACM Int. Conf. Proceeding Ser. (2012). https://doi.org/10.1145/2185354.2185356.
7. Mohd Shariff, S., Zhang, X., Sanderson, M.: User Perception of Information Credibility of News on Twitter. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 8416 LNCS, 513–518 (2014). https://doi.org/10.1007/978-3-319-06028-6_50.
8. Finding true and credible information on Twitter | IEEE Conference Publication | IEEE Xplore, https://ieeexplore.ieee.org/abstract/document/6915989, last accessed 2021/08/30.
9. Kareem, I., Awan, S.M.: Pakistani Media Fake News Classification using Machine Learning Classifiers. 3rd Int. Conf. Innov. Comput. ICIC 2019. (2019). https://doi.org/10.1109/ICIC48496.2019.8966734.
10. Liu, S., Liu, S., Ren, L.: Trust or Suspect? An Empirical Ensemble Framework for Fake News Classification.
11. Hakak, S., Alazab, M., Khan, S., Gadekallu, T.R., Maddikunta, P.K.R., Khan, W.Z.: An ensemble machine learning approach through effective feature extraction to classify fake news. Futur. Gener. Comput. Syst. 117, 47–58 (2021). https://doi.org/10.1016/J.FUTURE.2020.11.022.
12. Othman, M.F. Bin, Yau, T.M.S.: Comparison of different classification techniques using WEKA for breast cancer. IFMBE Proc. 15, 520–523 (2007). https://doi.org/10.1007/978-3-540-68017-8_131.
13. Rubin, V.L., Conroy, N.J., Chen, Y., Cornwell, S.: Fake News or Truth? Using Satirical Cues to Detect Potentially

Misleading News. 7–17.
14. Komarek, P., Moore, A.: Fast Logistic Regression for Data Mining, Text Classification and Link Detection. 1–8 (2003).
15. Alsari, A.B.A.: Short Text Classification using Machine Learning Techniques. (2018).
16. Qadi, L. Al, Rifai, H. El, Obaid, S., Elnagar, A.: Arabic text classification of news articles using classical supervised classifiers. 2019 2nd Int. Conf. New Trends Comput. Sci. ICTCS 2019 - Proc. (2019). https://doi.org/10.1109/ICTCS.2019.8923073.
17. Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. Part F1288, 847–855 (2013). https://doi.org/10.1145/2487575.2487629.
18. Ahmed, A.A.A., Aljabouh, A., Donepudi, P.K., Choi, M.S.: Detecting Fake News Using Machine Learning : A Systematic Literature Review. Psychol. Educ. J. 58, 1932–1939 (2021).
19. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on Twitter. Proc. 20th Int. Conf. Companion World Wide Web, WWW 2011. 675–684 (2011). https://doi.org/10.1145/1963405.1963500.
20. Buntain, C., Golbeck, J.: Automatically Identifying Fake News in Popular Twitter Threads. Proc. - 2nd IEEE Int. Conf. Smart Cloud, SmartCloud 2017. 208–215 (2017). https://doi.org/10.1109/SMARTCLOUD.2017.40.
21. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. Science (80-.). 359, 1146–1151 (2018). https://doi.org/10.1126/SCIENCE.AAP9559.
22. Fakenews Dataset(csv and arff) | Kaggle, https://www.kaggle.com/abkafi/fakenews-datasetcsv-and-arff, last accessed 2021/09/10.
23. Bisaillon, Clément: Fake and real news dataset | Kaggle, https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset, last accessed 2021/08/24.
24. Granik, M., Mesyura, V.: Fake news detection using naive Bayes classifier. 2017 IEEE 1st Ukr. Conf. Electr. Comput. Eng. UKRCON 2017 - Proc. 900–903 (2017). https://doi.org/10.1109/UKRCON.2017.8100379.
25. Keskar, D., Palwe, S., Gupta, A.: Fake News Classification on Twitter Using Flume, <Emphasis Type="Italic">N</Emphasis>-Gram Analysis, and Decision Tree Machine Learning Technique. 139–147 (2020). https://doi.org/10.1007/978-981-15-0790-8_15.
26. Ali, S., Smith, K.A.: On learning algorithm selection for classification. https://doi.org/10.1016/j.asoc.2004.12.002.
27. Goldberg, Y.: Neural Network Methods for Natural Language Processing. http://dx.doi.org/ 10.2200/S00762ED1V01Y201703HLT037 . 10, 1–311 (2017). https://doi.org/10.2200/S00762ED1V01Y201703HLT037.
28. Saritas, M.M., Yasar, A.: Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification. Int. J. Intell. Syst. Appl. Eng. 7, 88–91 (2019). https://doi.org/10.18201//IJISAE.2019252786.
29. Pregibon, D.: Logistic Regression Diagnostics. https://doi.org/10.1214/aos/1176345513. 9, 705–724 (1981). https://doi.org/10.1214/AOS/1176345513.
30. Kleinbaum, D.G., Klein, M.: Logistic Regression. (2010). https://doi.org/10.1007/978-1-4419-1742-3.
31. Garner, S.R.: WEKA: The Waikato Environment for Knowledge Analysis. Proc New Zeal. Comput. Sci. Res. Students Conf. 57–64 (1995).
32. Flake, G.W., Flake, G.W., Lawrence, S.: Efficient SVM Regression Training with SMO. Mach. Learn. 46, 1--3 (2000).
33. Lee, T.-H., Ullah, A., Wang, R.: Bootstrap Aggregating and Random Forest. Adv. Stud. Theor. Appl. Econom. 52, 389–429 (2020). https://doi.org/10.1007/978-3-030-31150-6_13.
34. Safavian, S.R., Landgrebe, D.: A Survey of Decision Tree Classifier Methodology. IEEE Trans. Syst. Man Cybern. 21, 660–674 (1991). https://doi.org/10.1109/21.97458.
35. Ivan, N., Ahishakiye, E., Omulo, E.O., Taremwa, D.: Crime Prediction Using Decision Tree (J48) Classification Algorithm. Int. J. Comput. Inf. Technol. 2279–0764 (2017).
36. Ozbay, F.A., Alatas, B.: Fake news detection within online social media using supervised artificial intelligence algorithms. Phys. A Stat. Mech. its Appl. 540, 123174 (2020). https://doi.org/10.1016/J.PHYSA.2019.123174.
37. Kaur, G., Chhabra, A.: Improved J48 Classification Algorithm for the Prediction of Diabetes. Int. J. Comput. Appl. 98, 975–8887 (2014).
38. Bouckaert, R.R.: Properties of Bayesian Belief Network Learning Algorithms. Uncertain. Proc. 1994. 102–109 (1994). https://doi.org/10.1016/B978-1-55860-332-5.50018-3.
39. Buntine, W.: Theory Refinement on Bayesian Networks. Uncertain. Proc. 1991. 52–60 (1991).

https://doi.org/10.1016/B978-1-55860-203-8.50010-3.
40. Cohen, J.: A Coefficient of Agreement for Nominal Scales: http://dx.doi.org/10.1177/001316446002000104. 20, 37–46 (2016). https://doi.org/10.1177/001316446002000104.
41. Ben-David, A.: Comparison of classification accuracy using Cohen's Weighted Kappa. Expert Syst. Appl. 34, 825–832 (2008). https://doi.org/10.1016/J.ESWA.2006.10.022.
42. Landis, J.R., Koch, G.G.: An Application of Hierarchical Kappa-type Statistics in the Assessment of Majority Agreement among Multiple Observers. Biometrics. 33, 363 (1977). https://doi.org/10.2307/2529786.