A Comparative Approach to Detecting COVID-19 Fake News through Machine Learning Models

Zyad Al-Azazi and Ramzi A. Haraty
Department of Computer Science and Mathematics
Lebanese American University
Beirut, Lebanon
Email: zyad.alazazi@lau.edu, rharaty@lau.edu.lb

***Abstract*--Identifying fake news has become an increasingly challenging task in recent years, with the proliferation of digital media and the ease of spreading misinformation. The problem has only become more complex with the global pandemic situation, as false information about COVID-19 can have serious consequences for public health and safety. Fortunately, the same technological advancements that have made it easier to spread fake news have also enabled potential solutions to this problem. In this work, we aimed to test and evaluate approaches for automatically classifying fake news. We focused specifically on fake news related to COVID-19, given its widespread impact on public health and the urgency of addressing misinformation in this area. To do this, we trained and evaluated several machine learning models using a dataset of news articles labeled as either "fake" or "real." Our goal was to identify the most accurate and effective model for detecting COVID-19 related fake news. After testing several models, we found that an SVM classifier performed the best, achieving an accuracy of 93.83%. We also conducted an analysis of each model's performance, examining factors such as feature selection and model complexity that may have influenced their results.**

***Keywords**: Machine Learning, SVC Classifier, COVID-19.*

I. INTRODUCTION

Alongside the health and economic crisis that the COVID-19 pandemic is still causing in the whole world, there is another form of crisis that started before the pandemic but has grown to be more threatening to all people under the current circumstances – the COVID-19 fake news crisis. According to Cambridge dictionary, "fake news" is a term that refers to false stories that appear to be news, spread on the internet or other media, usually created to influence political views or as a joke [1]. This phenomenon has been evolving and most people have been exposed to fake news at some point. The main reason for our exposure to fake news is social media. According to [2], the use of social media has increased among the American adults by 1 – 2 additional hours per day as of March 2020. Furthermore, 46% of respondents on a survey stated that their use of social media was to stay up-to-date with the news [3]. Other reasons for the continuously increased widespread of fake news, include political influence, lack of awareness towards fake news, the higher levels of uncertainty and distrust and the existence of fake-news-spreading bots [4].

Experts have come to categorize fake news; according to [5], not all fake news is considered the same. One type of fake news is caused by "disinformation," which is an intentional spread of fake news with the intention of causing tension for a pre-set agenda. For example, the rumors affiliating certain minority groups or communities to the spread of the disease belong to this category. The other category is caused by "misinformation." This type of fake news is usually spread unintentionally due to the lack of awareness by the people, who believe they are correct. Unfortunately, both categories are harmful as they could lead, in worst case scenarios, to the loss of people's lives. Nevertheless, different social media platforms have been trying to combat the spread of fake news through different policies and mechanisms. However, despite all the efforts, fake news still finds its way to reach and affect us and our beloved ones.

In this paper, we aim to contribute to the already-existing approaches by testing and evaluating the effectiveness of different machine learning models to identify fake news regarding COVID-19. First, we will start by reviewing the previous attempts to successfully detect fake news presented by researchers in the field. Then, we will explain our methodology in detail. Next, we will showcase the results we have obtained in our work and try to analyze and explain the underlying reasons. Finally, we will be talking about the different limitations we have encountered in this project.

II. LITERATURE REVIEW

There has been great interest in attempting to employ the machine learning methodologies to classify and detect fake news, whether it be on social media or from different articles taken from the self-proclaimed news agencies and websites. This interest has translated into literature demonstrating diverse approaches to this problem. In this section, we will dive deeper into some of these attempts and discuss their results.

In [6], the focus was to understand the influence of different features to distinguish between real and fake news. The dataset used consisted of news articles concerned with the 2016 US elections along with Facebook shares and reactions towards these articles. The study also considered

any media content that was embedded in these articles and extracted text from them. The total number of features studied was 141 textual features categorized according to the following: features from news content, features from news source and features from the environment (social media). The results indicated the supremacy of the RF and XGB models in their classification with both yielding an F-1 score of approximately 81%.

A different approach was introduced in [7] by considering deep learning techniques. The dataset used in this study included tweets about the 2010 Chile Earthquake. The study utilized different approaches for featurization (TF-IDF and Count Vectors), in addition to Word Embeddings. TF-IDF showed to be a better featurization technique with the accuracy of models relying on it being higher than models that depended on Count Vectors. The highest F-1 scores were achieved by the SVM and Naïve Bayes models with scores of 94%. On the other hand, LSTM reported better accuracy results than RNN with a score of 76%.

There was also another attempt by [8] that was primarily focused towards evaluating the performance of machine learning and deep learning models in detecting fake news in social networks. The approaches to featurization in this study were TF-IDF and PCFG (Probabilistic Context-free Grammar) using bigram frequency. As for the dataset, it consisted of labeled tweets related to five rumor stories. In [8], it was noted that the use of TF-IDF solely showed significant predictive power despite the neglection of named entities. It was concluded that the hybrid model of CNN (Convolutional Neural Networks) and RNN (Recurrent Neural Networks) with TF-IDF showcased the best performance out of all the approaches surveyed in the study.

Another recent extensive research tackled the problem from two different angles. In [9], one methodology was concerned with classifying fake news using machine learning models and transformer-based deep learning models. The results showed a huge advantage for the transformer-based models over the ML models with the XLNET-base scoring an F-1 score of 98%; the highest ML model was the XGBoost scoring 90%. The other angle was concerned with detecting twitter bots specialized in spreading fake news. The solution to this problem was a voting classifier depending on the output of three ensemble classifiers: RFC, AdaBoost and XGBoost. It was found that an accuracy score of 96% was achievable by an RFC model on one feature, which was the "duration between the account creation and tweet date." Adding other features resulted in accuracy scores ranging between 96% and 99%.

There was only one paper that focused on classifying fake news related to COVID-19. [10] created and annotated a textual dataset themed around the fake news of COVID-19. Moreover, they attempted to benchmark the dataset using different machine learning algorithms. The results showed that the highest F-1 scores were scored by the SVM and the LR models – 93.46% and 92.75%, respectively.

III. METHODOLOGY

This section will be discussing the details regarding the methodology applied in this project. We will be describing the dataset, the data cleaning process, the file ingestion and schema validation technique and the machine learning models used.

A. Dataset

The dataset contains 10,700 rows of textual English news labeled as "real" or "fake." The phrase "real news" under the context of our dataset is used to describe tweets about COVID-19 that were true and came from verified sources, whereas "fake news" is used to describe tweets, posts and articles that have been proved to be not true. The dataset was collected from different sources; the "fake news" was collected from different fact-checking platforms and tools, such as PolitiFact, Google fact-check-explorer and IFCN explorer, while the "real news" was from the verified twitter accounts [10].

The dataset is divided among three files: training, testing and validation. Training data constitutes 60% of the overall data, while testing, as well as validation, are each 20% of the total percentage of the data. We combine both the validation and training sets, such that we have 80% of the data for training and validation purposes and 20% for testing. The percentage of news classified as "real" is 52.34%, whereas the news classified as "fake" is 47.66%. We used word cloud to present the most frequent words in both the real news and fake news as can be observed in figures (1) and (2).

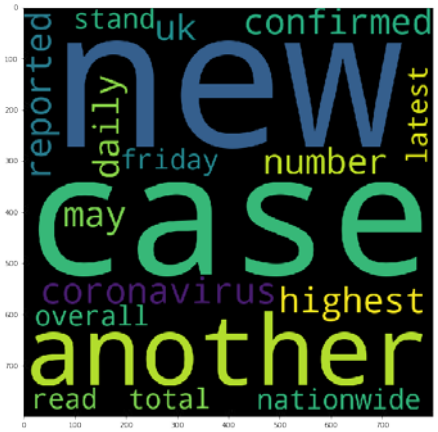

Fig. 1. Most frequent words in "real" news.

B. File Ingestion and Schema Validation

One of the best practices in software development, especially in data science since the majority of work relies on datasets, is to use YAML files. YAML is a recursive acronym that stands for (Ain't a Markup Language); this type of files is commonly used for configuration files. The purpose of such a file is to store information about the properties of the dataset or datasets we are expecting to work with, such as the names of the columns, the data types, the file name, and the

file type. Hence, it is very practical since we only need to change the name of the dataset file in the YAML instead of directly changing our code.

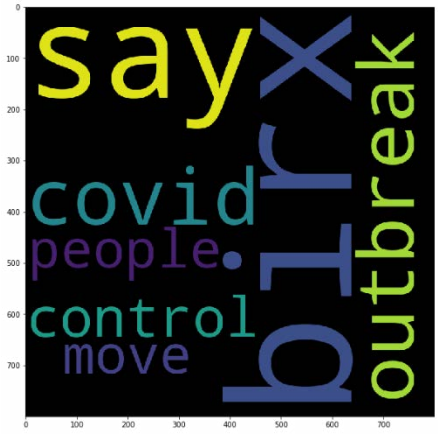

Fig. 2. Most frequent words in "fake" news.

C. Data Preprocessing and Cleaning

The preprocessing of data goes through multiple steps to obtain the final cleaned text. First, we start by preprocessing hashtags to convert them to words by removing the '#' and '_' characters, then we separate any words that are not separated by space characters using the upper-case letters as an indication of a new word's start character. The next step is to remove any URLs or special twitter characters, such as the retweet character ("RT"). Then, we lower the case of all the letters and expand any contracted words, for example the phrase "should've" will be separated to "should have." Next, we remove any non-alphanumeric characters, English stop words and extra space characters. Finally, we lemmatize the words using the "WordNetLemmatizer." It should be noted that a great portion of the cleaning is done through regular expressions.

D. Featurization Method

We use the TF-IDF vectorization method to represent the frequency of the term (t) using a value that assesses the importance of a word in a set of documents according to the following formula:

$$TF - IDF(t) = \mathrm{TF(t, d)} \times IDF(t)$$

where (d) is the document that the term (t) occurs in and:

$$TF(t) = \frac{Number\ of\ occurrences\ of\ \boldsymbol{t}\ in\ \boldsymbol{d}}{Total\ number\ of\ terms\ in\ a\ document}$$

$$IDF(t) = \log_e \frac{1 + Total\ number\ of\ documents}{1 + Number\ of\ documents\ containing\ \boldsymbol{t}} + 1$$

IV. MACHINE LEARNINH MODELS

A. Gaussian Naïve Bayes

The Naïve Bayesian classifier is one of the most primary classifiers. It is based on Bayes' theorem; however, it also assumes that all predictors are independent from each other. For example, for events A and B, the $P(A)$ does not affect the $P(B)$; hence:

$$P(A/B)P(B) = P(B/A)P(A)$$

Since:

$$P(A/B) = P(A) \text{ and } P(B/A) = P(B)$$

Consequently, if we have more than one condition (predictors in our case):

$$P(A/BCD) = P(A/B) \times P(A/C) \times P(A/D)$$

B. Logistic Regression

This is one of the classical and most used statistical models in binary classifications. Assuming that we have two classes (0 and 1), and we have sample (X) that we are attempting to classify; the logistic function will look like this:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

When fitting the logistic model, we use the maximum likelihood method, where we try to estimate best values for (β_0) and (β_1) so that we can obtain a predicted probability $\hat{p}(x_i)$ that closely corresponds to the default class of the sample (X) [11]. As for the maximum likelihood function:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} \hat{p}(x_i) \prod_{i:y_i=1} (1 - \hat{p}(x_i))$$

Figure 3 shows the difference between linear regression and logistic regression. We can notice that logistic regression has a sigmoid function, and the points are either classified as part of class 0 or class 1. Meanwhile, in linear regression, points do not have to lie within the boundaries of these two classes.

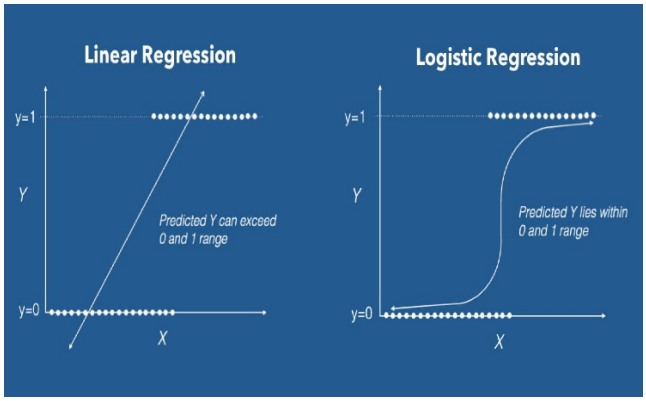

Fig. 3. The Difference between Linear and Logistic Regression (taken from machinelearningplus.com).

C. Support Vector Machines (SVM)

An SVM is an extension of the Support Vector Classifier (SVC) that results from using kernels. While the original SVC uses linear decision boundaries to distinguish between classes, SVMs allow non-linear decision boundaries. The main factor behind this robustness is the diversity in kernels;

each kernel has a different decision boundary. These decision boundaries do not conform to a specific number of dimensions as they are "hyperplanes." For example, if we have (p) dimensions, a hyperplane would be a flat affine subspace of dimension (p-1) [11].

To illustrate the concept of kernels in a better way, let us observe figure 4. We can notice the significant difference between the different kernels in bounding the different classes in the example. The linear kernel has the most rigid boundaries between the classes; whereas, the RBF (radial basis function) kernel has the most flexible decision boundaries.

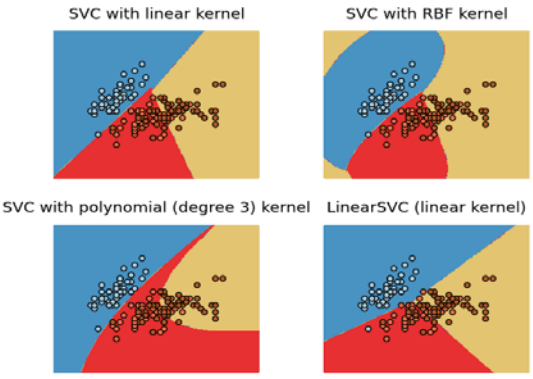

Fig. 4. Different Types of SVM Kernels (taken from scikit-learn.org).

D. Random Forest

This is a bagging (bootstrap and aggregating) statistical model that relies on constructing multiple decision trees; each of these trees will randomly select a few samples and several features. Accordingly, the training will be performed in each tree independently. The final classification decision of each testing sample is usually a majority vote among all the decision trees [11]. The way the samples are placed in a single decision tree is usually based on one of the features (as demonstrated in figure 5).

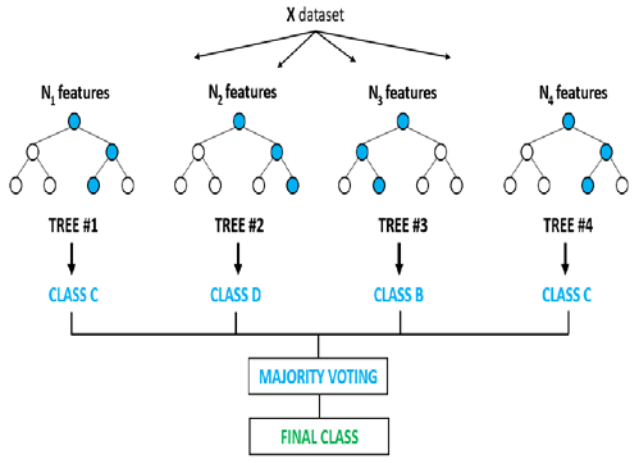

Fig. 5. Majority Voting in Random Forest Classifier (taken from medium.com).

E. Training Models

After finalizing the distribution of data, 80% for training and validation with 20% for testing, we started by training the models that do not require any hyperparameters to be tuned – Gaussian Naïve Bayes and Logistic Regression. Then, we started with the rest of the models that required hyperparameter tuning. For the SVC with linear kernel, we had only one parameter that needed to be tuned, which was 'C.' This hyperparameter controls the error in our SVM; the lower the C value is, the less error our model will report.

As for the SVC with the RBF kernel, we also had the 'Gamma' hyperparameter besides 'C.' Gamma is an indicator of the curvature of the decision boundary; the higher the Gamma value, the more flexible the decision boundary is. On the other hand, Random Forest had the largest number of hyperparameters to tune. These hyperparameters were the number of trees, the maximum depth a tree can reach, the minimum number of observations to split a node and the minimum number of samples to be present in the leaf node after splitting. To tune our hyperparameters, we used grid search cross-validation with k-fold cross-validation, where k = 3. All the training and testing of the models was performed on a local machine. Also, the models were directly imported from the sci-kit learn module in Python.

V. EXPERIMENTAL RESULTS

After an aggregate training time of over 34 hours, the models were tested on the testing dataset – a portion of the data that none of the models have encountered during the training process. For performance measurement purposes we are using multiple metrics:

- **Accuracy:** the number of correctly classified samples over the total number of samples.

$$\boldsymbol{Accuracy} = \frac{True\ Positives + True\ Negatives}{Total\ Number\ of\ Observations}$$

- **Precision:** the number of correctly classified positives over the total number of positive classifications by the model.

$$\boldsymbol{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

- **Recall:** the number of correctly classified positive observations over the total number of positive observations in the original dataset.

$$\boldsymbol{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- **F-1 score:** the weighted average of precision and recall.

$$\boldsymbol{F-1\ Score} = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

Table (1) demonstrates the results we obtained from testing the previously discussed models. We can clearly notice the supremacy of the SVM models, especially the SVM with the radial kernel, over all the other models with an F-1 score of 93.83%. On the contrary, we can see that the lowest results belong to the Naïve Bayes model with an F-1 score of 84.51%

Table 1: Comparison of Model Prediction Results

Model	Accuracy (%)	Precision (%)	Recall (%)	F-1 Score (%)
Gaussian Naïve Bayes	84.63%	85.17%	84.63%	84.51%
Logistic Regression	92.06%	92.1%	92.06%	92.06%
SVM (Linear Kernel)	93.5%	93.51%	93.5%	93.51%
SVM (RBF Kernel)	93.83%	93.9%	93.83%	93.83%
Random Forest	87.66%	88.26%	87.66%	87.65%

A. Discussion and Analysis

If we compare the results obtained in our work with the results obtained in [10], we will clearly notice the slightly better performance achieved by the SVM model with radial kernel. There are multiple reasons that could have contributed to this performance; the most obvious would be the hyperparameter tuning we performed on the SVM and Random Forest models. Another reason could be the cleaning process we applied on the data; it removed numerous meaningless features that we could have captured, such as contracted words and meaningless hashtags that could have added more features to our TF-IDF implementation with no actual semantic value. Nevertheless, the overall supremacy of SVM corresponds to the results in [7] and [10]; this could be explained by the robust and high-dimensionality considerate nature of the SVM. However, our preprocessing approach may have negatively affected the performance of the logistic regression model as it was recorded achieving a slightly less accuracy and f-1 scores than the logistic regression model in [10]. The reason behind this setback could be the number of features that resulted; too many features or even less features than needed could have had a negative effect on the classification power of the logistic regression model. As for the RF model, one of the hyperparameters – number of trees – was tuned to the highest value we put in the set of potential values, which may be an indicator of the higher predictive power the model could have achieved if the highest bound of the potential values was raised. The relatively poorer performance by the GNB model can be explained by the independence condition that the statistical model is based on; we noticed the existence of a set of highly frequent common words among the news in the same class, which is a feature that the model does not consider; hence, less predictive power was attained.

B. Limitations

This project has many limitations but the main limitation that affects the whole problem of attempting to classify fake news is the definition of fake news. Defining the term "fake" in the context of news and information transfer must take into consideration many factors, including the time when the news was released. For example, one sample of the dataset was about the existence of a COVID-19 vaccine; in the dataset, this news was classified as "fake." However, in our present context, this news could be actually "real." Thus, it is important to consider the context, which is a feature that was considered in [9].

Another limitation is the dataset nature that hindered the exploration and utilization of other hand-crafted features. For example, we believe that the inclusion of the number of words in a single statement of news would have created a clear bias since the "real" news in the dataset were mainly taken from Twitter [10] – a social platform that restricts the length of the single tweet to 280 characters. Meanwhile, the "fake" news was taken from other news platforms that had no character number restrictions.

A final limitation that could have enhanced the performance of the models with hyperparameters is the available computational resources. As mentioned above, the models were run on a local machine with a core i7 CPU; more computational resources could have provided more flexibility and more hyperparameter tuning options in acceptable times.

V. CONCLUSION

In this paper, we were able to test and evaluate the performance of multiple machine learning models. The results showed the SVM to have the best performance out of all the models we tested. We also provided our analysis on the reasons behind the performance of the models, in comparison to other models proposed by previous research.

Future work should focus on the mining of new datasets that reflect the current context. Also, more exploration with deep learning techniques should be evaluated. Additionally, we believe that more datasets in other languages could provide interesting insights and have the potential to enrich the field of automatic fake news detection.

REFERENCES

[1] "fake news." https://dictionary.cambridge.org/us/dictionary/english/fake-news (accessed May 30, 2021).

[2] "U.S. increased time spent on social due to coronavirus 2020," *Statista*. https://www.statista.com/statistics/1116148/more-time-spent-social-media-platforms-users-usa-coronavirus/ (accessed May 30, 2021).

[3] "Frequency of using social media for news in the U.S. 2020," *Statista*. https://www.statista.com/statistics/263498/use-of-social-media-for-news-consumption-among-hispanics-in-the-us/ (accessed May 30, 2021).

[4] "Infodemic: The Rise of Fake News During Covid-19," *Tech.co*, Oct. 27, 2020. https://tech.co/news/fake-news-covid-19 (accessed May 30, 2021).

[5] A. Murray, "Fake News in the Age of COVID-19," *Faculty of Business and Economics*, Jan. 14, 2021. https://fbe.unimelb.edu.au/newsroom/fake-news-in-the-age-of-covid-19 (accessed May 30, 2021).

[6] J. C. S. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, and E. Cambria, "Supervised Learning for Fake News Detection," *IEEE Intell. Syst.*, vol. 34, no. 2, pp. 76–81, Mar. 2019, doi: 10.1109/MIS.2019.2899143.

[7] Abdullah-All-Tanvir, E. M. Mahir, S. Akhter, and M. R. Huq, "Detecting Fake News using Machine Learning and Deep Learning Algorithms," in *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, Sarawak, Malaysia, Malaysia, Jun. 2019, pp. 1–5. doi: 10.1109/ICSCC.2019.8843612.

[8] W. Han and V. Mehta, "Fake News Detection in Social Networks Using Machine Learning and Deep Learning: Performance Evaluation," in *2019 IEEE International Conference on Industrial Internet (ICII)*, Orlando, FL, USA, Nov. 2019, pp. 375–380. doi: 10.1109/ICII.2019.00070.

[9] W. Antoun, F. Baly, R. Achour, A. Hussein, and H. Hajj, "State of the Art Models for Fake News Detection Tasks," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Doha, Qatar, Feb. 2020, pp. 519–524. doi: 10.1109/ICIoT48696.2020.9089487.

[10] P. Patwa *et al.*, "Fighting an Infodemic: COVID-19 Fake News Dataset," in *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, vol. 1402, T. Chakraborty, K. Shu, H. R. Bernard, H. Liu, and M. S. Akhtar, Eds. Cham: Springer International Publishing, 2021, pp. 21–29. doi: 10.1007/978-3-030-73696-5_3.

[11] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-7138-7.