COVID-19 Fake News Detector

Mohsen Bahremani, Daniel Berezovski, Rini Perencsik, Youjia Zhang
Wilfrid Laurier University

Abstract The increasing number of Novel Coronavirus (COVID-19) cases has given rise to a proliferation of misinformation related to COVID-19. This misinformation makes it difficult for individuals to find reliable news sources, resulting in protest against government measures to control the virus, social turmoil, and even death. To help alleviate the spread of misinformation, we built and evaluated a variety of machine learning models to help predict the reliability of COVID-19 related news. We combined data from two sources, which include news articles and website posts from official institutions. As our final method, we presented an ensemble of methods that achieves an Area Under Curve (AUC) of 0.97 and an F1-score of 0.92. Additionally, we created a website for readers to interact with the model at: `www.modellingcomp.com`.

1 Introduction

As the Novel Coronavirus (COVID-19) pandemic continues to infect thousands of individuals globally, misinformation related to the pandemic spreads in parallel. COVID-19 brought the world's first 'infodemic', short for information epidemic, according to the World Health Organization (WHO) [1].

The infodemic is making it difficult for individuals to find trustworthy information on the pandemic, therefore dangerously distorting their views and actions to the point where measures to control the pandemic are in jeopardy. Further, misinformation poses a threat to social cohesion as hate speech circulates through the media and heightens political polarization. Fake news is costing lives, and social media firms and government officials are failing to take effective actions on its elimination as approximately 90% of fake news still remains online [2]. As such, we created and evaluated several COVID-19 machine learning classifiers to predict the reliability of news articles and hopefully contribute to the combat against misinformation.

We used a variety of linear classifiers including Logistic Regression, Naïve Bayes, Passive Aggressive Classification, and Support Vector Machine. In addition to linear classifiers, we built a Convolutional Neural Network, a Transfer Learning Neural Network, and two Recurrent Neural Networks, one with Long Short-Term Memory and another with Gated Recurrent Unit. To capitalize on the unique advantages of each model, we created an ensemble of models as a more robust method.

The rest of the paper is in the format as follows. In Section 2, descriptions and visualizations of the data sets are given. Section 3 explains the preprocessing methods applied to the data sets. Section 4 provides a description of each method used. Section 5 discusses the results and a comparison of model performance. Section 6 contains a conclusion summarizing the report.

2 Data sets

2.1 Introduction to data sets

There are two COVID-19 data sets that were combined and studied in this project; the first data set (Data Set1) was collected by Susan Li [3], and the second data set (Data Set2) was from Limeng Cui [4].

The raw data from Data Set1 is shown in Figure 1. The news information is contained in the text column along with a corresponding label, 'Fake' or 'True' in the label column. Additionally,

	title	text	source	label
509	Herbs and Essential Oils to Fight Coronaviruse...	One of the biggest challenges designing a stra...	https://web.archive.org/	fake
342	Coronavirus conspiracy video spreads on Instag...	Instagram and Facebook have made a concentrate...	https://www.nbcnews.com/	TRUE
324	Mapping the Social Network of Coronavirus	To slow the virus, Alessandro Vespignani and o...	https://www.nytimes.com/	TRUE
640	Trump' s advisors are pushing coronavirus treat...	The true agenda of Dr. Anthony Fauci, the curr...	https://www.naturalnews.com/	Fake
555	What' s happening with a vaccine?	A vaccine for Covid-19 isn' t around the corner...	https://www.wired.co.uk/	TRUE

Figure 1: Raw data from data set1

Figure 2 includes the raw real and fake news data accordingly from Data Set2. Data comprising of real news were given the label 'True' and 'False' is given to the fake news. News information is stored in the 'title' and 'content' columns, and 'news titles' column is the same as 'title'. This data set is more diverse than Data Set1 as it contains website posts, news articles, and tweets. However, after analyzing the tweets, we noticed that many of them were not labeled correctly. For instance, all tweets comparing COVID-19 to the flu are labeled as fake news even if it is an accurate comparison, such as "COVID is NOT exactly like the flu", which in many cases were labeled as fake. Therefore, tweets data were not studied in this project. Both data sets were concatenated to create a larger data set to build the COVID-19 related fake news detector. After removing duplicates, the final data set has 3542 real news observations and 1411 fake news observations, indicating a slightly imbalanced data set with a 4:10 ratio. The 'title_text' column combined 'title' and 'text' columns in Data Set1, and combined 'title' and 'content' in Data Set2. The 'title_text' column is the unstructured input data and 'label' column is the binary output with 0 denoting fake news and 1 denoting real news.

2.2 Data Visualization

There are various visual analysis tools that allow us to better understand our text data. A popular visual is a word cloud which works in a straightforward way: the larger and bolder a word appears, the more frequently it is mentioned within the text data. In Figure 3, the 50 most frequent words in the fake and real news text are depicted for comparison. It can be easily seen that numerous potentially deceiving words are emboldened in fake news Figure 3a, such as vaccine, china, chinese, and bill gates, all of which the media commonly provides false information on. Interestingly, many fake news articles frequently use the words claim, government, research, and study to fortify their fake news and appear credible.

Another way to observe the data is to visualize which words and phrases are more characteristic of a single category–that is fake news and real news–than others. Characteristic metrics were calculated using the formula F-score = Harmonic Mean(P(term | category) , P(category |

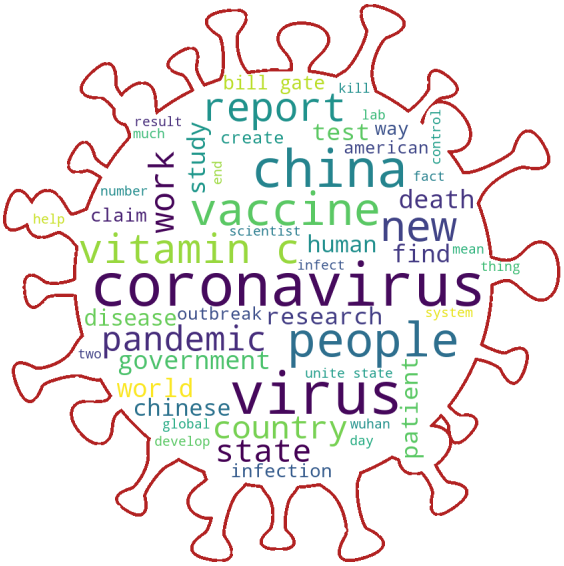
	type	title	newstitle	publish_date	content
0	article	Coronavirus Outbreak: Live Updates	"Coronavirus Outbreak: Daily Updates"	NaN	experts warn most states that are reopening st...
1	article	Screen Time Doesn’ t Hurt Kids’ Social Skills, ...	"Screen Time Doesn't Hurt Kids' Social Skills,...	NaN	a new study finds an increase in screen time d...
2	article	Here’ s Exactly Where We Are with Vaccines and ...	"Where We're at with Vaccines and Treatments f...	NaN	scientists around the world are working on a n...
3	article	Here’ s What Happens to the Body After Contract...	"Here's What Happens to the Body After Contrac...	NaN	coronaviruses cause respiratory illnesses so t...
4	article	Why You Should Be Skeptical of Early Results o...	"Why You Should Be Skeptical of Leaked Results...	NaN	medical experts are studying if remdesivir can...

(a) Real News Raw Data

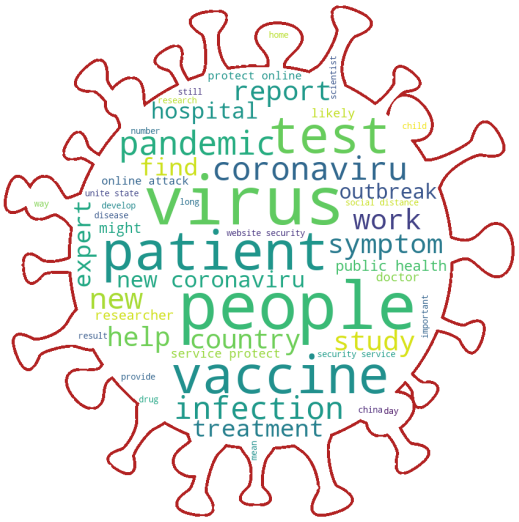
	type	title	newstitle	publish_date	content
0	post	Facebook posts shared in at least three countr...	NaN	NaN	NaN
1	article	Wisconsin is Òclearly seeing a decline in COVI...	"Wisconsin Legislature Takes Gov. Evers to Court"	NaN	speaker robin vos r rochester and senate major...
2	post	Facebook posts claim a child who is infected w...	NaN	NaN	NaN
3	post	IndiaÕs Ministry of Home Affairs banning citiz...	NaN	NaN	NaN
4	post	42 Democratic senators, plus two Independents,...	NaN	NaN	NaN

(b) Fake News Raw Data

Figure 2: Raw data from Data set2



(a) COVID-19 Fake news Word Cloud



(b) COVID-19 True news Word Cloud

Figure 3: Word Cloud of COVID-19 Fake and Real News

term)) and the plot was drawn with *Kessler's scatter plot* tool [5].

In Figure 4, each point corresponds to a word used in the news data. The closer a point is to the upper-left of the plot, the more characteristic it is to fake news data. On the contrary, the closer a point is to the bottom-right, the more characteristic it is of real news. Additionally, terms are colored by their association; those that are more associated with fake news are in blue, and those more closely associated with real news are in red. It appears that words which are characteristic of false news tend to be less related to COVID-19 as the real news. For example, many false news characteristic words tend to be centered more around politics and are unrelated to the virus, such as bill gates, military, weapons, china, economic, and chinese. However, many of the words that are characteristic of real news such as guidance, service, distancing, transmission and healthcare are more informative of COVID-19 and how to protect oneself from the virus.

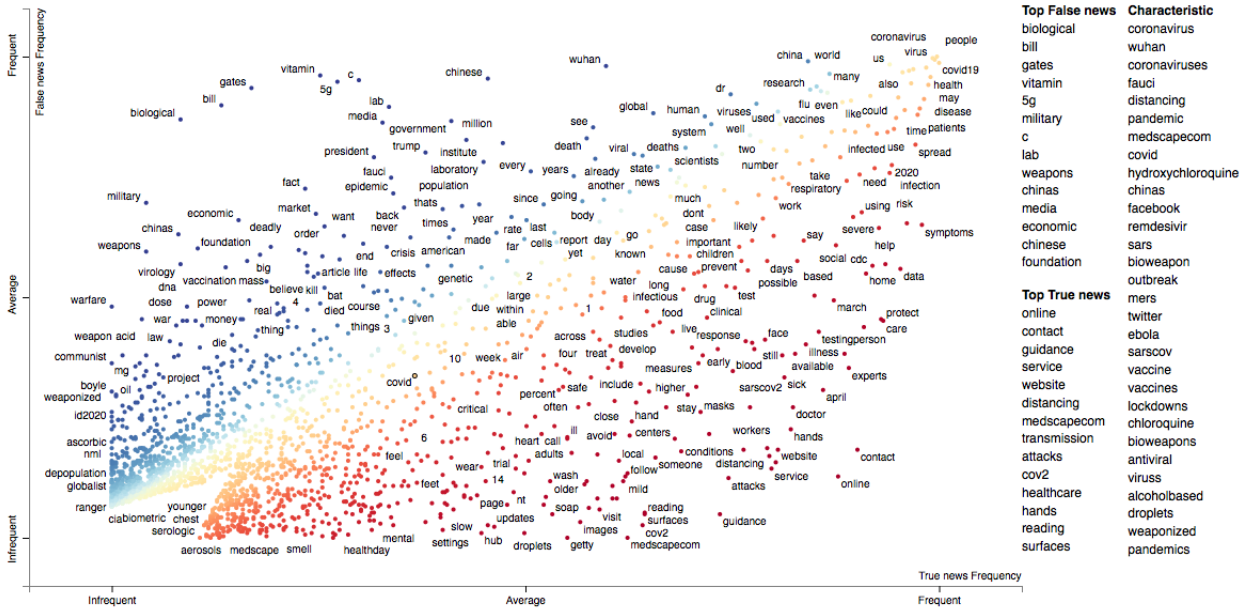


Figure 4: Fake Real News Scatter Plot

3 Data Preprocessing

The 'title_ text' column necessitates preprocessing, which simply means to process the given text into a form that is predictable and analyzable for the upcoming machine learning algorithms. To this end, data cleaning and normalization are put into practice as follows:

Data Cleaning Of the data collected, the project studies only the content and title of each news article. The following preprocessing techniques remove unwanted information.

1. Html is removed from articles using python *RegEx* [6] functions. The news data are collected by human beings with potential preferences from selected sources, hence the html should not be a feature to be considered.
2. Stop words such as *'a'*, *'the'*, *'and'* are removed using the python *nltk* [7] package as they contribute very little to the meaning of text.

3. Punctuation, non-English characters, numbers, and extra white spaces are removed as they are not relevant to the analysis.

After the data cleaning stage, the sentence, "The coronavirus, a man made virus, was created in China according to https://website.com/covid", for example, is transformed into "coronavirus man made virus created China according".

Normalization Normalization is the process of reducing text into a 'standard' form so that the input is consistent before applying machine learning algorithms. The normalization process contains the following steps.

1. **Tokenization** The tokenization step splits longer strings of text into smaller pieces such as sentences or words which are called tokens. In this project, each word in a news article is tokenized. First, each sentence is tokenized by splitting the sentence whenever there is a white space. Then, each word is stored sequentially in a list that now represents the original sentence. The clean sentence above would be ['coronavirus', 'man', 'made', 'virus', 'created', 'China', 'according'] after tokenization.
2. **Lower case** All text data are transformed to lower case to be a remedy for the sparsity issue, that is '*PANDEMIC*' is treated the same as '*pandemic*'. After this step, the tokenized sentence becomes ['coronavirus', 'man', 'made', 'virus', 'created', 'china', 'according'].
3. **Stemming** Stemming is the method of reducing inflection in words to their canonical form, which benefits from a crude heuristic process that removes the very ends of vocabularies. This is done using *PorterStemmer* [8] from the *NLTK* package. For instance, compute, computer, computed, and computing are replaced by "comput". After stemming, the final preprocessed sentence becomes ['coronaviru', 'man', 'made', 'viru', 'creat', 'china', 'accord'].

4 Methodologies

4.1 Training Test Split

To combat the class imbalance problem, each individual model performance was validated using the stratified 5-fold Cross Validation (CV) [9] method shown in Figure 5. This splitting method endeavors to guarantee that each fold represents the same proportion of the fake and real news classes (4:10) as the original data set for the test and training splits. Ron Kohavi [10] believes that stratified CV is generally a better model validation technique, both in terms of bias and variance, than regular cross-validation which can give too much weight to over-represented classes. Since there are 3542 real news observations and 1411 fake news observations, each test set consisted of 283 fake and 708 real news, and each training set included 1128 fake and 2834 real news, with a 70%/30% training-test split. Moreover, 30% of the data in every training set were split into a validation set to fine-tune hyperparameters using the stratified method.

4.2 Implemented Methods

Logistic Regression (LR) This is a linear classifier where the decision boundary is made based on a linear function of features. The logistic regression model was trained on the preprocessed data with term frequency-inverse document frequency (TF-IDF). TF-IDF was used to transform

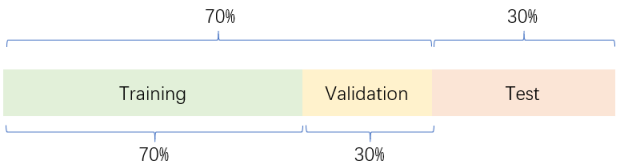


Figure 5: Training Test Split used the stratified method (For brevity, only 1 fold is shown).

words into a numerical representation by evaluating how relevant a word is to a data set or document in a collection of them. Hyperparameters such as regularization coefficient C and L1, L2 penalty term, were tuned by the *GridSearch* [11] method using the *sklearn* [12] package.

Naïve Bayes (NB) The classic statistical Naïve Bayes model is widely used in classification tasks and assumes the independence among features. Because of the independence assumption, Naïve Bayes performance was hindered in some cases [13]. The *GaussianNB* model from the *sklearn* package was applied.

Passive Aggressive Classification (PAC) Passive Aggressive algorithm proposed by Crammer et al. [14]. is categorized as an online learning algorithm. It is simple to implement as it follows a closed-form update rule. The core concept is that the classifier adjusts the weight vector for each misclassified training sample in order to improve its ability. That is, the algorithm is passive when there is a correct classification but aggressive in the event of a miscalculation. The *PassiveAggressiveClassifier* from the *scikit* package was trained with TF-IDF representation of words along with Grid Search.

Support Vector Machine (SVM) This model is a supervised machine learning algorithm used to classify binary and categorical response data. SVM models classify data by means of optimizing a hyperplane that separates the classes. This can also be considered as finding the hyperplane that maximizes the margin between the classes [15]. SVM models from the *sklearn* package with linear and polynomial kernels were studied in this project.

Gated Recurrent Unit (GRU) Recurrent Neural Network (RNN) are broadly used in Natural Language Processing (NLP) as they are powerful in sequence based predictions. The problem with the basic RNNs is that they have trouble remembering information for long periods of time as they are prone to the vanishing gradient problem. GRU, first introduced in 2014 by Kyunghyun Cho et al [16], is a type of RNN that helps combat this problem using internal mechanisms called gates that can remember more important information. A Word2Vec model was trained on the data set and created a Word2Vec embedding matrix used in the embedding layer of the GRU from the *keras* [17] package. Then, two GRU layers with 128 units and 64 units respectively were added, each with 20% drop out. The final dense layer used a sigmoid activation function as this is a binary classification. The parameters were tuned based on binary crossentropy loss function, adam optimizer, and accuracy metrics.

Long Short-Term Memory (LSTM) LSTM is another network that helps deal with the vanishing gradient problem. LSTM is similar to GRU but differs in that LSTM has three gates whereas GRU has two gates [18]. LSTM was built using the *keras* package. The first layer in the LSTM consisted of a Word2Vec embedding layer, followed by a dropout layer dropping 30% of the

units which has a regularizing effect, therefore reducing overfitting. Directly after this, there is an LSTM layer followed by another dropout layer, which allows the neural network to carry information across multiple time steps, alleviating the loss of information from earlier words.

Convolutional Neural Network (CNN) Although the CNN is well known for its applications in computer vision, it also has applications in NLP [19]. The first layer of the CNN is a word embedding layer, produced by Word2Vec. Since the data are sequential, the model works with one dimensional convolutions. As such, we included a 1D CNN directly after the embedding layer with a ReLu activation. A convolutional window size of 8 and 128 output filters produced the highest performance. Further, we added a global pooling layer that combines the vectors from the previous layer into a single layer by taking the maximum values in order to reduce the size of the input layer, speed up computation, and avoid overfitting. The CNN model was built using the *keras* package and the remaining hyperparameters including epochs and learning rates were determined using the *keras callbacks* function.

4.3 Transfer Learning

With only 4953 COVID-19 news observations, the small size of data set can limit model performances. As such, we explored transfer learning which can be useful for applications where there is a lack of a large data set. In transfer learning, an already existing data set similar to the data set of interest is leveraged, the useful information is extracted from auxiliary domains to help classify COVID-19 related fake news [20]. Figure 6 displays a sample of the raw general news data that were retrieved from a Kaggle competition. The data set has 20, 800 observations; 10, 387 are real news data and 10, 413 are fake news data. Both the 'title' and 'text' columns contain news information. Label 1 denotes fake news and label 0 denotes real news.

	id	title	author	text	label
0	0	House Dem Aide: We Didn' t Even See Comey' s Let...	Darrell Lucus	House Dem Aide: We Didn' t Even See Comey' s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

Figure 6: Raw data from the general news data set

A GRU model with two hidden layers was first trained using the general fake news data. The first layer is a Word2Vec embedding layer, followed by a GRU layer with 128 units and a drop out of 20%. The second hidden layer include 64 units with a drop out of 20%. The final dense layer uses sigmoid activation. After being tuned using binary crossentropy loss function and adam optimizer, the general fake news model achieved the results in Table 1. The model was then tested on the COVID-19 news data, which it had not yet seen, and had poor performance, as expected. As part of the transfer learning, the first two layers of the GRU were frozen and the output layer was removed. Therefore, one hidden layer with 65 trainable parameters was trained using the COVID-19 data. When tested using the COVID-19 data, the final transfer learning model performance was improved, illustrated in Table 1, although it did not perform as well as anticipated. The transfer learning model performed poorly and after observing a word cloud of each data set, we believe the low performance is due to the lack of similarity between the general

fake news data and the COVID-19 fake news. The general fake news data contain a large amount of political news items regarding Trump, Clinton, and the 2016 presidential election in general, which are unrelated to the COVID-19 virus. We suspect that transfer learning would work well with a fake news data set specifically on an arbitrary virus, rather than any fake news in general.

Table 1: GRU Transfer Learning Model Test Set Results

	Accuracy	Precision	Recall	F1-score
General fake news model	0.97	0.97	0.97	0.97
General fake news model tested on COVID-19 news	0.31	0.51	0.50	0.27
Final model tested on COVID-19 news	0.69	0.48	0.50	0.44

4.4 Ensemble Method

Ensemble methods have become popular as the solutions produced usually have a higher accuracy, AUC, and F1-score than a single model. The effectiveness of the average ensemble method has been demonstrated [21]. From the models discussed in the previous sections, the four best probabilistic models were selected according to their 5-fold cross validation results. These models are LR, GRU, CNN, and LSTM. Several combinations of these models were chosen to create an ensemble of models to test their impact on performance. These combinations were selected subjectively, and each model's prediction was weighted equally. Averaging out the probability outputs from all base models and classify news into real or fake, the optimal thresholds were determined by AUC. The results of the ensemble models are compared in Table 4 and discussed in Section 5.

4.5 Model Evaluation Criteria

Area Under Curve (AUC) AUC is one of the most important metrics to evaluate classification models' ability of distinguishing between classes.

F1-score Recall measures the percentage of positive classes that are correctly predicted. Precision measures how much of predicted positive classes are actually positive. When data sets have imbalanced classes, Recall and Precision can be misleading. Hence F1-score ($F1 = \frac{2*Precision*Recall}{Precision+Recall}$) is used to balance these two criteria.

5 Results

The optimal parameters and hyperparameters as well as individual model results are summarized in Table 2 and Table 3 respectively.

Table 2: Optimal Parameters and Hyperparameters Chosen

LR	inverse of regularization coefficient C = 10, L2 ridge penalization
PAC	maximum step size = 6, max iteration = 300, Class weight = chosen automatically
NB	variance smoothing = 1e-9
SVM	kernel: linear, Class weight = None
GRU	dropout = 0.2, fully connected layer activation function: sigmoid, loss function: binary crossentropy, optimizaer: adam
CNN	kernel size = 5, convolutional layer activation function: relu, optimiazer: adam fully connected layer activation function: sigmoid, loss function: binary crossentropy
LSTM	dropout = 0.3, fully connected layer activation function: sigmoid, loss function: binary crossentropy, optimizaer: adam

Table 3: Individual Models 5-Fold Stratified Cross Validation Test Results

	LR	PAC	NB	SVM	GRU	CNN	LSTM
CV AUC	0.91	0.90	0.81	0.79	0.82	0.87	0.87
CV F1-score	0.91	0.90	0.81	0.81	0.81	0.88	0.86

Among the linear classifiers, SVM and NB performed more poorly than anticipated. NB's assumption of independence among features might explain the low performance. PAC and LR reached much higher F1-scores than SVM. Since PAC is not a probabilistic classifier and only returns predicted classes, we removed it from the base models.

In general, the neural networks had lower performance than the linear classifiers due to our smaller data set as neural networks usually perform better on larger data sets. We believe a reason that the LSTM model outperforms the GRU model is that LSTM has three gates whereas GRU has two gates. Since the data set contains news articles which are lengthy, modeling long distance relations becomes important and the LSTM having one more gate better achieves this.

The CNN model has a fast training time relative to the RNNs as it is able to simultaneously process all the elements whereas the RNNs processes each word sequentially. Further, CNNs specialize in detecting key expressions and RNNs specialize in discovering key information in the sequential form of the data. We believe that the CNN outperformed the RNN because it is more important that a model notices key phrases. For example, a sentence in a fake news article might be, "The coronavirus is a man made virus that was created in a laboratory in China." In this example, the information needed to correctly make the prediction lies in the phrase "man made", rather than the nature of the sequence.

The results for the ensemble models are presented in Table 3. The highest values of the CV accuracy are highlighted.

Table 4: Ensemble Models

	LR, GRU, LSTM, CNN	GRU, LSTM, CNN	LR, CNN	LR, GRU, CNN	LR, GRU, LSTM	LR, CNN, LSTM
CV Accuracy	0.9170	0.9041	0.9204	**0.9245**	0.9164	**0.9267**
Standard Deviation	0.0103	**0.0046**	0.0142	0.0115	0.0098	**0.0079**
Optimal Threshold	0.6565	0.6366	0.5939	0.6304	0.5089	0.6448
AUC	0.9717	0.9608	0.9757	0.9713	0.9696	0.9744
F1-Score	0.90	0.89	0.91	0.91	0.90	0.91

The LR, CNN, LSTM combination had the highest CV accuracy with a small standard deviation. The 95% confidence interval of the CV accuracy was between 91.89% and 93.46% and reaches 0.9744 AUC score which was much higher than those of individual models. Hence, this ensemble model was chosen as our final model. With the optimal thresholds 0.6401, the final model had precision 0.91, recall 0.93, and F1-score 0.92. By applying the ensemble method, the final model leverages the unique advantages of LR, CNN, and LSTM.

6 Conclusion

This report presents the results of several algorithms on modeling COVID-19 fake or real news data. After cleaning and preprocessing the data, we trained linear classifier models such as LR, NB, PAC, and SVM. The highest performance of linear models was Logistic Regression with 0.91 for both AUC and F1-score. Then, we built neural networks including GRU, CNN, and LSTM. LSTM outperformed the other neural networks with an AUC and F1-score of 0.87 and 0.86 respectively. With a data set limited in size, the linear classifier models were more successful than neural networks. A transfer learning model was trained but performed poorly which we believe to be a result of the general news data set that is not similar enough to COVID-19 news data. Finally, to create a more robust model, we created several combinations of the existing models. The best performing ensemble model include LR, CNN, and LSTM, and therefore was chosen to be our final model as it achieved the best results with an AUC of 0.9744 and an F1-score of 0.92. The combination of the three models allows for a reliable model.

Further Potential Improvement Model performance was hinged as a result of data set size limitations. As more data regarding COVID-19 news become available and are labeled appropriately, the neural network based models are expected to be improved. The high feature dimensional space makes classification more challenging, therefore, feature selection methods such as information gain can be proposed. In addition, model hyperparameters can be further tuned. For instance, although we studied linear and polynomial kernels, we recently found that the SVM overall performance can be improved greatly by choosing random basis function (rbf). Furthermore, there are numerous state-of-the-art models that have not been studied yet, such as K-Nearest Neighbors (KNN), Bidirectional Encoder Representations from Transformers (BERT), and Attention models. Finally, developing a multi-language COVID-19 fake news detector can help debunk misinformation on a worldwide level.

Website for Knowledge Dissemination To disseminate our discoveries from this project faster, we created a website for readers to interact with the final chosen model as well as an additional feature of providing similar news articles. To use the website, one simply enters a COVID-19 related news article and clicks the submit button. The same preprocessing techniques mentioned in the report will be applied before the data are passed to our final model. Since the average character length the data set used to train our model is 950 characters, we believe that the model's accuracy will be higher with articles of similar length. Additionally, the most similar news articles in our data set to the one entered will appear upon submission along with the fake/real news classification result. The Cosine Similarity metric is applied to calculate the similarity between different news articles based on the orientation of two documents [22]. The website can be found at: `www.modellingcomp.com`.

A Appendix

Acronyms

AUC Area Under Curve. 1

BERT Bidirectional Encoder Representations from Transformers. 10

CNN Convolutional Neural Network. 7

COVID-19 Novel Coronavirus. 1

CV Cross Validation. 5

GRU Gated Recurrent Unit. 6

KNN K-Nearest Neighbors. 10

LR Logistic Regression. 5

LSTM Long Short-Term Memory. 6

NB Naïve Bayes. 6

NLP Natural Language Processing. 6

PAC Passive Aggressive Classification. 6

rbf random basis function. 10

RNN Recurrent Neural Network. 6

SVM Support Vector Machine. 6

TF-IDF term frequency-inverse document frequency. 5

B Code

All codes and data sets are available upon requests.

References

[1] Novel coronavirus (2019-ncov) situation report – 13. *WHO*, April 18, 2020. 1

[2] Social media firms fail to act on covid-19 fake news. *BBC News*, June 3, 2020. 1

[3] Susan Li. Nlp-with-python. Github. https://github.com/susanli2016/NLP-with-Python. July, 2020. 2

[4] Limeng Cui. Coaid. Github. https://github.com/cuilimeng/CoAID. July, 2020. 2

[5] Jason S. Kessler. Scattertext: a browser-based tool for visualizing how corpora differ. 2017. 4

[6] Guido Van Rossum. *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020. 4

[7] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. 4

[8] Peter Willett. The porter stemming algorithm: Then and now. *Program electronic library and information systems*, 40, 07 2006. 5

[9] N.A. Diamantidis, D. Karlis, and E.A. Giakoumakis. Unsupervised stratification of cross-validation for accuracy estimation. *Artificial Intelligence*, 116, 10 1997. 5

[10] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, page 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. 5

[11] Steven M LaValle, Michael S Branicky, and Stephen R Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):673–692, 2004. 6

[12] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011. 6

[13] Irina Rish. An empirical study of the naïve bayes classifier. *IJCAI 2001 Work Empir Methods Artif Intell*, 3, 01 2001. 6

[14] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms, 2006. 6

[15] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes - which naive bayes? 01 2006. 6

[16] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. 6

[17] Francois Chollet et al. Keras, 2015. 6

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. 6

[19] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. Understanding convolutional neural networks for text classification. *CoRR*, abs/1809.08037, 2018. 7

[20] Charu C. Aggarwal and Chengxiang Zhai. *Mining Text Data*. Springer, 2012. 7

[21] E. Alpaydin. Multiple networks for function learning. In *IEEE International Conference on Neural Networks*, pages 9–14 vol.1, 1993. 8

[22] Baoli Li and Liping Han. Distance weighted cosine similarity measure for text classification. In Hujun Yin, Ke Tang, Yang Gao, Frank Klawonn, Minho Lee, Thomas Weise, Bin Li, and Xin Yao, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, pages 611–618, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 10