Learning To Choose A Classifier For Fake News Detection

Aditya Thagarthi Arun
Computer Science
Texas A&M University
College Station, TX, USA
adityata@tamu.edu

Vishruth Raghuraj Gollahalli
Computer Science
Texas A&M University
College Station, TX, USA
vishruth@tamu.edu

***Abstract*—Support Vector Classifiers[1] and Random Forest Classifiers[2] make good candidates to classify news as being real or fake. But each of them has its shortcomings when it comes to certain data points in which case one model can do better than the other at correct classification. We propose a novel model to teach a neural network to learn which classifier would perform better at classifying a news article as being real or fake for a given article. This is a fine-grain approach to classifying with multiple candidate classifiers.**

Neural networks[3] provide an excellent opportunity to classify an underlying classifier for a data point. One of the main advantages of this model is that it offers flexibility regarding which and how many classifiers it can be trained on. In this manner we combine the prowess and characteristics of multiple classifiers.

Index words-*machine learning, neural network, random forests, support vector classifiers, fake news detection*

I. Introduction

Fake news refers to false or misleading information that may or may not be intentional. A fake news article can be a piece of disinformation that intends to manipulate its readers and may be harmful and cause disorder by wrongfully changing the perception and attitude of the people reading it. With the volume of information in today's digital age, it has become very easy to manipulate a vast majority of the population with fake news and propaganda. The faster spread[4] of false news as compared to real news makes the challenge of distinguishing false news from all the more difficult and simultaneously, vastly more necessary.

Manually fact checking every article that is published is virtually impossible. Automating this fact checking will be incredibly helpful in mitigating the harmful effects of fake news. Machine learning techniques are one of the best methods to implement a model for automating news detection. A machine learning model is able to consume vast amounts of information/news articles and learn from it. These models can learn patterns in fake news articles and can help us differentiate between real and fake news articles there by eliminating the need for manual intervention.

Models such as random forests, probabilistic models, artificial neural networks and others are quite common when classification problems are being addressed. Given the vast majority of such models available to us, there arises a need to identify what models would be the best suited for a given classification problem. This can be addressed by testing these models and picking the one that does performs the best (measured on heuristics such as accuracy, precision, recall, and ROC curves) on a given set of data. This is a rather course-grained solution. A much more fine-grained solution would be to pick a model that would be best suited for each individual data point. We hypothesise that this fine-grained approach will yield better results in classifying data rather than the course-grained approach. This necessitates a higher-level machine learning model, so to say, that "learns" which classification model would be the best suited for a data point. This is the motivation of our project.

II. Related Work

In this paper, we have taken inspiration from Zhang's paper on FAKEDETECTOR[5] which uses a deep diffusive neural network to learn representation of news articles. While they propose a new neural network model, we have taken a detour in trying to apply a neural network to learn from performances of underlying classification models.

Horne and Adah[6] identified characteristics of fake news in detail. The analysis of systemic stylistic differences between fake and real news by them sheds a lot of light on the features that can be extracted to identify the integrity of news articles. Although we do not actively extract these features, they certainly give insight as to how these features may aid in choosing the right underlying classifiers.

III. Methodology

A. Dataset collection

We merged two Kaggle datasets in this project, totalling 51233 news articles. We extracted the articles' title, text, and the corresponding labels. The datasets we used are as follows:

- real_or_fake[7]: 6335 data points(3164 fake and 3171 real)
- Fake and real news dataset[8]: 44898 data points(23481 fake and 21417 real)

B. Dataset Vectorization

We used the statistic from the term frequency-inverse document frequency technique to weight the words in our corpus of news articles. The statistic helps gauge how important a given word is to a document in a collection/corpus(taken from Wikipedia). We used the TfidfVectorizer[7] function in the Python sklearn library to vectorize our news text. The resulting vectorized text was used for training and testing the support vector classifier, random forest classifier and the multilayer perceptron classifier.

C. Dataset Preparation

After obtaining the Tfidf vectorization of our comprehensive dataset, we split the data into subsets.

1) Training for support vector and random forest classifiers (*40% of the total vectorized data*)
2) Testing support vector and random forest classifier to record accuracy, precision and recall scores (*60% of the total vectorized data*)
3) Training the multilayer perceptron to learn which classifier to choose (*25000 data points out of the 60% split for testing random forest and support vector classifiers*)
4) Testing the multilayer perceptron (*5740 data points*)

The data was split in such a way that testing data for any of the models were not used from the same pool of data that were trained on any model to avoid data leakage and loss of integrity of any classification model.

D. Hyper Parameter Tuning

To identify the best hyperparameters, we wrote scripts to iterate over a range of possible hyperparameters (max depth and number of iterations for random forest; kernel type for support vector classifier). We identified the most optimal hyperparameters for each of the models.

- Support Vector Classifier (SVC): *linear kernel*
- Random Forest Classifier (RFC): *max depth = 500, number of estimators = 200*

E. Random Forest and Support Vector Classifiers

With the tuned hyperparameters, we continue with training the random forest and support vector classifiers on vectorized news data points. These trained models will be used two times. The first time will be to test the model's base performance by using a test set and the second will be when it will be used to predict labels during multilayer perceptron training. To gauge the performance of the neural network it is important to first evaluate the performance of these underlying classifiers.

F. Architecture

Figure 1 shows the architecture of our model. We first pass the vectorized data to the SVC and the RFC classifiers to train them on real and fake labels. Another set of vectorized labels (see point 3 under *Dataset Preparation*) were then used to train the neural network. The parameters for the MLP neural network are as follows:

- 3 hidden layers of size 100, 25 and 25 neurons
- 1000 iterations
- L-2 regularization parameter (α) $10{-}5$
- Optimizer: Limited memory Broyden-Fletcher-Goldfarb-Shanno (lbfgs)

The purpose of the neural network is to learn which underlying classification model is better to classify a given news article. On input a vectorized news article, the neural network outputs the label of the classifier that is better suited. To achieve this, we do the following:

1) Pass vectorized news article as input to the neural network
2) Predict label of the news article using both the underlying classifiers (SVC and RFC).
3) Compare the underlying classifiers' predicted label to the true label of the data point
4) If:
   - SVC predicts correctly and RFC does not, train neural network to pick SVC (assigned label 1)
   - RFC predicts correctly and SVC does not, train neural network to pick RFC (assigned label 0)
   - If both classifiers predict wrong label, train neural network by picking a random classifier
   - If both classifiers predict true label, train neural network by picking a random classifier

At the end of training, the neural network will be able to output 0 or 1 corresponding to RFC or SVC as the most suitable classifier to classify the given news article.

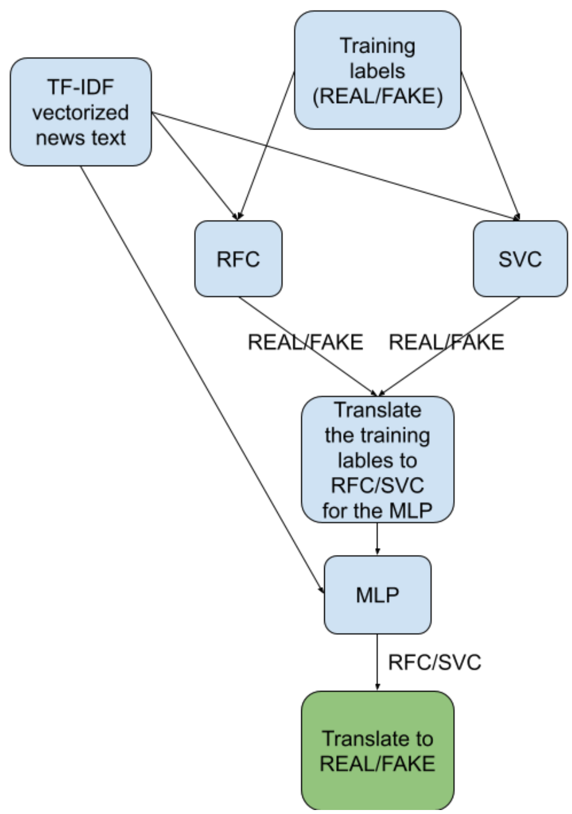

Fig. 1. Architecture flowchart.

IV. Experiment

Once the labels were ready for the neural network, we trained it on 25000 data points. Owing to the high accuracy of the underlying classifiers, the neural network also achieved high accuracy. The error was measured as the distance between the predicted label (0 or 1) and the label of the classifier that predicted the correct outcome of the data point. If both underlying classifiers predicted correctly (which was the most often case considering the high accuracy rates), the neural network performance would be decided based on the label that was randomly chosen.

The network was trained over 1000 iterations so that it would be able to converge. The test set for the neural network consisted of 5740 vectorized data points that were seen by none of the models earlier. The test labels were correspondingly selected from the complete set of generated labels. Note that the neural network predicted only the best underlying model to be used and not the "real"/"fake" label of the data point.

To concretely test our model, we first retrieve the 0 and 1 predictions from the neural network. For the corresponding data point, we predict the "real"/"fake" label using the classifier that was predicted by the neural network (i.e, RFC for 0 and SVC for 1).

V. Results

Figures 2 and 3 show the confusion matrices of the support vector classifier and the random forest classifier respectively. It is clear from observing the two figures that both models have high accuracy and recall with SVC doing better than the RFC on the given dataset.

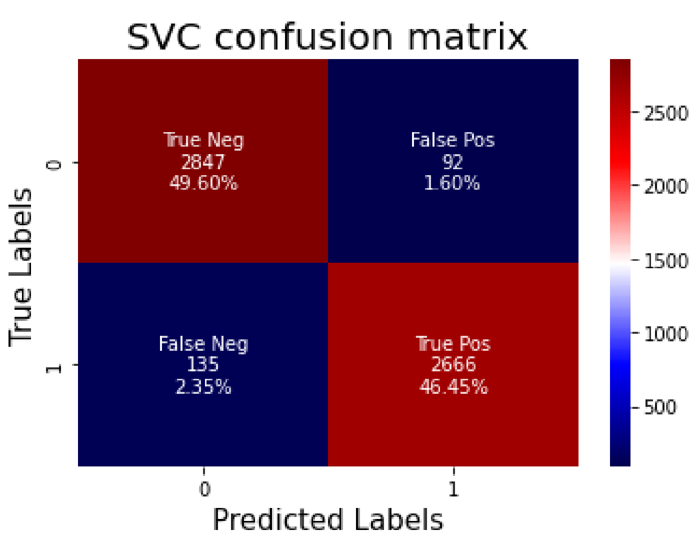

Fig. 2. SVC confusion matrix.

The classification with the MLP leads to a lower number of false positives but a slightly higher number of false negatives. See Figure 4 for this result.

The interesting find here is that when faced with the decision to pick a classifier to do a classification, the MLP picked the classifier that predicts the correct label nearly 69% of the time (Figure 5). This means that, in the situation where two

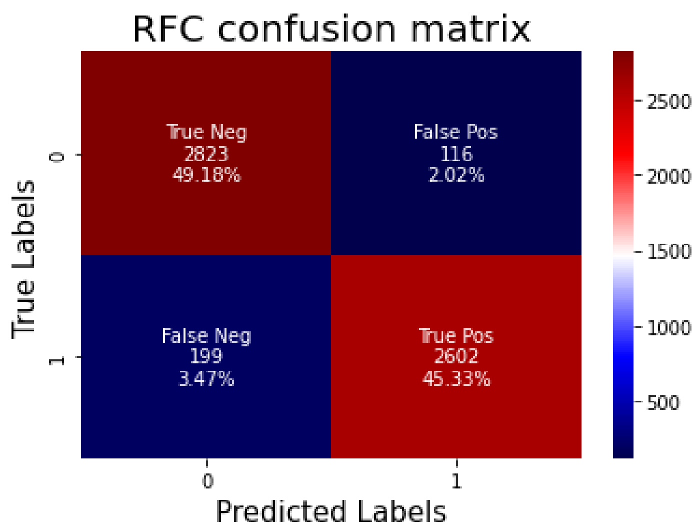

Fig. 3. RFC confusion matrix.

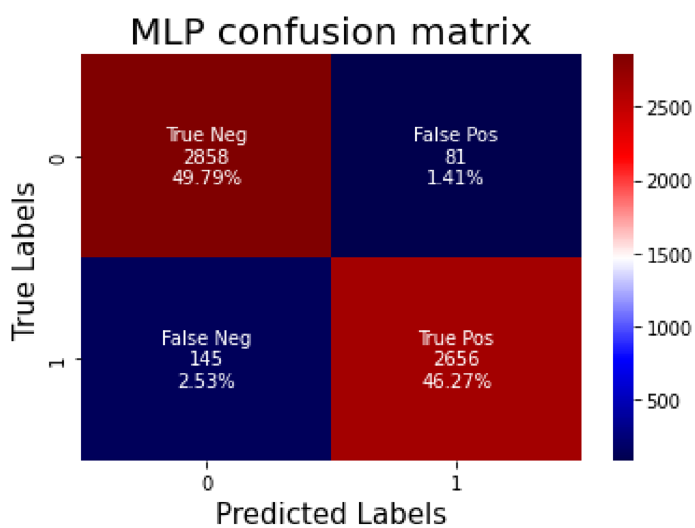

Fig. 4. MLP confusion matrix.

classifiers differ on the predictions for a data point, the neural network is able to pick the classifier that predicted correctly 69% of the time. Although this is not an overwhelming proportion, it is significant.

As an aside, we would like to point out that, during the base training of SVC and RFC, it was observed that SVC performed better in general as compared to RFC in classifying news data. The results of MLP can be used to infer similar

Models	**Real**		
	precision	**recall**	**f1-score**
RFC	0.96	0.93	0.94
SVC	0.97	0.95	0.96
MLP(ours)	0.97	0.95	0.96

TABLE I
Classification report for real news

Models	Fake		
	precision	recall	f1-score
RFC	0.93	0.96	0.95
SVC	0.95	0.97	0.96
MLP(ours)	0.95	0.97	0.96

TABLE II
CLASSIFICATION REPORT FOR FAKE NEWS

Model	Accuracy
RFC	0.95
SVC	0.96
MLP(ours)	0.96

TABLE III
ACCURACY REPORT OF THE TWO BASELINE CLASSIFIERS AND OUR ARCHITECTURE

information. The MLP has picked SVC more often than RFC while choosing a classifier as can be seen from Figure 6 and Figure 7. SVC was picked 2949 times and RFC was picked 2791 times.

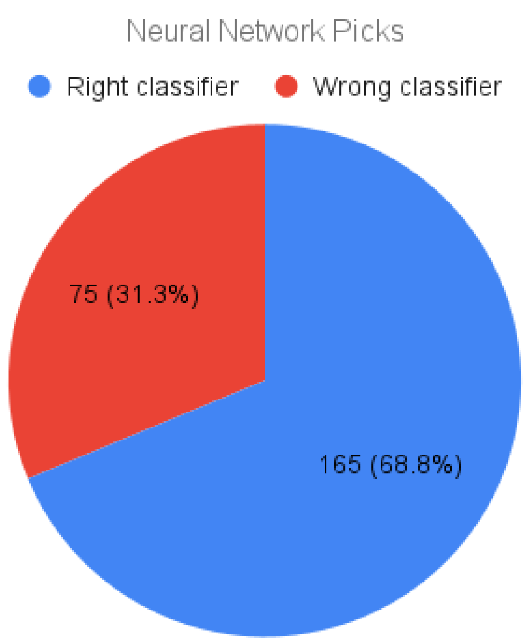

Fig. 5. What the neural network picked where there was a disagreement between the SVC and the RFC.

VI. CONCLUSION

Our hypothesis that fine-grained selection of classifiers rather than course-grained selection is validated to a certain extent using the proposed model. We would want to investigate this further with multiple classifiers rather than just two. We would like to include Naive-Bayes and K-nearest neighbor classifiers along with the existing classifier models to observe how the performance of the model improves and in what way it would impact the classification of real and fake news.

The second point we would like to concentrate on is identifying better models of neural networks that would be better suited to learn classifiers. Along with varying the models and architectures, we are also interested to tweak the optimizers, hidden layer numbers and adding various activation functions and compare what factors of the model architecture would lead to the best classification performance.

Finally, there is scope to improve the size of the dataset. A larger and a more diverse corpus of news data would aid training models better. A larger test set also would help assess performances of varied underlying classifiers better.

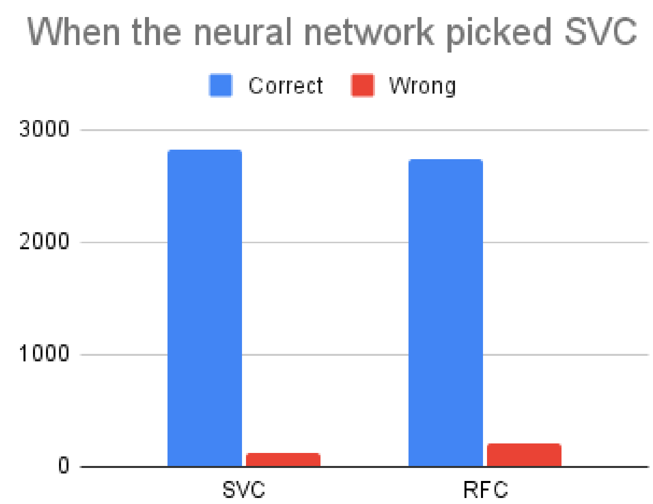

Fig. 6. How the two classifiers performed when SVC was picked by the MLP (2949 times)

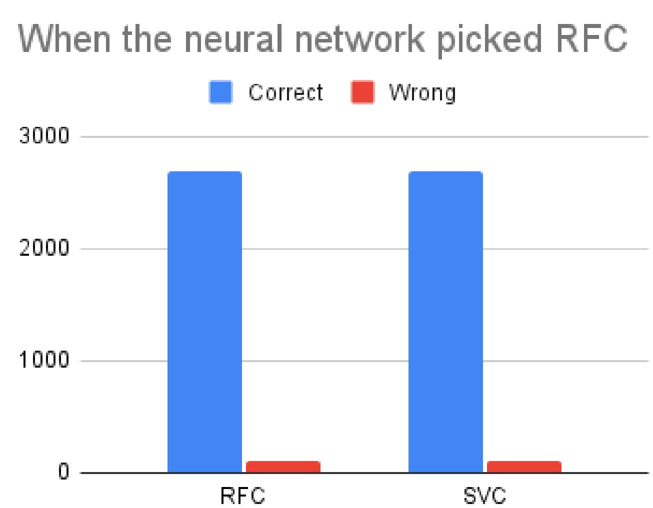

Fig. 7. How the two classifiers performed when RFC was picked by the MLP (2791 times)

REFERENCES

[1] Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297. CiteSeerX 10.1.1.15.9362. doi:10.1007/BF00994018. S2CID 206787478

[2] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995
[3] Hastie, Trevor. Tibshirani, Robert. Friedman, Jerome. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY, 2009.
[4] Zhou and Zafarani. ”A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities”. Syracuse University, 2020.
[5] Zhang, Dong, and Yu. ”FakeDetector: Effective Fake News Detection with Deep Diffusive Neural Network”. 2019.
[6] Horne, Benjamin; Adah, Sibel (2017). ”This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News”. Eleventh International AAAI Conference on Web and Social Media
[7] https://www.kaggle.com/rchitic17/real-or-fake
[8] https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset