Truth or Lie: Pre-emptive Detection of Fake News in Different Languages Through Entropy-based Active Learning and Multi-model Neural Ensemble

Md. Saqib Hasan, Rukshar Alam, and Muhammad Abdullah Adnan
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
{msaquibhasan,ruksharalam7,abdullah.adnan}@gmail.com

***Abstract*—In recent times, the circulation of fake news on social networks has increased exponentially with spikes in propagation seen during and after the 2016 US elections. Hence, there has been a surge in research into automated fake news detection. However, most research tends towards supervised learning which requires a significant amount of labeled data which is difficult to obtain. Thus, in this paper, we develop a semi-supervised learning method for fake news detection incorporating active learning based on entropy as a query strategy to train a multi-model neural ensemble architecture. The goal of the research is to achieve high accuracy on fake news detection while using lower amounts of data. Our experiments against other standards indicate promising results, with our model achieving high accuracy with 4% to 28% of the dataset.**

***Index Terms*—Fake news detection, Active learning, Deep Learning, Ensemble Methods**

I. INTRODUCTION

Nowadays online platforms have become leading sources of news consumption due to being cheap and easily accessible. More and more people spend a considerable amount of time using social media or online news platforms [1]. However, due to the absence of regulatory authority on such platforms, the viewers are presented with lower quality news pieces compared with traditional sources of news [2]. As such, online news media enables widespread dissemination of fake news. Fake news [2] means false information publicized with the intent to mislead people. Fake news negatively influences both the individuals and the society as a whole [3] by polluting the news ecosystem with false and harmful information. Fake news also cleverly tricks readers to consider biased information as authentic for financial and political gains [2]. Thus, fake news detection on online platforms has become a prime problem that is being addressed by researchers nowadays.

Detecting fake news has been a considerable challenge due to two main reasons: difficulty in distinguishing fake from real due to the creator's agenda to misinform; lack of comprehensive datasets with enough metadata such as user reaction. Still, researchers are proposing ample amounts of automated fake news solutions, particularly based on supervised classification using extra metadata such as user connectivity, propagation of news, etc. However, these solutions present new problems on their own. Firstly, these solutions do not pre-emptively identify and flag fake news as they require extra information for detection which can only be obtained after the news has spread to a certain extent. Secondly, it is a cumbersome task to collect a large news dataset and then label them [4] for supervised classification.

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

Addressing both these issues, we propose a pool-based active learning method utilizing an entropy-based query strategy to train a multi-model neural ensemble network. Using active learning, our algorithm selects the best data points for training at each iteration, thereby drastically reducing the amount of annotated data required. The proposed classifier is a deep learning ensemble of multi-layer, convolutional, and recurrent neural networks which are all connected by a stacking layer. The neural ensemble is designed to combine the best standard models, thereby giving better performance and saving time required for grid-searching.

Our proposed method is examined on three real-world English-based fake news datasets (George McIntire, Liar, and Twitter Dataverse) and a Bangla fake news corpus. Moreover, our proposed algorithm is evaluated against several other standards (SVM, Naive Bayes, Gradient Boosting, CNN, LSTM, and MLP) and state-of-the-art fake news methods (dEFEND [5] and Neural User Response Generator [6]) to compare performance. Results show that our model performs similar to or better than the other standards during supervised learning and our active learning method can achieve high accuracy on dataset fractions as low as 4%.

If the major contributions of this paper are summarized, it is as follows:

- To the best of our knowledge, we propose the first active learning-based method that utilizes the power of entropy sampling to select the best data points for the classification of fake news.
- The proposed method significantly reduces the time and energy required to provide large amounts of labeled data to models for training.
- Experimentation shows that our method can achieve state-of-the-art accuracy on real-world datasets of different languages compared to other benchmark models with only a fraction of the data.

The paper is divided into many sections. In section II, we describe our proposed method in detail. We discuss the experimental setup, datasets used, how data is prepared, and the models and metrics used for comparison in our experiments in Section III. In Section IV, we show the results of our simulation with Section V concluding the paper and discussing future prospects of the work.

II. Methodology

In this section, we discuss the method that is used to solve the problem at hand. We formalize the problem and discuss some concepts pertaining to our problem in Section II-A. In Section II-B, we describe our method and model architecture in detail.

A. Background

Our task is, given a dataset of statements x_i and the label y_i, where $1 \leq i \leq n$, to develop an active learning-based method for the optimal classification of whether the statement x_i is true or false. The veracity of statement x_i is determined by the label y_i. y_i can only have binary values $[0, 1]$ where 0 constitutes as real and 1 constitutes as fake. For datasets with a range of fuzzy values from $[0, 1..., n]$ where 0 denotes true with increasing values denoting a gradual decrease of the truthfulness of the statement, we convert the labels into a binary format.

Active learning is a semi-supervised learning algorithm where the model interactively queries the user to obtain the desired outputs at new data points [7]. It is a very useful technique for small datasets or where the annotation is difficult. The most popular active learning method is **pool-based sampling** where the learner is exposed to a large pool of unlabelled data and, at each iteration, has to select a set of the most informative data points by utilizing a **query strategy** to learn further and then send it to the user/machine for labeling. There are a number of query functions but **entropy sampling** is the most popular. We will be using pool-based entropy sampling, which will be further elaborated in Section II-B.

Since active learning is a meta-algorithm functioning on top of a supervised learning algorithm, a supervised model is required. Extensive research has shown deep neural networks (DNN) to be the state-of-the-art supervised model for such natural language tasks like ours. However, selecting the perfect DNN architecture is a trial and error process and can be taxing. To tackle this, researchers have come up with various **ensemble** methods. Popular methods of ensembling neural networks include **averaging** and **stacking**. As such, we developed a multimodel ensemble neural network, inspired by [8], where multiple models are connected by a stacking layer. Details regarding our novel classifier are further given in Section II-B.

B. Proposed Algorithm

Let us discuss our proposed algorithm. We are using the **pool-based sampling** method of active learning and our query strategy will be **entropy sampling**, where new data points are selected based on the entropy of the classifier output for that data point, as mentioned in Section II-A. Entropy, as per information theory, is defined as a metric to measure how much information exists in a random variable using its probability distribution. The amount of information conveyed by each class in the classifier output, and thereby that data point, can thus be measured using entropy. The formula for entropy is:

$$S = \frac{-\sum_{i=1}^{n} P_i \log_2 P_i}{\log_2 n}$$

where P_i is the probability that the data point belongs to the class i and n is the total number of classes. As we are dealing with a binary situation of fake or real, so $n = 2$ and the entropy formula reduces to:

$$S = -x \log_2 x - (1 - x) \log_2 (1 - x)$$

where x is the probability of “fake” class.

In Section II-A, we also discussed ensembles and how multi-model neural networks help us overcome the problem of grid searching through the best model for a particular dataset. Inspired by Random Multimodel Deep Learner in [8], which is currently regarded as an efficient solution to the problem of text classification, we have designed a neural network model to tackle this issue. A brief visual of the network is shown in Figure 1.

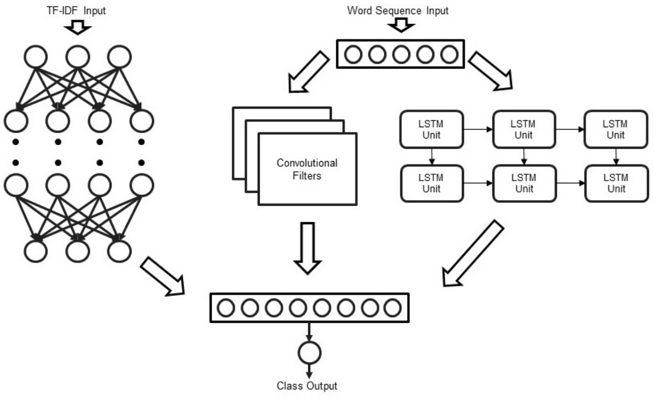

Fig. 1: Our Proposed Multi-model Ensemble Neural Network

Our multimodel ensemble neural network, as shown in Figure 1 consists of 3 different types of neural networks connected by a concatenation layer, which acts as the stacking layer, followed by the output perceptron. The 3 architectures in question are multi-layer perceptron (MLP), one-dimensional convolutional neural network (CNN), and a recurrent neural network (RNN) of Long-Short Term Memory (LSTM) units. These three architectures are considered the base standard DNNs for the task of text classification. MLP has long been the standard for DNN based solution to natural language tasks before the arrival of sequential models like RNNs. RNNs have since become the de-facto model for natural language problems due to their ability to capture temporal relations in sequence data. Further development, in the form of LSTMs, came to improve upon vanilla RNNs ability to recall important context from the past when calculating the current output. However, CNNs, in particular 1D-CNN, have shown great promise in recent years in the field of natural language processing although they have been developed extensively for computer vision. This is because 1D-CNN consists of sets of 1D filters, of variable size, which can be trained to detect certain word n-grams and then use them in cohesion to classify the text. Our model incorporates these individual DNNs into a single model to harness the ability and power of each architecture.

The input to the MLP is the **Term Frequency-Inverse Document Frequency** vector of the data while the input to the CNN and RNN comes from a single trainable word embedding layer. Details about the algorithms for calculating these input vectors is given in Section III-A. These three networks are eventually connected in the end by a dense layer of perceptrons, called the concatenation layer. This layer can be generalized as the stacking layer that connects all the three models into an ensemble. This concatenation layer is connected to a single output perceptron. Only one perceptron is needed to provide output since our classification task is a binary classification problem. The output from this perceptron is used to calculate the binary cross-entropy loss:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i log(y_i) + (1-y_i) log(1-y_i)$$

where y_i is the output of the perceptron after application of the activation function, which is sigmoid in this case, for the i^{th} input and N is the total size of the dataset. This loss is then used to calculate the gradient updates for each of the layers in the three networks and the concatenation layer and then the weights and bias are updated using **backpropagation** algorithm.

We have described our multimodel ensemble network and our active learning system. Now we move on to describe our proposed algorithm as a whole, as seen in Algorithm 1. We start by initializing our multi-model neural network θ_m. First, N samples are picked from our unlabelled dataset D using random sampling. These N samples are then labeled by the user/machine manually to form the training dataset L, while the rest of the data points in D collectively form the query dataset Q. We set a target accuracy A till which we want to train our model using active learning or till we have exhausted all our hardware resources or dataset points. In each iteration of the loop, the classifier θ_m is trained on L and then used to predict the class values p from Q. These class probability values are then used to calculate the entropy of each of the data points in Q using the equation described at the start of this section. A hyperparameter k determines the number of new data points that will be selected from Q which correspond to the highest k entropy values in e. This new set of data points, l, is then labeled by the user/machine manually and added to the training set L. The loop is then continued till our desired accuracy is reached or the budget is exhausted. Once out of the active learning loop, our trained model $\theta_m^{'}$ is returned.

III. Experimentation and Setup

In this section, we discuss the various benchmark datasets that our algorithm has been evaluated upon, other models that have been used for comparison, and the metrics that are calculated and presented for comparison. (code here [1])

Algorithm 1 Proposed Algorithm

Input: N: initial sample size, D: unlabelled dataset, A: target accuracy, θ_m: untrained classifier, k:number of entropy sample
Output: $\theta_m^{'}$: trained classifier

```
Select $N$ data points from $D$ by random sampling.
Using $N$ data points, divide $D$ into training dataset $L$ and query dataset $Q$
loop: Repeat if $\theta_m.accuracy \leq A$ or budget exhausted
$\theta_m.train(L)$
$p = \theta_m.predict(Q)$
$e = ENTROPY(p)$
Select $k$ data points in $Q$ corresponding with the $k$ maximum values in $e$, forming $l$
$L = L \cup l$
goto loop
$\theta_m^{'} = \theta_m$
return $\theta_m^{'}$
```

A. Dataset and Pre-processing

In order to test our proposed method, we trained our model and algorithm on five datasets, each divided into training and test set in the **80:20** ratio. To simulate the scenario of unlabelled datasets, the label of each row is provided only if it is part of the initial sample or when included in the update during an iteration of the active learning algorithm.

The first dataset was the George McIntire[2] dataset, consisting of 3171 real and 3164 fake news collected during the 2016 election cycle. The second was the Liar dataset from [3], restricting the class categories by keeping the top 3 categories as true and the bottom 2 as false and dropping the rest. The third dataset is from [9] consisting of 0.95m rumor and 1.1m non-rumor tweets from which we sampled 50k each. The final dataset is a Bangla fake news dataset from [10].

The datasets were prepared through a series of lowercase conversion, punctuation, number, hashtag, and URL removal, stop words elimination, and lemmatization. After preprocessing, the textual data is vectorized using TF-IDF and word embedding. Term frequency-inverse document frequency (TF-IDF) is a vectorization method that converts a sentence into a vector of unigram, bigrams, etc. where the weight of each term is proportional to frequency in the sentence and inversely proportional to the frequency in the dataset. Word embedding models convert sentences into a vector of integers, each corresponding to a word in the dataset. For our task, we utilized **trainable model embeddings** instead of pre-trained ones as they accommodate better to the nature of the task.

[1] https://github.com/DataAnalyticsLab/ActiveLearningFakeNews

[2] https://opendatascience.com/how-to-build-a-fake-news-classification-model/

B. Models and Metrics for Comparision

Our proposed method is also evaluated against a number of other standard baseline supervised models for fake news detection: naive bayes [11], support vector machines (SVM), gradient boosting, convolutional neural network (CNN) [4], long-short term memory network (LSTM) [12],dEFEND [5] and neural user response generator (NURG) [6]. Our model is compared in two forms: one based on our method and another using supervised learning. A number of metrics are calculated for comparison: accuracy and F1-score for comparing the different models; the fraction of dataset used against the accuracy of the model to measure efficacy of the active learning method. To compare the entropy query strategy, we also run our algorithm using the least confidence, another query strategy. To analyze the performance of our ensemble model, we perform an analysis of the active learning algorithm on two other baselines along with our ensemble model in Section IV-C. The results using these metrics are presented in Section IV.

IV. Results

The results are presented in three folds in this section. In Section IV-A, we present the comparative results between our model and the other models that we trained for comparison on benchmark datasets. In Section IV-B, we observe the ability of our semi-supervised model in gaining accuracy with different amounts of data present with different query strategies. In Section IV-C, we observe the ability of our ensemble model against state-of-the-art models when subjected to the same active learning algorithm.

A. Comparision of the classifiers

The results, in terms of test-set accuracy and F1 score, of training our model along with the comparison models on benchmark datasets, as discussed in Section III, are shown in Tables I and II. Results from our model are done in two folds: first, we train a supervised version of our model on the whole dataset; second, we train using our active learning method to a predefined target accuracy and measure the fraction of dataset used along with the other metrics. From the tables, we observe that our ensemble architecture performs close to or better than all the other models during supervised training. Looking at the results from the active learning portion, we see that our method can reach within 2-3% of the supervised model accuracy with only about 4% to 28% of the dataset. This proves the efficacy of our algorithm in achieving high accuracy while significantly reducing the burden of having a large amount of annotated data provided during training.

B. Performance of Active Learning Model

In this section, we observe the performance of our active learning algorithm with the ensemble model on the benchmark datasets as the amount of annotated data provided increases through Figure 2. To observe how entropy query strategy fares, we compared it to another popular query strategy, known as the least confidence, in the graphs. The curves represent the accuracy against the fraction of the whole dataset for training while the straight line, in red, represents the accuracy of our model when trained in a supervised manner. The blue curve corresponds to when the entropy query strategy is used while the green curve corresponds to when the least confidence query strategy is used.

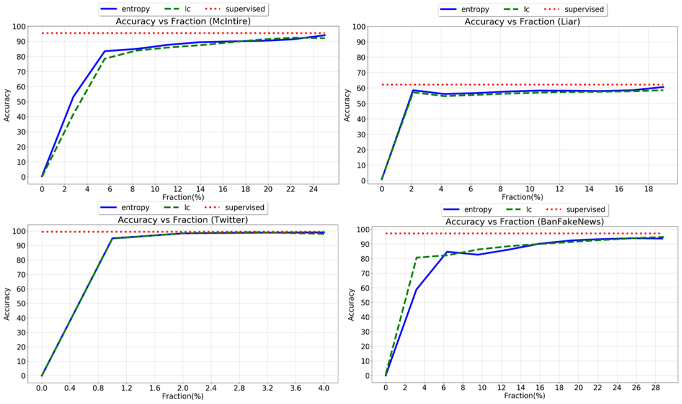

Fig. 2: Accuracy against fraction of dataset for different query strategies on our model

From the graphs, we observe that our active learning model converges towards the accuracy of the supervised model at fairly low fractions of the whole dataset. The percentages are 25% for McIntire, 18% for Liar, 4% for Twitter, and 28% of dataset for BanFakeNews. Also, we find that the entropy query strategy performs better for McIntire and Liar dataset while showing similar performance to the least confidence for Twitter. Only in the case of Bangla fake news do we see a marginal lower performance. Finally, we see that, for our entropy-based training, accuracy is consistently increasing with the fraction of the dataset used, indicating that our method is picking the best data points at each iteration. Hence, our algorithm can be said to be **statistically consistent**.

C. Performance of Ensemble Model with Respect to Active Learning

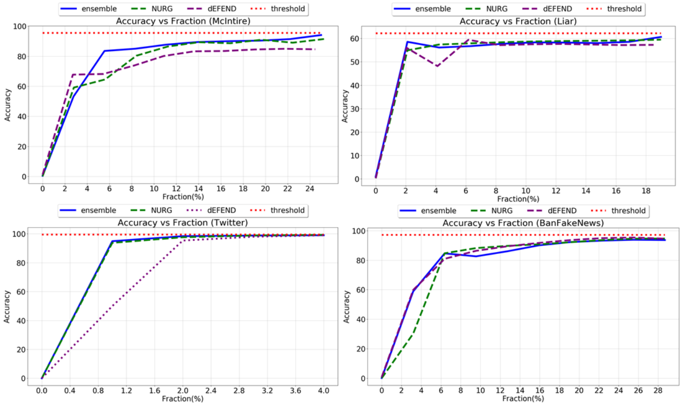

Fig. 3: Accuracy against fraction of dataset for active learning applied to different models

	Naive Bayes		Support Vector Machine		Gradient Boosting		Multi-layer Perceptron		CNN	
Dataset	Test Accuracy	F1 Score	Test Accuracy	F1 Score	Test Accuracy	F1 Score	Test Accuracy	F1 Score	Test Accuracy	F1 Score
George McIntire	86.19	0.8608	92.03	0.9173	88.16	0.8786	91.87	0.9137	93.29	90.9284
Liar	62.67	0.6972	61.68	0.6728	60.01	0.703	60.99	0.6446	59.17	0.6324
Twitter	98.11	0.9811	98.96	0.9896	95.09	0.9486	98.99	0.9898	99.45	0.9945
BanFakeNews	87.69	0.8735	90.19	0.8990	90.48	0.9011	91.54	0.9080	96.35	0.9598

TABLE I: Data from the Naive Bayes, SVM, Boosting, MLP and CNN models in our experiments

	LSTM		dEFEND		NURG		Supervised		Active Learning		
Dataset	Test Accuracy	F1 Score	Test Accuracy	F1 Score	Test Accuracy	F1 Score	Test Accuracy	F1 Score	Test Accuracy	F1 Score	Fraction of Dataset
George McIntire	90.92	0.9053	93.37	0.9320	93.69	0.9320	95.42	0.9541	**94.00**	**0.9407**	**25%**
Liar	58.19	0.6033	60.60	0.6285	58.48	0.6310	62.17	0.7132	**61.04**	**0.6792**	**18%**
Twitter	99.48	0.9948	99.48	0.9948	99.51	0.9951	99.41	0.9941	**99.0**	**0.99**	**4%**
BanFakeNews	97.60	0.9739	97.31	0.9703	97.31	0.9698	97.12	0.9676	**94.13**	**0.9413**	**28%**

TABLE II: Data from LSTM, dEFEND, Neural User Response Generator and our proposed system in our experiments

In this section, we look at how our ensemble model fairs against the two other state-of-the-art models, Neural User Response Generator (NURG) and dEFEND, when all of them are trained using the proposed entropy-based active learning method. Figure 3 shows the results of this analysis through the accuracy against the fraction of the dataset used to train the model for the benchmark datasets. The blue, green, and purple curves correspond to the ensemble, NURG, and dEFEND respectively while the red line is the accuracy of the ensemble model during supervised training.

Observing the graphs in Figure 3, we wish to evaluate the performance of our ensemble model compared to the others under the same active learning settings. This can be done by observing whether, using the same active learning method, the comparison models achieve the threshold accuracy at a much lower fraction than our model. From Figure 3, we can see that our model performs better or the same for each of the datasets. For the McIntire and Liar dataset, the ensemble model achieves the threshold accuracy at a lower fraction before the other two models. For the Twitter and BanFakeNews datasets, all three models show similar performance as they converge to the threshold accuracy at the same fraction of the dataset.

V. Conclusion

In this research, we implement a novel active learning-based multi-model neural ensemble architecture for automated fake news detection using low amounts of data to pre-emptively prevent the detrimental spread of fake news. We tested our methods on several real-life datasets and against benchmarks to prove the efficacy of our model. The simulation results show that our proposed method significantly reduces the burden of labeling large news datasets for training a classifier. As part of future work, we wish to develop similar architecture based on active learning to other problems where data categorization and annotation are difficult and cumbersome such as political NLP research, hate speech, etc.

Acknowledgment

This work was supported by the ICT Division, Government of the People's Republic of Bangladesh.

References

[1] T.-S. Chang and W.-H. Hsiao, "Time spent on social networking sites: Understanding user behavior and social capital," *Systems Research and Behavioral Science*, vol. 31, no. 1, pp. 102–114, 2014.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[3] W. Y. Wang, "" liar, liar pants on fire": A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, 2017.

[4] S. D. Bhattacharjee, A. Talukder, and B. V. Balantrapu, "Active learning based news veracity detection with feature weighting and deep-shallow fusion," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 556–565.

[5] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "defend: Explainable fake news detection," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 395–405.

[6] F. Qian, C. Gong, K. Sharma, and Y. Liu, "Neural user response generator: Fake news detection with collective user intelligence." in *IJCAI*, 2018, pp. 3834–3840.

[7] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[8] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, "Rmdl: Random multimodel deep learning for classification," in *Proceedings of the 2nd International Conference on Information System and Data Mining*, 2018, pp. 19–28.

[9] S. Kwon, M. Cha, and K. Jung, "Rumor detection over varying time windows," *PloS one*, vol. 12, no. 1, 2017.

[10] M. Z. Hossain, M. A. Rahman, M. S. Islam, and S. Kar, "Banfakenews: A dataset for detecting fake news in bangla," *arXiv preprint arXiv:2004.08789*, 2020.

[11] M. Granik and V. Mesyura, "Fake news detection using naive bayes classifier," in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. IEEE, 2017, pp. 900–903.

[12] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks." in *Ijcai*, 2016, pp. 3818–3824.