problem is to get the claims corroborated by professionals such as journalists based on evidence that is available. One such organization that does the fact checking is PolitiFact [5] which has three editors to inspect a news byte. However, with increasing amount of data, this process becomes more tedious, cumbersome requires more man power and leads to inflated costs. Fake news detection as an autonomous task has been researched in different perspectives owing to the increasing development in areas of Data Mining, Machine Learning (ML) and NLP.

Even though Deep Learning (DL), a subset of ML performs terrifically well in both Computer Vision (CV) as well as NLP domains [6-10], there are certain constraints that need to be addressed before utilizing DL techniques them for any task. These constraints that need to be taken care of are, firstly, these algorithms need a huge chunk of labelled data (comparable to ImageNet dataset [11]) and heavy computational resources for appropriate training which can be sometimes difficult as well as expensive to acquire. Secondly, DL can create the problem of underfitting and overfitting which leads to poor results. To overcome these constraints researchers have introduced an approach notably referred to as "Transfer learning" (in this paper, fine-tuning is used as a transfer learning technique) [12-13]. The fundamental rule in this approach is to reuse a model trained for some task as an initial point for a model which is to be trained for the target downstream task. Transfer learning has already shown exceptional results for most of the CV tasks [14-15] in the past two-three years and in practice nowadays, researchers rarely train a DL model from scratch. Transfer learning which was earlier limited to CV tasks is now possible to do in the domain on NLP also with the introduction of recent language representation models such as ELMO [16], ULMFiT [17], Open AI transformer [18] and the most recent state-of-the-art, Google's BERT [19]. Transfer learning has performed well on natural language understanding tasks like that of Common-sense Machine Comprehension (CMC); that enables learning of temporal and causal learning [20].

In this paper, we propose to fine-tune the 12-layered BERT language representation model for downstream task of classification of fake news. BERT is a bidirectional model which itself is based upon a transformer architecture [21]. In addition, we train 2 more classification models, one of which is based on an ML algorithm, gradient boosted trees and other is a single-layered RNN (LSTM) model. Experimental evaluations of these models demonstrate that proposed method of fine-tuning BERT is able to outperform other 2 models on the classification task by a good margin.

The paper follows the following structure. In Section 2, the relevant work in NLP domain and in the detection of fake news is discussed. Following it, is the Section 3, in which an in-depth background for BERT language representation model and provide the step-by-step directions for fine-tuning the model using GPUs is given, and after which our proposed methodology is demonstrated. After the previous section, Section 4 gives the results and comparative analysis with different approaches. Finally, to conclude, Section 5 and 6 have the discussion and final conclusion of this research study highlighting its application and the future scope.

2. Related works

In this section, we start by discussing how definition of "Fake news" has evolved over time. Then, we discuss existing works and methods that are applicable for the task of fake news classification. Different researchers have applied significantly different approaches for tackling fake news and to achieve decent progress on combating this challenge, so we study these approaches based on their type i.e. whether the method applied is content-based, feedback-based or is based on the social media engagement of users. After this, we give an overview of available datasets that have been used in the past for this task.

Disinformation and misinformation which is colloquially known as "Fake News", isn't a new phenomenon. It has recently garnered much attention due to 2016 US presidential elections, as can be observed by looking at the term on Google Trends [22]. Misinformation was present before 2016 election as well, as is evident by studies conducted on misinformation before 2016 which shows that misinformation has wide ranging effects that range from financial loss, to politics. One such instance is a 2008 false bankruptcy story about UAL parent company which led to 76% drop in stock price [23].

In content-based based methods, the fundamental basis is that the textual and linguistic features of a real news will differ from that of a fake news. There are hand-engineered ways of extracting these cues as well as more recent DL methods. One of the earliest hand-engineered features-based methods, Scientific Content Analysis (SCAN) proposed in 1987 was primarily developed for polygraph examinations and consisted of cues such as grammatical errors, continuity in written paragraphs and provided information [24]. While the method did seem promising in its early days but was later proved ineffective [25, 26]. SCAN also required experts to rigorously analyse the content. As, there always has been efforts to decrease human labour for these tasks, another linguistic-based method was developed by Fuller et al. [27]. Authors created a comprehensive set of 31 linguistic cues which were further refined using 3 classifiers to have only 8 cues for deception detection. These cues were based on the previously proposed different cue sets in the linguistic field [28 – 30]. One main limitation of this work was that the cues were largely dependent upon topic or domain of the text and the model was not able to generalize well when tested on contents from different domains [31]. Relatively recent feature-based methods include analysis-based methods such as punctuation marks [32], regular expressions [33], platform (Twitter, Facebook, Wikipedia, etc.) specific cues such as like counts, hashtags [34, 35].

While the method of hand-designing the features and cues is much interpretable but also has disadvantages such as need of re-drawing based on domain, platform or situation

of the content, human involvements and lack of generalization. To improve generalization ability of detection models, researchers have also utilized more effective ways of extracting features such as N-gram [36, 37]. Term frequently vectors are created using N-grams and then these are sent to different classifiers like Support Vector Machine (SVM). While using N-grams did improve the performance but being a simple approach could not capture all the features in the different writing styles. Some researchers also devised the classification models that instead of being word-based like N-grams, are based on the syntactical part of writing that exploits Part of speech (POS) tags or are derived from Probabilistic Context Free Grammars (PCFG) [38-40]. These approaches lacked ability to capture clues across long news articles and were even weaker as compared to word-based approaches.

Process of feature extraction is now automated with the advent of DL. Deep neural networks are able to extract simple as well as complex features that are not intuitive. Wang et al. used two popular forms of neural networks, one is Convolutional Neural Network (CNN) and the other is bidirectional LSTMs for embedding the statement text and speaker metadata information into lower dimensions and then fed to classifier for classifying fake news based on the content [41]. They also made use of word embedding known as word2vec for capturing useful contextual properties [42]. Quian et al. proposed a Two-level CNN which first generates sentence embedding using words and then utilize the sentence embedding to create article embedding [43]. Their proposed variant of CNN was able to outperform the generic CNN. Variants of CNNs and RNNs have occasionally been used over the past decade for the same task [44 - 46].

While most of the work done on detection of Fake News has been on building supervised models, there are unsupervised techniques that have been employed to detect the credibility of a post. Yang et al. uses opinions of users on social media towards authenticity of a story and uses Bayesian networks to build a probabilistic graphical model that treats truth of news and user credibility as latent random variables [47].

The underlying basis of feed-back based methods are the secondary information such as user's comments, news' propagation graph in the social media and other user-related information. Researchers have tried developing hand-engineered features for these methods such as number of followers, content of tweets, depth of retweets, geographical location etc. [48 - 50]. The route of retweets or shares of a news articles and how it propagates through the social media web has been extensively analysed by researchers. Ma et al. utilized Jaccard similarity to compute similarity scores of propagation trees of users [51, 52]. Texts of user's comment along with article's text also give rise to an informative model for fake news classification as it is highly likely that fake news articles will have fewer positive comments as compared to real news articles [53]. Shu et al. proposes a novel method of detection of fake news that uses TriFN which is a tri relationship embedding framework between the users, publishers and news pieces; this auxiliary information improves significantly improves upon the baseline models [54]. Propagation patterns of articles can also be useful features in detection of fake news, as was demonstrated by Monti et al. where geometric deep learning was used for creating a model to detect fake news. Heterogenous data like user profile and activity, content, news spreading patterns and structure of social network is fused together underlying by using algorithms that are a generalization of classical convolutional neural networks to graphs [55].

Methodologies discussed so far have been applied on variety of datasets in the past. There are several novel datasets that have been made available solely for the task of fake news detection. These datasets do vary largely with each other as some may solely comprise of articles related to politics while some may be related to any other particular domain. Additionally, datasets also vary on the kind of data present in them as some may contain very short statements while other can have large articles. In the following paragraph, we summarize some of the popular datasets. Dataset that we use is discussed in detail in the later section.

LIAR dataset available for detection of fake news has 12.8k labelled short political statements collected over a period of 9 years (from 2007 to 2016) from POLITIFACT.COM. Precisely, labels in this dataset are: true, mostly-true, half-true, barely-true, false and pants-fire. Number of claims per class are roughly equal in size. Another dataset, FEVER, short for Fact Extraction and Verification has 185,445 claims. These claims were created by extracting data from Wikipedia and then the claims were verified without prior knowledge of the sentences of origin. These claims have been classified into three classes: supported, refuted or notenoughinfo and have also been verified by skilled annotators [56]. As present form of fake news is mostly present on social media, datasets such as BUZZFEEDNEWS contain 2282 samples published using Facebook by 9 news agencies one week before the 2016 US elections. Every post or link is checked and verified by 5 BuzzFeed journalists. Labels in this dataset are: mostly true, mixture, mostly false and no factual content [57]. A similar dataset, Some-like-it-hoax dataset consists of 15,500 Facebook posts and 909,236 users that are classified as either hoax or not hoax [58]. PHEME dataset is a collection of 6425 tweets that are rumours and non-rumours and were posted during the time of some breaking news. 60% of samples are non-rumours, 16% are true rumours, 10% are false and rest are unverified. Most of the contents in the dataset have been verified by journalists and via crowd-sourcing. The CREDBANK dataset is a set of tweets that were traced over a period of around 4 months during 2014-2015. Along with the tweet's content, it consists of topics classified as events or non events that are annotated with ratings stating their credibility [59]. FAKENEWSNET [60] is yet another popular database of News Content and gives a better understanding of how fake news is present on the social media.

3. Methodology

Fake news keeps evolving every day, which brings the need of creating an end-to-end classification model which is robust and requires minimal computation and pre-processing. In order to achieve this we have leveraged the power of transfer learning in context of NLP by fine-tuning BERT for downstream task of fake news classification, to show how the technique of fine-tuning fares in accomplishing the task of classification of fake news its performance has been compared to that of gradient boosted trees and LSTMs.

3.1. Overview of BERT

BERT is a language representation model, which was introduced by Google AI. BERT is first of its kind language representation that can be utilized to pre-train deep bidirectional representations by taking into consideration left and right contexts.

Previous work in pre-training representations like in OpenAI GPT and ELMo are unidirectional and shallow bidirectional respectively, as opposed to BERT which is deeply bidirectional. BERT removes the constraint provided by the unidirectional approach by using Masked LM (Masked Language Model) as a pre-training objective. BERT crossed the threshold of eleven state-of-art NLP tasks. Hence, BERT provided us with an approach that can yield state-of-art results without using heavily engineered and task-specific architectures. BERT in its input representation uses three embedding layers, they are described below. The final input embedding is a summation of the three embeddings. Figure 1 demonstrates the block diagram for BERT-based classification model.

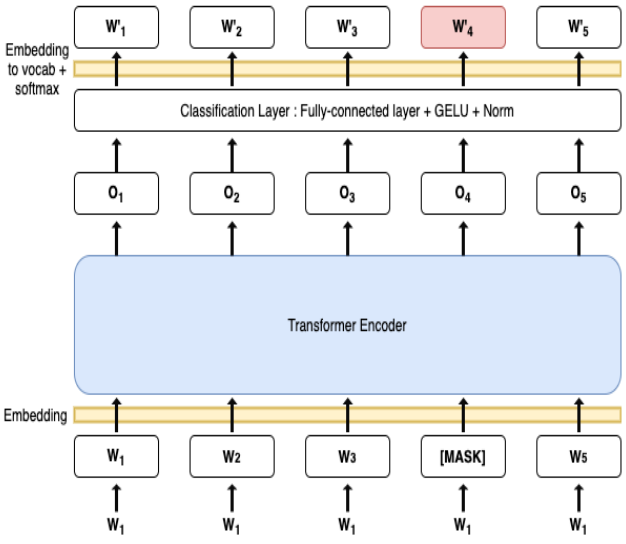


Figure 1. Block Diagram of BERT for classification.

• Token Embeddings – In BERT it transforms the words into a fixed 768-dimension vector representation. [CLS] and [SEP] tokens are appended to the start and end of the tokenized sentence; they serve as input representations for classification tasks and to separate input texts. BERT cleverly used WordPiece tokenizer which enables BERT to store only 30522 words and rarely encounter any out-of-vocabulary word.

• Segmentation Embeddings - BERT can perform text classification tasks if given a pair of text input. An example of this is to classify if two pieces of text are semantically similar. So, the text is concatenated and fed to BERT, BERT distinguishes between text with the help of Segment Embedding. Segment Embedding layer only has two vector representations; first vector (EA) is assigned to tokens of input1 and second vector (EB) is assigned to tokens of input2.

• Positional Embeddings – Used learned positional embeddings were used in BERT; this was done using the functions that were used to calculate positional encodings in transformer. Positional Embeddings used here understands the relative positions instead of just the absolute ones. This is done by adding a sinusoidal function depending on the position of token i in the sequence of sentence and j for the position of embedding feature to the 768-dimensional vector representation of words, which yields a slightly different position of the same word in different positions.

$$p_{i,j} = \begin{cases} \sin\left(\dfrac{i}{10000^{\frac{j}{d_{embeddingdimension}}}}\right) & when\ j\ is\ even \\ \cos\left(\dfrac{i}{10000^{\frac{j-1}{d_{embeddingdimension}}}}\right) & when\ j\ is\ odd \end{cases} \quad (1)$$

Being an attention-based architecture, BERT uses Encoder which is introduced in the architecture of Transformer (contains Encoder and Decoder). In BERT, N encoders are stacked together to give the Encoder output. Different encoding block finds different relationships between the input representations and encodes them in its output.

BERT used a novel approach to use bidirectionality, by pre-training on “masked language model” and “next sentence prediction” instead of pre-training the model on a language model. MLM (Masked Language Model), forces the model to predict the masked tokens. 15% of all WordPiece tokens are masked, of which 80% of the time word is replaced by [MASK] token, 10% of the time it is replaced by a random word and rest of the time the word is left unchanged. The model tries to predict the correct value of masked words, based on the context given by words that aren’t masked in the sequence. Technically, three steps are required to predict output words. Firstly, on top of an encoder layer a classification layer is to be added. Secondly, Multiplying the embedding matric by the output

vectors, transforming them into the dimension of the vocabulary. Lastly, Calculating the probability of every single word in the vocabulary using softmax.

MLM cannot successfully capture the relationships between two sentences, which is important for tasks like question answering and natural language inference. To better understand the relationships between two sentences BERT authors used next sentence prediction which is nothing but a classification task that finds if sentence B follows sentence A or not. 50% of the training examples are correct, and rest are chosen at random to generate a wrong pair of sentences.

The corpus for pre-training as built entirely from BookCorpus (800 million words) [61] and English Wikipedia (2,500 million words). The pre-training samples were generated in batches of two such that the length of two sentences that were chosen was less than 512 tokens and in our case 256 tokens. The training loss was computed as the sum of mean MLM likelihood and mean next sentence prediction likelihood.

3.2. Dataset

At present, there are various datasets available to work upon in the domain of fake news classification. Each of the datasets has its advantages as well as disadvantages, as we discussed in Section 2. In this paper, the proposed method has been applied on NewsFN Dataset [62]. This dataset has 6335 items, consisting of the headline and text of the news articles on politics from a wide range of news sources that are classified as either “Fake” or “Real”. Precisely, 3164 articles are labeled as “Fake” and 3171 as “Real”. The ratio of the number of “Fake” articles to that of “Real” articles is roughly 1:1 hence dataset is well balanced with respect to the two classes, and there is no need for oversampling or under sampling. Figure 2 describe the word clouds corresponding to each of the two classes. It illustrates the most frequent words other than the stop words that are present in the dataset.

(a) (b)

Figure 2. Wordclouds representing most frequent words other than stopwords in (a) Fake News and (b) Real News articles.

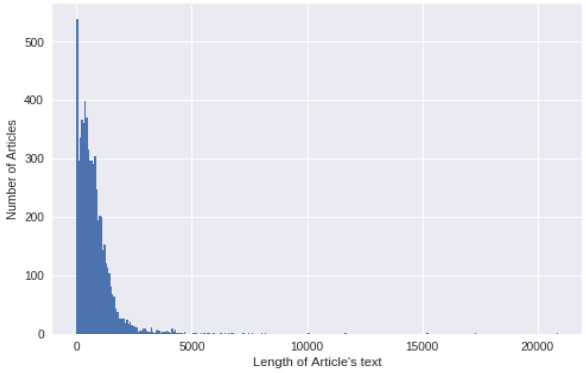
This dataset does not comprise of any kind of URLs. However, our work is primarily concerned with the textual context rather than external links. Additionally, we checked the authenticity of most of the news articles by randomly picking them from the dataset and comparing them with the reputed online news sources.

3.3. Data pre-processing

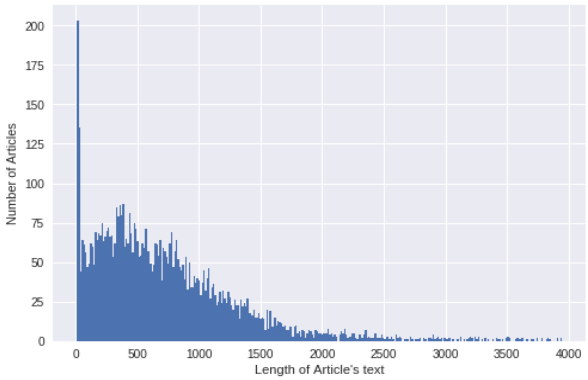
Before passing the data through the classification model, it is required to do some pre-processing of the texts so that our model’s performance doesn’t get hindered by some level of noise that is present in the dataset. First, we remove outliers from the dataset to decrease overall variance in the data.

3.3.1 Removing outliers

An outlier is a data point that is at a distant from other data points in a particular dataset. After analysis of the dataset, it was found out that the length of articles’ texts varies largely in terms of number of words. Median text length in the dataset is 597, 1% of the dataset has length < 10 and 1% has length > 3958. So, in order to remove such high variance in the dataset, only those articles that don’t fall in either of the above categories were kept, resulting in total of 6210 items. Figure 3 shows the distribution of text lengths before and after removing outliers. Table 1 summarizes the related statistical information about the dataset.



(a)



(b)

Figure 3. Distribution of length of article’s texts in number of words (a) Before removing outliers; (b) After removing outliers

Table 1: Statistical information about dataset before and after removing outliers

	Before removing outliers	After removing outliers
No. of articles labeled as "Fake"	3164	3073
No. of articles labeled as "Real"	3171	3137
Median length of articles' text	597.0	597.0
Average length of articles' text	776.30	731.49
Maximum length of articles' text	20891	3947
Minimum length of articles' text	0	10

3.3.2. Removing noise from text

To increase the value of performance metrics, unimportant entities or noises in the texts such as punctuation marks, numerical values, new-line characters were removed. Removal of these entities decreases size of sample space of possible feature sets and hence improves the level of performance.

3.3.3. Partitioning dataset into training and testing sets

To ensure that the classification model doesn't overfit on the dataset and that testing is done on a batch of data that hasn't been seen by the training model, the dataset was divided in two parts, namely, training and testing sets with ratio 80:20 i.e. eighty percent of the data is utilized for training and rest twenty percent of data is utilized for testing. Demographics of these datasets are demonstrated in Table 2. The training set will be used by the model to minimize error rate i.e. to find patterns in the data. While test set will be the one on which assessment of performance of the model is done, and which is not seen by the model during training.

Table 2: Demographics of Training and Testing sets

	Training set	Testing set
No. of articles labeled as "Fake"	2453	620
No. of articles labeled as "Real"	2515	622
Total no. of articles	4968	1242

The main goal of this paper is to test the robustness of the proposed method of fine-tuning BERT for classification of fake news, so text pre-processing is kept as minimal as possible. In the next section, little tweaks that are to be made in the model and how parameters of this state-of-the-art language model can be fine-tuned for our downstream task of fake news classification are demonstrated.

3.4. Preparing BERT model for binary classification

In the past, researchers have demonstrated how pre-trained language representations can be applied to improve on many supervised downstream tasks such as natural language inference, question answering, etc. The main advantage of this method over training from scratch is that there are only a few parameters that are required to be trained from scratch. Specifically, researchers have devised two inexpensive strategies for making use of pre-trained language representations. First strategy is feature-based approach where pre-trained representations are used as additional features for the downstream task. Second strategy is fine tuning in which we train the downstream tasks by fine-tuning i.e. tweaking the pre-trained parameters.

Feature-based approaches such as ELMo are effective but also require task-specific architectures, while it is not the case for fine-tuning. Since Google's BERT is one of the present state-of-the-art and most powerful models which has improved benchmark for several datasets, henceforth, we primarily focus on applying a pre-trained BERT model for binary classification by fine-tuning it.

Google Research has open-sourced the implementation of BERT in TensorFlow and has also released several pre-trained models. At the time of our study, there are 6 pre-trained models that have been released by Google. Table 3 gives an overview of these models. The models primarily vary in the total number of parameters and how computationally expensive they are, BERT-Large is more compute-intensive as opposed to BERT-Base.

For our experiment and considering the amount of computational power available, the smallest and most simplistic, BERT-Base, uncased model for task of fake news classification is used. This model comprises of 12 attention layers with a total of 110M parameters. Moreover, before fine-tuning, all the text is converted to lowercase by the tokenizer which comes along the implementation provided by Google.

BERT uses WordPiece embedding [63] with a vocabulary of 30,000 tokens where split word pieces are denoted with "##". Additionally, it uses learned positional embeddings for transforming text and the maximum supported sequence length by the model is 512 tokens. As it can be seen from Table 1, median text (sequence) length in our case after removing outliers is 597, also, due to the limitation of resources and better (lesser) running time, we set the sequence length to be 256 tokens for each of the text. Articles that have sequence length less than 256 are padded

with zeros while those with greater than 256 tokens are stripped down to 256.

Table 3: Variation of BERT models open-sourced by Google Research

Type	No. of layers (L)	No. of hidden units (H)	No. of self-attention heads (A)	Total no. of parameters
BERT-Base, Uncased	12	768	12	110M
BERT-Large, Uncased	24	1024	16	340M
BERT-Base, Cased	12	768	12	110M
BERT-Large, Cased	24	1024	16	340M
BERT-Base, Multilingual cased	12	768	12	110M
BERT-Base, Chinese	12	768	12	110M

As the BERT model can be used for multiple tasks, for specifying the type of task as classification, the first token of every sequence in the training as well as test set is fixed as a special classification embedding ([CLS]). The output of the transformer i.e. the last hidden state corresponding to this special token is then used as a cumulative sequence representation for performing classification by the model. This output can be represented as a vector $C \in \mathbb{R}^H$ where H is the hidden size.

New parameters that are added at the time of fine-tuning for classification are $W \in \mathbb{R}^{K \times H}$, where K is the number of classes, here 2 ("Fake" and "Real"). Probabilities for these K labels are computed as:

$$P = \frac{e^{C.W^T}}{\sum_K C.W^T} \quad (2)$$

Where, $P \in \mathbb{R}^K$ are the label probabilities. So, the pre-trained parameters of BERT-Base, uncased model and parameters of classification layer W are jointly fine-tuned for maximizing the log-probability corresponding to the correct label that is either "Fake" or "Real".

For training of parameters, Adam optimizer is chosen, which is also recommended by Google [64]. Adam optimizer is an effective optimization algorithm which has the ability of computing adaptive learning rates for each of the parameters and is more specifically a combination of RMSprop and traditional stochastic gradient descent with momentum [65, 66]. This was specifically designed to train deep neural networks and update the parameter value as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (3)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (4)$$

Where m_t and v_t are mean estimate and variance at t_{th} time step of the gradients g_t respectively. β_1 and β_2 are the decay rates. Moments m_t and v_t are then corrected for bias as:

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t} \quad (5)$$

$$\widehat{v_t} = \frac{v_t}{1 - \beta_2^t} \quad (6)$$

Where, $\widehat{m_t}$ and $\widehat{v_t}$ are the corrected m_t and v_t respectively. These are then used to update parameter W as:

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{\widehat{v_t}} + \epsilon} \widehat{m_t} \quad (7)$$

Where η is the learning rate and ϵ is a smoothing term.

Except learning rate, batch size and number of epochs, most of the hyper-parameters of the model are kept the same as the loaded pre-trained model. During our experimentation, we found out that performance of the model is best when learning rate η is 2e – 5, batch size is 16 and number of epochs E which implies the number of times training set is passed through the model is kept 4. Furthermore, for avoiding overfitting, we use dropout regularization with dropout probability ratio of 0.1. The overall methodology is summarized in Figure 4 below.

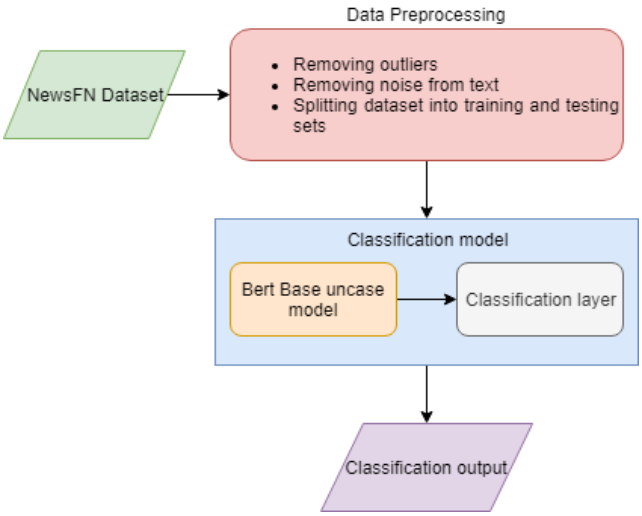


Figure 4. Proposed methodology

4. Experimental results

For performance evaluation of the model, accuracy is chosen as the primary metric for evaluation since the training set, as well as the test set, are completely balanced. It took around one hour for fine-tuning the model with

training set along with the evaluation of test set on NVIDIA Tesla K80 accelerated GPU. Final loss on training set came out to be 0.1450 while for the test set, resultant loss value is 0.1457 which is nearly the same and hence model is able to generalize excellently. Accuracy on the test set is 97.021% which itself is promising and amongst state-of-the-art models for classification of fake news.

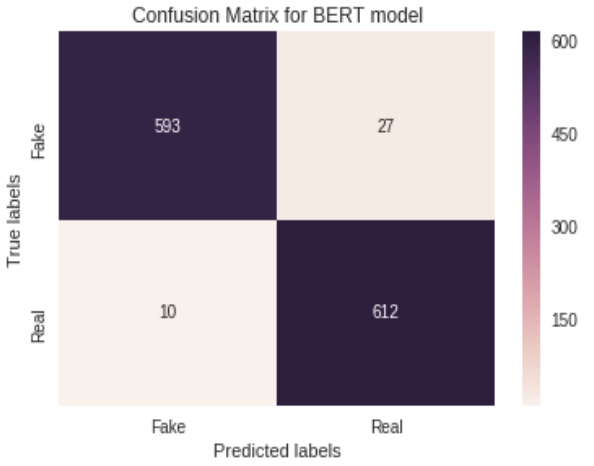
Furthermore, for comparing the performance of fine-tuned BERT model with other relatively older approaches, we trained two more classification models from scratch i.e. without any kind of transfer learning. First, we used a popular machine learning technique, gradient boosting decision trees and implemented it using highly optimized XGBoost library [67]. Briefly, gradient boosting yields a prediction model in the form of an ensemble of weak prediction models, such as decision trees. To represent words of the text in numerical form, we used tf-idf which stands for the product of term frequency and inverse document frequency. Tf-idf basically reflects how important a word is to a document in a corpus. For this classifier, most of the hyperparameters are kept as default except number of decision trees that are equal to 100 in our case and maximum depth of a particular tree is kept as 6.

The second classifier for comparison is LSTM (Long Short-Term Memory) network which is a special kind of RNN. LSTMs are usually the first choice when it comes to NLP problems as they have the ability to remember key information for a longer period as compared to other sequence models. Similar to the BERT, LSTM also requires a fixed sequence length. In our case, we use a sequence length of 512 for LSTMs which is much closer to median sequence length. Each of the words in the texts needs to be represented in numerical form hence before passing data through LSTM layers, an embedding layer of size 400 is also added whose parameters are trained along with LSTM's parameters. To avoid overfitting, a dropout layer with dropout ratio 0.5 is added after LSTM layers. Dropout layer is then followed by classification layers. This model is also trained with Adam optimizer. Batch size for training this model is 43. Accuracies for all three models on the test set are shown in Table 4. Values for other evaluation metrics are compared in Table 5.

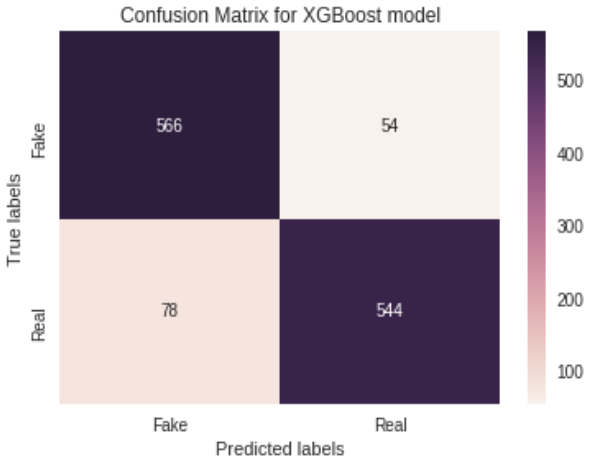
Table 4: Accuracy comparison for three models

Model	Accuracy (in percentage)
Fine-tuned BERT	97.021%
XGBoost	89.372%
LSTM	86.231%

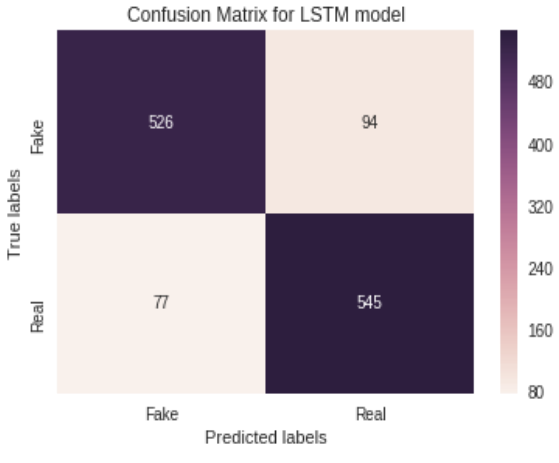
Figure 5 below illustrates the comparison of a number of articles that are correctly or incorrectly predicted by the three models that are trained.



(a)



(b)



(c)

Figure 5. Confusion Matrix showing the number of articles from Test set that are predicted as either "Fake" or "Real" by (a) BERT model; (b) XGBoost model and (c) LSTM model

We futher computed Precision, Recall and F1-Score for evaluation and comparative analysis of the above defined classification models. These metrics are computed as:

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (10)$$

Figure 6 gives the comparison of ROC (Receiver Operating Characteristic) curves for three models which are created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. Values for the area under the ROC curves (ROC AUC score) are computed using prediction probabilities for "Real" class and are shown in Table 5.

Table 5: Comparison of Precision; Recall; F1-Score and ROC AUC Score for predictions on Test set

Model	Precision		Recall		F1-Score		ROC AUC Score
	Fake	Real	Fake	Real	Fake	Real	
Fine-tuned BERT	0.98	0.95	0.95	0.98	0.97	0.97	0.99
XGBoost	0.87	0.91	0.91	0.87	0.89	0.89	0.96
LSTM	0.87	0.85	0.84	0.87	0.86	0.86	0.92

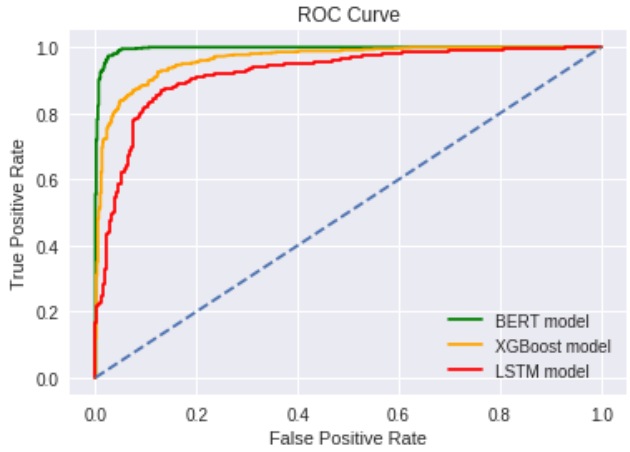


Figure 6. ROC Curve Comparison

5. Discussion

Detection of Fake News is gaining a lot of traction among researchers because of its complexity, and the requirement for an algorithm that can filter thousands of news articles and judge their authenticity in a matter of minutes. Earlier techniques to deal with this problem involved a lot of hand-engineering of features, this led to poor generalization of the models which is very important in the case of this task. With the advances made in DL the extraction of features is not something that has to be hand engineered, since, deep neural networks can extract non-intuitive complex features which may not have been possible even with hand-engineering of features. This promises a better chance of generalizability of a model.

One of the most recent advances in the field of NLP has been the introduction of transformer architecture and the BERT model which when fine-tuned with the addition of one layer was able to set benchmarks on various NLP tasks without explicitly being trained for doing those tasks.Transfer learning is a powerful approach that can adapt well to different tasks. In this paper a framework based on natural language processing (NLP) is proposed to address this task of classifying news articles as either fake or real using Fine Tuned BERT model. The BERT model considerably outperforms other approaches even with minimal to no engineering of features. The results show us that transfer learning can yield good results in the case of detection of fake news as well. The fine-tuned BERT system can achieve an accuracy of 97.021 per cent on NewsFN data and is capable of surpassing the other two models by approximately eight per cent.

6. Conclusion and Future Scope

In this paper, a powerful and time efficient approach is proposed for accurately classifying news articles into two classes: Fake and Real. Here, in this research study, we utilized the robustness of pre-trained BERT language model and applied a highly successful approach of transfer learning for converting the model into a classification model. We also built two more models for comparison and computed values for various evaluation metrics to support the performance of the proposed method of classifying fake news. Specifically, on NewsFN dataset the pre-trained BERT model is able to classify news articles with accuracy of 97.021% which is a significant improvement over other traditional approach.

For the future, we recommend researchers to try the same approach on datasets that comprise of much diverse news articles. Also, instead of limiting the number of classes to only two, researchers can include various more nuanced classes. Moreover, the overall performance of this approach can be improved by fine-tuning larger BERT models, provided the available dataset is large enough and there are enough computational resources to handle the increased computational complexity. One of the most concerning factors for this research task is to get a properly labeled dataset, as currently, there is no particular dataset that is diverse enough to build a state-of-the-art mechanism for fake news detection.

Needless to say, there is a long way to go tackle the problem of fake news detection, transfer learning promises to be a strong means of progress in the field. In our research we have chosen a binary dataset that has news labelled as fake or real, in reality news isn't as black and white and there are certain nuances that are associated with different articles that aim to spread propaganda or fake news, for instance an article may not be entirely fake, just a part of it may be fake. There is a need of large datasets that should be labelled at sentence or paragraph level, so that a more fine-grained level of classification can be achieved.

References

[1] Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of economic perspectives*, *31*(2), 211-36.

[2] Swift, A. (2016). Americans' trust in mass media sinks to new low. Gallup News, 14(6). Retrieved from https://news.gallup.com/poll/195542/americans-trust-mass-media-sinks-new-low.aspx.

[3] Reilly, R., & Nye, R. (2012). Power, principles and the press. Retrieved from http://www. The open road. com/wp-content/uploads/2012/09/Powerprinciples-and-thepress-Open-Road-and-Populus1. pdf.

[4] Oshikawa, R., Qian, J., & Wang, W. Y. (2018). A survey on natural language processing for fake news detection. arXiv preprint arXiv:1811.00770.

[5] Fact-checking U.S. politics. PolitiFact. Retrieved from https://www.politifact.com/.

[6] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[8] Khan, M. A., Khan, M. A., Ahmed, F., Mittal, M., Goyal, L. M., Hemanth, D. J., & Satapathy, S. C. (2020). Gastrointestinal diseases segmentation and classification based on duo-deep architectures. Pattern Recognition Letters, 131, 193-204.

[9] Gautam, R., & Sharma, M. (2020). Prevalence and Diagnosis of Neurological Disorders Using Different Deep Learning Techniques: A Meta-Analysis. Journal of Medical Systems, 44(2), 49.

[10] Mittal, M., Goyal, L. M., Kaur, S., Kaur, I., Verma, A., & Hemanth, D. J. (2019). Deep learning based enhanced tumor segmentation approach for MR brain images. Applied Soft Computing, 78, 346-354.

[11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S.,& Berg, A. C. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[12] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In Advances in neural information processing systems (pp. 3320-3328).

[13] Bawa, V. S., & Kumar, V. (2019). Emotional sentiment analysis for a group of people based on transfer learning with a multi-modal system. Neural Computing and Applications, 31(12), 9061-9072.

[14] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. In International conference on artificial neural networks (pp. 270-279). Springer, Cham.

[15] Ramírez, I., Cuesta-Infante, A., Pantrigo, J. J., Montemayor, A. S., Moreno, J. L., Alonso, V., & Palombarani, L. (2018). Convolutional neural networks for computer vision-based detection and recognition of dumpsters. Neural Computing and Applications, 1-9.

[16] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[17] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.

[18] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.

[19] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[20] Han, W., Peng, M., Xie, Q., Hu, G., Gao, W., Wang, H., ... & Liu, Z. (2019). DTC: Transfer Learning for Common sense Machine Comprehension. Neurocomputing.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N.,& Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[22] Fake news - Explore - Google Trends. Retrieved from https://trends.google.com/trends/explore?date=2013-12-06 2018-01-06 &geo=US & q=fake news.

[23] Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In Proceedings of the 20th international conference on World wide web (pp. 675-684).

[24] Sapir, A. (1987). The LSI course on scientific content analysis (SCAN). Phoenix, ZA: Laboratory for Scientific Interrogation.

[25] Bogaard, G., Meijer, E. H., Vrij, A., & Merckelbach, H. (2016). Scientific content analysis (SCAN) cannot distinguish between truthful and fabricated accounts of a negative event. Frontiers in psychology, 7, 243.

[26] Nahari, G., Vrij, A., & Fisher, R. P. (2012). Does the truth come out in the writing? Scan as a lie detection tool. Law and Human Behavior, 36(1), 68.

[27] Fuller, C. M., Biros, D. P., & Wilson, R. L. (2009). Decision support for determining veracity via linguistic-based cues. Decision Support Systems, 46(3), 695-703.

[28] Zhou, L., Burgoon, J. K., Nunamaker, J. F., & Twitchell, D. (2004). Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. Group decision and negotiation, 13(1), 81-106.

[29] Buller, D. B., & Burgoon, J. K. (1996). Interpersonal deception theory. Communication theory, 6(3), 203-242.

[30] Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. Mahway: Lawrence Erlbaum Associates, 71(2001), 2001.

[31] Ali, M., & Levine, T. (2008). The language of truthful and deceptive denials and confessions. Communication Reports, 21(2), 82-91.

[32] Rubin, V. L., Conroy, N., Chen, Y., & Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. In Proceedings of the second workshop on computational approaches to deception detection (pp. 7-17).

[33] Zhao, Z., Resnick, P., & Mei, Q. (2015). Enquiring minds: Early detection of rumours in social media from enquiry posts. In Proceedings of the 24th international conference on world wide web (pp. 1395-1405).

[34] Morris, M. R., Counts, S., Roseway, A., Hoff, A., & Schwarz, J. (2012). Tweeting is believing? Understanding microblog credibility perceptions. In Proceedings of the ACM 2012 conference on computer supported cooperative work (pp. 441-450).

[35] Chen, Y., Conroy, N. J., & Rubin, V. L. (2015). Misleading online content: recognizing clickbait as" false news". In Proceedings of the 2015 ACM on workshop on multimodal deception detection (pp. 15-19).

[36] Mihalcea, R., & Strapparava, C. (2009). The lie detector: Explorations in the automatic recognition of deceptive language. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers (pp. 309-312). Association for Computational Linguistics.

[37] Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1 (pp. 309-319). Association for Computational Linguistics.

[38] Biber, D., Johansson, S., Leech, G., Conrad, S., Finegan, E., & Quirk, R. (1999). Longman grammar of spoken and written English (Vol. 2). London: longman.

[39] Rayson, P., Wilson, A., & Leech, G. (2002). Grammatical word class variation within the British National Corpus sampler. In New Frontiers of Corpus Research (pp. 295-306). Brill Rodopi.

[40] Johnson, M. (1998). PCFG models of linguistic tree representations. *Computational Linguistics*, *24*(4), 613-632.

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

[42] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[43] Qian, F., Gong, C., Sharma, K., & Liu, Y. (2018). Neural User Response Generator: Fake News Detection with Collective User Intelligence. In *IJCAI* (pp. 3834-3840).

[44] Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 2931-2937).

[45] Volkova, S., Shaffer, K., Jang, J. Y., & Hodas, N. (2017). Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 647-653).

[46] Chopra, S., Jain, S., & Sholar, J. M. (2017). Towards automatic identification of fake news: Headline-article stance detection with LSTM attention models. In *Stanford CS224d Deep Learning for NLP final project*.

[47] Yang, S., Shu, K., Wang, S., Gu, R., Wu, F., & Liu, H. (2019). Unsupervised fake news detection on social media: A generative approach. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 5644-5651).

[48] Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In Proceedings of the 20th international conference on World wide web (pp. 675-684).

[49] Kwon, S., Cha, M., & Jung, K. (2017). Rumor detection over varying time windows. *PloS one*, *12*(1).

[50] Yang, F., Liu, Y., Yu, X., & Yang, M. (2012). Automatic detection of rumour on Sina Weibo. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (pp. 1-7).

[51] Ma, J., Gao, W., & Wong, K. F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

[52] Ma, J., Gao, W., & Wong, K. F. (2018). Rumour detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

[53] Chen, T., Li, X., Yin, H., & Zhang, J. (2018). Call attention to rumours: Deep attention based recurrent neural networks for early rumour detection. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 40-52). Springer, Cham.

[54] Shu, K., Wang, S., & Liu, H. (2019). Beyond news contents: The role of social context for fake news detection. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (pp. 312-320).

[55] Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

[56] Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

[57] Silverman, C. (2016). Viral fake election news outperformed real news on Facebook in final months of the us election.

BuzzFeed News, 16. Retrieved from https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperforme-real-news-onFacebook.

[58] Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*.

[59] Mitra, T., & Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. In Ninth International AAAI Conference on Web and Social Media.

[60] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. arXiv preprint arXiv:1809.01286.

[61] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision (pp. 19-27).

[62] fake_real_news_dataset · GitHub. Retrieved from https://github.com/GeorgeMcIntire/fake_real_news_dataset.

[63] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W.,& Klingner, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

[64]Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[65]Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, *14*(8).

[66] Robbins, H., & Monro, S. (1951). A stochastic approximation method. The annals of mathematical statistics, 400-407.

[67] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).