EasyChair Preprint

№ 15828

DEFENSE: Enhancing Fake News Detection on COVID Through Transformer Based Feature Engineering and Sentence Embedding Approach

C Oswald and Allen Puthenparambil Alex

February 14, 2025

DEFENSE: Enhancing Fake News Detection on COVID through Transformer based Feature Engineering and Sentence Embedding Approach

Journal Title
XX(X):1–20

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Abstract

The current global COVID-19 pandemic has wreaked havoc in our daily lives, both physically and mentally. An enormous amount of fake news and misinformation about COVID-19 has spread fast across social media platforms, as people rely heavily on them for current updates. Inestimable harm on human lives can be caused by the surmise, misconceptions, fear and the spread of rumours. Detecting such fake news and blocking their spread is of predominant importance and an influential research problem as well. Some primary challenges in fake news detection involve lack of contextual understanding of the social media post and the absence of a concrete feature engineering mechanism in analysing the contents of the post. In this article, we present DEFENSE, a Transformer-based model for fake news detection in social media posts. We focus on constructing a precise and concrete feature engineering model to extract the textual and sentimental features like sentiment polarity and sentiment subjectivity, of a post. Moreover, we use an efficient mechanism to extract the contextual meaning of the post using various sentence embedding methods. In order to reduce overfitting and increase accuracy, our model is trained to remove multi-collinearity through dimensionality reduction, before classifying with an extensive set of classifiers. Comprehensive experiments on the benchmark dataset namely Contraint@AAAI 2021 COVID-19 Fake News Detection Dataset(20) are performed to evaluate our method. The results of our experiments demonstrate the efficacy of DEFENSE in detecting fake news, which significantly outperforms a few of the state-of-the-art baselines with an Accuracy of 0.9472, increase in Precision by 8% and Recall by 3%, and an F1-score of above 0.9.

Keywords

Accuracy; Classification; COVID-19; Fake news detection; Sentence Embedding; Social media.

Introduction

For most of the last century, our news was mainly consumed from a few sources, like the television, newspapers and radios that had people dedicated to fact check every bit of information before it is displayed to the public. The development and the increasing accessibility to the Internet has dramatically changed the way we get our information. With a drastic increase in the usage of social media applications and increase in the scope of its activities, there has been a shift in the consumption of news from traditional news outlets to online social media platforms. The simple access and swift dissemination of information through social media is useful for people to consume news quickly and take the necessary and appropriate action required. However, as we rely more on social media platforms for our news, the amount of misinformation spread can rise due to lack of a proper information verification system. Quantifying this situation, the study by Vosoughi et al.(28) found that true content took six times longer than fake content to be seen by 1500 people. With the increasing evolution of technology and the transmission of fake content through social media, it is crucial to develop an efficient computation framework to distinguish between the real and fake posts on social media.

In addition to that, the COVID-19 pandemic took the entire world by storm and forced everyone around the globe to stay indoors. The time spent by people in front of social media platforms during the COVID-19 pandemic is drastically increasing owing to the quarantine restrictions, lockdown measures and a work from home environment becoming more common. The spread of the virus brought with it the fear, panic, speculations, and rumours. Due to being isolated from the rest of society, people all over the world depend on their online social media networks for new information. New information comes out for solutions e.g. medicines, vaccines, mask usage, or regarding the spread and dangers. Misinformation about this can lead to problems where erroneous or flawed medical information cause harm. A post with misinformation that gains enormous traction or shared by an influential person can be potentially harmful to people without the proper information. Hence, the objective of this work is the detection of fake posts based on COVID-19 on social media.

The problem of fake news detection on social media has been studied using various approaches for the past two decades. Zhou et al.(32) found that methods to detect fake news can be broadly classified into four main types: (a) knowledge-based methods, that deal with the authenticity of the content, (b) propagation methods, that deal with how posts spreads, (c) style-based methods, that deal with the style of the writing and (d) source-based methods, that deal with the credibility of sources. The work by Castillo et al.(4) in 2011 is one of the initial attempts at assessing the credibility (fake news detection) of a social media post

(Twitter). A work by Lukasik et al.(16) uses Bag of Words (BOW) feature representation for classifying the posts and identifies certain patterns in its content. Later, works by Perez-Rosas et al.(21) using the linguistic features of the post and Wang et al.(30) using neural networks were proposed. Lately, transformer based language models like BERT have also beed used in fake news detection algorithms in some of the work by H Jwa et al.(9). However, all these methods are limited in their efficiency as they do not fully consider the contextual meaning of the text when it converts the text into its numerical representations.

Some attempts have been made by researchers in the past two years in detecting the fake COVID-19 related social media posts. Paka et al.(18) used user and tweet metadata to detect the fake news through a semi-supervised attention neural model. Though the accuracy of the model seems to be high, the metadata requirements seems to be complex. The extension of this work by Bansal et al.(3) named ENDEMIC results in the factuality of the external knowledge being doubtful. Kar et al.(11), Das et al.(5) and Wani et al.(31) are some of the other recent works towards this direction. Some of the existing techniques suffer from the need to have a lot of external knowledge for classification. In some cases, they require a lot of supporting metadata regarding the the propagation of the post or the user data to detect the fake news. In this work, we attempt to efficiently solve the problem of detection of fake COVID-19 related posts in social media with the feature engineering of precise and relevant feature sets to achieve high accuracy and less overfitting. The problem of detecting whether a social media post contains fake or real content is formulated as a binary classification problem. Classification(7) is a supervised machine learning techniques that categorises an input set of data points into class labels. Binary in binary classification implies that the data points can be classified into two class labels.

Problem Definition Given a set of social media posts aggregated from different social media platforms (Twitter, Facebook, Instagram, etc) related to the COVID-19 pandemic $S = \{P_1, P_2, P_3, ..., P_n\}$ $(1 \leq i \leq n)$, where P_i is the i^{th} post, the aim of our work is to implement a binary classification model using supervised learning to find out whether a post is real or fake. Each post is classified as real or fake and assigned a label l_i, where $l = \{0\ (real), 1\ (fake)\}$ and l_i is the label for the i^{th} post.

Our work tries to handle some of the bottlenecks in the existing methods through a feature engineering technique to concretely classify posts based on sentimental and textual features. Moreover, our model learns and extracts the contextual meaning from the post in numerical form to detect whether they are fake or real. An efficient way of embedding is performed using appropriate transformer based machine learning models and the embeddings extracted are processed further using suitable dimensionality reduction techniques. We experiment our architecture with various classification models and evaluated the performance of the proposed approach on Contraint@AAAI 2021 Covid-19 Fake News Detection Dataset(20), a dataset containing 10,700 posts related to COVID-19 from different social media applications. An exhaustive experimentation analysis with various parameters demonstrate the better working of DEFENSE in terms of accuracy and F1-score, in comparison to the existing methods. The main contributions of our work are highlighted and summarised below:

- A novel method named as DEFENSE (**De**tection of COVID-19 Fake News on Social Media Posts using **F**eature **En**gineering and **S**entence **E**mbedding) to detect the credibility of COVID-19 related posts on social media is proposed.
- We employ a precise and concrete feature engineering model to extract the textual and sentimental features of a post in the classification process.
- An efficient way to extract the contextual meaning of the post is performed using various sentence embedding methods and further processed by reduced the dimensions of the embedding using appropriate dimensionality reduction technique.
- The effectiveness of DEFENSE is tested exhaustively on various factors such as (i) dataset size, (ii) ratio of test size and train size, (iii) the pre-trained model used to get the sentence embedding of the posts, (iv) the variance to be maintained after dimensionality reduction (v) the classification algorithm used.

Further, the organisation of the paper as follows. Section 2 reviews existing works related to the detection of fake COVID-19 posts in social media platforms. The problem statement is defined and the details of our proposed architecture is described in Section 3. Section 4 presents the experimentation details followed by the performance evaluation of our proposed model. Finally, Section 5 concluded the paper with a summary of the contributions and possible future directions.

Related Work

In this section, we explore the various approaches for the detection of fake news and discuss their limitations. Additionally, we also look at research papers that tackled fake news detection in the context of COVID-19 and a review of language models.

Detection of Fake News

Since the launch of social media applications, the spread of fake news online has been extensively researched upon. We classify the different approaches for detecting the fake posts based on the two main type of features used: features based on the content of the post and features based on the context of social media. Approaches to detect fake news use either one of these two type of features or a combination of both.

Content based Features: Content features are either linguistic features that are related to the style of writing or knowledge-based features that deal with the authenticity of the content. Lukasik et al.(16) identified patterns in the content of the posts using Bag of Words (BOW) feature representation for classification. Perez-Rosas et al.(21) extracted several sets of linguistic feature: ngrams, punctuation, psycholinguistic features

(psychological components in language), readability (text understandability) and syntax to build a model to detect fake news. Wang et al.(30) built a neural networks based model that detects fake news by extracting the visual and textual features from a post using a multi-modal feature extractor. However, these methods do not consider the context of the text and the position of a word when it converts the text into its numerical form for computation.

Social Media based Features: Social media based features are either user-based features of the source account of the post, propagation-based features that deal with how posts spreads or user engagement based features that quantify the engagement of posts between users. Liu et al.(13) built a model to detect fake news by focusing on the propagation based features by modelling the propagation path of each news story as a multivariate time series. The engagement of the posts is also incorporated in the model by each tuple in the time series representing the user characteristics of those who spread the news. Another work that focuses on the engagement of posts was a deep neural network in a meta-learning framework was proposed by Shu et al.(26) to exploit weak signals based on the user and content engagements. Helmstetter et al.(8) attempted to predict whether a tweet was true by predicting the trustworthiness of the source of the tweet using both social and content features. A method to detect fake tweets based on the sequence of retweets using a graph network and attention mechanism to extract explanations was proposed by Lu et al.(15). The techniques that take social features into consideration face a few limitations in their approach due to lack of obvious patterns. Classification based on users' credibility can be misleading because there can be fake posts spread by accounts that usually post true content. Furthermore, techniques that focus on the spread and propagation of the news to distinguish between fake and real news may not perform well as we have seen in real life that fake news spreads equally fast as real news, if not more.

There are some methods that use a combination of both the set of features together to better classify the posts as fake or real. Castillo et al.(4) uses machine learning classification algorithms to assess whether a set of tweets is credible by taking into consideration a combination of user-based, content-based and propagation-based features. A knowledge based approach was used by Atodiresei et al.(2) in computing the credibility of a tweet by comparing it to news sources that are trustworthy and computing the users' credibility. Ramezani et al.(22) developed a model to label news as fake or true as early as possible by utilising a novel loss function on recurrent neural networks that requires user information, message context and time of posting. Kang et al.(10) built a model that detects fake news based on the relation in time, source and content between multiple news.

COVID-19 Fake News Detection

In the context of the COVID-19 pandemic, some attempts have been made to detect fake COVID-19 related posts on social media. Paka et al.(18) developed a cross-attention attention model that works in a semi-supervised manner that obtains an accuracy of 0.9540. This work was extended by Bansal et al.(3) for the purpose of early detection of fake news by building a model named ENDEMIC that uses endogenous and exogenous signals to attain an accuracy of 0.9370. Both these methods are highly computationally intensive and require a huge amount of metadata like the features regarding the user that posted the news and knowledge from external websites in addition to the post as its inputs for classification. A BERT based model to detect fake news about COVID-19 from Twitter was proposed by Kar et al.(11) obtaining an F1-score of 0.8947. This method also requires features regarding the users account and additionally, it has a poor F1-score below 0.9, despite testing on a comparatively smaller dataset. Das et al.(5) achieve a high accuracy of 0.9892 using an ensemble model consisting of a novel heuristic algorithm, pretrained models and a statistical feature fusion network to detect fake posts. Wani et al.(31) use two main approaches to classify the posts as real or fake, namely the approach of deep learning and the approach of transformer models, reporting an accuracy of 0.9841. However, since the transformer based models were pretrained on a COVID-19 corpus before testing it on a COVID-19 fake news dataset, there is a possibility of such a high accuracy value occurring due to overfitting by their model.

Language Models

Transformer(27) based models are the current state-of-the-art language models that have been used effectively in a wide array of text-based classification applications. A huge amount of data is used to train these models that perform excellently in comparison with the previous language processing approaches that use techniques like Gated Recurrent Units (GRU), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM). However in the context of sentences, a massive computational overhead makes it unsuitable for computing the semantic similarity. Therefore, Sentence BERT was proposed by Reimers et al.(24) to derive sentence embeddings that captured the contextual meanings of sentences and were semantically meaningful.

Although most of the existing research works are focused on detecting the fake posts, they involve the use of features more than just the post such as the user features, external knowledge, propagation patterns and user engagements. In this research, we focus on classifying the posts as fake or real, specifically based the post's content. Solving the problem by features engineering precise and relevant features, we ensure that our model effectively detects the fake posts even with very little supporting metadata. Therefore, in this work, the patterns and behaviours of COVID-19 related information in posts is studied primarily using the features engineered and the contextual embeddings learnt from the post using Transformer based language models. These patterns alone sufficiently help in detecting the fake posts circulating online and can be used to prevent the spread of fake content online.

DEFENSE: A Fake COVID-19 Social Media Posts Detection Model

In this section, we present the details of the proposed model for fake COVID-19 news detection in social media, named as

DEFENSE. Each module is described and explained using a sample post as an example.

Detailed Description of DEFENSE

Given a social media post on COVID-19 as data, the goal of DEFENSE is to detect whether the content of the post is real or fake. The architecture of our model consists of five main modules and is shown in Figure 1.

1. Pre-processing of the Post
2. Feature Engineering of Textual and Sentimental Features
3. Sentence Embedding of the Post
4. Dimensionality Reduction of the Sentence Embeddings using Principal Component Analysis (PCA)
5. Classification of the Post

Pre-processing of the Post The social media posts might be unstructured and contain some objects that are part of the social media vocabulary, as mentioned below.

- *Hashtags* - A hash sign (#) followed by a word or group of words. It is used in social media applications to connect the content of the post to a particular topic, occasion or subject.
- *URLs* - A Uniform Resource Locator is an address used to locate specific web page.
- *Mentions* - It is how a specific social media account, either individual, brand or community, can be referenced in the post.
- *Retweet (RT)* - It is used in Twitter to indicate that the post was originally posted by someone else.
- *Emojis and Smileys* - Small images used to denote facial expressions, animals, food items, common objects, vehicles, places and weather etc.

To obtain the numeric representation of the post, these objects are not required and must be removed. 'Tweet-preprocessor'(17), which is a library in Python, pre-processes the text and makes it proper for further analysis and processing.

Feature Engineering of Textual and Sentimental Features There are two main groups of features that are obtained from the post - sentimental features and textual features.

1. **Sentimental features** - They are the features that help us in understanding the sentiment of the post and give us an insight into the post. The text of the post obtained after pre-processing is given as an input to the textual data processing python library - TextBlob to obtain the sentiment polarity and sentiment subjectivity of the post, which is defined below.
   (a) **Sentiment Polarity** - Polarity is a numeric value between [-1, 1] that quantifies the emotions (positive or negative) expressed in a sentence. -1 defines a negative sentiment, 0 denotes a neutral sentiment and 1 defines a positive sentiment. For example, let us consider a sample post *"Major League Baseball is Now Considering Tearing Down Coronavirus-Infested Marlins Park https://t.co/de19ZssN07 #coronavirus #baseball"*. The sentiment polarity obtained for this post is -0.44. This denotes the sentiment expressed is tending more towards a negative emotion.
   (b) **Sentiment Subjectivity** - Subjectivity is a numeric value between [0, 1] that quantifies whether the text contains factual information or personal opinion. A sentiment subjectivity value closer to 0 implies that the post contains information and is highly objective. A sentiment subjectivity value closer to 1 implies that the post is highly subjective and contains an opinion. A value of 0.5 means that the post is neutral in its subjectivity. For the sample post considered previously, the sentiment subjectivity attained is 0.39. This denotes that the subjectivity of the post is between neutral to mildly objective.
2. **Textual Features** - The text of the post obtained after pre-processing is fed to the python library - NLTK (Natural Language Toolkit) to obtain the textual features as mentioned below. The textual features obtained for the post *"Major League Baseball is Now Considering Tearing Down Coronavirus-Infested Marlins Park https://t.co/de19ZssN07 #coronavirus #baseball"* are also mentioned below alongside the features.
   (a) Number of Characters - 134
   (b) Number of Words - 11
   (c) Number of Sentences - 1
   (d) Retweet: Denotes if the post is a retweeted post or an original post - False
   (e) Number of Hashtags - 2
   (f) Number of User Mentions - 0
   (g) Number of URLs - 1
   (h) Number of Upper Case Characters - 13
   (i) Number of Upper Case Words - 11
   (j) Number of Punctuation Marks - 8
   (k) Number of Unique Words - 11
   (l) Number of Stop Words - 1
   (m) Ratio of Number of Unique Words/Number of Words - 1.00

Sentence Embedding of the Post

The text of the post obtained after pre-processing needs to be represented in a numeric form for further processing. Embedding models are used to obtain the contextual

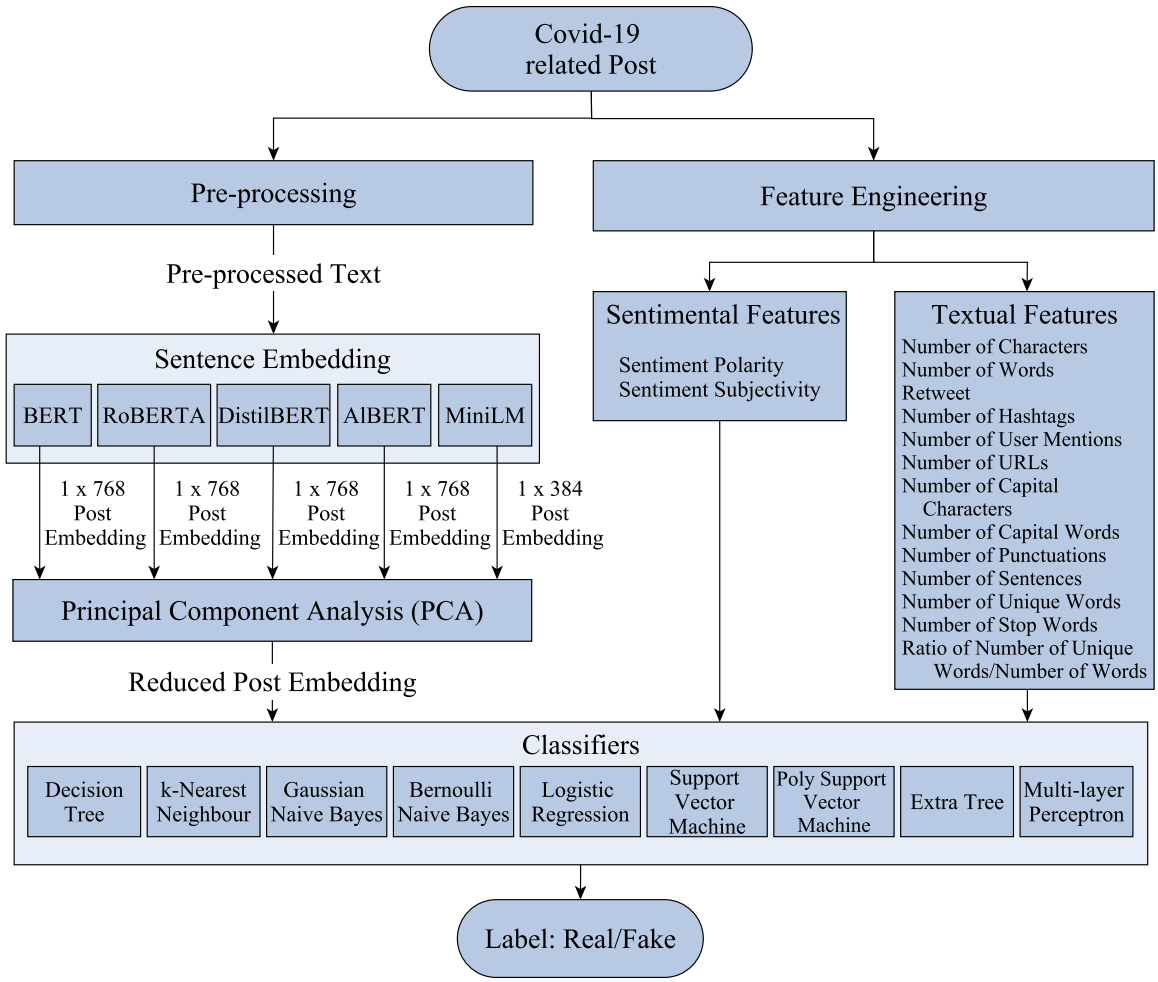


Figure 1. Architecture of DEFENSE: Fake COVID-19 Social Media Posts Detection Model

meaning of the post. However, using word embedding techniques on each word of the post and combining all the individual word embeddings to get the embedding of the post would limit the information extracted from the post as a whole. Hence, Sentence Transformers(24) are used in an attempt to obtain the sentence embedding of the post. Sentence Transformers map entire sentences to semantically meaningful sentence embedding vectors that capture the similarity or relatedness between sentences. Sentence Transformers have a few pre-trained models to obtain the sentence embedding and the pre-trained models used by our model are shown in Table 1, along with the dimensions of embedding obtained.

Table 1. Pre-trained Models used for Sentence Embedding

Model(23)	**Dimension of embedding**
BERT	768
RoBERTa	768
DistilBERTa	768
AlBERT	768
MiniLM	384

Bert-base-nli-mean-tokens(24) is a pre-trained model used by sentence transformers that converts sentences into a 768 dimensional dense vector. A pooling layer in transformers model enable us to create a fixed-size representation for input sentences of varying lengths. The pooling method used in this model is mean pooling. This sentence transformers model is trained on Natural Language Inference (NLI) dataset with a Semantic Textual Similarity (STS) benchmark score of 77.12. The base model for this sentence transformer is bert-base-uncased(6) which is trained on English Wikipedia and BookCorpus (a repository of 11,038 unpublished books). The base model is trained with the masked language modelling (MLM) and next sentence prediction (NSP) objectives.

Nli-roberta-base(24) is a pre-trained model used by sentence transformers that converts sentences into a 768 dimensional dense vector. The pooling method used in this model is mean pooling. This sentence transformers model is trained on NLI dataset with a STS benchmark score of 77.49. The base model for this sentence transformer is RoBERTa-base(14) which uses a different pretraining scheme to BERT by eliminating the NSP objective and trained with higher learning rates and larger mini-batches. RoBERTa is pre-trained on a larger corpus which includes English Wikipedia, OpenWebText (an open-source recreation of the WebText dataset), CC-News (a repository of 63 millions English news

articles from September 2016 to February 2019), Stories from Common Crawl and BookCorpus (a repository of 11,038 unpublished books).

Nli-distilbert-base(24) is a pre-trained model used by sentence transformers that converts sentences into a 768 dimensional dense vector. The pooling method used in this model is mean pooling. This sentence transformers model is trained on NLI dataset with a STS benchmark score of 78.69. The base model for this sentence transformer is distilBERT-base-uncased(25), which is a distilled version of bert-base-uncased. It uses a compression technique called knowledge distillation where the behaviour of a larger model is reproduced by a small model. It uses 40% less parameters than bert-base-uncased while preserving over 97% of BERT's performances and performs 60% faster. DistilBERT is pretrained on the same data as BERT.

Paraphrase-albert-small-v2(24) is a pre-trained model used by sentence transformers that converts sentences into a 768 dimensional dense vector. The pooling method used in this model is mean pooling. This sentence transformers model has a STS benchmark score of 83.40(23) and is trained on AllNLI, sentence-compression, SimpleWiki, S2ORC_citation_pairs, msmarco-triplets, quora-duplicates, flickr30k_captions, yahoo_answers_title_question, altlex, coco_captions, stackexchange_duplicate_questions and wiki-atomic-edits. The base model for this sentence transformer is albert-base-v2(12) which is pretrained on the same data as BERT using a masked language modelling (MLM) objective.

Paraphrase-MiniLM-L6-v2(24) is a pre-trained model used by sentence transformers that converts sentences into a 384 dimensional dense vector. The pooling method used in this model is mean pooling. This sentence transformers model has a STS benchmark score of 84.12(23) and is trained on AllNLI, sentence-compression, SimpleWiki, S2ORC_citation_pairs, msmarco-triplets, quora-duplicates, flickr30k_captions, yahoo_answers_title_question, altlex, coco_captions, stackexchange_duplicate_questions and wiki-atomic-edits. The base model for this sentence transformer is MiniLM-L6-H384-uncased, which is a pre-trained model proposed by Wang et al.(29). The smaller MiniLM (6-layer) obtains 5.3x speedup and produces very competitive results over BERT-Base. These sentence transformer models take the pre-processed text and produce the sentence embedding vectors for each post. The dimensions of the embedding vectors depend on the sentence-transformers model used for sentence embedding.

Dimensionality Reduction of the Sentence Embeddings using PCA

The contextual embedding vector of the social media post obtained is a d-dimensional vector, where $d = \{768, 384\}$ depending on the model chosen for sentence embedding. To reduce the time and storage space, the embedding vector is reduced into a smaller vector of n dimensions, where $n < d$. Dimensionality reduction will convert the data into a lower-dimensional space from a higher-dimensional space, retaining the meaningfulness of the original data to a certain extent. The method of Principal Component Analysis (PCA) processes the higher dimensional embedding of the post and produces a lower dimensional embedding of the post, while still retaining as much of the variance in the data as possible. The size of the lower dimensional embedding depends on the amount of variance v that is set to be retained from the original dimensional embedding after dimensionality reduction. n would be the minimum number of dimensions required to maintain variance v from the original d-dimensional vector space.

Classification Model

The sentimental features, the textual features and the embedding of the post after PCA are fed into the classifier model to detect whether the post is real or fake. The various supervised algorithms used for classification are mentioned below. (7) (1)

- Decision Tree takes a dataset with features and classes to produce a sequence of rules for classification. The sequence of rules are in form of a tree with two kinds of nodes (leaf and decision). Decision nodes denote the attributes of a dataset and have multiple divisions depending on the possible values of the particular attribute and each leaf node represents a class. Choosing a feature as the decision node is based on metrics like Gini index.

- k-Nearest Neighbours Classifier is a type of lazy learning as it does not build a model, but simply stores each record of the training set as a point. Classification is done for each new record by choosing the label occurring the most among the k nearest neighbours of the point.

- Naive Bayes Classifier is a classifier built on Bayes theorem that determines conditional probability (the chance of an event happening based on the previous occurrence of the event). It is a probabilistic classifier that works with the assumption of independence between every pair of features. Two types of naive bayes model is used in our model - Gaussian model, which supposes a normal Gaussian distribution exhibited by the features and Bernoulli model, which assumes that the features are independent Booleans variables.

- Logistic Regression uses a logistic function to model the probabilities of the possible outcomes of the class label. Rather than producing the exact value as 0 and 1, it computes the probabilistic values that would be between 0 and 1 which are mapped to appropriate class labels.

- Support Vector Machine Classifier creates the best possible decision boundary called a hyperplane that breaks the n-dimensional space into class labels with the widest gap possible. We test SVM on two different kernels - linear and polynomial. A linear kernel works best when the data points are linearly separable. When they are not clearly linearly separable, a polynomial kernel applies curved lines to separate the data points.

- Extra Tree Classifier is an ensemble learning technique which accumulates numerous unique decision trees for

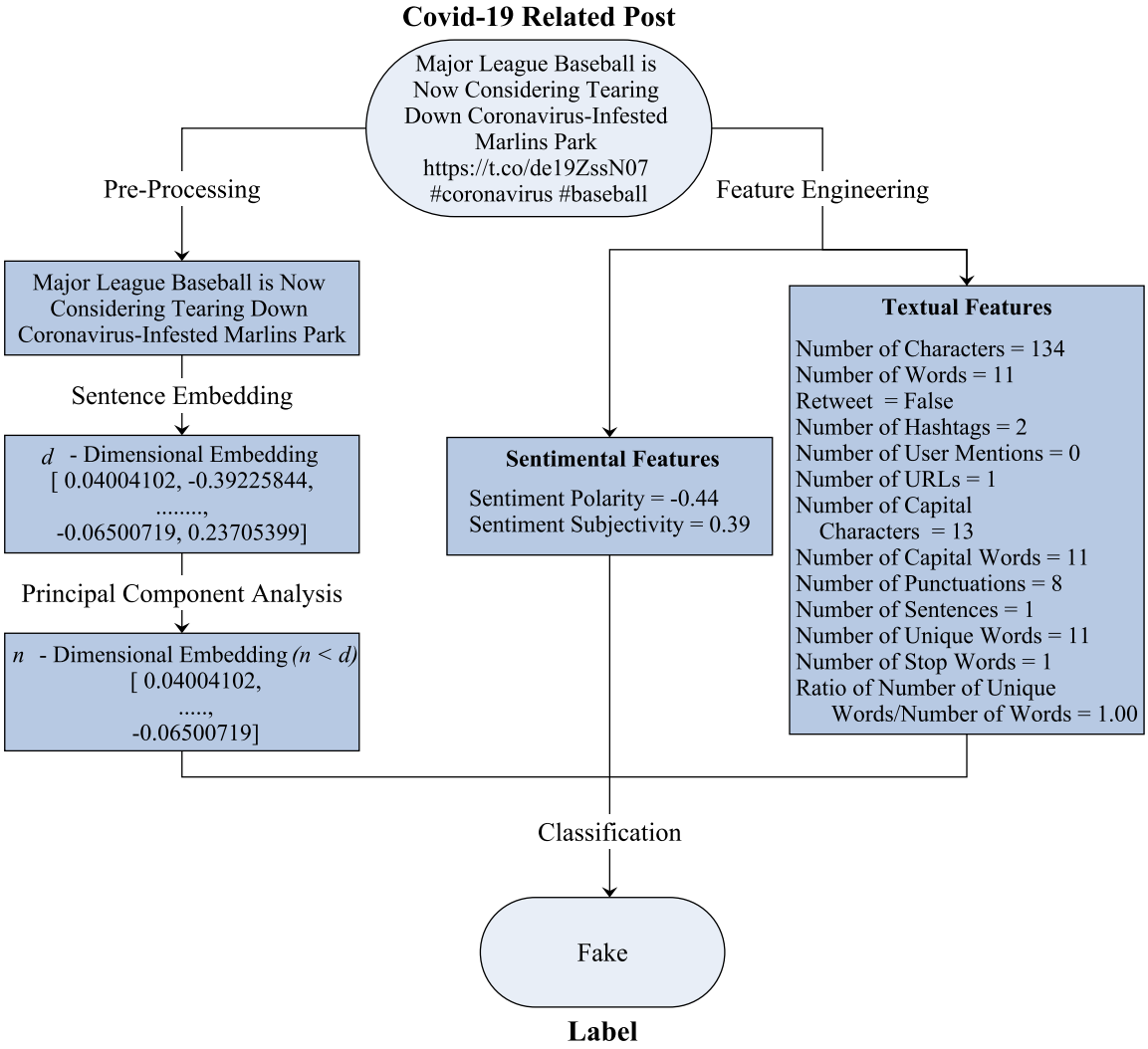


Figure 2. Work Flow of DEFENSE for a Sample Post

classification. At each decision node of the tree, the best feature among a random group of k features out of the feature set is used. Since every decision node will have a different random sample of features, multiple unique decision trees can be constructed.

- Multi-layer Perceptron Classifier is a neural network that performs the task of classification. It trains on a given set of features and class labels and learns a non-linear function approximator for classification. The possible activation functions for the hidden layer are: identity ($f(x) = x$), logistic ($f(x) = 1/(1 + exp(-x))$), tanh ($f(x) = tanh(x)$) and rectified linear unit function ($f(x) = max(0, x)$). After training, the model can predict labels for new samples.

Working of DEFENSE

The working of DEFENSE is demonstrated using a sample post *"Major League Baseball is Now Considering Tearing Down Coronavirus-Infested Marlins Park https://t.co/de19ZssN07 #coronavirus #baseball"* as it passes through the model and undergoes classification as shown in Figure 2. The post undergoes preprocessing to obtain the text of the post "Major League Baseball is Now Considering Tearing Down Coronavirus-Infested Marlins Park". This text is passed to the sentence transformer module to obtain its sentence embedding in a d-dimensional vector form. (768 or 384 dimensional vector, depending on the sentence embedding model chosen). This d-dimensional vector undergoes dimensionality reduction through PCA, which produces a n-dimensional vector; where $n < d$ and n depends on the value of variance v that is set to be retained from the original dimensional embedding after dimensionality reduction. Simultaneously, the post undergoes feature engineering to obtain the sentimental and textual features of the post. These sentimental features and textual features, along with the n-dimensional sentence embedding of the post, is passed to the classifier module. According to the classification algorithm chosen, it builds the model and classifies the post as either real or fake.

Experimental Results and Discussions

In this section, we evaluate the effectiveness of the proposed DEFENSE model by present the details of the experiments,

the various parameters on which the model was tested upon and the results obtained.

Description of Dataset

The proposed model has been implemented using Python3 and tested on Contraint@AAAI Covid-19 Fake News Detection dataset(20) which was provided on the website(19) of the competition by the organisers of the workshop. The dataset has 10,700 social media posts related to COVID-19 that have been aggregated from a variety of social media applications such as Facebook, Instagram and Twitter. The posts in the dataset have been given class labels appropriately as real or fake. The dataset is balanced with respect to both the classes with 5600 (52.3%) real posts and 5100 (47.6%) fake posts.

Experimentation on Benchmark Dataset

The various factors (parameters) considered during experimentation that could have significant impact in the performance study are (i) dataset size, (ii) ratio of test size and train size, (iii) the pre-trained model used to get the sentence embedding of the posts, (iv) the variance to be maintained in the dataset on applying principal component analysis on the sentence embeddings of the post and (v) the classification algorithm used. The model was tested on datasets of size 2500, 5000, 7500 and 10700 maintaining the class balance as shown below in Table 2.

Table 2. Tested Dataset Sizes

Dataset Size (# of Posts)	# of Real Posts	# of Fake Posts
2500	1297	1203
5000	2629	2371
7500	3907	3593
10700	5600	5100

Three different train-test ratios of 0.7-0.3, 0.75-0.25 and 0.8-0.2 were tested. The variance in the sentence embedding to be maintained were varied from 0.8 to 0.98 in increments of 0.02. Five different models were tried out for sentence embedding - BERT, RoBERTa, DistilBERT, AlBERT and MiniLM along with a number of different classification models - Decision Tree, Extra Tree Classifier, Gaussian Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, k-Nearest Neighbours Classifier (KNN), Support Vector Machine Classifier (SVM), Poly Support Vector Machine Classifier (Poly-SVM) and Multi-layer Perceptron Classifier. In this model, posts that are real are assigned the label as 0 and the fake posts are assigned the label as 1. Every classification of a post can have one of the following four outcomes.

- True Positive (TP): A fake news is predicted as fake news.
- True Negative (TN): A true news is predicted as true news.
- False Negative (FN): A fake news is predicted as true news.
- False Positive (FP): A true news is predicted as fake news.

Evaluation Metrics

The metrics used for evaluation are:

1. Accuracy is the ratio of the number of accurately predicted classifications to the total number of classifications. It highlights how many posts have been correctly predicted in total.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (1)$$

2. Precision is the ratio of the number of accurately predicted positive classifications to the total number of positive classifications. It highlights how many posts are actually fake out of all the posts that have been predicted as fake.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3. Recall is the ratio of the number of accurately predicted positive classifications to the total number of positive data points. It highlights how many posts have been predicted as fake out of all the posts that are actually fake.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4. F1-score is is a metric which takes into consideration both precision and recall simultaneously. It is the harmonic mean of precision and recall.

$$Accuracy = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

Performance Analysis of DEFENSE

Effect of Dataset Size on Accuracy The accuracy of all the classifiers in detecting the social media post as real or fake is tested on dataset sizes of 2500, 5000, 7500 and 10700 with different embedding model and plotted as seen in Figure 3. As the size of the dataset is increased, the accuracy of all classifiers increases although the amount of increase isn't the same across all the classifiers. This observed trend is due to the fact that as the dataset size is increased, the data available for training of the classifier models is more. The highest accuracy is attained as 94.72% with dataset of size 10700.

A few exceptions to this trend are observed. In Figure 3a, Gaussian Naive Bayes classifier on AlBERT embedding model gives an accuracy of 0.7536 for dataset of size 2500, 0.7256 for dataset of size 5000, 0.7259 for dataset of size 7500 and 0.7226 for dataset of size 10700 which shows a steady decrease in accuracy as the size increases. In Figure 3b, Gaussian Naive Bayes classifier on BERT embedding model gives an accuracy of 0.7936 for dataset of size 2500, 0.8024 for dataset of size 5000, 0.7867 for dataset of size 7500 and 0.7634 for dataset of size 10700 showing an initial rise in accuracy followed by a decrease. This could be due to the assumption by Naive Bayes that all the features are independent, which could negatively impact the accuracy of the classifier as the number of data points were increased.

In Figure 3d, PolySVC on MiniLM embedding model gives an accuracy of 0.7040 for dataset of size 2500, 0.6613 for dataset of size 5000, 0.6991 for dataset of size 7500 and 0.6526 for dataset of size 10700 which shows a continuous decrease in accuracy as the size increases. The polynomial curved boundary used by the poly-SVC to separate the classes performs poorly in this binary classification setup, with the accuracy worsening as the dataset size is increased.

Effect of Dataset Size on F1-Score The F1-scores of all the classifiers in detecting the social media post as real or fake is tested on dataset sizes of 2500, 5000, 7500 and 10700 with different embedding model and plotted as seen in Figure 4. The trend observed is that as the size of the dataset is increased, the F1-score of all classifiers increase. This is owing to the fact that as the dataset size increases, there is more data available for training and the recall and precision values increases. Subsequently, the F1-score which depends on the precision and recall, also increases. The highest F1-score is attained as 94.72% with dataset of size 10700.

There are a few exceptions observed to this trend. As seen in Figure 4a, Bernoulli Naive Bayes Classifier on AlBERT embedding is observed to have a steady F1-score even as the dataset size increases. Conversely, there are a few instances where the opposite of the general trend is observed. As seen in Figure 4a, Gaussian Naive Bayes classifier on AlBERT embedding model gives an accuracy of 0.734 for dataset of size 2500, 0.6938 for dataset of size 5000, 0.6894 for dataset of size 7500 and 0.6673 for dataset of size 10700 which shows a continuous decrease in accuracy as the size increases. The reason behind the poor behaviour by the Bernoulli and Gaussian Naive Bayes Classifiers could be due to the assumption by Naive Bayes that all the features are independent, negatively impacting the F1-score as the number of data points considered were increased. Additionally, PolySVC on MiniLM embedding model gives an F1-score of 0.7007 for dataset of size 2500, 0.6554 for dataset of size 5000, 0.6897 for dataset of size 7500 and 0.6385 for dataset of size 10700 which shows a drop in F1-score as the size increases as seen in Figure 4d. This is due to the poor performance of the polynomial curved boundary used by the poly-SVC to separate the classes leading to the F1-score decreasing as the number of data points are increased.

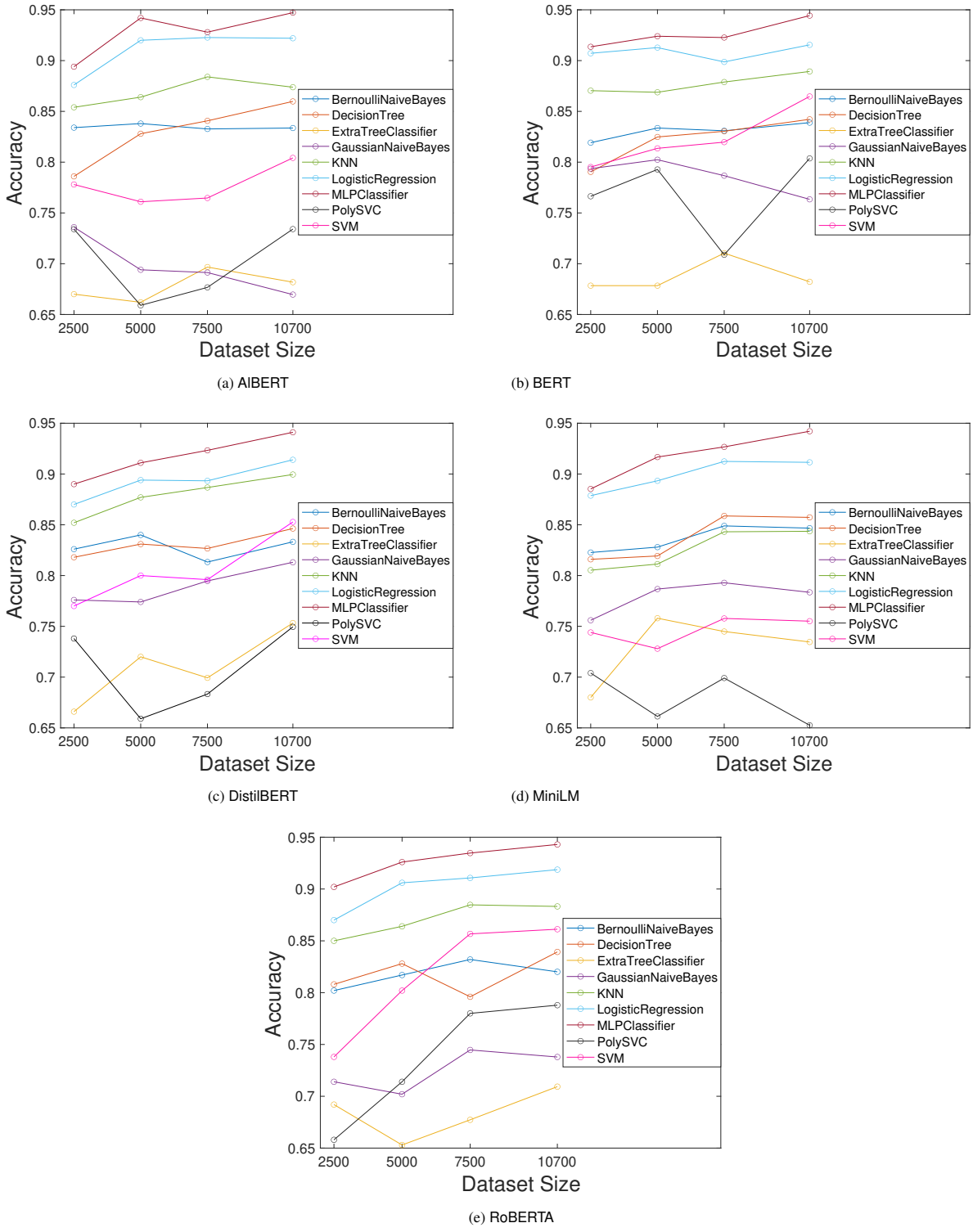
Effect of Datasplit on Accuracy The performance of all the classifiers in detecting the social media post as real or fake is tested on train-test ratios of 0.7-0.3, 0.75-0.25 and 0.8-0.2 with different embedding model and the accuracy values are plotted as seen in Figure 5. The general intuition is that accuracy will increase as the ratio of training data is increased. However, we observe that the best performing data split with reference to accuracy varies according to the embedding model used. As seen in Figure 5a, classifiers on AlBERT embeddings showed their best accuracy with a data split of 0.8-0.2, with the notable exceptions of Gaussian Naive Bayes (with highest accuracy of 0.6707 at 0.75-0.25), K-Nearest Neighbours (with highest accuracy of 0.8763 at 0.7-0.3) and Bernoulli Naive Bayes Classifier (with highest accuracy of 0.8355 at 0.7-0.3).

As seen in Figure 5b, classifiers on BERT embeddings showed their best accuracy with a data split of 0.75-0.25, with the notable exceptions of Gaussian Naive Bayes (with highest accuracy of 0.8184 at 0.7-0.3), K-Nearest Neighbours (with highest accuracy of 0.8916 at 0.8-0.2) and Extra Tree Classifier (with highest accuracy of 0.7393 at 0.7-0.3). Classifiers on DistilBERT embeddings showed their best accuracy with a data split of 0.8-0.2, with the sole exception of Poly SVC (with highest accuracy of 0.772 at 0.75-0.25) as seen in Figure 5c. Classifiers on RoBERTa embeddings evenly showed their best accuracy performances with five classifiers showing their best accuracy on 0.8-0.2 and the remaining four classifiers observed their best accuracy on 0.75-0.25 as seen in Figure 5e. Classifiers on MiniLM did not show any pattern in the different data splits in terms of accuracy values.

Effect of Datasplit on F1-Score The performance of all the classifiers in detecting the social media post as real or fake is tested on train-test ratios of 0.7-0.3, 0.75-0.25 and 0.8-0.2 with different embedding model and the F1-scores are plotted as seen in Figure 6. The general intuition is that F1-scores will increase as the ratio of training data is increased. We observe that the best performing data split with reference to F1-scores varies according to the embedding model used. Figure 6a shows classifiers on AlBERT embeddings with their best F1-score using a data split of 0.8-0.2, with the notable exceptions of Gaussian Naive Bayes (with highest F1-score of 0.6688 at 0.7-0.3), K-Nearest Neighbours (with highest F1-score of 0.8753 at 0.7-0.3) and Bernoulli Naive Bayes Classifier (with highest F1-score of 0.8354 at 0.7-0.3).

Classifiers on BERT embeddings showed their best F1-score with a data split of 0.75-0.25, with the notable exceptions of Gaussian Naive Bayes (with highest F1-score of 0.7959 at 0.7-0.3), K-Nearest Neighbours (with highest F1-score of 0.8913 at 0.8-0.2) and Extra Tree Classifier (with highest F1-score of 0.6881 at 0.7-0.3) as seen in Figure 6b. In Figure 6c, classifiers on DistilBERT embeddings show their best F1-score with a data split of 0.8-0.2, with the sole exception of Poly SVC (with highest F1-score of 0.7669 at 0.75-0.25). Classifiers on MiniLM embeddings showed their best F1-score with a data split of 0.7-0.3, with the notable exceptions of Gaussian Naive Bayes (with highest F1-score of 0.7855 at 0.8-0.2), SVM Classifier (with highest F1-score of 0.755 at 0.75-0.25) and Extra Tree Classifier (with highest F1-score of 0.7396 at 0.75-0.25) as seen in Figure 6d. Classifiers on RoBERTa embeddings evenly showed their best F1-scores with five classifiers showing their best accuracy on 0.8-0.2 and the remaining four classifiers observed their best F1-scores on 0.75-0.25 as seen in Figure 6e.

Effect of Variance on Accuracy The performance of all the classifiers in detecting the social media post as real or fake is tested for values of variance from 0.8 to 0.98 in increments of 0.02 with each embedding model and the accuracy values are plotted as seen in Figure 7. The intuition is that as the variance is increased, more will be the dimensions of the embedding available for training and hence the performance will be better. It is observed that on increasing the variance, the accuracy either increases consistently throughout or it steadily increases until a threshold variance, following which



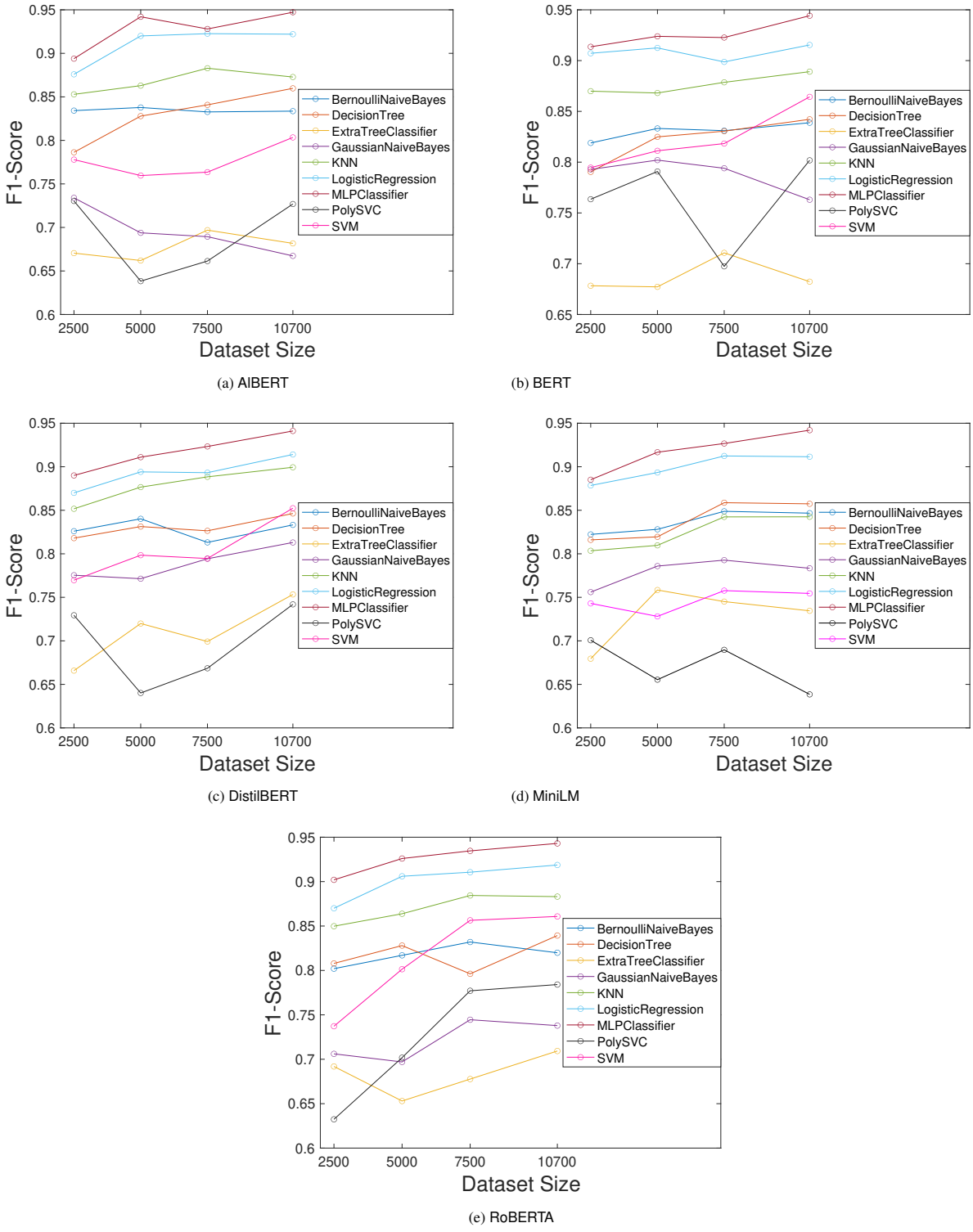
(a) AlBERT (b) BERT

(c) DistilBERT (d) MiniLM

(e) RoBERTA

Figure 3. Accuracy versus Dataset Size for Various Embedding Models

it decreases. Among the classifiers, the highest degree of variation in accuracy values is observed for the Extra Tree Classifier. Figure 7c shows that the variation in accuracy values by Extra Tree Classifier is highest in DistilBERT, with an accuracy of 0.8056 for variance of 0.84 and an accuracy of 0.6790 for variance of 0.98. This variation in accuracy values by the Extra Tree Classifier is due to the random sampling of attributes from the attribute-set at each test node for every decision tree. This leads to creation of different forests at each instance of testing. Conversely, Gaussian Naive Bayes Classifier on AlBERT, DistilBERT and MiniLM embedding show a behaviour opposite to the observed trend, with the accuracy value constantly decreasing as the variance was increased. This is because as the variance increases, the number of dimensions of embedding fed into the classifier also increases. Gaussian Naive Bayes performs poorly on large number of dimensions if all the dimensions are not guaranteed to be mutually independent.



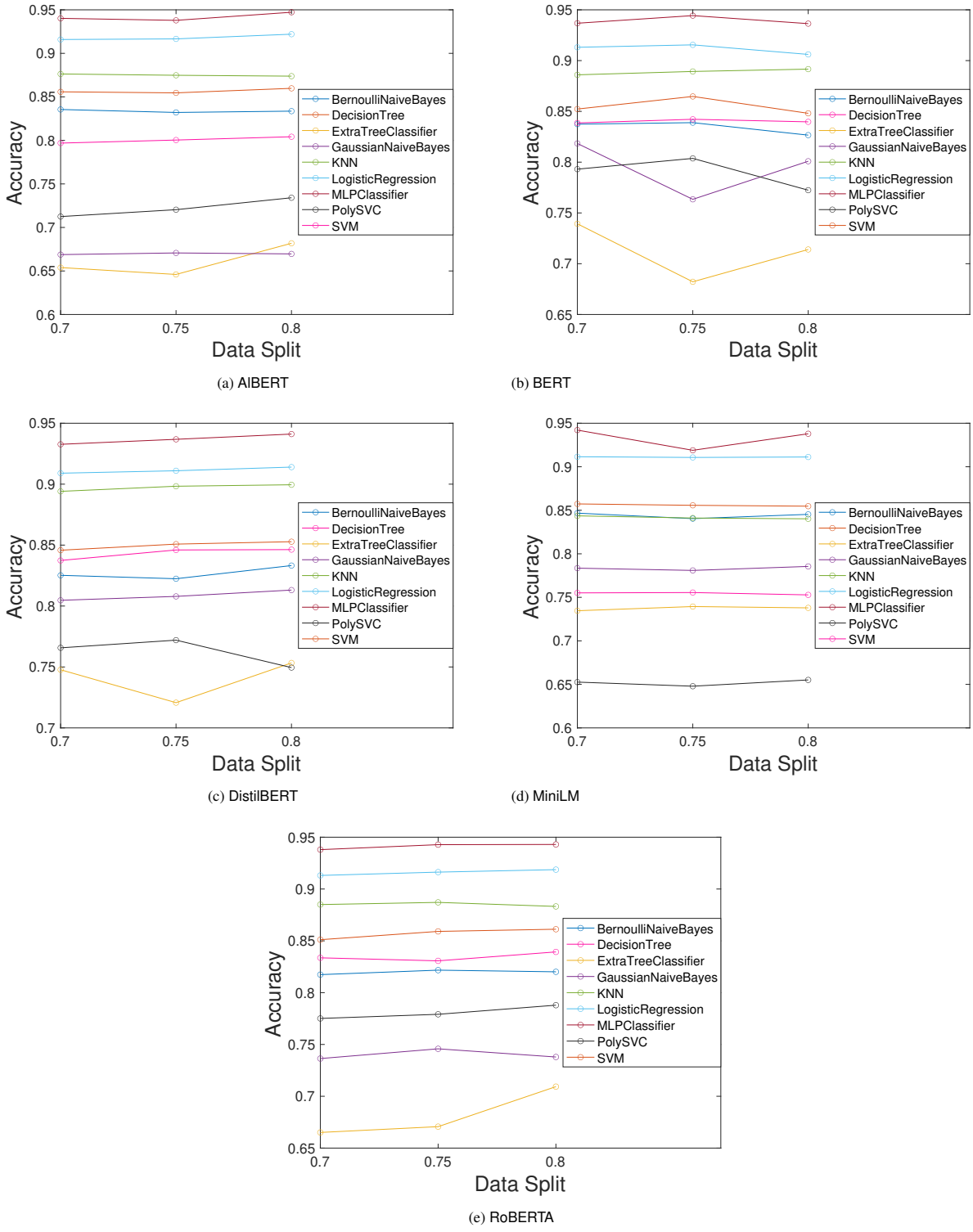
(a) AlBERT (b) BERT

(c) DistilBERT (d) MiniLM

(e) RoBERTA

Figure 4. F1-score versus Dataset Size for Various Embedding Models

Effect of Variance on F1-Score The performance of all the classifiers in detecting the social media post as real or fake is tested for values of variance from 0.8 to 0.98 in increments of 0.02 with each embedding model and the F1-scores are plotted as seen in Figure 8. The intuition is that as the variance is increased, more will be the dimensions of the embedding available for training and hence the performance will be better. It is observed that on increasing the variance, the F1-scores of the various classifiers either increases consistently throughout or it steadily increases until a threshold variance, following which the F1-score decreases. Similar to the effect of variance on accuracy, the highest degree of variation in F1-scores among the classifiers for different values of variance is observed for the Extra Tree Classifier across all the different embedding models. Among the embedding models, the variation in F1-scores by Extra Tree Classifier is highest in DistilBERT, with a F1-score of 0.8057 for variance of 0.84 and a F1-score of 0.679 for variance of 0.98 as seen in Figure 8c. Similar to its accuracy, the F1-scores of Gaussian Naive Bayes Classifier



(a) AlBERT (b) BERT

(c) DistilBERT (d) MiniLM

(e) RoBERTA

Figure 5. Accuracy versus Training-Test Data Split for Various Embedding Models

on AlBERT, DistilBERT and MiniLM embedding constantly decrease as the variance is increased.

Effect of Embedding models and Classifier models on Accuracy and F1-Score As observed from Figures 3-8, each classifier algorithm tested gives a different accuracy of detecting the post as real or fake as each classifier works on a different strategy of classification. Among the nine classifiers tested, multilayer perceptron classifier is observed to give the best results with accuracy values consistently above 0.9. The highest accuracy obtained is 0.9472 and highest F1-score obtained is 0.9472 using AlBERT as the embedding model for a 0.8-0.2 data split for training and testing. The reason behind the excellent performance is the adaptive learning capability and the presence of one or more hidden layers providing levels of abstraction during training to correctly assign the class labels. Next to multilayer perceptron, logistic regression classifier gives an impressive performance with accuracy values consistently above 0.9. Logistic regression attains a best accuracy value of 0.922

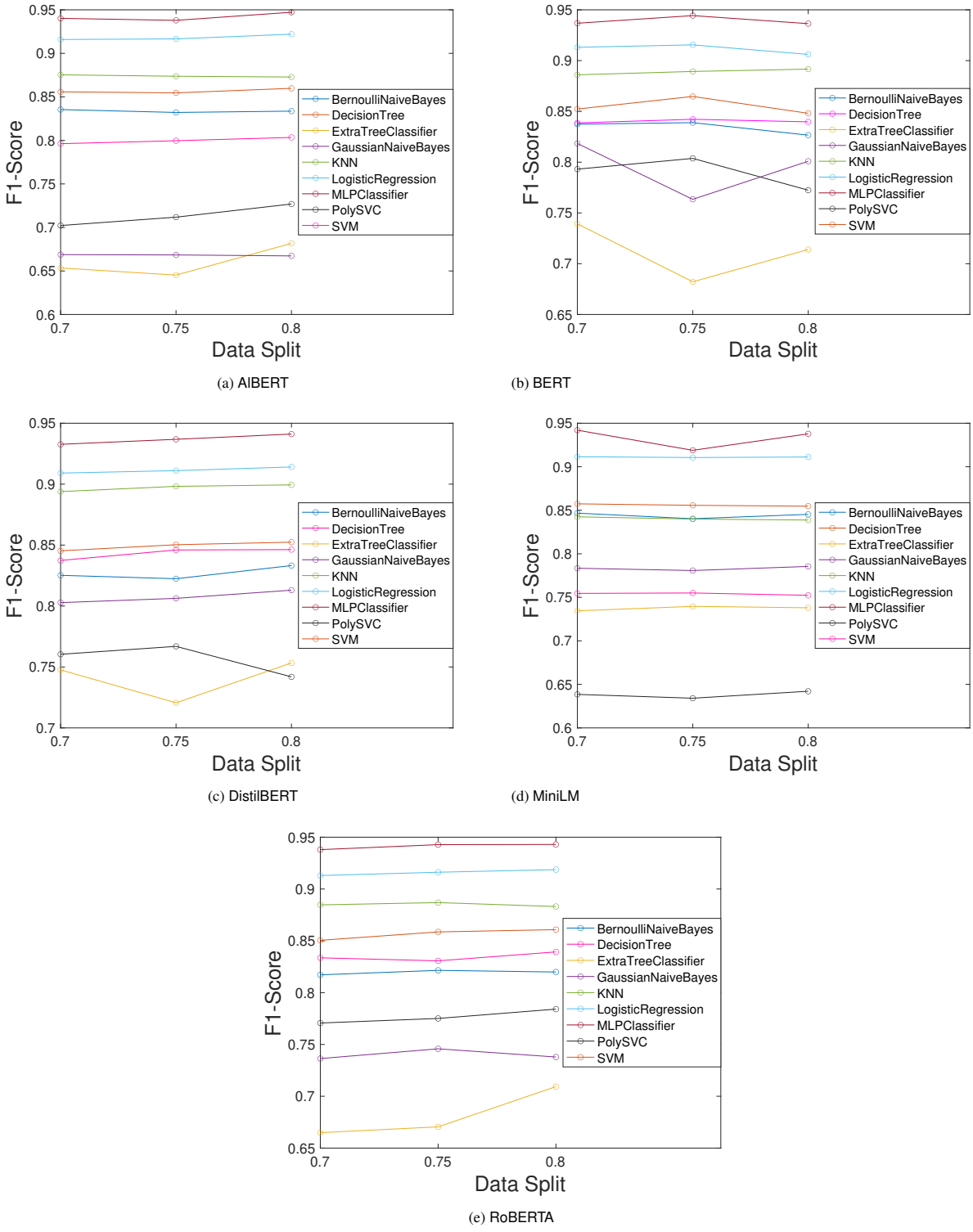
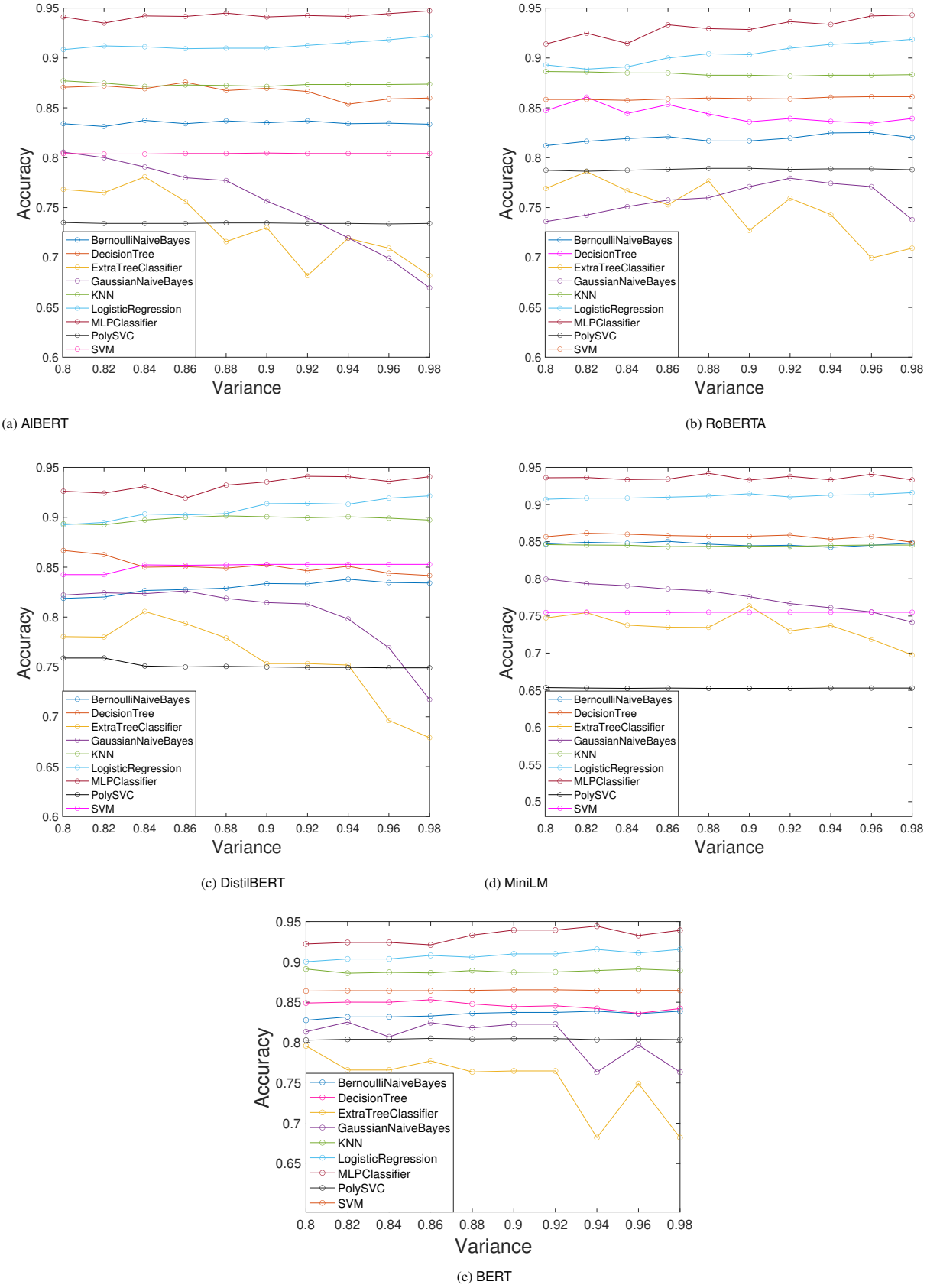


Figure 6. F1-score versus Training-Test Data Split for Various Embedding Models

and F1-score of 0.922 with AlBERT embedding on a 0.8-0.2 data split for training and testing. Logistic regression performs well due to its statistical approach that uses well-calibrated probabilities for classification. The accuracy and F1-scores for different dataset sizes, different training-testing data splits and different variance values for the multi layer perceptron classifier is shown in Figure 9 and for the logistic regression classifier is shown in Figure 10.

As seen from Figures 9-10, among the five pre-trained embedding models used, AlBERT consistently performs better than the others, with the highest accuracy of 0.9472 and highest F1-score of 0.9472. BERT embedding attains an highest accuracy of 0.9443 and highest F1-score of 0.9442. RoBERTa embedding attains the best accuracy of 0.943 and best F1-score of 0.943. DistilBERT embedding performs with the best accuracy of 0.9411 and F1-score of 0.9411. MiniLM embedding performs with the best accuracy of 0.9421 and F1-score of 0.942.



(a) AlBERT

(b) RoBERTA

(c) DistilBERT

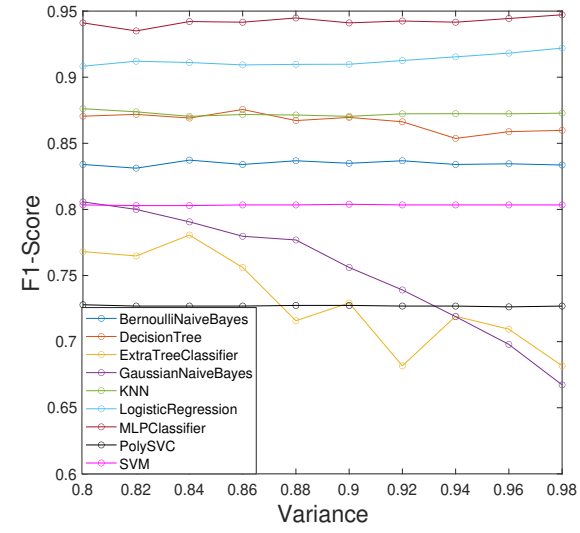
(d) MiniLM

(e) BERT

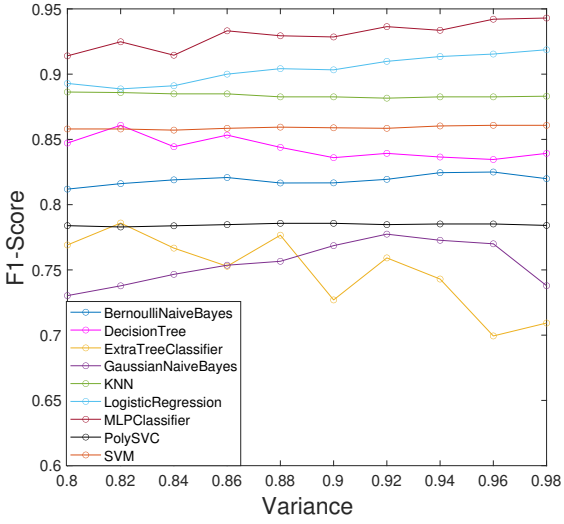
Figure 7. Accuracy versus Variance for Various Embedding Models

Comparative Analysis of DEFENSE with Existing Methods

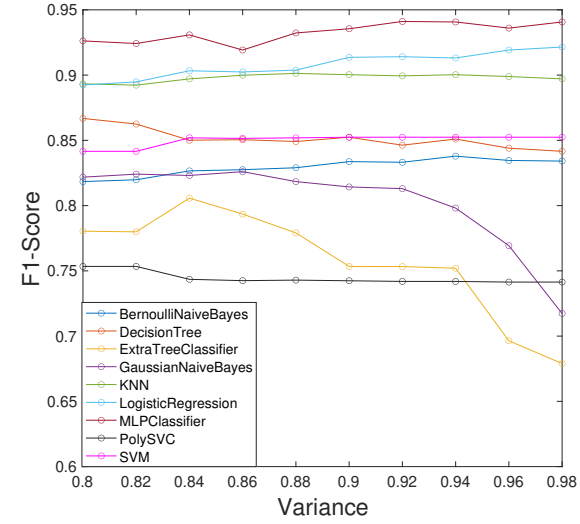
The performance of DEFENSE in predicting COVID-19 social media posts as fake or real is evaluated based on evaluation metrics like accuracy, F1-score, recall and precision. Table 3 shows a comparative analysis between our proposed approach and the existing approaches on fake COVID-19 post detection. Accuracies and F1-scores



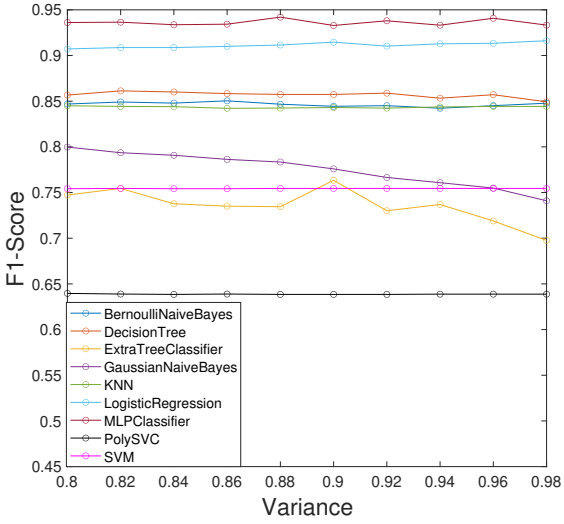
(a) AlBERT



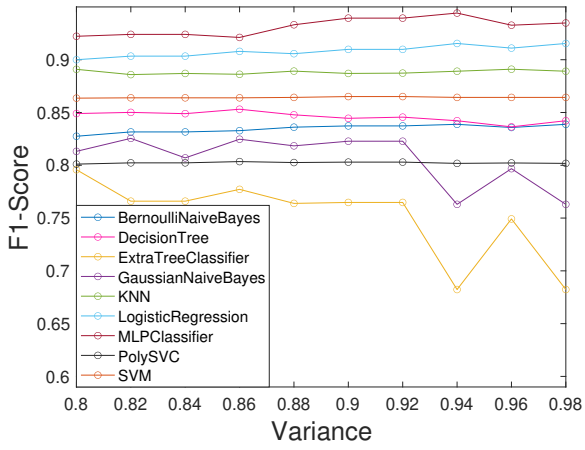
(b) RoBERTa



(c) DistilBERT



(d) MiniLM



(e) BERT

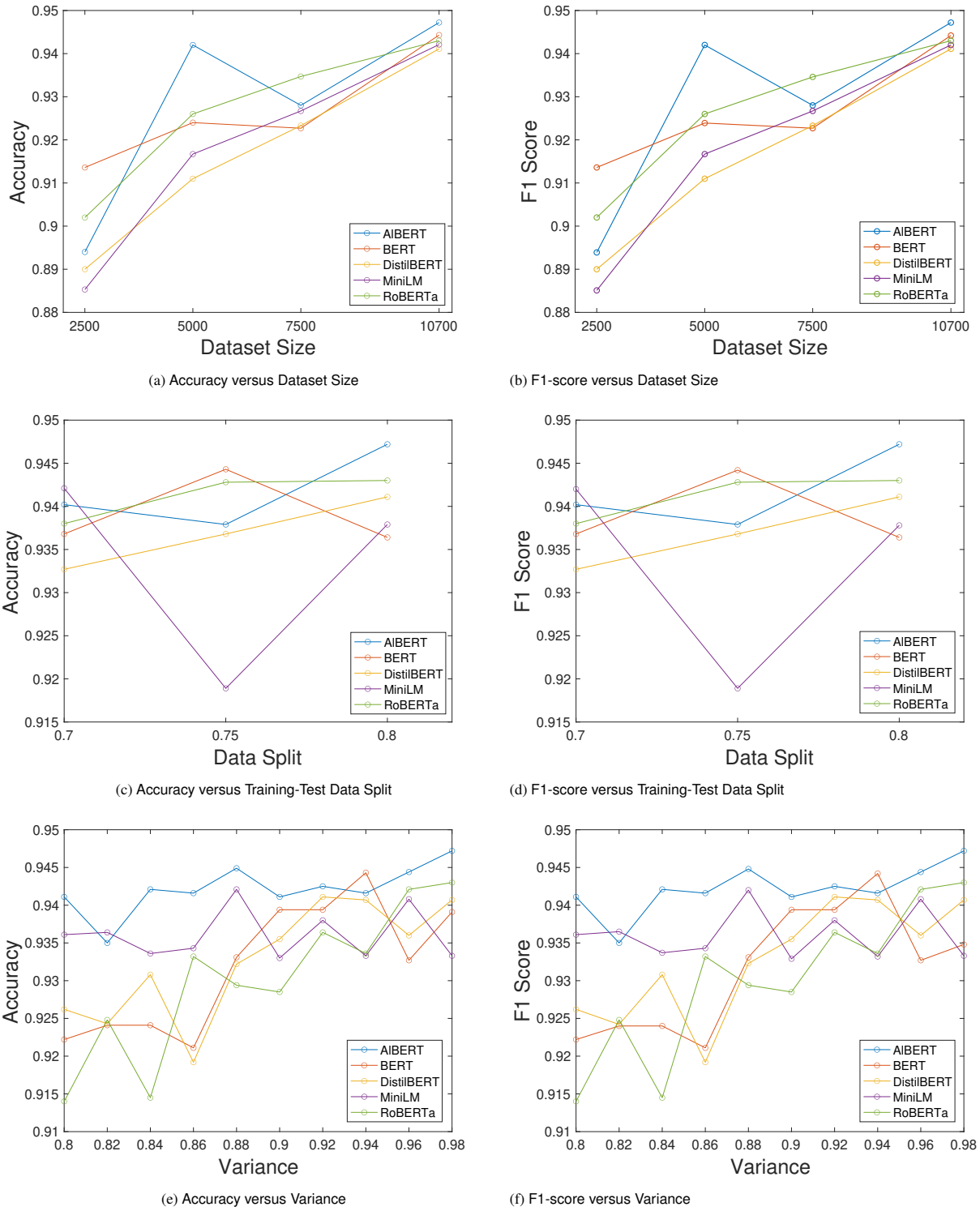
Figure 8. F1-score versus Variance for Various Embedding Models

of various proposed models and existing approaches are compared and plotted in Figure 11.

Among the various approaches of DEFENSE, using AlBERT embeddings along with Multilayer Perceptron Classifier provides the highest accuracy of 0.9472, precision

of 0.9473, recall of 0.9472 and F1-score of 0.9472. This is a better result in comparison with the 0.9370 accuracy and 0.9320 F1-score obtained by Bansal et al.(3), despite them using a co-attention network approach that combines exogenous and endogenous signals to classify posts. Our approach also performs better than a BERT based model proposed by Kar et al.(11), which was tested on a relatively smaller dataset to attain a F1-score of 0.8947 (accuracy value was not mentioned). The dimensionality reduction of contextual embeddings using PCA, in combination with the semantic and textual features, enables our approach to perform well and attain a higher accuracy and F1-score.

Wani et al.(31) attempts two approaches, firstly using deep learning techniques and secondly using Transformer based models that are pretrained with a COVID-19 tweets corpus to detect fake posts with the highest accuracy of 0.9841. This is in contrast to our approach, where no pre-training of the language models in the context of covid has been attempted. Das et al.(5) achieve a high accuracy of 0.9892 using an ensemble model consisting of a novel heuristic algorithm,



(a) Accuracy versus Dataset Size

(b) F1-score versus Dataset Size

(c) Accuracy versus Training-Test Data Split

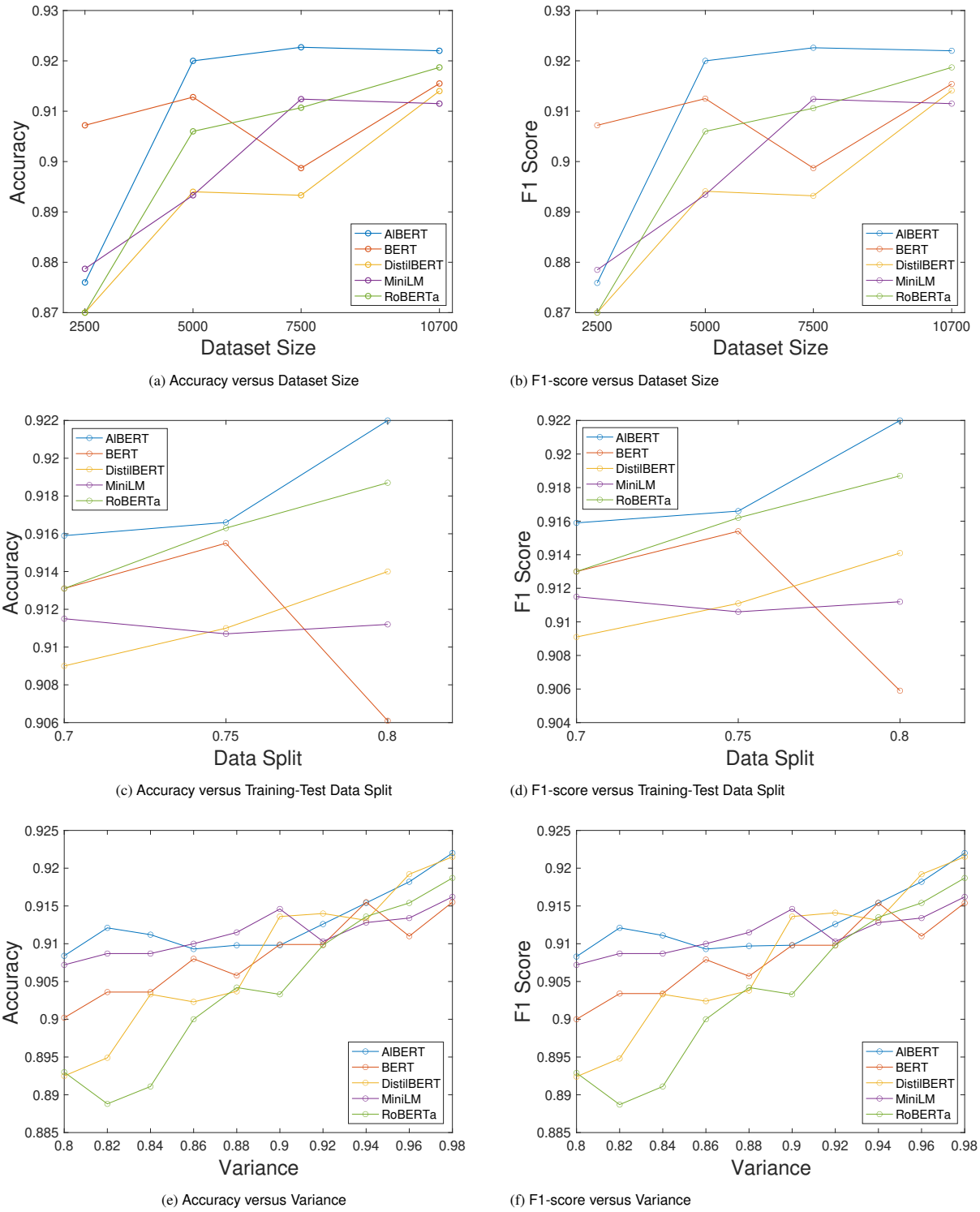
(d) F1-score versus Training-Test Data Split

(e) Accuracy versus Variance

(f) F1-score versus Variance

Figure 9. Effect of Various Factors on Multi-Layer Perceptron Classifier

pretrained models and a statistical feature fusion network. A cross-stitch based attention neural model proposed by Paka et al.(18) for classification attains a slightly better accuracy of 0.9540. On the other hand, this method requires more data such as user metadata and knowledge from external websites in addition to the post as its inputs before it can classify whether the post is fake or real.



(a) Accuracy versus Dataset Size

(b) F1-score versus Dataset Size

(c) Accuracy versus Training-Test Data Split

(d) F1-score versus Training-Test Data Split

(e) Accuracy versus Variance

(f) F1-score versus Variance

Figure 10. Effect of Various Factors on Logistic Regression Classifier

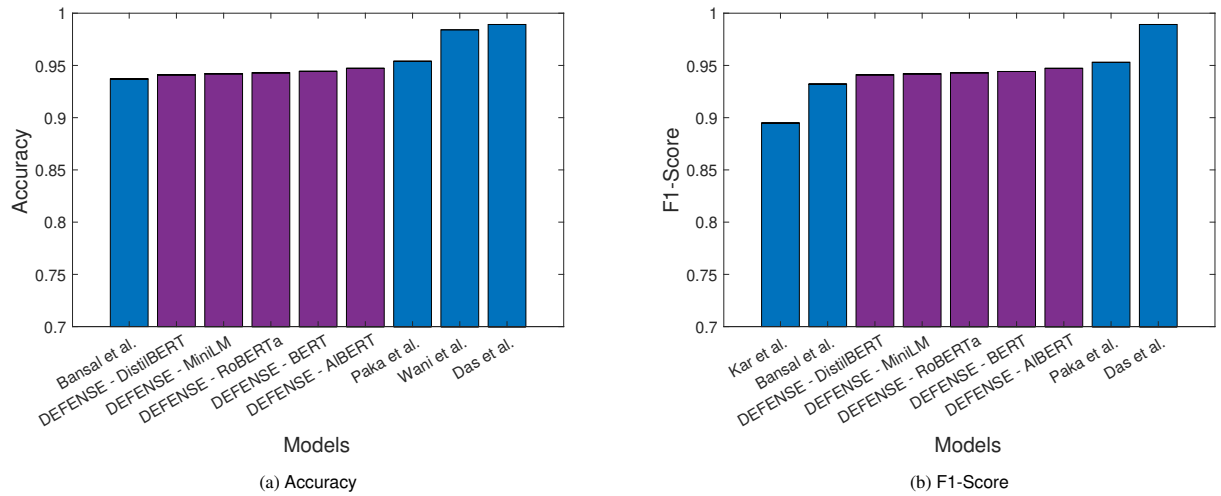
Table 3. Comparative Performance Study of Existing Results with our Approach

Approach	Author	Accuracy	Precision	Recall	F1-score
Bi-LSTM + Cross-Stitch + SentenceBERT	Paka et al.(18)	0.9540	0.9460	0.9610	0.9530
Bi-LSTM + SentenceBERT + FFN + Co-Attention + Softmax	Bansal et al.(3)	0.9370	0.9220	0.9430	0.9320
mBERT + Softmax	Kar et al.(11)	not mentioned	0.8717	0.9189	0.8947
BERT pretrained on Covid-19 tweets corpus	Wani et al.(31)	0.9841	not mentioned	not mentioned	not mentioned
SFFN (with MCDropout) + Heuristic Post-Processing	Das et al.(5)	0.9892	0.9892	0.9892	0.9892
DEFENSE					
DEFENSE: Features + BERT + PCA + MLP		0.9443	0.9447	0.9443	0.9442
DEFENSE: Features + RoBERTa + PCA + MLP		0.943	0.943	0.943	0.943
DEFENSE: Features + DistilBERT + PCA + MLP		0.9411	0.9411	0.9411	0.9411
DEFENSE: Features + AlBERT + PCA + MLP		**0.9472**	**0.9473**	**0.9472**	**0.9472**
DEFENSE: Features + MiniLM + PCA + MLP		0.9421	0.9421	0.9421	0.942

Conclusions and Future Work

In this paper, we propose DEFENSE, a model to solve the task of COVID-19 fake news detection in social media. Given a set of COVID-19 related posts from social media as input, DEFENSE determines whether the post is real or fake. The proposed method engineers the textual features, semantic features and the textual embeddings of the post. The embeddings are generated using pre-trained models of Sentence Transformers such as BERT, RoBERTa, DistilBERT, AlBERT and MiniLM and then reduced using Principal Component Analysis technique. The model was tested on different dataset sizes, data splits, a range of variance values and a number of classifier algorithms such as Decision Tree Classifier, Extra Tree Classifier, Gaussian Naive Bayes Classifier, Bernoulli Naive Bayes Classifier, Logistic Regression, k-Nearest Neighbours Classifier (KNN), Support Vector Machine Classifier (SVM), Poly Support Vector Machine Classifier (Poly-SVM) and Multi-layer Perceptron Classifier. The experimental results performed comparably to other models attempting the same task of COVID-19 fake news detection, with the best accuracy of 0.9472 obtained using Multi-layer Perceptron Classifier with AlBERT embedding technique.

Possible future works could be to build on this model by implementing it into a full-scale tool or using it as a base for other ventures. Although our proposed model is fine-tuned for the detection of COVID-19 related fake news, we could test to observe the performance of DEFENSE on general fake news datasets. Other possible directions to build on could be to extend this to other languages and



(a) Accuracy

(b) F1-Score

Figure 11. Performance of Proposed Models with Existing Approaches

analyse if sentence transformer models are able to extract the contextual meaning of texts well in a multilingual context. Moreover, testing out more rigorous embedding models that can capture the semantic meaning of the posts in a better manner is another option to look at. Finally, building a method to stop the spread of the posts that are detected as fake is a great way to ensure this work helps to minimise the circulation of fake news online.

References

[1] Charu C. Aggarwal and ChengXiang Zhai. An introduction to text mining. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 1–10. Springer, 2012.

[2] Costel-Sergiu Atodiresei, Alexandru Tănăselea, and Adrian Iftene. Identifying fake news and fake users on twitter. *Procedia Computer Science*, 126:451–461, 2018.

[3] Rachit Bansal, William Scott Paka, Shubhashis Sengupta, Tanmoy Chakraborty, et al. Combining exogenous and endogenous signals with a semi-supervised co-attention network for early detection of covid-19 fake tweets. *arXiv preprint arXiv:2104.05321*, 2021.

[4] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684, 2011.

[5] Sourya Dipta Das, Ayan Basak, and Saikat Dutta. A heuristic-driven uncertainty based ensemble framework for fake news detection in tweets and news articles. *arXiv preprint arXiv:2104.01791*, 2021.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining concepts and techniques, third edition, 2012.

[8] Stefan Helmstetter and Heiko Paulheim. Weakly supervised learning for fake news detection on twitter. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 274–277. IEEE, 2018.

[9] Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Applied Sciences*, 9(19):4062, 2019.

[10] Zhezhou Kang, Yanan Cao, Yanmin Shang, Tao Liang, Hengzhu Tang, and Lingling Tong. Fake news detection with heterogenous deep graph convolutional network. In *PAKDD (1)*, pages 408–420. Springer, 2021.

[11] Debanjana Kar, Mohit Bhardwaj, Suranjana Samanta, and Amar Prakash Azad. No rumours please! a multi-indic-lingual approach for covid fake-tweet detection. *arXiv preprint arXiv:2010.06906*, 2020.

[12] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019.

[13] Yang Liu and Yi-Fang Brook Wu. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[15] Yi-Ju Lu and Cheng-Te Li. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv preprint arXiv:2004.11648*, 2020.

[16] Michal Lukasik, Trevor Cohn, and Kalina Bontcheva. Classifying tweet level judgements of rumours in social media. *arXiv preprint arXiv:1506.00468*, 2015.

[17] Said Özcan. Tweet preprocessor. https://pypi.org/project/tweet-preprocessor/.

[18] William Scott Paka, Rachit Bansal, Abhay Kaushik, Shubhashis Sengupta, and Tanmoy Chakraborty. Cross-sean: A cross-stitch semi-supervised neural attention model for covid-19 fake news detection. *Applied Soft Computing*, 107:107393, 2021.

[19] Parth Patwa, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas PYKL, Amitava Das, Asif Ekbal, Shad Akhtar, and Tanmoy Chakraborty. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *Proceedings of the First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (CONSTRAINT)*. Springer, 2021.

[20] Parth Patwa, Shivam Sharma, Srinivas Pykl, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. Fighting an infodemic: Covid-19 fake news dataset. *arXiv preprint arXiv:2011.03327*, 2020.

[21] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*, 2017.

[22] Maryam Ramezani, Mina Rafiei, Soroush Omranpour, and Hamid R Rabiee. News labeling as early as possible: Real or fake? In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 536–537. IEEE, 2019.

[23] Nils Reimers. Pre-trained sentence embedding models. https://www.sbert.net/docs/pretrained_models.html.

[24] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[25] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[26] Kai Shu, Guoqing Zheng, Yichuan Li, Subhabrata Mukherjee, Ahmed Hassan Awadallah, Scott Ruston, and Huan Liu. Leveraging multi-source weak social supervision for early detection of fake news. *arXiv preprint arXiv:2004.01732*, 2020.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[28] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[29] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*, 2020.

[30] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857, 2018.

[31] Apurva Wani, Isha Joshi, Snehal Khandve, V Wagh, and R Joshi. Evaluating deep learning approaches for covid19 fake news detection. In *Combating Online Hostile Posts in Regional Languages during Emergency Situation: First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers*, page 153. Springer Nature, 2021.

[32] Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40, 2020.