Misinformation Detection with Graph Neural Networks

Zain Zain
Department of NLP
MBZUAI, Abu Dhabi, UAE
zain.mujahid@mbzuai.ac.ae

Muhammad Arslan Manzoor
Department of NLP
MBZUAI, Abu Dhabi, UAE
muhammad.arslan@mbzuai.ac.ae

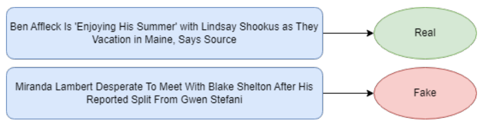
Abstract

Social media is a popular and easily accessible source that publish news with a fast rate of dissemination. Misinformation is the precarious product of social media that impacts society in different negative ways. Recently, researchers have presented the repository known as FakeNewsNet to conduct research on the detection of misinformation. This report presents the problem formulation, motivation, dataset structure and evaluation of machine learning baselines on the benchmark mentioned above. The results depict that Bernoulli Naive Bayes achieves the highest accuracy and F1 score among all approaches. We later explore multiple ensemble learning techniques with topK classifiers, and BERT embedding with multiple model designs to do the same. The graphical structure of the FakeNewsNet has already motivated the research community to address this task through geometric deep-learning approaches. Therefore, we explore different varients of these graphical networks to see how graphical structure of FakeNewsNet helps in detecting misinformation.

1 Introduction

Social media usage, particularly on sites like Twitter has recently ingrained itself into our everyday lives. Users may conveniently communicate private messages, images, and videos on social networks. However, while many individuals love social networks, dishonest practices like spreading rumors or fake news can persuade users to believe false information. Unintentionally affecting public opinion and endangering social and political development, the vast distribution of misinformation on social networks has emerged as a worldwide problem. As a result, misinformation detection (MID) in social networks has drawn a lot of interest and is now seen to be an emerging topic of research.

Figure 1: Fake News Classfication Example



Many studies have been done for MID, even yet, it might be challenging to identify misinformation automatically since a sophisticated model is needed to determine how relevant or unrelated the reported information is to actual information. We can treat this problem as a text classification problem an example of which can be see in Figure 1

In this project, we use a linear approach to understand how previous researchers have been detecting misinformaiton using traditional machine learning methods. We explore ensemble methods and BERT embeddings to do the same. At the end, due to the graphical structure of the dataset that we use, we explore multiple graph neural networks variants to see how they are useful for the very task.

2 Literature Review

In this part, we first go through the most popular methods for identifying fake information. We then examine several applications of geometric deep learning to achieve the same goals. Many researchers have previously tried to do misinformation detection using context and content based methods.

A variety of models fall under the category of **traditional statistical ML models** which employ statistical analysis to categorise an input. For instance, the Naive Bayes (NB) classifier (Oraby et al., 2017) uses the Bayes theorem to determine the result that is most likely to occur from a finite set of possible outcomes using input numeric characteristics. By maximising the width between any two categories, the Vapnik-Chervonenkis theory is used by the support vector machine (SVM) classifier (Zhang et al., 2012) to classify data into numerous groups. A decision tree (DT) (Lyu and Lo, 2020) seeks to identify the decision process in which each numerical characteristic determines a potential path to a resolution. The random forest (RF) model (Natarajan et al., 2022), on the other hand, comprises of many DTs in order to absorb more information and provide more complicated paths for determining the potential classification outcomes.

One can drive features at words level, like word counts and TF-IDF. (Benamira et al., 2019) did the same, they extracted features at word level and treated the problem of MID as a binary text classification. They faced the problem of low recall on the fake class due to limited tagged data. We will be trying to overcome this problem by using contextual information by extracting relations between different nodes in our data.

Other approach can be to extract features at sentence level. (Huang et al., 2020b) did exactly the same. They introduced a syntactic tool which can be used to extract syntax from syntax. They utilized the transfer learning approach to apply existing models for MID on different media applications. Their technique was good in detecting false information across different platforms but with the unseen facts the recall was low. We will try to generalize our model to detect any future unseen fake news as well.

Deep learning models, such as neural networks, are now recognised as a successful technique for spotting fake news (Ruchansky et al., 2017). Recurrent neural networks (RNNs), which include basic RNN, GRU, and LSTM, are among the deep learning techniques that take time series into account when categorising news since it might be crucial to monitor changes and the emergence of false news. For instance, (Ma et al., 2016) trained a multi-layer GRU based on the time series of the tweets, using each tweet's input of a 5000 dimension TF-IDF score to eventually predict the veracity of an event. Comparing this strategy to non-deep learning methods, it produces an accuracy performance gain of 10%. (e.g., DT ranking, SVM, RF classification).

In order to identify fake news, convolutional neural network (CNN) models have also been used. To identify news items that are not factual, (Wang, 2017) suggested employing a CNN model to analyse the textual information from news articles. According to the content of the news items, (Fang et al., 2019) advocated using self attention-based CNN, which performed better than RNN-based models on the job of identifying non-factual articles.

In addition, there are characteristics obtained from the entire text, such as topic or sentiment analysis, as well as more generic representations. (Cheng et al., 2020) introduced a LSTM based variational autoencoder (Kingma and Welling, 2013) misinformation detection system based on these methods. They were able to detect unseen labels with good accuracy.

Based on the work from previous author, (Han et al., 2020) used GNN (Scarselli et al., 2008) excluding the textual information. They found out that even without any textual data, the GNNs can learn to discriminate real and fake news way better than many state of the art existing models.

(Huang et al., 2020a) utilized the overall text content semantic relationships and made a multimodal graph based on rumor spread from source tweets and text content. Their research on actual Twitter data proved the effectiveness of their suggested method for early rumor detection. We will be using the same content semantic to increase the performance of our final graph.

(Lu and Li, 2020) suggested a co attention graph model for misinformation detection that is built on deep neural networks. By highlighting the evidence of suspicious retweeters and the phrases they concern, authors were able to determine if the tweet was genuine or not.

(Wu et al., 2020) tried to explain the claim verification process by using Decision Trees Co Attention models.In addition to claim verification, their suggested design delivers cutting-edge performance.

The **syntax of writing**, commonly thought of as grammar, is useful in spotting false information (Kasami, 1966). With this technique, text is converted into a syntax tree, revealing and allowing for further study of the grammatical relationships between nouns, verbs, subjects, and objects. For instance, (Feng et al., 2012) demonstrated that the deception detection task can be completed with 70%-90% accuracy utilising features generated by Context Free Grammar parse trees across four datasets.

The goal of **sentiment analysis** is to determine if a text conveys positive, neutral, or negative emotion. It is useful to comprehend the sentimental view by assigning a single sentiment score or by using a label (positive, negative, neutral). Sentiment has been demonstrated to be beneficial in false news identification tasks. As an illustration, (Hamidian and Diab, 2019) work demonstrated how sentiment analysis enhanced the model's performance when it came to identifying bogus news. Additionally, (Ghenai and Mejova, 2017) found the most useful characteristics for determining if a tweet contains rumours regarding the Zika infection by combining information gain (IG) with the greedy backward elimination approach. They demonstrated that, according to IG, sentiment score is the fourth-best feature.

Automatic fact-checking systems seek to eliminate the need for human journalists by labelling news as fake or not directly by the algorithms, maximising the automation of the fact-checking process. For instance, FakeNewsTracker (Shu et al., 2019) collected tweets linked to bogus news that has already been exposed by Politifact and BuzzFeed News, and then extracted helpful textual elements to create machine learning models for spotting fake news.

Based on a collection of explicit and latent properties derived from the textual data, FAKEDETECTOR (Zhang et al., 2020) constructed a deep diffusive network to represent news items, producers, and subjects all at once.

According to ACT (Aloshban, 2020), it is possible to classify whether an article contains nonfactual information by employing a two-dimensional matrix that incorporates the combined credibility of the claim-article combination and textual elements from language models.

In order to undertake fact-checking utilising the Wikipedia Knowledge base, WikiCheck (Trokhymovych and Saez-Trumper, 2021) suggested finding evidence that either supports or contradicts statements.

To effectively index, filter, and compare content against pre-existing false news, WebChecker (Trummer, 2021) uses a variety of cost-accuracy tradeoffs and a reinforcement learning-based optimizer to discover the best checking plans.

Humanin-the-loop systems, which take into account the challenges of developing a completely automatic fact-checking system, use journalists to manually verify news with information that has been automatically collected by machine learning models.

For instance, Scrutinizer (Karagiannis et al., 2020) decreased the manual fact-checking time by organising the precise queries that need human editors to explain and automatically identifying and marking the components of the assertions that need to be reviewed.

Journalists can target a specific claim on Twitter using ClaimPortal (Majithia et al., 2019), which offers an integrated online platform where algorithms trained on a database of false claims can assist categorise the claim as true or false.

FactCatch (Nguyen et al., 2020) suggested that a claims poll be used to identify a set of claims that it deemed valuable to be fact-checked, and those valuable claims would then be sent to human fact-checkers. A final judgement for the claim would then be calculated using the automatically inferred the truthfulness of the claim and the input of the human judges.

In order to distinguish fake news about the COVID-19 pandemic, CoVerifi (Kolluri and Murthy, 2021) created a platform where fact-checkers' categorization findings could be combined with a GPT-2 model. To recognise machine-generated text and determine whether claims are false or true, the GPT-2 model is trained using the CoAID dataset (a freshly created COVID-19 dataset (Cui and Lee, 2020)).

ClaimHunter (Beltrán et al., 2021) adopted a reinforcement learning strategy, where the system sends claims and the machine generated predictions to journalists, and also uses the journalists' final judgments to improve the system.

Watch' n' Check (Cerone et al., 2020) proposed a platform that provides a keyword-filtering process where journalists can monitor and follow the discussion of a specific topic.

Last but not least, WhistleBlower (Ramachandran et al., 2020) suggested to identify false news using textual attributes produced by language models while also enabling the fact-checking community to update the labelling of bogus news using blockchain nodes.

Knowledge graphs are utilised for a variety of downstream tasks, including entity linking [(Chen et al., 2020), (Moreno et al., 2017)], relation prediction [125, 192], and knowledge graph completion [(Lin et al., 2015), (Shi and Weninger, 2017)]. Knowledge graphs are able to give structured information about entities and relations. The approaches now in use for knowledge graphs-based fake news detection mainly concentrated on named entity linking [(Cheema et al., 2020), (Hassan et al., 2015)], creating facts using Wikipedia (Trokhymovych and Saez-Trumper, 2021), and entity similarity (Su et al., 2019). Limited study has been done on embedded entities' potential for identifying bogus news, though. Therefore, the ideal entity embedding models and how to customise entity representations for fake news detection are still up for debate.

3 Dataset

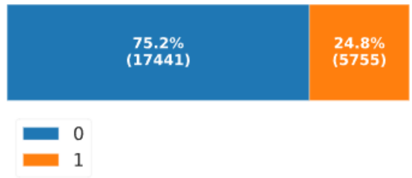
In order to detect Misinformation we use FakeNewsNet (Shu et al., 2018), a repository consisting of two multilingual extensive datasets with news content, social context, and dynamic data. In this section we describe the overall overview structure of the data.

We started with the basic publicly available version of the dataset. There are 4 csv files in total, 2 for each new content, 1 for real news and 1 for fake. Each csv contains id, news url, the title of the news along with its tweet it. We combine all these csv, assign labels 1: Fake, 0: Real and get a final of **23196** labeled examples. We use this datasets to do the preliminary testing to make test 7 machine learning algortihms, discussed further in the report, to make a comparison of baseline model.

To get the contextual and graphical information between tweets and users, one can add the Twitter API in the dataset repository. As per (Shu et al., 2018) the complete version of the dataset have the following structure:

- All of the meta data for the news articles was compiled using the specified news source URLs and is contained in the file *news content.json* This is a JSON object containing the following attributes:
  - **text** contains the text about the news articles
  - **images** is the list containing the url for all the images used in the webpage
  - **publish date** is the date the news article got published
- The information for the list of tweets linked to the news story are stored in the **tweets folder** as individual files for each tweet. These files contain the tweet objects that Twitter API returns.

Figure 2: Category Frequency Plot of Target Variable



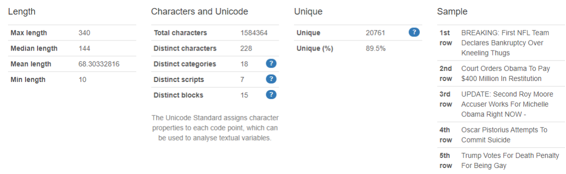
- A list of files in **retweets folder** contains tweets that have been re-posted and are related to the news story. Each file is given the name **tweet id.json** and use Twitter API to gather a list of retweet objects connected to a certain tweet.
- The **user profiles folder** contains files with all the user metadata from the dataset. Each file in this directory is a JSON object that was gathered from the Twitter API and contains details about the user, such as the time the profile was created, the URL of the profile image, the user's location, the number of tweets that have been posted, the number of followers and followings, and the number of favorited tweets.
- The **user timeline tweets folder** contains JSON files with a list of the user's most recent N tweets. This contains the whole tweet object, including all tweet-related data.

This part of the data can only be gathered by running the scripts provided by the authors using our personal Twitter API. Our initial attempts to crawl this data failed. So for the later part of the project, graphical neural networks, we use already provided embeddings of the same dataset by a different source.

3.1 EDA & Pre Processing

For exploratory data analysis we explored the whole dataset in depth. In this section we share the analysis on the columns of interest. For the first part of the project we are using the tweet titles and the target variable which have two classes, 1: Fake, 0: Real. The distribution of these classes can be seen in the Category Frequency Plot in Figure 2. Note that we have approx. 23K samples out of which only 5K are labelled as Fake. The dataset is highly imbalanced. The informaiton about the Tweets be seen in Figure 3 The distribution of tweet title lengths can be seen in its length histogram in Figure 4

Figure 3: Overview of Tweet Titles



Length

Max length	340
Median length	144
Mean length	68.30332816
Min length	10

Characters and Unicode

Total characters	1584364
Distinct characters	228
Distinct categories	18
Distinct scripts	7
Distinct blocks	15

The Unicode Standard assigns character properties to each code point, which can be used to analyse textual variables.

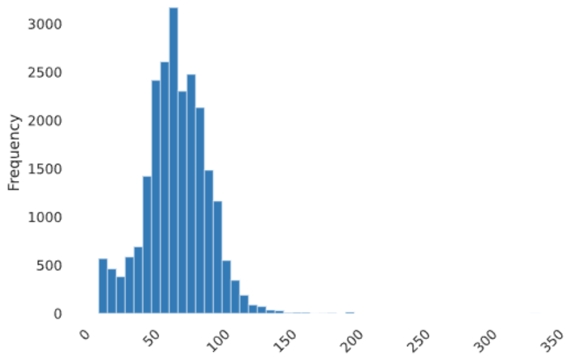
Unique

Unique	20761
Unique (%)	89.5%

Sample

1st row	BREAKING: First NFL Team Declares Bankruptcy Over Kneeling Thugs
2nd row	Court Orders Obama To Pay $400 Million In Restitution
3rd row	UPDATE: Second Roy Moore Accuser Works For Michelle Obama Right NOW -
4th row	Oscar Pistorius Attempts To Commit Suicide
5th row	Trump Votes For Death Penalty For Being Gay

Figure 4: Histogram of Tweet Title Length



For the pre-processing part we did all the usual text-cleaning steps including removing stopwords, emoticons & emojis, extra spaces, punctuations and contractions etc along with lower casing the text.

4 Software Packages Used

In this section we list the python packages we have used until now.

1. Sklearn (Buitinck et al., 2013) is used for designing pipeline, to get features from text, and to train and evaluate 10 baseline models
2. NLTK (Bird et al., 2009) is used to remove stopwords from the text.
3. Transformers (Wolf et al., 2020) is used to get BERT embeddings.
4. PyTorch Geometric (Fey and Lenssen, 2019) is used to construct 4 varients of GNNs.

5 Experiments

In this section we describe the methodology we have used to make different experiments, their evaluation and comparison. We took a linear approach and started with traditional machine learning algorithms, then moving towards ensemble methods to classify target variable using BERT embeddings. At the end we explored graph neural networks to do the same.

5.1 Methodology

First of all we assign the target variable to each CSV, Fake = 1, Real = 0, concatenate the resultant data into 1 big dataframe. We then shuffle and split the data into training and testing test. The reserved size for testing set is 20% of the total data. Note that for the feature extraction we are only using tweet tiles for now and our target variable is the label of the tweet either real or fake.

Since this dataset contains English language, we remove the stopwords from the tweet titles, it is previously proved to enhance the ability in pattern recognition (Munková et al., 2013).

We then use the approach described by (Patel and Meehan, 2021) to utilize **CountVectorizer** (HB et al., 2017) and **Term Frequency-Inverse Document Frequency** (Aizawa, 2003) to classify fake news on Reddit. We pass the cleaned tweet titles to the pipeline containing:

1. CountVectorizer Object
2. TF-IDF Object
3. Classifier Object

We **train** the final pipleine on training data, get **predictions** on the test data. And **evaluate** models based on their final **recall** and **f1-score**. Most of the models are the ones covered during 1st Semester in **AI701** & **NLP701**.

5.2 Multinomial Naive Bayes Model

We started our experimentation with the Multinomial Naive Bayed Model (Kibriya et al., 2004). We get a final **accuracy** of **82%** but the model is biased towards the Real News class and the **recall** on the Fake News Class is very low, **0.32**.

5.3 Bernoulli Naive Bayes Model

We then use a simplified Naive Bayes designed to work with 2 classes (Wang et al., 2015). It slightly improved the **accuracy**, **83%** and the **recall** on the fake class **0.55**.

5.4 Decision Tree Classifier

We then use more explainable models, Decision Trees (Rokach and Maimon, 2005) to classifiy the tweets. Model performed good on the fake class with a **recall** of **0.53** and a final **accuracy** of **78%**.

5.5 Random Forest Classifier

We went on with using Random Forest (Parmar et al., 2018) which use multiple Decision Trees. With **100 n_estimators** we were able to get **0.51 recall** on fake class and a final **accuracy** of **84%**. This didn't turn out to improve very much than the normal decision trees.

5.6 Perceptron Classifier

We used Perceptron algorithm (Kanal, 2003) for our problem. We got a final **accuracy** of 78% with a **0.56 recall**. It was amazing to see how can a very easy classifier can classify tweets just with the tweet title information.

5.7 Logistic Regression

With continued the experimentation by using the logistic regression algorithm (Wright, 1995). However we didn't get any improved results than normal perceptron algorithm. The final **accuracy** turned out to be **84%** with a **recall** on fake class of **0.45**.

5.8 K-Neighbors Classifier

USing K-Neighbors Classifier (Sun and Huang, 2010). We were able to get a final accuracy of **80%** with a **recall** of **0.46** on fake class.

5.9 Stochastic Gradient Classifier

We explored stochastic gradient classifier (Ranjeeth et al., 2021) to classify fake news. We were able to get a final accuracy of **84%** with a **recall** of **0.46** on fake class.

5.10 Support Vector Classifier

Our last baseline model was support vector classifier (Lau and Wu, 2003). We were able to get a final accuracy of **84%** with a **recall** of **0.47** on fake class

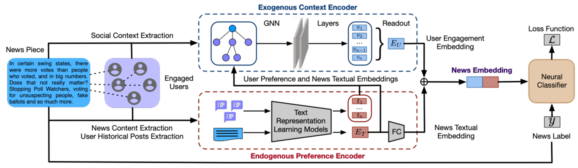
5.11 Ensemble Learning

We used topK algorithms from our baseline with different values of K and ensembled them. We used soft voting classifier (Karlos et al., 2020), hard voting classifer (Habib and Tasnim, 2020), stacking (Sesmero et al., 2015), bagging (Zareapoor et al., 2015) and boosting (Chen et al., 2015) to classify fake news detection, and were able to get 0.76, 0.73, 0.74, 0.75, 0.71 f1-score respectively

5.12 BERT Embeddings

We used bert-base-uncased from HuggingFake to get sentence embedding using different combine strategies, and used different algortihms, a total of 5, to classify these embeddings into real and fake. Our results were not good, so we tried AutoTrain from huggingFace to do this for us and the output results were amazing. All 5 models had f1-score in range of 0.87-0.89.

Figure 5: UPFD Framework



5.13 Graph Neural Networks

For this part of the project we used the embedding of FakeNewNet provided by (Dou et al., 2021). And explored multiple graph neural networks varients on these. The 768-dimensional bert and 300-dimensional spacy features are encoded using pre-trained BERT and spaCy word2vec, respectively. There are four different node feature types included in the dataset. A Twitter account's profile is used to create the 10-dimensional profile feature. The 310-dimensional content feature is made up of a 10-dimensional profile feature in addition to a 300-dimensional user comment word2vec (spaCy) embedding.

Each network is a hierarchical tree-structured graph, with the news as the root node and Twitter users as leaf nodes who have retweeted the article. If a user node retweeted the news tweet, it has an advantage over the news node. If one user retweeted the other user's news tweet, the two user nodes are in a better position. The UPFD framework, together with the specifics of dataset generation, is shown in Figure 5.

We started with Bi-Directional Graph Convolutional Networks adopted from the original implementation from the paper authors (Bian et al., 2020) It makes use of a GCN with an opposing directed graph of rumour diffusion and a GCN with a top-down directed graph of rumour spreading to learn the patterns of rumour propagation and to capture the structures of rumour dispersion. We used all **profile**, **Word2Vec** and **Bert** embeddings individually to get final f1-score of **0.78, 0.79 and 0.80** respectively.

We then used GCN-FN (Monti et al., 2019) which is implemented using two GCN layers and one mean-pooling layer as the graph encoder. With

3 encodings we were able to get final f1-scores of **0.77, 0.83, 0.81**

Finally we used GNN-CL (Han et al., 2020) which is implemented using DiffPool as the graph encoder and were able to get final f1-scores of 0.67, 0.38 and 0.69 on our 3 embeddings.

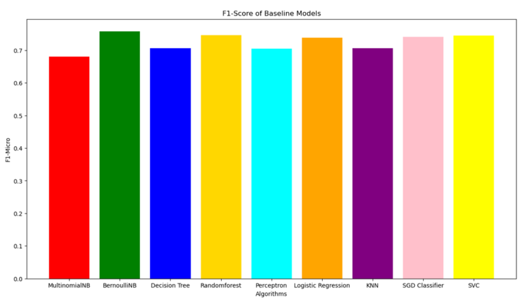
6 Evaluation

We evaluated all our models based on their final **f1-score**. The f1-score of traditional machine learning experiments can be seen in Table 1

Traditional Models	**F1-Score**
MultinomialNB	0.6798
BernoulliNB	**0.7576**
Decision Tree	0.7056
Random Forest	0.7465
Perceptron	0.7043
Logistic Regression	0.7378
K-Nearest Neighbors	0.7065
SGD Classifier	0.7410
Support Vector Classifier	0.7451

Table 1: F1-Score of Baseline Models

Figure 6: F1-Score of Baseline Model



A bar plot in Figure 6 is made to better visualize the f1-score of each baseline model. From both table 1 and bar plot 6 it can be seen that Bernoulli Naive Bayes baseline model performed the best in order to detect misinformation with a final **f1-score** of **0.7576**

The f1-scores of **Ensemble learning** experiments can be seen in Table 2

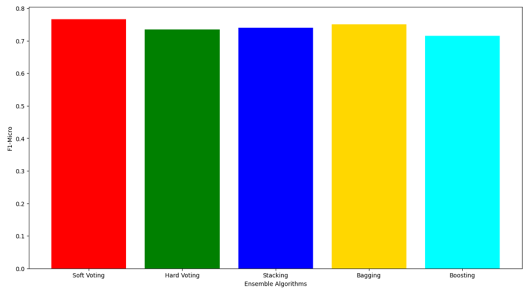
A bar plot in Figure 7 is made to better visualize the f1-score of each baseline model. From both table 2 and bar plot 7 it can be seen that Soft Voting ensemble slightly improved the f1-score of traditional methods to detect misinformation with a final **f1-score** of **0.7662**

The f1-score of 5 models auto trained using BERT embeddings were between range 0.87-0.89

Traditional Models	**F1-Score**
Soft Voting	**0.7662**
Hard Voting	0.7349
Stacking	0.7407
Bagging	0.7505
Boosting	0.7154

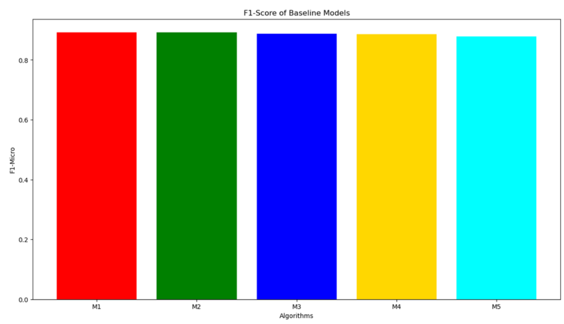
Table 2: F1-Score of Ensemble Models

Figure 7: F1-Score of Ensemble Model



as can be seen in the bar plot from Figure 8. We can see that bert out-performed all the methods that we used so far.

Figure 8: F1-Score Models using BERT Embeddingsl



The f1-scores after using Graph Neural Networks can be seen in Table 3. We can see that GCN-FN architecture performed the best with a final f1-score of **0.8349** on BERT embeddigns.

7 Conclusion & Future Work

We saw that even with no context information many classifiers can perform really good in classifying misinformation. These models were merely trained on the titles of the tweet. Imagine getting tweet text and do the same. The ensemble methods incorporate multiple traditional models and slightly increased the f1-score. With BERT sentence embeddings we got the maximum f1-score. If we compare graphical methods which uses user behavior as a factor, with traditional ML and ensemble

Models	f1-macro		
	profile	Word2Vec	Bert
BiGCN	0.7804	0.7907	**0.8044**
GCN-FN	0.7722	**0.8349**	0.8177
GNN-CL	0.6767	0.3845	**0.6996**

Table 3: F1-Score of GNN Models

model, we will know that GNN are much better in classifying fake news.

Continuing forward, we will try deep learning methods and convolutional neural networks on the same baseline data that we used above. We will run all the models again on tweet title+text data once we are able to scrape everything perfectly. And finally we will try multiple GNN models and combining them to get the new state of the art f1-score on this dataset.

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Nujud Aloshban. 2020. Act: Automatic fake news classification through self-attention. In *12th ACM Conference on Web Science*, pages 115–124.

Javier Beltrán, Rubén Míguez, and Irene Larraz. 2021. Claimhunter: An unattended tool for automated claim detection on twitter. In *KnOD@ WWW*.

Adrien Benamira, Benjamin Devillers, Etienne Lesot, Ayush K Ray, Manal Saadi, and Fragkiskos D Malliaros. 2019. Semi-supervised learning and graph neural networks for fake news detection. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 568–569. IEEE.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 549–556.

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

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Assunta Cerone, Elham Naghizade, Falk Scholer, Devi Mallal, Russell Skelton, and Damiano Spina. 2020. Watch'n'check: Towards a social media monitoring tool to assist fact-checking experts. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 607–613. IEEE.

Gullal S Cheema, Sherzod Hakimov, and Ralph Ewerth. 2020. Check_square at checkthat! 2020: Claim detection in social media via fusion of transformer and syntactic features. *arXiv preprint arXiv:2007.10534*.

Shuang Chen, Jinpeng Wang, Feng Jiang, and Chin-Yew Lin. 2020. Improving entity linking by modeling latent entity type information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7529–7537.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Mingxi Cheng, Shahin Nazarian, and Paul Bogdan. 2020. Vroc: Variational autoencoder-aided multi-task rumor classifier based on text. In *Proceedings of the web conference 2020*, pages 2892–2898.

Limeng Cui and Dongwon Lee. 2020. Coaid: Covid-19 healthcare misinformation dataset. *arXiv preprint arXiv:2006.00885*.

Yingtong Dou, Kai Shu, Congying Xia, Philip S Yu, and Lichao Sun. 2021. User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2051–2055.

Yong Fang, Jian Gao, Cheng Huang, Hua Peng, and Runpu Wu. 2019. Self multi-head attention-based convolutional neural networks for fake news detection. *PloS one*, 14(9):e0222713.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Amira Ghenai and Yelena Mejova. 2017. Catching zika fever: Application of crowdsourcing and machine learning for tracking health misinformation on twitter. *arXiv preprint arXiv:1707.03778*.

Al-Zadid Sultan Bin Habib and Tanpia Tasnim. 2020. An ensemble hard voting model for cardiovascular disease prediction. In *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pages 1–6. IEEE.

Sardar Hamidian and Mona T Diab. 2019. Rumor detection and classification for twitter data. *arXiv preprint arXiv:1912.08926*.

Yi Han, Shanika Karunasekera, and Christopher Leckie. 2020. Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th acm international on conference on information and knowledge management*, pages 1835–1838.

Barathi Ganesh HB, Soman KP, et al. 2017. Vector space model as cognitive space for text classification. *arXiv preprint arXiv:1708.06068*.

Qi Huang, Junshuai Yu, Jia Wu, and Bin Wang. 2020a. Heterogeneous graph attention networks for early detection of rumors on twitter. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Yen-Hao Huang, Ting-Wei Liu, Ssu-Rui Lee, Fernando Henrique Calderon Alvarado, and Yi-Shin Chen. 2020b. Conquering cross-source failure for news credibility: Learning generalizable representations beyond content embedding. In *Proceedings of The Web Conference 2020*, pages 774–784.

Laveen N Kanal. 2003. Perceptron. In *Encyclopedia of Computer Science*, pages 1383–1385.

Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. 2020. Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification. *arXiv preprint arXiv:2003.06708*.

Stamatis Karlos, Georgios Kostopoulos, and Sotiris Kotsiantis. 2020. A soft-voting ensemble based co-training scheme using static selection for binary classification problems. *Algorithms*, 13(1):26.

Tadao Kasami. 1966. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*.

Ashraf M Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. 2004. Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence*, pages 488–499. Springer.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Nikhil L Kolluri and Dhiraj Murthy. 2021. Coverifi: A covid-19 news verification system. *Online Social Networks and Media*, 22:100123.

KW Lau and QH Wu. 2003. Online training of support vector classifier. *Pattern Recognition*, 36(8):1913–1920.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

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

Shikun Lyu and Dan Chia-Tien Lo. 2020. Fake news detection by decision tree. In *2020 SoutheastCon*, pages 1–2. IEEE.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Sarthak Majithia, Fatma Arslan, Sumeet Lubal, Damian Jimenez, Priyank Arora, Josue Caraballo, and Chengkai Li. 2019. Claimportal: Integrated monitoring, searching, checking, and analytics of factual claims on twitter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

Jose G Moreno, Romaric Besançon, Romain Beaumont, Eva D'hondt, Anne-Laure Ligozat, Sophie Rosset, Xavier Tannier, and Brigitte Grau. 2017. Combining word and entity embeddings for entity linking. In *European Semantic Web Conference*, pages 337–352. Springer.

Daša Munková, Michal Munk, and Martin Vozár. 2013. Influence of stop-words removal on sequence patterns identification within comparable corpora. In *International Conference on ICT Innovations*, pages 67–76. Springer.

Rathika Natarajan, Abolfazl Mehbodniya, Kantilal Pitambar Rane, Sonika Jindal, Mohammed Faez Hasan, Luis Vives, and Abhishek Bhatt. 2022. Intelligent gravitational search random forest algorithm for fake news detection. *International Journal of Modern Physics C*, 33(06):2250084.

Thanh Tam Nguyen, Matthias Weidlich, Hongzhi Yin, Bolong Zheng, Quang Huy Nguyen, and Quoc Viet Hung Nguyen. 2020. Factcatch: Incremental pay-as-you-go fact checking with minimal user effort. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2165–2168.

Shereen Oraby, Lena Reed, Ryan Compton, Ellen Riloff, Marilyn Walker, and Steve Whittaker. 2017. And that's a fact: Distinguishing factual and emotional argumentation in online dialogue. *arXiv preprint arXiv:1709.05295*.

Aakash Parmar, Rakesh Katariya, and Vatsal Patel. 2018. A review on random forest: An ensemble classifier. In *International Conference on Intelligent Data Communication Technologies and Internet of Things*, pages 758–763. Springer.

Ankitkumar Patel and Kevin Meehan. 2021. Fake news detection on reddit utilising countvectorizer and term frequency-inverse document frequency with logistic regression, multinominalnb and support vector machine. In *2021 32nd Irish Signals and Systems Conference (ISSC)*, pages 1–6.

Gowri Ramachandran, Daniel Nemeth, David Neville, Dimitrii Zhelezov, Ahmet Yalçin, Oliver Fohrmann, and Bhaskar Krishnamachari. 2020. Whistleblower: towards a decentralized and open platform for spotting fake news. In *2020 IEEE International Conference on Blockchain (Blockchain)*, pages 154–161. IEEE.

S Ranjeeth, TP Latchoumi, and P Victer Paul. 2021. Optimal stochastic gradient descent with multilayer perceptron based student's academic performance prediction model. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, 14(6):1728–1741.

Lior Rokach and Oded Maimon. 2005. Decision trees. In *Data mining and knowledge discovery handbook*, pages 165–192. Springer.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

M Paz Sesmero, Agapito I Ledezma, and Araceli Sanchis. 2015. Generating ensembles of heterogeneous classifiers using stacked generalization. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 5(1):21–34.

Baoxu Shi and Tim Weninger. 2017. Proje: Embedding projection for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Kai Shu, Deepak Mahudeswaran, and Huan Liu. 2019. Fakenewstracker: a tool for fake news collection, detection, and visualization. *Computational and Mathematical Organization Theory*, 25(1):60–71.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 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*.

Ting Su, Craig Macdonald, and Iadh Ounis. 2019. Entity detection for check-worthiness prediction: Glasgow terrier at clef checkthat! 2019.

Shiliang Sun and Rongqing Huang. 2010. An adaptive k-nearest neighbor algorithm. In *2010 seventh international conference on fuzzy systems and knowledge discovery*, volume 1, pages 91–94. IEEE.

Mykola Trokhymovych and Diego Saez-Trumper. 2021. Wikicheck: An end-to-end open source automatic fact-checking api based on wikipedia. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4155–4164.

Immanuel Trummer. 2021. Webchecker: Towards an infrastructure for efficient misinformation detection at web scale. *IEEE Data Eng. Bull.*, 44(3):66–77.

Shasha Wang, Liangxiao Jiang, and Chaoqun Li. 2015. Adapting naive bayes tree for text classification. *Knowledge and Information Systems*, 44(1):77–89.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Raymond E Wright. 1995. Logistic regression.

Lianwei Wu, Yuan Rao, Yongqiang Zhao, Hao Liang, and Ambreen Nazir. 2020. Dtca: Decision tree-based co-attention networks for explainable claim verification. *arXiv preprint arXiv:2004.13455*.

Masoumeh Zareapoor, Pourya Shamsolmoali, et al. 2015. Application of credit card fraud detection: Based on bagging ensemble classifier. *Procedia computer science*, 48(2015):679–685.

Hu Zhang, Zhuohua Fan, Jiaheng Zheng, and Quanming Liu. 2012. An improving deception detection method in computer-mediated communication. *Journal of Networks*, 7(11):1811.

Jiawei Zhang, Bowen Dong, and S Yu Philip. 2020. Fakedetector: Effective fake news detection with deep diffusive neural network. In *2020 IEEE 36th international conference on data engineering (ICDE)*, pages 1826–1829. IEEE.