Identification of Fake News in Social Media

Mihir Rao
msrao@andrew.cmu.edu

Leah Chowenhill
lchowenh@andrew.cmu.edu

Aniket Sakpal
asakpal@andrew.cmu.edu

Shubham Dubey
srdubey@andrew.cmu.edu

Rucha Khopkar
rkhopkar@andrew.cmu.edu

Shreyas Piplani
spiplani@andrew.cmu.edu

Bhiksha Raj
bhikshar@andrew.cmu.edu

Abstract

With the growth of social media networks like Twitter, misinformation has become an increasingly prevalent issue. Moreover, social media algorithms reinforce confirmation bias, only showing users content which aligns with their views. As the amount on content shared on social networks is exponential, fact-checking an impossible task.Therefore, online platforms have begun to turn to Deep Learning as a means of classifying content as real or fake.

Fake news is known to have a different propagation pattern from real news. Therefore, Graphical Neural Networks (GNNs) are commonly used to identify fake news using both news content features and propagation features. More recently, endogenous features have been experimented with for classification, such as user profile information. While these models have obtained good results, it is unclear which features are actually the most valuable in the classification of fake news.

Considering Occam's Razor, or the principle that the simpler of two explanations is the one most likely to be true, we propose a multi-layer perceptron (MLP) model. The MLP model utilizes news content features from the source node of each graph and averages the tweet content features from succeeding nodes of each graph.

https://github.com/Aniket17Sakpal/Identifying-Misinformation-On-Social-Media

1 Introduction

The expansion of social media has caused an exponential increase in the spread of misinformation on these platforms. Whether it be about the presidential election, climate change, or Covid-19 vaccines, misinformation continues to spread rampantly across social media.

There are multiple approaches to detect misleading information; fact-checking is the most direct approach, which involves domain knowledge and is highly labor-intensive. Recently, Deep Learning has had promising results in the field of fake news detection. GNNs encode the propagation pattern of news on social media, utilizing both exogenous context such as text from tweets or shared news articles and endogenous content like the number of followers a user has. However, simpler models have not been explored. We propose to determine which features hold the most value in identifying misinformation in social media and to implement an MLP model to classify such false information.

2 Literature Review

"Fake news," or fabricated information that is patently false, has become a major phenomenon in the context of social media. It has received serious attention in a variety of fields, with scholars investigating the antecedents, characteristics, and consequences of its creation and dissemination [1]. There are two modes of problems currently being explored using neural networks. The first is fake news detection where it is identified whether a piece of news is truthful or falsified. While fake news detection can be accomplished through manually fact-checking, the sheet quantity of news content on social networks and the pace with which news spreads makes this method insufficient. Therefore,the application of Deep Learning models that can predict the validity of a particular news article is particularly useful for these platforms. The second problem surrounding fake news is the susceptibility of users. In other words, why do people fall for fake news, and how we can address this vulnerability? While it is equally important to find a solution to this problem, it is outside the scope of this project. Therefore, we shall focus on developing a new means of detecting fake news.

Fake news detection in social media using Deep Learning techniques has experienced many enhancements in recent years. GNNs, CNNs, Deep Markov Random Fields, and many more architectures have been proven successful. Existing approaches for fake news detection can be divided into three broad categories [2].

1. Content: Content-based approaches, which are used in the majority of works on fake news detection, rely on linguistic (lexical and syntactical) features which capture deceptive cues or writing styles [1,3,4,5,6]. The disadvantage of content-based approaches is that they can be bypassed by sophisticated content. In other words, the model can only recognize content in the specific formats which it has been trained on. Moreover, most linguistic features are language-dependent, limiting the generality of these approaches. These constraints make content-based approaches prone to adversarial attacks, or the intentional spread of misinformation which fools the model by malicious players.
2. Social Context: Social context features include user demographics - such as age, gender, education, and political affiliation [7,8] - and social network structure [9, 10] in the form of connections between user and user interactions. Connections between users are defined as friendship or follower/followee relations. Social context features help us to know the propensity of a user to spread fake news given their connections and demographics.
3. Propagation: There is empirical evidence that fake news propagates differently from real news. Thereby, the observed spreading patterns of news can be exploited for fake news detection [11]. Propagation patterns are also robust to adversarial attacks.

Graph Neural Networks (GNNs) can be incorporated to combine all three categories to detect fake news. The benchmark approach which we seek to improve upon with a simpler, MLP solution is User Preference Aware Fake News Detection (UPFD) [15]. UPFD uses the fact that a user is more likely to spread a piece of fake news when it confirms their existing beliefs or preferences. A users historical and social engagements, such as posts, provide rich information about their preferences toward news. These user-level attributes are used to generate Endogenous Preference Encodings. Attributes pertaining to the social media network, such as how news propagates through it and the content which is spread, are used to create Exogenous Context Encodings. These features are then combined and fed into the model to classify fake news.

2.1 Existing Model: GNN-CNN

The benchmark model whose performance we are seeking to surpass is a GNN-CNN using SAGEConv, a "general inductive framework that leverages node feature information (e.g. text attributes) to efficiently generate node embeddings for previously unseen data" [18]. This model architecture is shown below in Figure 3. The code for this model was available to us on CodeOcean [19]. We first sought to develop an understanding of how the GNN-CNN model works. We incrementally ran each class within the code and printed the size(s) of the relevant output to gain an understanding of the dimensionality of the architecture. Once the raw data is read from PolitiFact and GossipCop, BERT is used to extract the content features from every node and the profile features from each user. For example, there are 41054 nodes across all graphs in the Politifact data subset, and 300 content features are extracted per node. Additionally, there are 10 profile features extracted per user, such as followers count and friend count. Users which appear in multiple graphs are duplicated. These profile features are then concatenated to the content features and a 41054 x 310 dimensional tensor is created describing the nodes across all graphs in the subset. Next, this matrix is used to create a Pytorch Geometric data object for each graph and batches are created. Once we had an understanding of the code, we ran the UPFD GNN-CNN model using SAGEConv to replicate the published results.

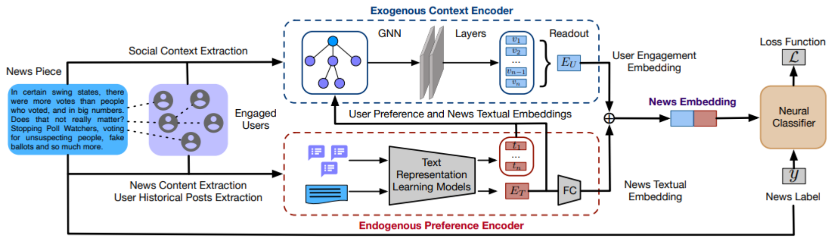

Figure 1: UPFD GNN-CNN Architecture [15]

3 Methodology and Experiments

As per the Literature Review, fake news detection can be categorized into three categories: news content, social context and propagation. The baseline model which we refer to in our experimentation is User Preference Aware Fake News Detection (UPFD), which uses a hybrid GNN-CNN model that incorporates all three of these categories. In an attempt to develop a simpler model which has equal or improved performance to this baseline, we went ahead with the approach detailed in this report.

3.1 Dataset

We use the FakeNewsNet dataset[13]. FakeNewsNet contains two subsets of social media posts collected from PolitiFact and GossipCop. There are samples of fake news and samples of real news from each source. FakeNews datasets contain the news content information, propagation graphs for each news article. We operate on the pre-processed dataset created by our baseline UPFD research paper [15]. The pre-processing steps converts all the information into feature embeddings using BERT and spaCy techniques[15]. Detailed information on the approach data-processing can be found in sections 2.1, 2.2 and 3.1.1 of the UPFD research paper. BERT is pre-trained on Wikipedia and book corpus, whereas spaCy is more generally pre-trained on instances of the English language. To understand the data-format, updated Politifact and GossipCop contain data on the node level of the propagation graphs for each news. A propagation graph for a news is shown in the following figure.

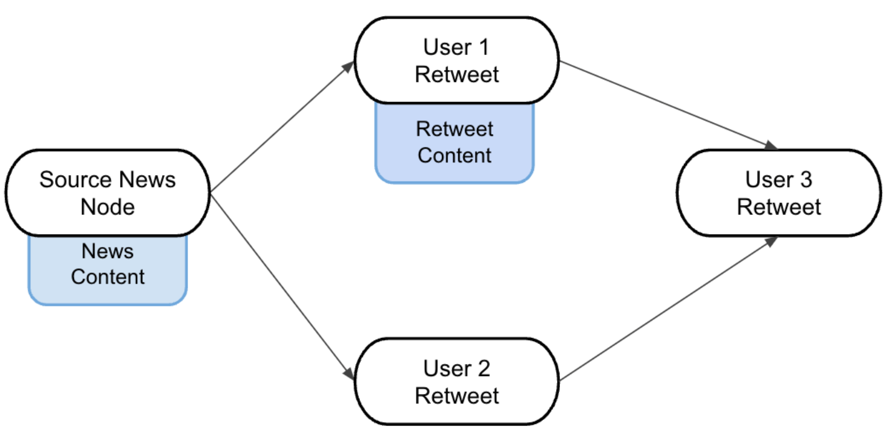

Figure 2: A propagation graph for a news to understand the level of data

The features are present for each node in the graph whose dimension will be 300 or 768 depending on the encoding technique (spaCy and BERT) used. There are following two types of Nodes:

1. Source News node: Source news node contains the features encoding the news content.
2. User nodes: User Nodes are the users who retweeted the news that contain the retweet content. Retweet content contains concatenation of its recent 200 tweets/retweets content, which includes the quoted tweets, the OP content of retweets, and the content of tweets from the user. Please refer to the Twitter API documentation[20] about how the crawler gets the user's recent tweets.

The dataset stats are as shown in Table 1.

Table 1: FakeNewsNet dataset summary

Data Source	Number of Graphs	Number of Fake News	Number of Total Nodes	Total Number of Edges	Average Number of Nodes per News Instance
PolitiFact	314	157	41,054	40,740	131
GossipCop	5464	2732	314,262	308,798	58

3.2 Baseline Model performance

UPFD-GNN baseline model achieves an accuracy of 84.6% and F1 score of 0.84 on the Politifact data subset with BERT embeddings. The same model achieves an accuracy of 97.2% and an F1 score of 0.97 on the GossipCop data subset using BERT embeddings.

Table 2: Baseline Model Results

Model	Dataset	Feature	Accuracy	F1 Score
UPFD-GNN Model	PolitiFact	BERT	84.6%	0.84
UPFD-GNN Model	GossipCop	BERT	97.2%	0.97

3.3 Evaluation Metrics

Four metrics will be used to evaluate our model's results. These metrics are accuracy, precision, recall, and F-measure. The first three of these metrics are calculated using the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) in the predictions. The F-score is then calculated using the precision and recall values. The following equations summarize these calculations:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

$$F - measure = \frac{2(precision * recall)}{precision + recall} \tag{4}$$

3.4 Ablation Studies

3.4.1 Hypothesis 1: Propagation pattern of the news contain major signal for fake news detection

Baseline GNN model relied heavily on the propagation of the news. Adhering to the principle of Occam's Razor, the belief that simpler explanations are more likely to be true. In order to understand the significance of the propagation, we considered all contents of data and transformed the graph network into a flattened layer (input to MLP model). This flattened data will be at the news

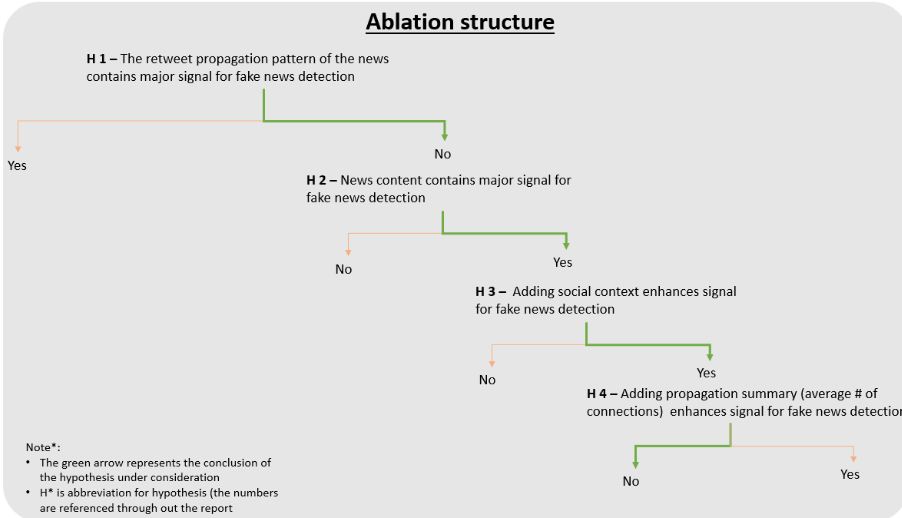

Figure 3: Ablation Structure [15]

Table 3: Hypothesis 1- Without Propagation Pattern

Model	Dataset	Feature	Accuracy	Precision	Recall	F-1 Score
MLP	PolitiFact	BERT	72%	0.92	0.58	0.72
MLP	GossipCop	BERT	89%	1.00	0.89	0.89
MLP	PolitiFact	spaCy	92%	0.87	0.96	0.92
MLP	GossipCop	spaCy	72%	0.92	0.58	0.72

level, consisting of the content of the news and social context of the users without consideration of propagation.

As observed in the above table, we ignored the propagation pattern in the data, yet we were able to achieve a near baseline performance. This raises a major question- where does the majority signal in the data lie?

We will validate the importance of propagation pattern in fake news detection through subsequent ablations.

3.4.2 Hypothesis 2: News content contains major signal for fake news detection

In the search of identifying the source of the signal in data, we will consider only the source node of the data (consisting of news content) as input to the MLP model discussed in the Hypothesis 1.

Table 4: Hypothesis 2- with news content

Model	Dataset	Feature	Accuracy	Precision	Recall	F-1 Score
MLP	PolitiFact	BERT	97%	1.00	0.94	0.97
MLP	GossipCop	BERT	76%	0.79	0.68	0.76
MLP	PolitiFact	spaCy	91%	0.93	0.88	0.91
MLP	GossipCop	spaCy	76%	0.75	0.75	0.76

As observed in the above table, while considering only the news content information, we were able to perform better than the baseline. Logical next step- include social context to check if signal power can be enhanced?

3.4.3 Hypothesis 3: Social context (in the form of retweet content) contains major signal for fake news detection

Next we wanted to test whether adding information about the retweet content information (i.e. social context) improves performance. For this we used the embeddings from retweets of each user for all news graph. When we flatten the data of nodes to feed it into MLP we run into the issue of node ordering. Essentially the if the node order of the graph is changed and then flattened, the class of news (true/fake) still remains the same, but the MLP will interpret it differently.

The easiest way to achieve node order in-variance is to average the embeddings per feature. Hence with the source node embeddings in place, we add the averaged embedding of all other nodes. This data is then fed into the MLP to calculate the evaluation metrics

Following is the result:

Table 5: Hypothesis 3 - with news + retweet content (social context)

Model	Dataset	Feature	Accuracy	Precision	Recall	F-1 Score
MLP	PolitiFact	BERT	94%	1.00	0.88	0.94
MLP	GossipCop	BERT	94%	0.98	0.89	0.94
MLP	PolitiFact	spaCy	91%	1.00	0.81	0.91
MLP	GossipCop	spaCy	95%	0.98	0.92	0.95

As observed above when we add the social context features the politics related news does not show any significant improvement in performance. This would mean that for politics related news the major signal is not in the change in content that happens across retweets.

For GossipCop news the social context provides a significant improvement. This means that for celebrity related news there is significant improvement due to the social context features.

3.4.4 Hypothesis 4: Propagation summary contains major signal for fake news detection

Next we wanted to test whether any propagation information contains any signal. For this we calculated the average degree per node (avg. number of connections per node)

The easiest way to achieve node order in-variance is to average the embeddings per feature. Hence with the source node embeddings in place, we add the averaged embedding of all other nodes. This data is then fed into the MLP to get the evaluation metrics.

Following is the result:

Table 6: Hypothesis 4 - with news + retweet + propagation summary

Model	Dataset	Feature	Accuracy	Precision	Recall	F-1 Score
MLP	PolitiFact	BERT	97%	1.00	0.94	0.97
MLP	GossipCop	BERT	95%	0.97	0.92	0.95
MLP	PolitiFact	spaCy	91%	1.00	0.81	0.91
MLP	GossipCop	spaCy	94%	0.98	0.89	0.94

As observed above when we add the propagation summary the performance does not change significantly. And hence we can conclude that propagation pattern does not contain significant signal for classifying misinformation

3.5 Dimensionality Reduction

The data after it was made node order invariant (by averaging across nodes) had 600 dimensions or 1500 dimensions with 314 rows for PolitiFact approximately 5000 rows for GossipCop. Since number of dimensions is greater than (or almost equal to) to the number of rows the data can be considered very high dimensional.

After the first set of experiments we were observing very high performance on the training dataset. But the high dimensionality raised a question on whether the data is causing the model to overfitting. To counter this we applied PCA, a linear dimensionality reduction technique to reduce the dimensions that cover 85 percentage of the variance. The PCA gave us 100 components to work with. These distilled 100 features where then fed into the MLP for further ablations

3.6 MLP Architecture Experiments

To improve model performance, we tried multiple MLP architectures by varying structure (Cylinder, Diamond), number of hidden layers, activation and standard regularization techniques like different dropout values and Batch Normalization etc. Here are some notable results (only for GossipCop dataset):

Table 7: Notable MLP Ablations

Data	Neurons	Regularization	Accuracy	F1 Score
Gossipcop + Bert	(2048,2048,2048,2048,2048)	BatchNorm1d, Dropout(p=0.3)	88%	0.88
Gossipcop + Bert	(1000,1500,2500,1500,1000)	BatchNorm1d, Dropout(p=0.3)	89%	0.89
Gossipcop + Spacy	(2048,2048,2048,2048,2048)	BatchNorm1d, Dropout(p=0.3)	93%	0.93
Gossipcop + Spacy	(1000,1500,2500,1500,1000)	BatchNorm1d, Dropout(p=0.3)	93%	0.93
Gossipcop+Bert	(2048,2048,2048,2048,2048,2048,2)	Batchnorm1d, Dropout(p=0.5)	90%	0.90
Gossipcop+Spacy	(2048,2048,2048,2048,2048,2048,2)	Batchnorm1d, Dropout(p=0.5)	88%	0.88

The final model after all the ablations on various architectures was as follows:

1. Five linear layers, each having 2048 output channels
2. ReLU Activation
3. Batch Normalization applied after each linear layer
4. Dropout of 0.3 applied after each linear layer
5. Learning Rate of 0.001
6. Adam used as the optimizer

4 Results

The UPFD GNN-CNN model using SAGEConv achieved an F1 score of 0.84 and an accuracy of 84.60% on the PolitiFact data subset, whereas the same model achieved an F1 score of 0.96 and an accuracy of 96.79% on the GossipCop data subset [15]. We were able to verify these result.

The MLP model, which takes as input all source node embeddings and the average of all subsequent node embeddings, achieved an F1 Score of 0.94 and an accuracy of 94% on the PolitiFact data subset. Therefore, our model was able to exceed the performance of the baseline when identifying fake political news. On the GossipCop data subset, the MLP model achieved an F1 Score of 0.93 and an accuracy of 93%. When classifying false information in celebrity news, the MLP model slightly underperformed as compared with the baseline GNN. BERT embeddings were used on the PolitiFact data subset, whereas spaCy embeddings were used for GossipCop. This choice of feature extraction method was based on earlier ablations, which demonstrated that BERT resulted in higher performance on political news and spaCy resulted in higher performance on celebrity news.

The slight discrepancy in the performance of the MLP model is intuitive based on the results of our ablations. These experiments showed that, for political news, it is the source node embeddings which have the highest predictive power. Alternatively, for celebrity gossip, the average of subsequent node embeddings is most indicative of whether a piece of information is real or fake. (and since we are averaging, there will be some amount of information loss)

The results of the MLP model are shown in the following table

In general,

Table 8: Results

Model	Dataset	Feature	Accuracy	F-1 Score
UPFD	PolitiFact	BERT	84%	0.84
UPFD	GossipCop	BERT	97%	0.97
MLP	PolitiFact	BERT	94%	0.94
MLP	GossipCop	spaCy	93%	0.93

5 Conclusion

In general, it is possible to develop an MLP model which outperforms existing GNN methods of fake news detection once it is known which features are most indicative of the validity of a news instance. This result is critically important because simpler models are both easier and cheaper for social networks to implement on a large scale.

Our inference through the ablation study is:

1. News content embeddings have strong predictive power for political news.
2. Retweet content embeddings have strong predictive power for celebrity news.

In the future, it would be valuable to test our approach on larger test data (the results achieved here utilized only a portion of FakeNewsNet) and try different non-linear methods of dimensionality reduction such as Variational Autoencoders (VAE).

Additionally, it is hypothesized that implementing an attention mechanism may further improve fake news detection. This conjecture should also be studied as an improvement on the MLP.

References

1. Molina, Maria D., S. Shyam Sundar, Thai Le, and Dongwon Lee. 2019. "'Fake News' Is Not Simply False Information: A Concept Explication and Taxonomy of Online Content." American Behavioral Scientist 65 (2): 000276421987822. https://doi.org/10.1177/0002764219878224.
2. Monti, Federico, Fabrizio Frasca, Davide Eynard, Damon Mannion, Michael Bronstein, Fabula Ai, and Usi Lugano. 2019. "Fake News Detection on Social Media Using Geometric Deep Learning." https://arxiv.org/pdf/1902.06673.pdf.
3. Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In Proc. Computational Approaches to Deception Detection, pages 7–17, 2016.
4. Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In Proc. Empirical Methods in Natural Language Processing, pages 2931–2937, 2017.
5. adia Afroz, Michael Brennan, and Rachel Greenstadt. Detecting hoaxes, frauds, and deception in writing style online. In Proc. IEEE Symp. Security and Privacy (SP), pages 461–475, 2012.
6. Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. arXivarXiv:1708.07104, 2017.
7. Kai Shu, Suhang Wang, and Huan Liu. Understanding user profiles on social media for fake news detection. In Proc. Multimedia Information Processing and Retrieval, pages 430–435, 2018.
8. Yunfei Long, Qin Lu, Rong Xiang, Minglei Li, and Chu-Ren Huang. Fake news detection through multi-perspective speaker profiles. In Proc. Natural Language Processing, volume 2, pages 252–256, 2017.
9. Kai Shu, Suhang Wang, and Huan Liu. Beyond news contents: The role of social context for fake news detection. In Proc. Web Search and Data Mining, 2019.
10. Kai Shu, H Russell Bernard, and Huan Liu. Studying fake news via network analysis: detection and mitigation. In Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining, pages 43–65. Springer, 2019.
11. Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. Science, 359(6380):1146–1151, 2018.
12. Sahoo, Doyen, Wang Hao, Shu Ke, Wu Xiongwei, Hung Le, Palakorn Achananuparp, Ee-Peng Lim, and Steven Hoi. 2019. "FoodAI: Food Image Recognition via Deep Learning for Smart Food Logging." https://doi.org/10.48550/arXiv.1909.11946.
13. Shu, Kai. 2022. "KaiDMML/FakeNewsNet." GitHub. March 6, 2022. https://github.com/KaiDMML/FakeNewsNet
14. Nasir, Jamal, Khan, Osama, Varlamis, Iraklis. "Fake news detection: A hybrid CNN-RNN based deep learning approach." ScienceDirect.com. Accessed March 7, 2022. https://doi.org/10.1016/j.jjimei.2020.100007
15. Dou, Yingtong, Shu, Kai, Xia, Congying, S., Philip, Lichao, Lu. "User Preference-aware Fake News Detection." 2021. https://arxiv.org/abs/2104.12259
16. "GREENER: Graph Neural Networks for News Media Profiling Anonymous ACL Submission." n.d. https://openreview.net/pdf?id=5hEUhYxlIY2.
17. Mu, Yida, and Nikolaos Aletras. 2020. "Identifying Twitter Users Who Repost Unreliable News Sources with Linguistic Information." PeerJ Computer Science 6 (December): e325. https://doi.org/10.7717/peerj-cs.325.
18. Hamilton, William L., Ying, Rex, Leskovec, Jure. "Inductive Representation Learning on Large Graphs." 2018. https://arxiv.org/pdf/1706.02216.pdf
19. Dou, Yingtong. GNNFakeNews. 2021. https://codeocean.com/capsule/7305473/tree/v1
20. Developer.twitter.com. 2022. GET statuses/user$_{t}imeline.[online]Availableat :< https://developer.twitter.com/en/docs/twitter-api/v1/tweets/timelines/api-reference/get-statuses-user_{t}imeline > [Accessed7May2022$

Appendix

```
├── gossipcop
│   ├── fake
│   │   ├── gossipcop-1
│   │   │       ├── news content.json
│   │   │       ├── tweets
│   │   │       │       ├── 886941526458347521.json
│   │   │       │       ├── 887096424105627648.json
│   │   │       │       └── ....
│   │   │       └── retweets
│   │   │               ├── 887096424105627648.json
│   │   │               ├── 887096424105627648.json
│   │   │               └── ....
│   │   └── ....
│   └── real
│      ├── gossipcop-1
│      │       ├── news content.json
│      │       ├── tweets
│      │       └── retweets
│              └── ....
├── politifact
│   ├── fake
│   │   ├── politifact-1
│   │   │       ├── news content.json
│   │   │       ├── tweets
│   │   │       └── retweets
│   │   └── ....
│   │
│   └── real
│      ├── poliifact-2
│      │       ├── news content.json
│      │       ├── tweets
│      │       └── retweets
│      └── ....
├── user_profiles
│               ├── 374136824.json
│               ├── 937649414600101889.json
│               └── ....
├── user_timeline_tweets
│               ├── 374136824.json
│               ├── 937649414600101889.json
│               └── ....
└── user_followers
│               ├── 374136824.json
│               ├── 937649414600101889.json
│               └── ....
└──user_following
                ├── 374136824.json
                ├── 937649414600101889.json
                └── ....
```

Figure 4: FakeNewsNet Folder Structure

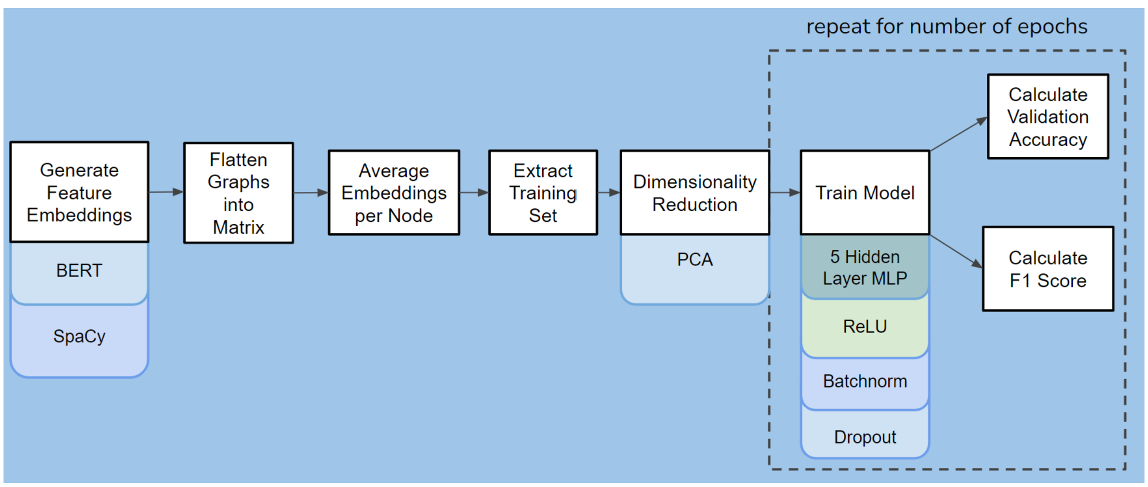

Figure 5: MLP Model Architecture