

# On the Feasibility of Using GANs for Claim Verification- Experiments and Analysis

Amartya Hatua<sup>a</sup>, Arjun Mukherjee<sup>a</sup> and Rakesh M. Verma<sup>a</sup>

<sup>a</sup>University of Houston, 4800 Calhoun Rd, Houston, TX 77004

## Abstract

The research on fact checking and claim verification has been explored using the Fact Extraction and VERification (FEVER) dataset. To supplement this research a Generative Adversarial Network (GAN) based model is used for fact checking on the FEVER dataset. The GAN based model generates synthetic data in an extended feature space of the FEVER dataset and gives leverage to new features. This synthetically generated data is further classified using positive-unlabeled (PU) learning considering supported facts as positive class and are added to the existing training dataset. Bidirectional Encoder Representations from Transformers (BERT) based encoding technique is used for both original and newly generated data to get the text's underline context. Due to the Information Gain in the synthetically generated data features, better performance is achieved in the fact checking and claim verification task. A thorough analysis of the model selection is done by comparing the GAN based model with BERT based classifier and other standard classifiers.

## Keywords

Fact checking, GAN, BERT, positive-unlabeled learning

## 1. Introduction

Fake news and misleading information are becoming a widespread phenomenon in our daily lives. Sometimes fake news is designed in such innovative ways, that it becomes difficult to separate the fake news from facts. To check the validity of such facts, other available resources are often used. To solve this problem, research on fact checking and claim verification is gaining a lot of attention. In most of the earlier research, this problem is treated as a classification task based on the patterns of the language [1, 2, 3], or the type of sources of the facts [4]. Sometimes external resources are used [5, 6] to supplement the classification task.

Fake claims and fake news often exhibit similar patterns and features. Earlier research attempted to find and use those patterns and features to perform fact checking tasks. In this research, we propose new features for fact checking and claim verification, and an attempt has been made to determine whether these new features help in fact checking tasks. To generate such synthetic features, generative models are used. In this research, a GAN [7] based model is used to create synthetic data, which helps to add new features to the existing dataset. The new features show an information gain which leads the model to produce better results. The class label of the newly generated data is given by the PU learning [8] method, where supported


---

ROMCIR 2021: Workshop on REDUCING ONLINE MISINFORMATION THROUGH CREDIBLE INFORMATION RETRIEVAL, April 01, 2021, Lucca, Tuscany, Italy (ONLINE EVENT)

✉ ahatua@central.uh.edu (A. Hatua); arjun@uh.edu (A. Mukherjee); rmverma@cs.uh.edu (R. M. Verma)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

claims or the true claims are considered as the positive class. The synthetically generated data is added to the existing training data so that the training feature space gains more features. BERT [9] plays a significant part in this experiment, as BERT is used for encoding of the training dataset. It helps to capture the underline context and the relationships between claims and evidence pairs. The diagrammatic representation of the proposed model is presented in Fig. 1.

## 1.1. Data

We use FEVER dataset for our fact checking and claim verification experiments. The FEVER dataset is open-source, and the research community is actively working on this dataset, so the FEVER dataset is selected for this research [10, 11, 12]. For every claim in the dataset, the evidence for the given claim and its class is given. Two of such (claim, evidence) pairs are presented in Table 1. In the FEVER dataset, the third category of data is also provided where the class of the claim is not mentioned as there is not enough information provided for or against the claim.

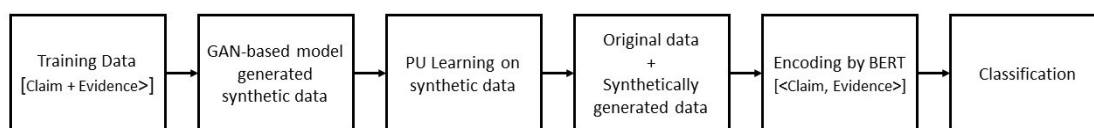
**Table 1**

Examples of claim verification

<b>Claim:</b> Tetris has sold millions of physical copies. <b>Evidence:</b> It was announced that Tetris has sold more than 170 million copies, approximately 70 physical copies and ... <b>Label:</b> True
<b>Claim:</b> Andy Roddick lost 5 Master Series between 2002 and 2010. <b>Evidence:</b> Roddick was ranked in the top 10 for nine consecutive years between 2002 and 2010, and won five Masters Series in that period. <b>Label:</b> False

Each data point in this dataset has three main elements: claim, evidence and label. For every claim there is one or multiple evidence from Wikipedia. The class label describes whether the evidence supports / refutes / do not provide enough information for the given claim. FEVER published different versions of the datasets. In this research we used FEVER 1.0 and FEVER 2.0 for training, validation and testing. FEVER 1.0 training dataset has 80,035 Supported claims, 29,775 Refuted claims, and 35,639 NotEnoughInfo claims. The FEVER 1.0 validation set and test set have 3,333 Support claims, 3,333 Refute claims, and 3,333 NotEnoughInfo claims respectively. FEVER 2.0 has 391 Support claims, 396 Refute claims, and 387 NotEnoughInfo claims respectively.

This research explores the possibilities of improvements in fact checking and claim verification tasks by adding extra features from synthetically generated data. Generating new data using GAN, which leads to improvement of the fact checking result, is novel to our knowledge. We also compare our result with prior research of Yang et al. [8] and other standard models like Long Short Term Memory (LSTM) [13], Convolution Neural Network (CNN) [14], Graph Neural Network (GCN) [15], Naive Bayes classifier [16], Support Vector Machine (SVM) [17], Random forest [18], Stochastic Gradient Boosting (SGB) [19].



**Figure 1:** Block diagram of the GAN based model

## 2. Related Work

Significant work has been done on fact checking with most of it focusing on the text’s linguistic patterns, the source of the fact, and occasionally some external information used as a supplement for a given claim.

**Internal features.** Throughout this research, we find that linguistic features are the most important and widely used features for this task. For example, in [1], H. Rashkin et al. analytically characterized the language of fake political news and determined political news’s truthfulness. A similar type of study is done by Ramy Baly et al. in [2] on multiple news resources. Apart from linguistic features and patterns; sentiment, mood, and other psychological factors can help to identify fake claims or news. In [3], Pérez-Rosas et al. showed a method of fake news detection using psycholinguistic features of the news. Using Linguistic Inquiry and Word Count software (LIWC), they extracted essential words in text that are part of psycholinguistic categories. These words are then used to identify fake news.

**Addition of external sources or metadata.** Some research shows that, other than the internal features of the text of the fact/news, external resources and meta data can play a vital role in identifying the truthfulness of a fact or claim. One such approach is in [4], where the meta-data is combined with the data to achieve a significant improvement in fake news detection. Furthermore, external sources include additional information about the news, user interaction, public opinion, etc., and helps in assessment of news or claims [5]. Moreover, previous work also proposed a method to find the truthfulness of news by collecting information from multiple related sources [6, 20]. These sources can either support or refute each other. In [6] Ravali et al. proposed a novel method of modeling the correlations between different sources of news and applied that in determining the truthfulness of the news. Similarly, Jeff Pasternack et al. introduced a generalized fact-finding framework in [20], which incorporates uncertainty in the information extraction of claims from documents, attributes of the sources, the degree of similarity among claims, and the degree of certainty expressed by the sources as additional information into the fact-finding process. For fact checking on inconsistent sources and information, Liang Ge et al. [21] proposed a two-step procedure. It calculates the degree of information consistency and identifies the underlined common reason for the inconsistency and calculates a consistent score for each item. Similarly, Q. Li et al. [22] proposed an optimization framework in which truths and reliable sources are considered as two sets of unknown variables and the framework aims to minimize the deviation between the truths and the multi-source observations. A generalized algorithm called TruthFinder is proposed in [23], which utilizes the information of different related web sites to perform fact checking.

**PU Learning.** In recent works on fact checking and claim verification, Yang et al. [8] proposed a GAN based PU learning technique on the FEVER dataset for claim verification task. This work is used as a baseline for this research, and their results are compared in this research. For our research we used FEVER as it is an open-source dataset for fact checking and claim verification. FEVER dataset contains evidence taken from Wikipedia pages, and the claims are constructed by crowdsourcing [24].

**Pipeline.** In some of the earlier fact checking research using the FEVER dataset, researchers followed a pipeline used in the base model [24]. The pipeline consists of identifying relevant wiki articles, extracting the appropriate supporting sentences, and determining the truthfulness of the claim. In prior research, document selection is made using different techniques. Some of the important techniques used by the Wikipedia API are the DrQA framework for document detection; token matching techniques and the AllenNLP framework. The second phase of the pipeline, i.e., sentence selection is done using TF-IDF based method, sequence matching neural network, and some ranking based methods. The third and final step of the task, i.e., classification is done using TF-IDF based approach in the base model. Neural network based models, different natural language inference models, and deep learning based models are also used later.

### 3. Model

As presented in Figure 1, the model’s pipeline consists of three central units: i) GAN, ii) PU Learning, iii) Classification unit. In this experiment the Leak GAN [25] model is used; for PU learning, a bagging based method is used with Random forest [26]. The BERT [9] encoding based classifier is used for the classification in the final step. A brief description of each of the units is given below.

#### 3.1. Leak GAN

A GAN [7] model consists of a generator ( $G$ ) unit and a discriminator ( $D$ ) unit. The generator unit generates synthetic data while the discriminator distinguishes between true data and synthetic data. The goal of the generator unit is to generate data that are very similar to the original data so that the discriminator cannot identify them as synthetic data. The optimization of GAN is done by  $D$  and  $G$  alternatively via a min-max game and  $p(z)$  denotes a simple distribution, such as  $\mathcal{N}(0, 1)$ .

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

Generation of long sentences using GAN is a challenging task, Leak GAN [25] is specially designed to generate long sentences and to produce good results on standard natural language processing based tasks [25]. Hence Leak GAN is used in this research. Leak GAN follows the standard adversarial training principle but in the standard method the scalar guiding signal to the generator unit becomes relatively less informative when GAN attempts to generate a long sentence. Leak GAN overcomes this problem because the discriminator unit leaks information of its own high-level extracted features to the generative unit for further guidance, which eventually helps Leak GAN to generate long sentences. A hierarchical reinforcement learning

(RL) architecture [27] is used in the generator unit to incorporate the leaked information from the discriminator unit. Although Leak GAN produces excellent results due to its hierarchical RL based architecture, it takes a long time to generate sentences. Another GAN model, LaTextGAN [28] is also used in this research to compare the results with Leak GAN. As LaTextGAN is not an RL based model, it converges faster than Leak GAN.

### 3.2. PU Bagging

Once the synthetic data is generated using the Leak GAN model, the next task is to label these data using the PU Learning method. PU bagging is one such PU learning method. In the first step, a training set is created combining all positive data points (here the supported claims in FEVER dataset) with a random sample from the unlabeled points, with replacement. In the second step, the dataset is used as a “bootstrap” sample to train a classifier considering unlabeled data points belonging to the negative class. In the next step, the classifier is used to classify the unlabeled data points that were not included in the random sample or the OOB (“out of bag”) points and record their assigned class. These three steps are repeated many times and finally each unlabeled point is labeled with the class it was assigned to the maximum number of times. While assigning the class label of the generated data, only ‘SUPPORTED’ and ‘REFUTED’ class labels are assigned, ‘NOT ENOUGH INFORMATION’ class label is ignored.

### 3.3. BERT

In this research BERT plays a significant role as an encoder and a classifier. BERT Huggingface [29] pretrained model is used to encode the training and test datasets. It ensures the consistency of the underlying semantic context and corresponding relations between claim, evidence and classes. The [SEP] token is used to separate the claim and the evidence. Some claims have multiple supporting statements, in such cases multiple claim evidence pairs are created. For a particular claim, its corresponding evidence is concatenated separately. For example, there is a data point with the following claim ( $C$ ), evidence ( $E$ ), and label ( $L$ ):  $[C, E < e_1, e_2, e_3 >, L]$ . The input data format to the BERT model will be:  $x = [< C; e_1, L >, < C; e_2, L >, < C; e_3, L >]$ .

## 4. Experimental Setup

### 4.1. Experiments

**GAN Based Models:** In this experiment, two different GAN based models are used to generate claims synthetically. One of the GAN based models (Leak GAN) uses reinforcement learning (RL), while the other model (LaTextGAN) does not use RL based models. Using each GAN based model, a total of 10,000 synthetic data points (claims) are generated. The length of the sentences generated by Leak GAN is longer than LaTextGAN. On average, the length of the generated sentences by LeakGAN is 20. Whenever LaTextGAN generates a long sentence, it exhibits a tendency to repeat some words multiple times. This problem is not observed in the synthetic data generated by the LeakGAN.

**PU Learning with Bagging using Random forest:** PU Learning with Bagging using Randomforest: The unlabeled synthetic data is labeled using the PU Learning technique. In the PU learning technique, a Random forest classifier is used for the initial step, and a bagging approach is followed to label the synthetic data in the final step.

**BERT Transformer:** Huggingface BERT pretrained transformer is used as tokenizer for the training, validation and testing dataset. The vocabulary size of the pretrained model is 30522 and the size of the hidden layer is 768. Later the pretuned model is fine tuned to classify the claims. BERT is used as an encoder for training (original and sythetic data), validation and testing datasets.

**Classifiers:** Other than BERT based classifiers some of the standard machine learning and deep learning classifiers are also used for the classification task. These classifiers are: GCN, LSTM, CNN, SVM, Random forest, Naive Bayes, SGB. The deep learning classifiers like GCN, LSTM and CNN are implemented. For GCN pointwise mutual information between words is calculated to generate the graph. To implement the CNN five kernels of sizes 2, 3, 4, 5 and 6 are used. For LSTM, the input data is encoded using GloVe [30]. The learning rate and batch size for these 3 models are 0.001, 64, respectively. In machine learning models, Categorical Naive Bayes classifier, SVM with RBF kernel, and Random forest is implemented. The Random forest is equipped with 1000 trees and entropy is used as supported criteria for the information gain. The SGB model utilizes hinge loss and L2 penalty. The deep learning models are implemented using PyTorch [31], and the Scikit learn library [32] is used for machine learning models.

**Evidence Sentence Selection:** The evidence for the synthetically generated sentences are selcted from the Wikipedia database [10] using cosine similarity [33]. In this case we have selected one evidence for every synthetically generated sentence.

**Table 2**

Examples of synthetically Leak GAN generated claims and respective evidence from the FEVER dataset

---

**Claim:** Colantoni Entertainment Singing dealt Wisin densely expertise Crooks Carthaginians toxoplasmosis Dextroamphetamine 313,000 1204 orphanages Illuminate Protestant Hackers Gupta 1917.

**Evidence:** Andrea Colantoni, quarter-finalist in Men’s Low-Kick at WAKO World Championships 2007 Belgrade-67 kg.

**Label:** False

---

**Claim:** Jackson’s singing citing blast Austrian Coppola direct 100 gruelling The screams Tick Mycroft FX Bacsinszky Orci MacShayne Castlevania Unkrich.

**Evidence:** “All in Your Name” is a song written and performed by Barry Gibb and Michael Jackson

**Label:** True

---

## 5. Results

The result of the above experiments on FEVER 1.0 and FEVER 2.0 is presented and discussed in this section. The precision, recall, and F1 score for all the models are reported in this section. All the results are compared with the previous research by Yang et al. The Leak GAN based model generates 10,000 data points, which are added to the initial training data. PU Learning

**Table 3**

Examples of synthetically LaTextGAN generated claims and respective evidence from the FEVER dataset

<p><b>Claim:</b> Steven Angele certified Kroll, burglary presidential Texas Lactobacillales Pont finding jumped knight population, Switzerland, person person person person.</p> <p><b>Evidence:</b> The 1877 Stevens Ducks football team represented Stevens Institute of Technology in the 1877 college football season.</p> <p><b>Label:</b> False</p>
<p><b>Claim:</b> Lionel messi reached breakdown now Barcelona “Bonet” philosophical Afghanistan Championship, adherents hook abandoned Kentuck Kentuck Kentuck Kentuck Kentuck Kentuck Kentuck</p> <p><b>Evidence:</b> Lionel Messi scored the winning goal in the fifth minute of the second half of extra time, securing Barcelona’s record sixth trophy for the 2009 calendar year.</p> <p><b>Label:</b> True</p>

is applied to these 10,000 synthetically generated data, and 6,823 data points are classified as Supported claims, and the rest 3,177 data points are classified as Refuted claims, and we ignore the NotEnoughInfo class. Hence, the new training dataset has 86,858 Supported claims, 32,952 Refuted claims, and 35,639 NotEnoughInfo claims. The test and validation dataset has 3,333 Supported claims, 3,333 Refuted claims, and 3,333 NotEnoughInfo claims. Training data for both FEVER 1.0 and FEVER 2.0 is the same; the test data different. All the experiments are repeated five times.

**Table 4**

Result of FEVER 1.0

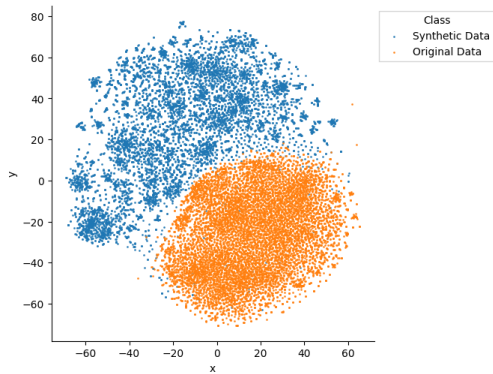
Classifiers	FEVER 1.0 Dataset		
	Precision	Recall	F1 Score
BERT Classifier	0.45 ± 0.011	0.44 ± 0.010	0.44 ± 0.009
Leak GAN Based Classifier	0.65 ± 0.003	0.64 ± 0.006	0.63 ± 0.003
LaTextGAN Based Classifier	0.41 ± 0.008	0.36 ± 0.016	0.30 ± 0.009
Graph Convolutional Network	0.45 ± 0.015	0.44 ± 0.013	0.44 ± 0.013
SVM	0.53 ± 0.013	0.42 ± 0.013	0.38 ± 0.013
Naive Bayes	0.41 ± 0.016	0.34 ± 0.014	0.24 ± 0.015
Random forest	0.33 ± 0.011	0.33 ± 0.010	0.28 ± 0.011
SGD	0.31 ± 0.023	0.22 ± 0.022	0.27 ± 0.023
LSTM	0.45 ± 0.003	0.42 ± 0.004	0.004 ± 0.004
CNN	0.46 ± 0.012	0.44 ± 0.011	0.43 ± 0.012
Yang et al. result	0.61	0.58	0.60

It can be observed in Table 4 and Table 5 that the performance of the Leak GAN based model is better than all other models on both the datasets. The F1 mean scores for the two datasets are 0.63 and 0.51. The Leak GAN based model performed better than the earlier published results and the results of the other standard classifiers.

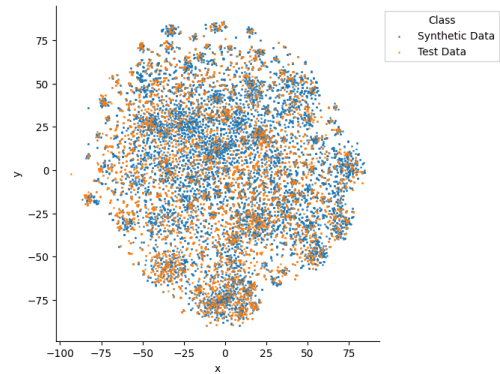
We have implemented two GAN based models, LaTextGAN and Leak GAN for synthetic data generation. Analysis is done on the synthetically generated data from both the GAN models.

**Table 5**  
Result of FEVER 2.0

Classifiers	FEVER 2.0 Dataset		
	Precision	Recall	F1 Score
BERT Classifier	$0.46 \pm 0.013$	$0.44 \pm 0.014$	$0.44 \pm 0.013$
Leak GAN Based Classifier	$0.52 \pm 0.023$	$0.51 \pm 0.019$	$0.51 \pm 0.021$
LaTextGAN Based Classifier	$0.42 \pm 0.02$	$0.39 \pm 0.019$	$0.39 \pm 0.019$
Graph Convolutional Network	$0.43 \pm 0.023$	$0.39 \pm 0.013$	$0.37 \pm 0.016$
SVM	$0.40 \pm 0.019$	$0.37 \pm 0.022$	$0.35 \pm 0.019$
Naive Bayes	$0.33 \pm 0.030$	$0.22 \pm 0.023$	$0.27 \pm 0.025$
Random forest	$0.33 \pm 0.014$	$0.26 \pm 0.017$	$0.29 \pm 0.015$
SGD	$0.30 \pm 0.025$	$0.22 \pm 0.029$	$0.26 \pm 0.027$
LSTM	$0.43 \pm 0.028$	$0.40 \pm 0.039$	$0.39 \pm 0.032$
CNN	$0.41 \pm 0.021$	$0.38 \pm 0.011$	$0.37 \pm 0.018$



**Figure 2:** t-SNE Plot of Original & Synthetic Data by LaTextGAN



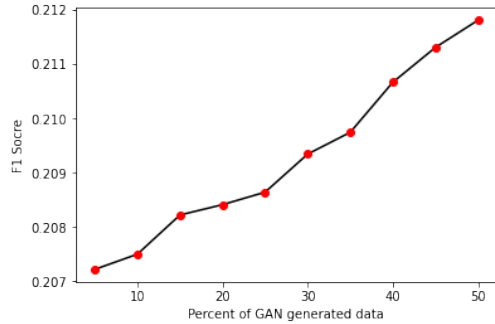
**Figure 3:** t-SNE Plot of Original & Synthetic Data by Leak GAN

In Figure 2 and Figure 3, the distribution of the original data and newly generated data (using LaTextGAN and Leak GAN) can be observed. To plot this graph, the dataset is encoded using BERT and t-SNE algorithm is used for visualization. The t-SNE plot is done using perplexity value 30, number of iterations 1000, and learning rate 200. In this visualization, 10,000 randomly selected data points are used from both the original and the synthetic datasets.

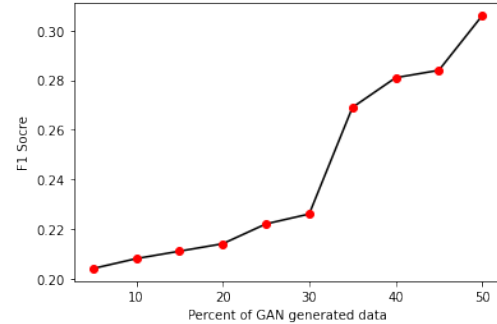
To understand the statistical significance of the synthetically generated dataset, paired t-Test is also performed on the same 20,000 data points. Here we considered the null hypothesis as there is no difference in the distribution between the two datasets. The paired t-Test produces a t-value of 1.811, and the respective p-value is 0.072 for the data generated by the Leak GAN based model. The result of the paired t-Test supports the null hypothesis with a p-value of 0.072. Moreover, the new features added by the synthetic data helps in information gain of 0.020 bits. On the other hand, the paired t-Test using the synthetic data from LaTextGAN produces a t-value of -3.37, and the respective p-value is 0.000737. The information gain is 0.008 bits. The information gain and result of t-Test shows that the synthetic data generated by Leak GAN has similar distribution to the original data and gives more information to the entire training dataset



than the dataset generated using LaTextGAN. In Table 2, it can be observed that the performance of the Leak GAN based model is better than all other models on both the datasets. The reason behind the better performance of the Leak GAN based classifier model is its synthetic data generation capability. All the analytical results discussed here helps us to draw this conclusion.



**Figure 4:** F1 Score of SVM Classifier on gradually increasing dataset



**Figure 5:** F1 Score of BERT Classifier on gradually increasing dataset

To confirm the synthetic data’s effectiveness on the improvement of the F1 score, we performed an empirical analysis with SVM and BERT classifier. To arrive at this analysis, a subset of the training dataset (original + synthetic) is used for training, and the test dataset provided by FEVER 1.0 is used for the testing. 10,000 original training data points are randomly selected for this experiment, where the ratio of the classes is kept the same as the entire dataset. Initially, the percentage of the synthetic data is 5% of the training data, and in the next 10 steps, the volume of synthetic data is gradually increased to 50% of the total training data. BERT based classifiers and SVM are applied to this subset of the training dataset, and F1 scores are recorded and plotted in Figure 4 and Figure 5. It can be observed from Figure 4 and Figure 5 that as the percentage of the synthetic dataset is increasing in the subset of the training dataset the F1 score is also increasing. This proves enhancement in effectiveness of classifiers due to addition of synthetic data with the original data.

## 6. Conclusion

This research attempts to employ the effectiveness of the synthetic data generation capability of the GAN. We proposed a GAN based model with PU bagging for fact checking and claim verification. The model is capable of generating synthetic data, which eventually helps the fact checking task. This research also discusses the distribution of the newly generated data and its statistical significance toward information gain and classification results. The entire research is carried out on FEVER 1.0 and FEVER 2.0 datasets, and the result of the proposed model is compared with several standard classifier’s results and previous results. In the future, a similar set of experiments can be carried out on other publicly available standard datasets to test this proposed model’s effectiveness.

## Acknowledgments

Research was supported in part by grants NSF 1838147, NSF DGE 1433817 and ARO W911NF-20-1-0254. Verma is the founder of Everest Cyber Security and Analytics, Inc. The views and conclusions contained in this document are those of the authors and not of the sponsors. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

## References

- [1] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, Y. Choi, Truth of varying shades: Analyzing language in fake news and political fact-checking (2017) 2931–2937.
- [2] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, P. Nakov, Predicting factuality of reporting and bias of news media sources, arXiv preprint arXiv:1810.01765 (2018).
- [3] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, R. Mihalcea, Automatic detection of fake news, arXiv preprint arXiv:1708.07104 (2017).
- [4] W. Y. Wang, "liar, liar pants on fire": A new benchmark dataset for fake news detection, arXiv preprint arXiv:1705.00648 (2017).
- [5] K. Popat, S. Mukherjee, J. Strötgen, G. Weikum, Where the truth lies: Explaining the credibility of emerging claims on the web and social media (2017) 1003–1012.
- [6] R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, D. Srivastava, Fusing data with correlations (2014) 433–444.
- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, *Advances in neural information processing systems* 27 (2014) 2672–2680.
- [8] F. Yang, E. Dragut, A. Mukherjee, Claim verification under positive unlabeled learning, *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (2020).
- [9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [10] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fact extraction and verification (fever) shared task, arXiv preprint arXiv:1811.10971 (2018).
- [11] J. Thorne, A. Vlachos, Adversarial attacks against fact extraction and verification, arXiv preprint arXiv:1903.05543 (2019).
- [12] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fever2.0 shared task (2019) 1–6.
- [13] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (1997) 1735–1780.
- [14] S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back, Face recognition: A convolutional neural-network approach, *IEEE transactions on neural networks* 8 (1997) 98–113.
- [15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Transactions on Neural Networks* 20 (2008) 61–80.

- [16] D. D. Lewis, Naive (bayes) at forty: The independence assumption in information retrieval (1998) 4–15.
- [17] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, *Advances in neural information processing systems* 9 (1996) 155–161.
- [18] M. Pal, Random forest classifier for remote sensing classification, *International journal of remote sensing* 26 (2005) 217–222.
- [19] J. H. Friedman, Stochastic gradient boosting, *Computational statistics & data analysis* 38 (2002) 367–378.
- [20] J. Pasternack, D. Roth, Making better informed trust decisions with generalized fact-finding (2011).
- [21] L. Ge, J. Gao, X. Li, A. Zhang, Multi-source deep learning for information trustworthiness estimation (2013) 766–774.
- [22] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, J. Han, Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation (2014) 1187–1198.
- [23] M. Wan, X. Chen, L. Kaplan, J. Han, J. Gao, B. Zhao, From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach (2016) 1885–1894.
- [24] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, Fever: a large-scale dataset for fact extraction and verification, *arXiv preprint arXiv:1803.05355* (2018).
- [25] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, J. Wang, Long text generation via adversarial training with leaked information, *arXiv preprint arXiv:1709.08624* (2017).
- [26] C. Li, X.-L. Hua, Towards positive unlabeled learning for parallel data mining: a random forest framework (2014) 573–587.
- [27] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, K. Kavukcuoglu, Feudal networks for hierarchical reinforcement learning, *arXiv preprint arXiv:1703.01161* (2017).
- [28] D. Donahue, A. Rumshisky, Adversarial text generation without reinforcement learning, *arXiv preprint arXiv:1810.06640* (2018).
- [29] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing (2020) 38–45. URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [30] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation (2014) 1532–1543.
- [31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems* 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher,

- M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [33] A. Huang, et al., Similarity measures for text document clustering 4 (2008) 9–56.