Gradient-based Adversarial Attacks on Current Fake-News Detection Systems

Jesper Jensen
Linköpings Universitet
Linköping, Sweden
jesperjen98@gmail.com

Kevin Scott
Linköpings Universitet
Linköping, Sweden
kevinscott98@gmail.com

***Abstract*—In this paper, the goal is to analyse how effective adversarial attacks can be on a fake-news detection system. The system under attack in this report is a neural network designed to identify fake news using style-based features. It is implemented using the BOW model and is trained on the ISOT dataset. It is attacked by different adversarial examples that generate input noise that in essence, tries to change a fake-news article to fool the neural network into believing it is real. By changing the amount of noise as well as how it is generated and applying it to the article itself, we were able to reduce the neural network-based model accuracy.**

***Index Terms*—Adversarial attacks, Bag of Words, Neural Networks, Fake-News.**

I. Introduction

Deep learning models are used in many different fields showing outstanding performance but recent research has shown that they can be exploited. Adversarial examples have been shown to be a weakness of these models and while there exists many research papers around this topic, very few are aimed towards evaluating the robustness of models used for natural language processing (NLP) and in particular simple Bag-Of-Words (BOW) models. This paper highlights the practicality of adversarial attacks against a fake news detection system and presents methods of sparse adversarial attacks with cardinality constraints on the BOW model.

II. Research questions

A. How can adversarial examples be used to attack a fake news detection network based on the BOW model.

B. Will the fast gradient sign method work on a fake news detection network based on the BOW model?

C. What level of noise is needed for an adversarial attack to work successfully and how will it be perceived by a human?

III. Limitations

We will not research fake news detection neural networks which are trained on images or amount of shares/like on a post. We will only research networks which are trained on sequential text data. The work on the report including the implementation of the network and attacks had a time limitation 160 hours.

IV. Background

This section covers concepts which lay the foundation of knowledge that the reader might need to understand the report.

A. Fake news

Fake news can be defined as information that is meant to mislead the reader into believe something that is not true. There are numerous reasons for this, one being for monetary gain by selling something through advertisements. The more prevalent reason seems to be harming the reputation of a person or entity. Fake news articles and posts are being spread through different mediums of social media with Facebook being the biggest offender.

B. Fake news detection

Detecting fake news can be done using multiple different metrics and these metrics are generally split into four different types of fake-news detection. These four types are knowledge-, propagation-, source- and style-based fake-news detection.

Knowledge-based detection works by extracting facts from the given article and comparing it to a knowledge database. This works well against articles that lie about statistics or fabricate quotes from well known entities.

Propagation-based detection works by assessing how fast the news is spread across social media. This works since fake news generally spreads much faster, further and more widely [1].

Source-based detection checks the news source to see how trustworthy the article is. For instance an article from The Guardian has a higher trust factor than a news article posted from a random Facebook group.

Style-based detection tries to assess if the intention of the article is to mislead the reader or not. This is based on the fact that fake-news articles are usually written in such a style that the reader is encouraged to share the text with other people. This style can be represented by different features such as containing emotionally loaded words, informality through netspeak or using certainty terms such as *never* and *always* [2].

C. Supervised learning

Supervised learning is a field in machine learning that utilizes methods for learning from examples. Formally, the idea is to teach an unknown function f to map an input x to an output y by looking at examples of input-output pairs *(x,y)*, so-called training data. Depending on the classification-task the output can either be discrete or continuous.

D. Bag of Words model

Bag of Words (BOW) is a model which is used to represent a text by simplifying its representation. The model simplifies the text by turning it into a multiset (also called a bag) that only contains the words used in the text and the number of times each separate word appears. The model does not care for grammar, sentence structure or in which order the words appear which removes layers of complexity. The words and the number of times they appear are then used as input features for a machine learning model to interpret. When training a machine learning algorithm using this model a vocabulary is gradually built upon the words from the training data. When executing a machine learning algorithm using this model, if a word shows up that is not in the vocabulary, it is disregarded as an input feature. There exists other more advanced natural language processing techniques such as Word2vec where this flaw does not exists, but as stated they are also more complex.

E. BERT model

Bidirectional Encoder Representations from Transformers (or BERT) is an NLP machine learning technique developed by Google [3]. BERT can learn contextual relations of words in a text. A directional model may read the input from left-to-right or the other way around, but BERT instead reads the entire text at once. This means that it can better learn the context of words in meanings by looking at both the words before it (to the left) and after it (to the right). Bert can be used for many things by fine tuning to a downstream task, one case being sentiment analysis where it can extract subjective information from text.

F. Adversarial examples

Adversarial examples are inputs deliberately designed to deceive a machine learning model. Generally, this is done by adding perturbation to an input sample. Techniques for generating perturbation can be categorised as either being white-box or black-box. White-box techniques require access to the models internal parameters including its gradient calculation. Black-box techniques are model-agnostic and generate perturbation based on the models input and output. The attacks presented in the following sections are white-box attacks and focus on maximizing the models loss function $J(x^{adv}, y)$ by some adversarial example x^{adv} and ground-truth label y.

G. Fast Gradient Sign Method

The fast gradient sign method (FGSM) [4] finds an adversarial example by adding perturbation according to the element-wise sign of the loss gradient w.r.t. the original input x as

$$x^{adv} = x + \epsilon \cdot \text{sign}(\Delta_x J(\theta, x, y)) \tag{1}$$

The perturbation amount is denoted ϵ and is some positive constant which constraints x^{adv} under the L_∞ norm bound $\left\| x^{adv} - x \right\|_\infty \leq \epsilon$. The attack assumes linearity by crafting adversarial examples from linearly extrapolating on the differentiated data point x. In this regard it is presumably the optimal attack against a *linear* binary classification model. But even though this linear assumption does not hold, the attack has shown to be very effective against nonlinear deep neural networks. With this assumption, the gradient only needs to be calculated once which makes the attack computationally efficient rather than optimal.

H. Iterative Fast Gradient Sign Method

An iterative approach can be used to combat the aforementioned shortcoming. The iterative fast gradient sign method (I-FGSM) [5] is an extension of FGSM, where instead of taking a single step in the direction of the gradient sign, the method will take T steps with the stepsize α and recalculate the gradient at each step. More specifically, the attack is initiates by setting $x_0^{adv} = x$ and then for each iteration $t < T$ computing

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(\Delta_x J(x_t^{adv}, y)) \tag{2}$$

The number of iterations T and stepsize α is decided heuristically. By convention (conventionally) the stepsize α is a fraction of ϵ and the number of iterations is a multiple of ϵ/α. To satisfy the L_∞ norm bound, each intermediate result x_t^{adv} can either be clipped within the allowed ϵ-vicinity or the stepsize α can be set to ϵ/T.

I. Momentum Iterative Fast Gradient Sign Method

In the article *Boosting Adversarial Attacks with Momentum* [6], the authors present a method of accelerating the I-FGSM attack by incorporating momentum. The gradient at each step in Eq. 2 is replaced with the accumulated velocity vector in the gradient direction as

$$g_{t+1} = \mu \cdot g_t + \frac{\Delta_x J(x_t^{adv}, y)}{\left\| \Delta_x J(x_t^{adv}, y) \right\|_1} \tag{3}$$

The velocity vector is zero-initialized $g_0 = 0$ and there is one additional parameter for the decay factor μ. Henceforth, this method will be referred to in its abbreviated form; MI-FGSM.

J. Adversarial Patch

An adversarial patch [7] is a targeted universal adversarial attack. The patch is derived from the gradient function $\Delta_x J(x + \text{patch}, \hat{y})$ where $\hat{y}$ is the targeted class. The patch is input-independent and exploits salient features to maximise the target loss.

V. Method

This section covers how the data-set was processed and how each attack got evaluated by the model on a high level.

A. Dataset

The ISOT [8] dataset was used to train and test the BOW model. The dataset contained in total 44 898 articles labeled either real or fake. There were 21 417 articles labeled real and 23 481 labeled fake. Other datasets considered were Kaggle, Liar and the FNC-1 dataset. The FNC-1 as well as the Liar dataset used multiclass-classification which would add undesirable complexity. The Liar dataset contained short statements instead of long articles and had been labeled real or fake using fact checking which was not suitable for style-based detection methods.

B. Data preprocessing

The texts for each article in the dataset were preprocessed in order for the BOW model to get more accurate results. Stopwords such as: *he*, *it*, *i* and *an* were filtered out and excluded since they do not add much substance to an article. Punctuation marks like for instance: *question marks*, *full stops*, *commas* and *colons* were removed in order to capture only words. The python package *nltk* [9] was used for identifying the stopwords and punctuation marks as well as removing them from the input data. A stemming process was used to retrieve the root form (or also called word stem, hence the name stemming) of all the words. This ensures that if the words *flying*, *flied* and *fly* appeared in the text they would all count as three instances of the word *fly* since that is the root form. This was done by the built in stemmer called SnowballStemmer in the *nltk* library. All articles were preprocessed in the same way and then randomly either put in the test- or training data. This was done so that there were the same amount of articles in the test- and training data respectively..

C. Feature scaling

Feature scaling or feature normalization is a common technique to reduce randomness in the results by making sure each feature can contribute to the model's prediction. Min-max normalization is one of the simplest methods used for this purpose and will normalise each feature by dividing it with the maxima feature in the particular data sample. This maxima feature is referred to as the normalization factor and one can invert the normalisation by multiplying the normalized feature vector with the normalization factor.

D. Model under attack

Based on current research on fake news detection systems, an appropriate model was found to be subject for attack in this research paper. The model under attack is based of a Brazilian research study [10]. The data-set was changed and parsing was changed from Portuguese to English to achieve higher accuracy for the ISOT data-set. The model in its essence was unchanged and is a simple feed-forward neural network with 2 fully connected layers. The model has four parameters

$$\theta \in \left\{ \begin{array}{ll} \mathrm{w}_0 \in \mathbf{R}^{32x1000}, & \mathrm{b}_0 \in \mathbf{R}^{32} \\ \mathrm{w}_1 \in \mathbf{R}^{1x32}, & \mathrm{b}_1 \in \mathbf{R}^{1} \end{array} \right\} \tag{4}$$

The model expects an input X in R^{1000} where 1000 corresponds to the number of input features. A trailing batch dimension may be added for handling multiple inputs at once and in this case the model will perform batch matrix multiplication as opposed to regular matrix multiplication. The models forward function f is given as

$$f(X) = layer_2(layer_1(X)) \tag{5}$$

where

$$\begin{aligned} layer_1(X) &= ReLU(w_0^T X + b_0) \\ layer_2(X) &= \sigma(w_1^T X + b_1) \end{aligned} \tag{6}$$

The models binary prediction is obtained by rounding the output from f. The binary cross-entropy loss function J was used to quantify the models accuracy.

$$J(X;\theta) = f_\theta(X) \cdot \log \hat{y}_i + (1 - f_\theta(X)) \cdot \log(1 - \hat{y}_i) \tag{7}$$

where $\hat{y}_i$ is the correct label for the i-th training sample. The model was trained for 50 epochs on all 22446 articles from the training set using the *Adam* optimizer with a learning rate and weight decay set to 0.001.

E. Problem formulation

Each attack method is considered as an optimisation problem, specifically the task of maximising

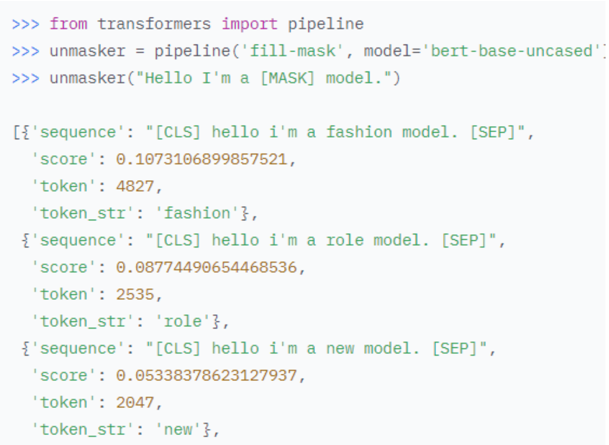
$$\underset{\|\delta\|_\infty \leq \epsilon}{\text{maximize}} \; J(f_\theta(X + \delta), \hat{y}) \tag{8}$$

The perturbation δ is generated by the attack itself and is confined by the maximum perturbation amount ϵ under the L_∞ constraint. For single-shot attacks like FGSM, the perturbation amount is directly correlated with the maximum perturbation amount. But for the other attacks (Iterate FGSM and MIFGSM), the perturbation amount can depend on multiple predefined parameters which will effect the perturbation amount. These attacks are still comparable with attacks like FGSM since they are both constrained under the maximum perturbation amount and finally the perturbed input is clipped between the normalised range to prevent illegal input to the model. To calculate the accuracy of the model over the maximum perturbation amount, the following procedure is used

1) Iterate over each test sample
2) calculate the models accuracy over a set of epsilon values

F. Deceiving the reader

All attack methods discussed in this paper are meant to attack the BOW model but does not take into account that the article will later on be read by a human. For instance, adding words randomly to an article might fool the BOW model into believing it is a real article, but that might not be the case for the reader. In order to avoid this problem a text representation function was created for the I-NGSM (iterative negative gradient sign method) attack. The I-NGSM attack works by iteratively removing a set number of words from the article that contain fake-news features. The job of the text representation function would then be to find suitable replacements for these words that are also not in the BOW vocabulary. The reason that the replacement can not exist in the vocabulary is that this could change the model prediction. The word replacement is done using the BERT model, more specifically the BERT-base uncased model. An instance of the given word in the article is removed and then the BERT model computes a list of words that may fit the sentence context ranked by the accuracy according to BERT. If every single word in the list is also in the vocabulary the word is not replaced and just removed instead. This was implemented using the python *transformers* [11] library using the pipeline function with BERT-base-uncased as model input.



```
>>> from transformers import pipeline
>>> unmasker = pipeline('fill-mask', model='bert-base-uncased'
>>> unmasker("Hello I'm a [MASK] model.")

[{'sequence': "[CLS] hello i'm a fashion model. [SEP]",
  'score': 0.1073106899857521,
  'token': 4827,
  'token_str': 'fashion'},
 {'sequence': "[CLS] hello i'm a role model. [SEP]",
  'score': 0.08774490654468536,
  'token': 2535,
  'token_str': 'role'},
 {'sequence': "[CLS] hello i'm a new model. [SEP]",
  'score': 0.05338378623127937,
  'token': 2047,
  'token_str': 'new'},
```

Fig. 1. Example of word replacement with the BERT model.

VI. Results

This section presents the results from experiments described in our method and findings from running the experiments.

A. Model baseline

The model ran for 50 epochs which resulted in an accuracy of 92 percent in the normalised domain and 96 percent in the unnormalised domain for the training and test data.

B. Gradient-based adversarial attacks

The gradient is used to minimise the loss when training the model by calibration of weights and biases. Since these calibrated parameters are fixed after training, the input is the only variable that can effect the loss. Gradient-based attack methods use the gradient of the loss w.r.t the input $\Delta_x J$to change the input in a way that will increase the loss, which is the opposite of what is being done during training.

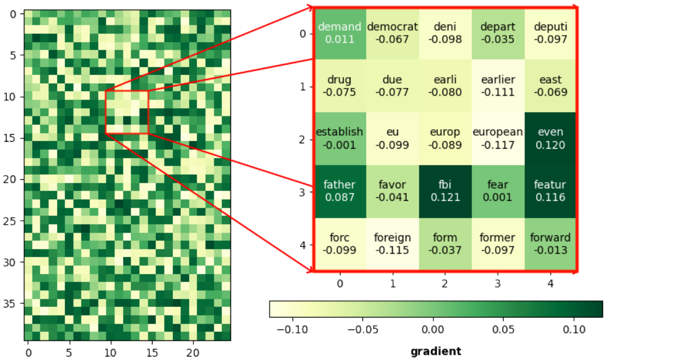


Fig. 2. The gradient $\Delta_x J$ for each word in the input space for one sample displayed as a 40x25 image (left), along with a 5x5 inset with feature labels and gradient value (right)

As the gradient in Fig 2 suggests, some features will have a higher influence on the models prediction that other features.

C. Adversarial Attacks with normalized features

The FGSM and MI-FGSM attack against the normalized BOW model were implemented as they are described in literature [4] [6]. The parameters for MI-FGSM was decided heuristically to be $\alpha = \epsilon/20$ stepsize and $T = 2 \cdot \epsilon/\alpha$ steps with the decay factor $\mu = 1$. Each intermediate result for the iterative method was clipped within the allowed perturbation vicinity. Finally the perturbed input x^{adv} from both attacks were clipped within the normalized range $0 \leq x^{adv} \leq 1$ to make sure no illegal input was produced.

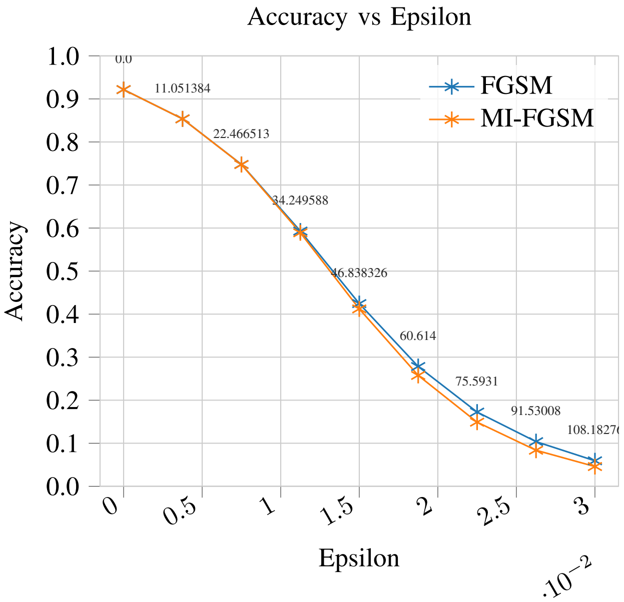


Fig. 3. Average accuracy of the normalized BOW model under attack by two different methods and different perturbation amount ϵ varying from 0 to 0.03 with a granularity of 0.00375. Each data point is accompanied with the average total amount of words replaced in the articles. The accuracy was computed for all 22 446 test samples.

The classification accuracy for adversarial news articles are outlined in Fig. 3. As shown in the figure 3 above, the models accuracy decreases when the perturbation amount ϵ increases. With large enough ϵ, the relative input won't matter. A more noteworthy observation is that the curve is not linear even though the ϵ values are linearly spaced. When ϵ is circa 0.015 the models prediction is effectively as good as randomly predicting if a news article is fake or not. At this data point when translating the perturbation back to its pre-normalized form as described in (V-C), the perturbation will add or remove $\sim$ 48.95 words on average from an article. Given that the model has 1000 input features, each feature will either increase or decrease its word count by $\sim$ 0.04895 on average. This poses a problem when translating the normalized perturbations back into the (unnormalized) word counts since the perturbation must land on a cardinal number for it to preserve input validity from multiplicity in the multiset of words.

A potential way of solving this problem is by rounding the word count (up or down) to the nearest integer, but even with a well-crafted rounding policy, the minimum amount of perturbation will be ± 1 (due to using the sign). Even though adding or removing one word for every input feature will likely mislead the classifier, it will inadvertently fail in misleading the victim who is reading the news article.

Based on the aforementioned arguments, there exist reasons not to add perturbation to every single input feature. The following sections present informal gradient-based attacks against the unnormalized model that preserves multiplicity. The attacks use the same principle of using the sign of the gradient w.r.t the cost function, but are more selective in which input features that need to be adjusted.

D. Negative Gradient Sign Method

Removing certain words from an article can be done under the assumption that the selected words are greater than or equal to the perturbation amount. The negative gradient sign method (NGSM) masks all input features with a positive word count and performs an indirect partition on the gradient of the cost function $\Delta_x J(x, y)$ in such a way that the gradient of the element in the kth position will be in its final sorted position in *ascending* order. The indices of the kth element and all smaller elements before it are considered the most appropriate to remove. The perturbation equates to decreasing all k elements with the corresponding indices with some perturbation amount ϵ. A formalised description is given as

$$x^{adv} = x - \epsilon \text{ where } x \geq \epsilon \ \wedge \ x \in \operatorname*{argmin}_k \Delta_x J(x, y) \tag{9}$$

The lowest accuracy over words removed was observed when setting k to a small fraction of the number of input features.

The classification accuracy for news article with $k \cdot \epsilon$ words removed is outlined in Fig 4 above. As the figure 4 shows, each attack converges asymptotically at some accuracy. This is likely due to the attack only considering k input features for removal and the word count for these k features will be zeroed with large enough perturbation amount ϵ since the word count cannot fall below zero.

To address this shortcoming, an iterative method with momentum is proposed were the k best input features are selected for removal at multiple time-steps. The method initiates by setting the momentum $g_0 = 0$ and first adversarial input $x_0^{adv} = x$ and then for each iteration $t \leq \epsilon$ begin by updating the accumulated velocity vector in the gradient direction as

$$g_{t+1} = \mu \cdot g_t + \frac{\Delta_x J(x_t^{adv}, y)}{\left\| \Delta_x J(x_t^{adv}, y) \right\|_1} \tag{10}$$

followed by updating the adversarial example (similar to Eq. 9) as

$$x_{t+1}^{adv} = x_t^{adv} - 1 \text{ where } x_t^{adv} > 1 \wedge g_{t+1} \in \operatorname*{argmin}_k g_{t+1} \tag{11}$$

The stepsize is set to -1 which is the minimum perturbation amount in the unnormalized domain. The number of iterations can conveniently be set to ϵ since there is not other usage of ϵ in Eq. 11 and increasing ϵ will implicitly increase the perturbation amount from using a constant stepsize. A notable difference between Eq. 11 and Eq. 9 is how the constraint on which words can be removed is being relaxed in Eq. 11. Through observation, the momentum term in Eq. 10 had a minuscule impact on increasing the effectiveness of the attack and the decay rate was set to $\mu = 0$, subverting it to a basic iterative method.

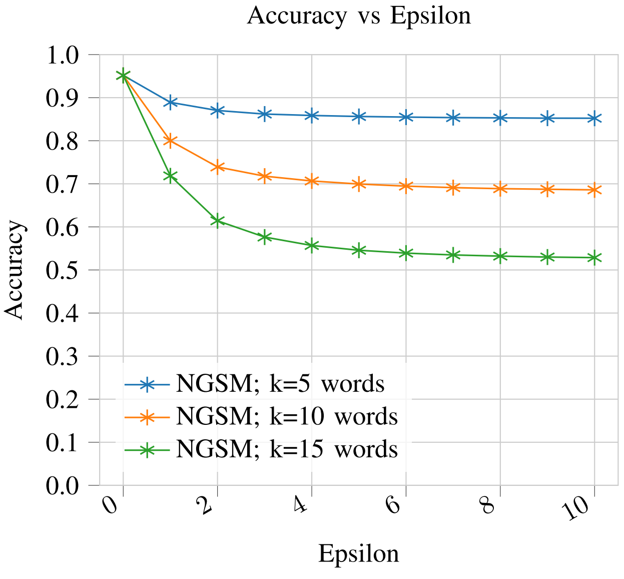


Fig. 4. Average accuracy of the unnormalized BOW model under attack by NGSM with different perturbation amount ϵ varying from 0 to 10 with a granularity of 1 and k set to either 5, 10 or 15. The accuracy was computed for all 22 446 test samples.

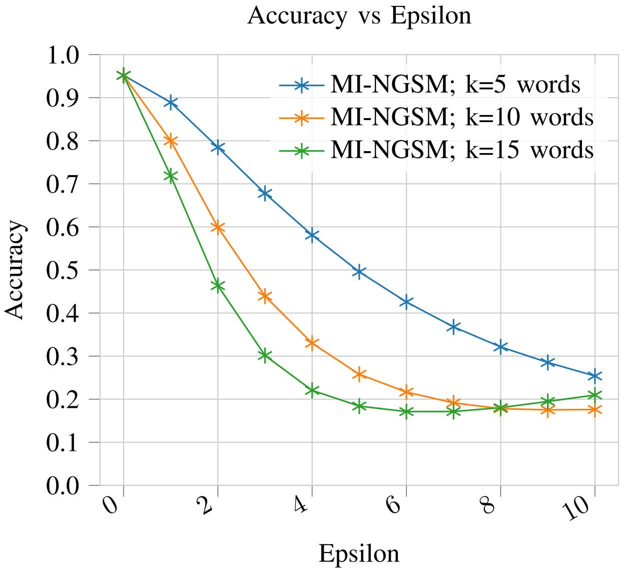


Fig. 5. Average accuracy of the unnormalized BOW model under attack by MI-NGSM with different perturbation amount ϵ varying from 0 to 10 with a granularity of 1 and k set to either 5, 10 or 15. The accuracy was computed for all 22 446 test samples.

The iterative method in Fig. 5 shows a clear improvement over NGSM in Fig. 4. The same asymptotic characteristic in Fig. 4 occurs for the iterative approach in Fig. 5, but the horizontal asymptot will occur at a lower accuracy which is likely because the iterative method adapts to which words should be removed and will zero out the word counts a lot later. It is also observable that both methods are exactly equally efficient when the perturbation amount ϵ equals 1.

E. Positive Gradient Sign Method

Adding certain words is the opposite of removing certain words from an article with the advantage that all input features can be used without any restrictions. The positive gradient sign method (PGSM) uses the same principles as NGSM, but replaces operation in Eq. 9 with their dual operations as

$$x^{adv} = x + \epsilon \text{ where } x \in \underset{k}{\operatorname{argmax}}\, \Delta_x J(x, y) \qquad (12)$$

The k parameter was set in the same way as NGSM and MI-NGSM to be a small fraction of the total number of input features.

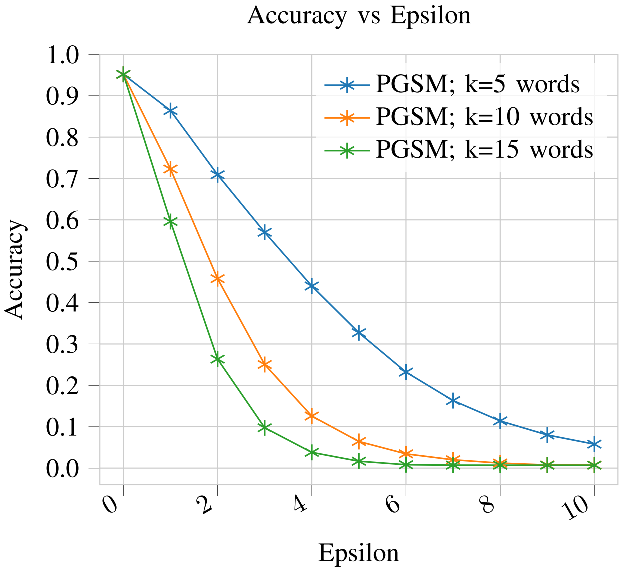


Fig. 6. Average accuracy of the unnormalized BOW model under attack by PGSM with different perturbation amount ϵ varying from 0 to 10 with a granularity of 1 and k set to either 5, 10 or 15. The accuracy was computed for all 22 446 test samples.

The classification error rate in Fig. 6 surpasses the two attacks in the normalized domain shown in Fig. 3. Adding 75 words to an article drops the models classification accuracy to 0%.

Moreover, there is no substantial benefit by applying the perturbations iteratively, as seen with NGSM.

F. Adversarial Patch

Adding the same *positive* perturbation independent of the models input is conceptually the same as an adversarial patch. The goal is to find a subset of input features that consistently decrease the models accuracy on all test samples. The proposed method is to select K words with the highest gradient amongst all test samples labelled fake and multiply them with the perturbation amount ϵ. This is done in two steps, The first step involves accumulating the occurrence of each word appearing in the K words with the highest (positive) gradient.

Algorithm 1 Adversarial Patch Logits

```
Input: A classifier $f$ with loss function $J$; a test sample $x$;
Input: unique words $K$;
Output: $logits$ (Increment for one test sample)

$x^* = x$
$y = 1$
for $k = 0$ to $K$ do
    Increment $logits_k$ with $+1$ where $\underset{x^*}{\operatorname{argmax}}\, \Delta_x J(x^*, y)$
    Remove $\underset{x^*}{\operatorname{argmax}}\, \Delta_x J(x^*, y)$ from $x^*$
end for
return $logits$
```

The logits of each fake news test sample are accumulated as described above.

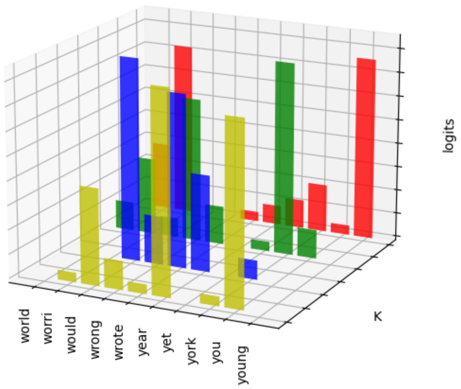


Fig. 7. The accumulated logits when using 10 input features and 4 unique words for the adversarial patch.

With $K = 4$ the accumulated logits may look like the figure above. The next step is to take the word with the max logit along the K axis which will give K words with the highest gradient. Finally, the perturbation amount is added to these words.

How effective the adversarial patch is will be determined by the two parameters. By increasing K the attack will consider more words to be adding perturbation to, while increasing the perturbation amount ϵ will increase the weight of all K words.

A heuristic approach was used to decide both parameters by comparing the accuracy of the model on the test data from varying both parameter's in the range $[0, 10]$ by the 2-fold Cartesian product. With $K = 7$ and $\epsilon = 7$ the models accuracy subsided to 0.08 percent, giving a much worse performance than randomly predicting if the article is fake. The 49 words used in the adversarial patch is displayed in the following quote below

> *friday, friday, friday, friday, friday, friday, friday, london, london, london, london, london, london, london, moscow, moscow, moscow, moscow, moscow, moscow, moscow, nov, nov, nov, nov, nov, nov, nov, reuter, reuter, reuter, reuter, reuter, reuter, reuter, some, some, some, some, some, some, some, washington, washington, washington, washington, washington, washington, washington*

Since the BOW model does not preserve semantic meaning for words in an article, the words in the adversarial patch can be rearranged and placed anywhere in the targeted fake news article.

G. Deceiving the reader

The results were mixed when running the text representation function. For some sentences, words were swapped out effortlessly, and the sentences fit the context of the article. For others, they ended up being completely drivel. The believed reason for this is examined under the discussion section. Examples of the result can be seen in table I and II.

VII. Discussion

This section covers the discussion of the results in relation to the given research questions.

A. Comparison of attacks

The non-targeted attacks on the pre-normalized model had varying success in decreasing the models accuracy with the least perceptible perturbations. The iterative negative gradient sign method VI-D gave best results when measuring the models accuracy over the number of words removed from the article. The real benefit of only removing words from an article is revealed when taking into account the reader of the article. Theoretically, it is possible to craft an adversarial news article without changing the linguistic semantics of the article.

When only considering the l_∞ norm bound for all non-targeted attacks, the positive gradient sign method VI-E had the best performance against the pre-normalized model. This is likely because the attack having a higher degree of freedom when choosing which words are to be added to an article. The adversarial patch had only a slightly worse performance than PGSM which might suggest that the data distribution of the training and test set are similar with small covariance. The words in the adversarial patch VI-F are interesting because we cannot know for certain why those particular words are chosen. According to the data-set description [8], all 12 600 true news articles are taken from reuter.com and this might explain why the word *reuter* occurs in the adversarial patch VI-F. For the other words like *friday* and *nov* there is no suitable explanation and the words might be occurring from an effort in memorising the ISOT dataset.

B. Deceiving the reader

As seen in the results deceiving the BOW model has been more successful than deceiving the reader. We believe there are five main reasons that caused the text representation function to not always work optimal.

1) Preprocessing not accounting for proper nouns and numbers: Proper nouns such as names of cities, countries or people are not removed in the preprocessing. This resulted in that a name of a country could be chosen to be swapped out in the text. This was bad since most names do not have synonyms and the BERT model would not be able to fill in the blank with a good replacement, and instead just solve it by selecting the name of a another country. The same problem comes with numbers as they would be replaced by other random numbers which makes the sentence lose its context to the article. This problem could be solved by doing a more intensive preprocess.

2) No suitable word replacement exists: If all the examples of words given from the BERT model already exist in the vocabulary the word in question is simply removed instead of being replaced with a new one. This works fine in some cases when for example and adjective is removed since it does not add much to the text, but when an verb or a noun is removed the sentence might lose its meaning entirely.

3) Does not account for where in the sentence the word is replaced: If the first word in the sentence is replaced the word returned by BERT is lowercase. This is an result of that the BERT model we chose (BERT-base-undercased) is only trained on uncapitalized words. This could be be solved by either swapping to the BERT-base-cased model which is trained on both uncapitalized and capitalized words or manually checking if the word is the first in the sentence and capitalizing it.

4) Only replaces one word at a time: Since only one word is removed at a time if a sentence has many of its words chosen to be removed it might lose its connection to the text. This is because the second replacement will be dependent on the result of the first replacement and so on. This means that the more words that are replaced the more important is it that replacements that comes first keep the context of the text. One way to solve this problem could be not swapping out the words until BERT has given examples for every word. This would remove the dependency between word replacements.

5) A bug in the implementation: Sometimes when two words next to each other are swapped out, double hashtags appear in the word. This bug could be solved by debugging and stepping through an iteration of word replacement in the text to find out what is causing it.

These problems are believed to not be that hard to solve but due to the time constraints of the course they can not be solved in time.

TABLE I
SENTENCES THAT WOULD LOOK NORMAL TO THE READER.

Before	After
The couple is licensed and has a **care** contract with the Catholic Charities **Community** Services.	The couple is licensed and has a **sponsorship** contract with the Catholic Charities **relief** Services.
It is **never really clear** just how **much truth** the journalists **receive** because the **news** industry has **become** complacent.	It is **only very obvious** just how **poor** the journalists **are** because the **tabloid-paper** industry has **grown** complacent.
The messages that it **presents** are shaped by corporate powers who often spend millions ...	The messages that it **broadcasts** are shaped by corporate powers who often spend millions ...

TABLE II
SENTENCES WOULD LOOK STRANGE FOR THE READER.

Before	After
Their spokespeople channel political **ideas** toward electoral cycles and ...	Their spokespeople channel political toward electoral cycles and ...
They are aborted into quick visual scans of an **image** or rapidly associated with a few words, like **black lives matter**, and then shared on social media.	**these** are aborted into quick visual scans of an **object** or rapidly associated with a few words, like **how ##dy**, and then shared on social media.
In Iran, 5 U.S. **prisoners** were **released**, with 4 of them **making** their **way** to **Germany via** Switzerland.	In Iran, 5 U.S. **5s** were **built**, with 4 of them their **export** to **france or** Switzerland.

REFERENCES

[1] S. Vosoughi. "The spread of true and false news online". In: *Science* 6380 (2018), pp. 1146–1150.

[2] Zhixuan Zhou et al. "Fake News Detection via NLP is Vulnerable to Adversarial Attacks". In: (2019). DOI: 10.5220/0007566307940800. URL: http://arxiv.org/abs/1901.09657.

[3] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[4] J. Shlens I. J. Goodfellow and C. Szegedy. "Explaining and harnessing adversarial examples". In: *ICLR* (2015). URL: https://arxiv.org/pdf/1412.6572.

[5] I. J. Goodfellow A. Kurakin and S. Bengio. "Adversarial examples in the physical world". In: *ICLR* (2017). URL: https://arxiv.org/pdf/1607.02533.

[6] Y. Dong et al. "Boosting Adversarial Attacks with Momentum". In: *cs.LG* (2018). URL: https://arxiv.org/pdf/1710.06081.

[7] T. Brown et al. "Adversarial Patch". In: *cs.CV* (2018). URL: https://arxiv.org/pdf/1712.09665.

[8] ISOT Research Lab. *ISOT Fake News Dataset*. 2017. URL: https://www.uvic.ca/engineering/ece/isot/datasets/fake-news/index.php.

[9] NLTK Project. 2021. URL: https://www.nltk.org.

[10] Gabriel Nogueira. *BR Fake News Detection*. Apr. 2021. URL: https://github.com/Talendar/br_fake_news_detection.

[11] The Hugging Face Team. 2020. URL: https://huggingface.co/transformers/.