ConTheModel: Can we modify tweets to confuse classifier models?

Aishwarya Ram Vinay[0000−0002−4540−1356], Mohsen Ali Alawami[0000−0002−1658−9716], and Hyoungshick Kim✉[0000−0002−1605−3866]

Department of Electrical and Computer Engineering, Sungkyunkwan University,
Suwon, South Korea
{aishwarya, mohsencomm, hyoung}@skku.edu

Abstract. News on social media can significantly influence users, manipulating them for political or economic reasons. Adversarial manipulations in the text have proven to create vulnerabilities in classifiers, and the current research is towards finding classifier models that are not susceptible to such manipulations. In this paper, we present a novel technique called ConTheModel, which slightly modifies social media news to confuse machine learning (ML)-based classifiers under the black-box setting. ConTheModel replaces a word in the original tweet with its synonym or antonym to generate tweets that confuse classifiers. We evaluate our technique on three different scenarios of the dataset and perform a comparison between five well-known machine learning algorithms, which includes Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and Multilayer Perceptron (MLP) to demonstrate the performance of classifiers on the modifications done by ConTheModel. Our results show that the classifiers are confused after modification with the utmost drop of 16.36%. We additionally conducted a human study with 25 participants to validate the effectiveness of ConTheModel and found that the majority of participants (65%) found it challenging to classify the tweets correctly. We hope our work will help in finding robust ML models against adversarial examples.

Keywords: Machine Learning, Social Media, Adversarial Examples, Tweets.

1 Introduction

Various social media platforms are used by a large number of population worldwide for communication as it is easily accessible. Statistics show that in 2020, approximately 3.6 billion people were using social media, and this number would increase by almost another billion by 2025 [6]. All is fine unless otherwise, when what is passed off as “news” on social media is often disinformation. Contrary to real news, fake news develops stories instead of reporting facts. Last October, a new law was passed in Singapore, which bans the spreading of false information. This law does so by allowing the government to instruct popular online social platforms to either remove or rectify the statements that are not in accordance with the general public's welfare [22].

Adversarial manipulations have taken the world by storm. To look at the basic, fake news has created havoc in people's lives since times immemorial, taking an even more serious turn during the 2016 US presidential election. When something as non-trivial as fake news can shake the world, one can imagine the severe impact of perturbations in the text. Even though adversarial learning has helped improve many models' performance, adversarial examples seem to attack state-of-the-art machine learning and deep learning models. The main agenda behind adversarial manipulations (a.k.a maliciously crafted inputs) is to fool the classifiers in producing a wrong output, and these manipulations have found their way into image classification [17, 27], speech recognition [4, 5], reinforcement learning [2], and natural language classification problems [3, 12, 30] amongst others. An attacker can arbitrarily create perturbations in the input which are usually unnoticeable to humans but can confuse the classifier by degrading its performance. As adversarial examples are a matter of security, recent research has focused on identifying machine learning classifiers that can be easily fooled and attacked. To contribute to the ongoing research, we propose a new approach called ConTheModel to study the extent to which machine learning classifiers can be confused using the well-designed adversarial examples, which results in the degradation of the models' F1 scores.

Without loss of generality, we develop an algorithm to replace the words with their synonyms or antonyms provided by NLTK corpus's WordNet package [21]. We train the ML models on source tweets collected from Twitter and labeled by professional journalists [31]. During the testing phase, we test the models' performance on (1) original samples (source tweets) and (2) successfully generated adversarial examples that are semantically and syntactically similar to that of the original samples. To show the effectiveness of ConTheModel, we used different categories of machine learning models, including Support Vector Machine (SVM), Naive Bayes (NB), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and Multilayer Perceptron (MLP) in our experiments. The test results show that the classifiers' performance is degraded after modification, indicating that generated adversarial examples can confuse models without changing the context of tweets. Through our experimental analysis and by considering all scenarios, we found the maximum drops in F1 scores of 13.18%, 16.36%, 12.34%, 12.09%, and 11.18% for RF, XGBoost, MLP, SVM and NB respectively. We also evaluate our approach by conducting a human study to show the modified tweets' effectiveness on human participants.

Our main contributions are as follows:

1. We propose an efficient approach called ConTheModel to investigate the feasibility of confusing classifier-based detection methods on tweets.
2. Our developed ConTheModel approach focuses on considering synonyms or antonyms as a replacement of the target words in tweets to confuse the classifiers.

3. We evaluate the performance of ConTheModel on the PHEME dataset with nine events using five different classifiers, and low results show the validity of our approach at confusing the classifiers under three different scenarios.

The rest of the paper is organized as follows: In Section 2, we provide the most recent papers related to our work, then we describe our system overview and algorithm in Section 3. The evaluation setup and information about the classifiers used are presented in Section 4. The details of the experimental results are shown in Section 5. The performance of human evaluation is discussed in Section 6. Finally, our conclusions and suggestions for future research are presented in Section 7.

2 Related work

In recent years, adversarial examples have been explored in various domains such as image, video, text classification, and others. The generation of these adversarial examples have been widely studied to bypass the neural networks and many other machine learning classification models. Primarily, it found its way into deep neural networks' classification algorithms exploring into image recognition domain. In this domain, prior work generated examples via optimization techniques by setting in motion unnoticeable changes to the pixel values until the distortion limit was reached [9, 18, 20, 26]. Also, Kurakin et al. [17] showed that machine learning systems could be misclassified if the examples were also constructed in the physical world and perceived through a camera.

Contrary to the image recognition domain, even the smallest change in the natural language domain is noticeable. In the information age, the internet community has become the hotspot for generating fake news due to the ease of creating and spreading this type of news. Consequently, it has garnered researchers' attention in finding efficient mechanisms to detect fake news [13]. Moreover, classifiers are susceptible to attacks where the attacker causes perturbations in the input, resulting in classifiers' misclassification. Adversarial examples have been created in a white-box setting such as the HotFlip method in [7] that performed character editing operations (such as flip, insertion, and deletion) as well as word-level operations against classifiers at both character-level and word-level deep neural network classifier, respectively. Assuming a black-box setting, Alzantot et al. [1] developed a gradient-free optimization algorithm to generate adversarial examples inspired by the natural selection process. They minimized the number of changes in the sentence and performed the attack on the IMDB dataset (Sentiment analysis task) and SNLI (Textual entailment task). In contrast to our work, Jia et al. [11] considered only antonyms from WordNet, and they worked on confusing the models into giving an incorrect answer by inserting sentences to the paragraphs on which questions have been asked. Papernot et al. [24] has shown that by replacing minimal words, an adversary can mislead the categorical and sequential recurrent neural networks without much importance given to grammatically correct adversarial examples. Similar to the generative adversarial network (GAN) [8], Ma et al. [19] proposed an approach where they promote information campaigns via uncertain and conflicting voices in twitter claims. The work in [12] considered a synonym based attack under black-box setting against various target models, including the powerful pre-trained BERT. Vijayaraghavan et al. [29] uses a reinforcement learning frame to generate adversarial examples that operate over characters and words of an input text.

Machine learning algorithms have succeeded in various tasks, amongst which is classification. The relationship between 13 different ML models on new and unseen rumors was investigated, and an ensemble solution of three models, Random Forest, XGBoost, and Multilayer Perceptron were created that overall produced a good F1 score [14]. However, many machine learning classifiers are susceptible to perturbations caused by adversarial attacks and are also vulnerable in various domains. Hu et al. [10] proposed an algorithm based on GAN, which generated adversarial malware examples that bypassed the machine learning-based detection models considering it as a black-box. Similarly, the vulnerability of machine learning classifiers in malware detection by generating five applications is shown in [28] that eventually resulted in yielding a higher misclassification rate. The work in [16] addresses the adversarial inputs on tasks such as spam filtering, sentiment analysis, and fake news detection against LSTM, CNN, and Naive Bayes classifiers. However, in our work, we focus on the tweets' modifications and conducted comprehensive experiments under three different scenarios to evaluate the performance of five categories of classifier-based ML models.

3 ConTheModel Overview

In this section, we present the dataset used in our experiments, the architecture of ConTheModel, and our developed algorithm.

3.1 Dataset

In our work, we consider the PHEME dataset, which is publicly available [15] and is associated with nine buzzworthy news events (Charlie Hebdo, Ferguson, Germanwings Crash, Sydney Siege, Putin Missing, Prince Toronto, Ottawa Shooting, Gurlitt, Ebola Essien) on Twitter. With the collected dataset, news events were classified into rumors and non-rumors. Rumors emerged as unverified at the time of posting and were later proven to be true, false, or remained unverified by professional journalists. According to the dataset's annotation, True news: "misinformation: 0, true: 1"; False news: "misinformation: 1, true: 0"; Unverified news: "misinformation: 0, true: 0".

The rumor news that was proven to be false were rumorous tweets, whereas the rumor news that was proven to be true were non-rumorous tweets. For our work, we ignore unverified news as there is not much information in the dataset but consider only verified tweets. Each tweet is a conversation that consists of the source tweet conveying the news and a thread of responses expressing their opinion to this tweet (response tweets). Table 1 shows the number of rumor and

Table 1: Distribution of Rumor and Non-Rumor tweets across all the events of PHEME dataset.

Events	**Rumors**	**Non-Rumors**
Charlie Hebdo	116	1814
Ferguson	8	869
Germanwings Crash	111	325
Sydney Siege	86	1081
Putin Missing	9	112
Prince Toronto	222	4
Ottawa Shooting	72	749
Gurlitt	0	136
Ebola Essien	14	0

non-rumor tweets of the PHEME dataset; as can be seen, except for Gurlitt event, all the other events have rumor tweets, and except Ebola Essien event, all the other events have non-rumor tweets. We considered source tweets which contain millennial slang words and hashes as the original sentences that need to be modified.

3.2 System methodology and algorithm

The architecture of the proposed ConTheModel is shown in Figure 1. We split the PHEME dataset into two portions where 75% of the dataset is trained, and the rest 25% is tested as both original and modified. To validate our technique, we compared categories of ML models such as Random Forest, perceptron-based (Multilayer Perceptron), statistical learning (Naive Bayes), Support Vector Machine, and eXtreme Gradient Boosting algorithm. From Figure 1, Original (1) and Modified (2) test sets are predicted by the ML models to observe the difference in score. We explain our developed modification algorithm shown in Algorithm 1 as follows. We consider 25% of original sentences for modification. Acronyms used to explain the algorithm is defined in **lines 1 - 7**.

In the beginning, every original sample (OS) in the dataset (X) is considered and tokenized into words (OW), and the score of each OW is calculated based on the probability distribution of n-grams in Gutenberg corpus using the smoothing technique of Kneser-Ney [23]. In **lines 11 - 21**, we find the position of OW in OS to ensure the substitute words are placed in the respective positions. For each OW, we check for its corresponding synsets in WordNet to replace the OW. WordNet is the NLTK corpus's lexical database and includes words from various parts of speech such as nouns, adjectives, verbs, adverbs but ignores prepositions, determiners, and other function words.

Synsets are a set of synonyms and identified by a 3-part name of the form: word.pos.nn. The first part of the 3-part name of synsets (word) is considered

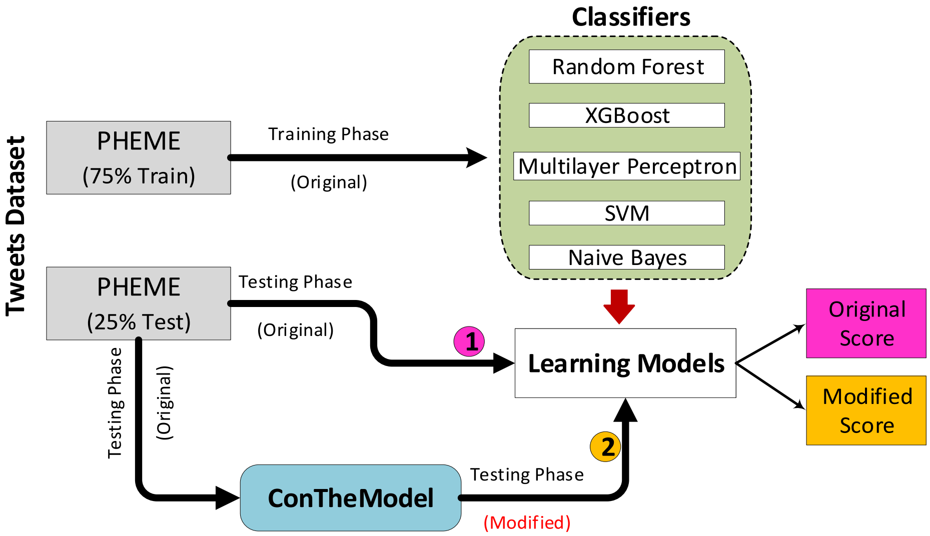

Fig. 1: Overview of ConTheModel.

and appended to the list of synonyms (SL) by avoiding repetition of words. Similarly, the list of antonyms (AL) is created by finding antonyms for the OW in WordNet, and these two lists are merged into a dictionary of the form "OW: SL or AL". For example, the word "hold" will have a dictionary of the form, hold: [clasp, grip, carry, give, let_go_of] where "hold" is the key and list on the left are the values. Therefore in our algorithm, the OW is selected for modification provided it has synonyms or antonyms in WordNet.

Furthermore, in **lines 23-25**, considering that there are n OWs in a sentence that can be replaced and each of these OWs consists of m values of variable length, we consider maximum length of m to restrict the number of modified samples (MS) that can be formed. For example, the word "forced" may have a length of m as two, whereas the word "hold" may have a length of five (maximum length). In **lines 26-38**, we replace n OWs in OS with its corresponding m values to form modified samples until maximum length is reached. These MSs are then tokenized, and score is calculated like in OS to ensure that we arrive at a single MS out of all the MSs for each OS. We selected the MS with a minimum score as we found that this showed better performance of our system. In **line 39**, a list of MS is created and are fed to the classifiers to predict the samples as in **line 41**.

For more illustration, Table 2 provides examples of original and modified tweets with multiple word replacements per tweet. As m values are a combination of both SL and AL, the OWs in the OS can be modified using synonym or antonym or union of both. The original words in the original tweets are highlighted in blue, and the modified words in the modified tweets are highlighted in red.

Algorithm 1: Modification of source tweets.

```
Input  : Original test samples(source tweets)
Output: Modified test samples(source tweets)
OS: Original test sample
MS: Modified test sample
X: 25% of the dataset
OW: Original word
SL: synonym_list
AL: antonym_list
z: values of both SL and AL
foreach OS in X do
    list_word ← Tokenize OS into words.
    Calculate the score of each original word.
    foreach OW in list_word do
        foreach k in wordnet.synsets(OW) do
            find the synonym word → build up SL.
            find the antonym word → buld up AL.
        end
        if SL and AL not empty then
            [OW: append(SL and AL)] → dictionary.
        else
            non empty list → dictionary.
        end
    end
    max_length = 0.
    foreach items in dictionary_values do
        max_length ← Length (longest list of values).
    end
    foreach i in range(max_length) do
        foreach OW, z in the dictionary_values do
            if i < max_length then
                Replace OW in OS with the values of (SL and AL) to form MS.
            end
        end
        if OS is not equal to MS then
            Tokenize MS into words.
            Calculate the score of MS.
            Scores←append (score).
            Get MS sentence with min(Scores).
        end
    end
    Append MS to list_MS.
end
Provide list_MS to the model to predict.
```

4 Evaluation

In this section, we describe the evaluation setup, metric and the classifiers that are used in our work.

Table 2: Examples of original tweets and the corresponding modified tweets using ConTheModel.

Relation	Original Tweet	Modified Tweet
Synonym	For those saying we don't yet know who's responsible for #CharlieHebdo killings, yes, you're right, it was probably fundamentalist atheists.	For those allege we don't nevertheless recognize who's responsible for #CharlieHebdo killings, yes, you're right, it constitute likely fundamentalist atheists.
Antonym	JUST IN: Germanwings plane crashes in southern France, up to 150 feared dead http://t.co/GIZjXnqBU3	JUST IN: Germanwings plane crashes in northern France, up to 150 feared alive http://t.co/GIZjXnqBU3
Synonym & Antonym	BREAKING a hostage siege in Sydney with a man with an IS flag or similar - live reports from @bkjabour http://t.co/VWugbceCAR	BREAKING a hostage siege in Sydney with a civilian with an IS flag or similar - dead reports from @bkjabour http://t.co/VWugbceCAR

4.1 Evaluation Setup

In order to evaluate our technique, 75% of the total dataset is used to train the models and the rest are used to test the performance of the models. We train the models assuming that the attacker has no prior knowledge of the models being targeted. Initially, we gather the performance results of the trained models on original samples, and then we use the modified samples to evaluate the performance using the same trained models. Efficiency of our work is evaluated based on the dissimilarity between the performance results of original and modified samples. During evaluation, we considered PHEME dataset which consists of tweets that were associated with nine buzzworthy news events on Twitter. While exploring the dataset, we observed that the dataset is unbalanced as it contains fewer rumors than non-rumors, as summarized in Table 1. We conducted our experiments on data from different events to investigate our ConTheModel technique's effectiveness in three different scenarios, i.e., *Individual*, *Topics*, and *All Events*. For each scenario, we compared our results between different classifiers such as Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Naive Bayes (NB) which were implemented using the scikit-learn library [25]. The three evaluation scenarios are summarized as follows.

Individual: Since the PHEME dataset consists of nine events, among them seven events (Charlie Hebdo, Ferguson, Germanwings Crash, Sydney Siege, Putin Missing, Prince Toronto, Ottawa Shooting) have both rumour and non-rumour tweets whereas the other two events have either rumour (Ebola Essien) or Non-rumour (Gurlitt) only tweets. Therefore, in this evaluation we consider the seven events which have both rumor and non-rumor tweets.

Topics: The dataset is associated with nine buzzworthy events which are then segregated into five rumor topics [14]; i.e., *Crime* (Charlie Hebdo, Ferguson, Sydney Siege, Ottawa Shooting), Politic (Putin Missing), Entertainment (Prince Toronto), Impact (Germanwings Crash), and *Mixed* (Ebola Essien, Gurlitt). Here, we consider topics with greater than one event.

All Events: In this evaluation scenario, in addition to the seven events considered in *Individual*, we considered Gurlitt and Ebola Essien events as well. So, all the nine events from the PHEME dataset were assessed.

4.2 Evaluation Metrics

We measured the performance of the models on our technique based on F1 score.

Assuming that original samples are labeled as positive and modified samples are labeled as negative, the definitions in the context of our paper can be as follows:

- *True Positive (TP)*: If the samples belonging to an actual class (positive) are correctly classified as the positive class.
- *True Negative (TN)*: If the samples belonging to an actual class (negative) are correctly classified as the negative class.
- *False Negative (FN)*: If the samples belonging to an actual class (positive) are incorrectly classified as the negative class.
- *False Positive (FP)*: If the samples belonging to an actual class (negative) are incorrectly classified as the positive class.

So F1 score is calculated as follows:

- *Precision*: The ratio of true positive samples and all tested samples.
- *Recall*: The ratio of true positive samples and all predicted samples.
- *F1 score*: The weighted average of the above two metrics.

4.3 Classifiers

In this section, we explain the classifiers used in our experiments.

Support Vector Machines (SVM). Support Vector Networks are advanced supervised learning algorithm that is used for classification and regression. The basic idea of SVM is of a margin (termed "hyperplane") where either side of the hyperplane form two different classes, and a newly observed object is classified into a class depending on which side of the hyperplane the object lies on.

Naive Bayes (NB). Naive Bayes is a collection of supervised learning classification techniques based on the probability of the Bayesian theorem. NB classifiers work with a strong assumption that given the class variable, the particular feature's value is independent of any other feature's value.

Multilayer Perceptron (MLP). Multilayer Perceptron (MLP) is a perceptron based technique that utilizes the supervised learning algorithm called backpropagation for training. MLP classifier can be viewed as a logistic regression classifier but differs from it so that MLP has at least one hidden layer between the input and the output layer.

Random Forest (RF). Random Forest is also a supervised learning algorithm that applies the bagging technique, which combines several base learners with low correlation to improve the overall result compared to an individual model. Random forest is an ensemble of decision trees where each decision tree is allowed to grow to its maximum extent, and each tree outputs a class, and the class with the maximum count becomes the model's output.

eXtreme Gradient Boosting (XGBoost). The eXtreme Gradient Boosting algorithm (XGBoost) is an ensemble learning method that applies the boosting technique. Boosting is one of the ensemble learners in which trees are built sequentially so that the succeeding tree in the sequence learns from the preceding trees.

5 Classification Results

As presented in Section 4, we used different categories of ML classifiers from the scikit-learn library that classify original samples and samples modified by our ConTheModel technique. The classification results provide insights into the performance of our modification technique. We noticed that our dataset is mostly comprised of tweets with a single sentence and has millennial words/ lingos, hashes, and links, as a result of which it is hard to find a replacement when compared to most of the other works which attempt to modify more than a single sentence or contains non-millennial words.

We evaluate the efficiency of our work by considering our technique's performance on samples from different events of the dataset under three evaluation scenarios (as mentioned in Section 4.1). The samples (column "Sentence") in all scenarios have been classified as *original* and *modified* with F1 score as the metric to measure the performance of the models. For the first evaluation scenario, we consider events with both rumor and non-rumor tweets (seven events) as shown in Table 1 and report the performance results for individual events in Table 3 on the classifiers. The classifiers' performance is low for samples modified by our ConTheModel technique compared to original samples across all seven events. In detail, we noticed the maximum drop (highlighted in bold) as follows: 7.18%, 8.39% and 10.18% for NB on three events, 14.64% and 16.36% for XGBoost on two events, whereas the other classifiers, i.e., RF, MLP showed a drop of 10.72% and 5.16% respectively for one event each. Also, we observed that RF had an average drop of 5.96% in F1 score; XGBoost shows an average drop of 8.78%; MLP shows an average drop of 4.77%; SVM had an average drop of 4.05%, and NB has an average drop of 8.09% across all the seven events.

In addition, to evaluate ConTheModel's effectiveness on data from a combination of events, we conducted our experiments on six events, which are grouped

Table 3: Adversarial evaluation on the classifiers for individual events.

Classifiers	Sentence	Charlie Hebdo	Ferguson	Germanwings Crash	Sydney Siege	Putin Missing	Prince Toronto	Ottawa Shooting
RF	**Original**	75.10	59.81	90.38	62.00	47.45	78.86	78.68
	Modified	**64.38**	59.50	79.65	59.49	47.42	74.62	65.50
XGBoost	**Original**	73.19	56.47	85.23	65.52	47.62	59.18	76.79
	Modified	63.41	49.79	**70.59**	61.34	47.42	49.55	**60.43**
MLP	**Original**	81.70	54.74	87.72	65.08	58.81	49.55	82.17
	Modified	77.42	54.70	75.38	62.24	**53.65**	49.10	73.89
SVM	**Original**	77.76	49.69	81.16	53.13	48.14	79.73	80.80
	Modified	73.33	49.60	69.07	51.03	48.10	79.63	71.25
NB	**Original**	49.17	51.31	70.38	54.81	52.23	69.55	67.33
	Modified	41.00	**44.13**	59.68	**46.42**	51.37	**59.37**	56.15

Table 4: Adversarial evaluation on the classifiers for Crime and Mixed topics.

Topic	**Sentence**	**Classifiers**				
		RF	**XGBoost**	**MLP**	**SVM**	**NB**
Crime	**Original**	60.52	62.16	64.74	56.39	34.85
	Modified	50.75	57.10	56.07	51.25	30.86
Mixed	**Original**	100	73.61	100	100	100
	Modified	97.41	70.90	100	89.29	91.66

as *Crime* and *Mixed* (explained in Section 4.1). Since the topics related to Politic, Entertainment, and Impact have one event each and have been already considered in Table 3, we pick topics with greater than one event for this scenario. In Table 4, considering the F1 score for modified samples of *Crime*, RF had the maximum drop of 9.77% whereas NB had the lowest drop of 3.99%, and under *Mixed*, SVM showed the maximum drop of 10.71% with RF classifier having the lowest drop (2.59%) whereas the F1 score remains the same for MLP. We hypothesize that since the number of tweets in both the events under *Mixed* is small, it can be a reason for no change in the F1 score for MLP. Overall, across all classifiers, there was a drop of 6.52% under *Crime* and a drop of 4.87% under *Mixed*. Finally, we provide the third evaluation scenario where all the nine events are evaluated, and the results of our system's performance on the classifiers are shown in Table 5. We observed that RF had the maximum drop of 2.71% in F1 score, whereas NB had the lowest drop of 1.81%, and the average drop across all classifiers was 2.25%. We observed that models had a performance drop in their F1 scores when tested with modified tweets. We found that in the: first scenario, XGBoost had the maximum drop of 16.36%; in the second scenario, SVM had

Table 5: Adversarial evaluation on the classifiers for all events.

Sentence	**Classifiers**				
	RF	**XGBoost**	**MLP**	**SVM**	**NB**
Original	75.47	74.15	76.01	71.60	48.16
Modified	72.76	72.04	73.34	69.65	46.35

the maximum drop of 10.71%, and in the third scenario, RF had a maximum drop of 2.71%.

6 Human Evaluation

We verified our technique's efficiency by conducting a human evaluation with 25 participants in a realistic scenario where we considered participants from various nationalities such as Canada, Ethiopia, India, Pakistan, South Korea, Ukraine, USA, Vietnam, and Yemen. Participants in our evaluation skewed male: male (17; 68%), female (8; 32%) with an age range of 23–45 years. The evaluation was conducted online for a week, and participants were instructed not to refer to any search engines to avoid colluding with our work. The participants were provided with 33 randomly chosen questions (samples/ tweets), a combination of either original or modified samples. Participants were then asked to choose an option on a 5-point Likert scale, ranging from "Surely Original" to "Not sure" to "Surely Modified" according to their perceptions. They were also given an option to choose "Somewhat Original" or "Somewhat Modified" if they were not 100 percent sure about the questions being original or modified.

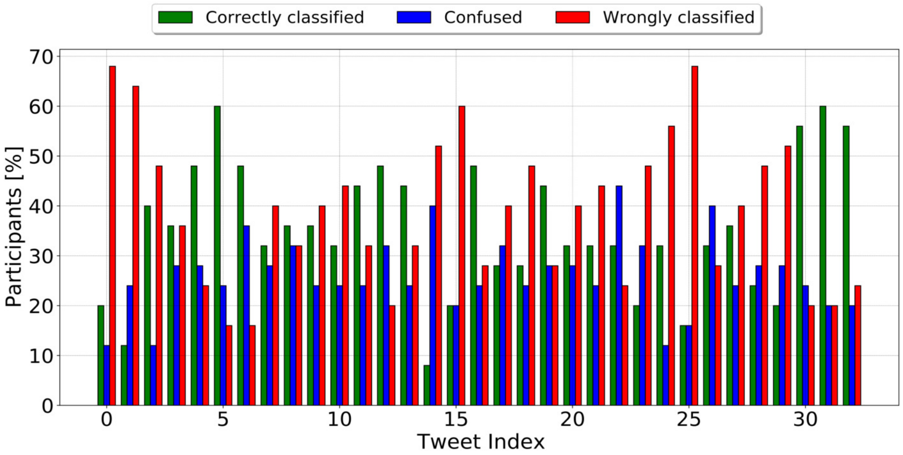

Fig. 2: Human evaluation of each sentence.

We assessed the results of our evaluation by iterating over each question's response by every participant, and we arrived at the number of participants who: "correctly classified" (if the question is actually original and the participant has responded as either "Surely Original" or "Somewhat Original") or "confused" (if the participant has responded as "Not sure") or "wrongly classified" (if the question is actually original and the participant has responded as either "Surely Modified" or "Somewhat Modified") each question. As shown in Figure 2, green represents samples that were correctly classified; red represents samples that were wrongly classified; blue represents samples that were classified as confused. We found that majority of participants: wrongly classified 18 (54.54%) samples, correctly classified 12 (36.36%) samples, confused about two (6.06%) samples, and one (3.04%) was both wrong as well as correctly classified (as in Tweet Index three).

Next, we averaged the response of all 25 participants across each of the 33 questions and found that our human evaluation resulted in 35% of the participants correctly classifying the samples to the original label and 39% of the participants wrongly classifying the samples. However, the remaining 26% were confused as they were not sure about the samples' labels. Our evaluation concluded that majority of the humans (65%) found it difficult to classify the samples correctly and could not detect the modification on tweets.

7 Conclusion

This paper proposed a technique named ConTheModel that generates adversarial examples of tweets to confuse five ML algorithms in three different evaluation scenarios. ConTheModel replaces target words in the original sentence with their respective synonyms or antonyms to form a modified sentence.

Our extensive experiments demonstrate that, on average most of the classifiers were confused in all three evaluation scenarios by showing a drop in performance. Furthermore, our human evaluation justified that humans found it challenging to classify the samples correctly. Our further research will be targeted towards finding other ML classifiers that can be confused by our technique and find robust models that defend against adversarial examples.

ACKNOWLEDGMENTS

This work was supported by the ICT R&D Programs (no. 2017-0-00545) and the ITRC Support Program (IITP-2019-2015-0-00403).

References

1. Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.J., Srivastava, M., Chang, K.W.: Generating natural language adversarial examples. arXiv preprint arXiv:1804.07998 (2018)

2. Behzadan, V., Munir, A.: Vulnerability of deep reinforcement learning to policy induction attacks. In: International Conference on Machine Learning and Data Mining in Pattern Recognition. pp. 262–275. Springer (2017)
3. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In: Joint European conference on machine learning and knowledge discovery in databases. pp. 387–402. Springer (2013)
4. Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., Zhou, W.: Hidden voice commands. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 513–530 (2016)
5. Carlini, N., Wagner, D.: Audio adversarial examples: Targeted attacks on speech-to-text. In: 2018 IEEE Security and Privacy Workshops (SPW). pp. 1–7. IEEE (2018)
6. Clement, J.: Number of social network users worldwide from 2017 to 2025 (2020)
7. Ebrahimi, J., Rao, A., Lowd, D., Dou, D.: Hotflip: White-box adversarial examples for text classification. arXiv preprint arXiv:1712.06751 (2017)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
10. Hu, W., Tan, Y.: Generating adversarial malware examples for black-box attacks based on gan. arXiv preprint arXiv:1702.05983 (2017)
11. Jia, R., Liang, P.: Adversarial examples for evaluating reading comprehension systems. arXiv preprint arXiv:1707.07328 (2017)
12. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Textfool: Fool your model with natural adversarial text (2019)
13. Kaliyar, R.K., Goswami, A., Narang, P.: Multiclass fake news detection using ensemble machine learning. In: 2019 IEEE 9th International Conference on Advanced Computing (IACC). pp. 103–107. IEEE (2019)
14. Kim, Y., Kim, H.K., Kim, H., Hong, J.B.: Do many models make light work? evaluating ensemble solutions for improved rumor detection. IEEE Access **8**, 150709–150724 (2020)
15. Kochkina, E., Liakata, M., Zubiaga, A.: All-in-one: Multi-task learning for rumour verification. arXiv preprint arXiv:1806.03713 (2018)
16. Kuleshov, V., Thakoor, S., Lau, T., Ermon, S.: Adversarial examples for natural language classification problems (2018)
17. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
18. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
19. Ma, J., Gao, W., Wong, K.F.: Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In: The World Wide Web Conference. pp. 3049–3055 (2019)
20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
21. Miller, G.A.: Princeton university "about wordnet." (2010), `https://wordnet.princeton.edu/.`
22. News, B.: Facebook bows to singapore's 'fake news' law with post 'correction' (30 November 2019), `https://www.bbc.com/news/world-asia-50613341`

23. Ney, H., Essen, U., Kneser, R.: On structuring probabilistic dependences in stochastic language modelling. Computer Speech and Language **8**(1), 1–38 (1994)
24. Papernot, N., McDaniel, P., Swami, A., Harang, R.: Crafting adversarial input sequences for recurrent neural networks. In: MILCOM 2016-2016 IEEE Military Communications Conference. pp. 49–54. IEEE (2016)
25. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
26. Sharma, Y., Chen, P.Y.: Attacking the madry defense model with l_1-based adversarial examples. arXiv preprint arXiv:1710.10733 (2017)
27. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
28. Taheri, R., Javidan, R., Shojafar, M., Vinod, P., Conti, M.: Can machine learning model with static features be fooled: an adversarial machine learning approach. Cluster Computing pp. 1–21 (2020)
29. Vijayaraghavan, P., Roy, D.: Generating black-box adversarial examples for text classifiers using a deep reinforced model. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 711–726. Springer (2019)
30. Wang, X., Jin, H., He, K.: Natural language adversarial attacks and defenses in word level. arXiv preprint arXiv:1909.06723 (2019)
31. Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., Tolmie, P.: Analysing how people orient to and spread rumours in social media by looking at conversational threads. PloS one **11**(3), e0150989 (2016)