International Journal of Informatics, Information System and Computer Engineering

TextGuard: Identifying and Neutralizing Adversarial Threats in Textual Data

Luay Albtsoh, Marwan Omar*

Dept. of Computer Science, Capitol Technology University, Maryland, USA
*Corresponding Email: derek.mohammed@saintleo.edu

ABSTRACTS

Adversarial attacks inside the text domain pose a serious risk to the integrity of Natural Language Processing (NLP) systems. In this study, we propose "Text-Guard," a unique approach to detect hostile instances in natural language processing, based on the Local Outlier Factor (LOF) algorithm. This paper compares TextGuard's performance against that of more traditional NLP classifiers such as LSTM, CNN, and transformer-based models, while also experimentally verifying its effectiveness on a variety of real-world datasets. TextGuard significantly surpasses earlier state-of-the-art methods like DISP and FGWS, with F1 recognition accuracy scores as high as 94.8%. This sets a new benchmark in the field as the first use of the LOF technique for adversarial example identification in the text domain.

ARTICLE INFO

Article History:
Received 14 Sept 2024
Revised 20 Oct 2024
Accepted 10 Dec 2024
Available online 14 Jan 2025
Publication date 01 Dec 2025

Keywords:
Text-Guard, Natural Language Processing (NLP), Local Outlier Factor (LOF), Textual Data

DOI: https://doi.org/10.34010/injiiscom.v6i2.15232
p-ISSN 2810-0670 e-ISSN 2775-5584

1. INTRODUCTION

Natural language processing (NLP) and machine learning (ML) models are increasingly being exposed to adversarial instances, which are purposefully changed inputs meant to deceive and impair their (Goodfellow, *et al*., 2014). In critical tasks like sentiment analysis and question answering, these adversarial methods have drastically reduced the effectiveness of NLP models (Farabet, *et al*., 2012; Jin, *et al*., 2020; Madry, 2017; Mozes, *et al*., 2020; Mrkšić, *et al*., 2016; Zhang, *et al*., 2019). Initially discovered in the picture domain (Goodfellow, *et al*., 2014), there is an urgent and crucial need for strong prevention and detection methods against these assaults in NLP (Omar, 2022; Omar, 2023; Omar, *et al*., 2022; Omar, *et al*., 2023; Omar & Sukthankar, 2022).

We provide TextGuard, a unique detection technique based on the Local Outlier Factor (LOF) algorithm, in recognition of the fact that NLP models are vulnerable to hostile attacks in real-world scenarios. By differentiating between hostile and regular inputs, our method provides a proactive protection against possible security flaws in NLP models (Sun, *et al*., 2020; Tsipras, *et al*., 2018). Inspired by approaches in the field of vision, TextGuard uses anomaly detection, a technology that has several uses, such as medical diagnosis and fraud (Cheng, *et al*., 2019).

Our approach innovatively applies the LOF algorithm in the NLP domain, a first of its kind, to identify adversarial examples based on their outlierness within datasets (Figure 1). This study makes several contributions:

1) The introduction of an LOF-based technique for the detection of adversarial examples in NLP.
2) An extensive evaluation of LOF on 1000 adversarial examples across various datasets and model architectures (BERT, WordCNN, LSTM), focusing on sentiment analysis and news classification tasks.
3) A comparison of TextGuard's performance with existing literature, demonstrating its superiority in detecting adversarial attacks in NLP.

Establishment The article is organized as follows: §II discusses relevant work, §III discusses methodology, §V discusses findings and discussion, §VI discusses important insights and unresolved issues, and §VII concludes.

2. RELATED WORK

Exploring protection and detection tactics against adversarial attacks has increased as a result of developments in the field of natural language processing. Developing strong defensive strategies has received a lot of attention, mostly through adversarial training, which creates hostile cases for retraining models to increase their resistance (Jones, *et al*., 2020; Keller, *et al*., 2021; Zhou, *et al*., 2019). Nonetheless, there is still a lack of research in the area of hostile example identification in NLP. The focus of recent study has started to move in this direction. Zhou et al. introduced DISP (learning to distinguish perturbations), a novel technique that employs contextualized word representations that have previously been trained to detect word-level perturbations without retraining the attacked model, to detect

DOI: https://doi.org/10.34010/injiiscom.vxix.xxxx
p-ISSN 2810-0670 e-ISSN 2775-5584

adversarial NLP attacks (Zhou, *et al.*, 2019). Li et al. did not broaden their technique to include a wider variety of attack frameworks; instead, they concentrated on identifying a particular kind of assault (Li, *et al.*, 2020). Semantic and grammatical limitations were not well addressed in Pruth et al.'s work on adversarial misspelling assaults (Pruthi, *et al.*, 2019).

A system for detecting novel adversarial assaults in text, TextFirewall by Wang et al., assesses discrepancies between model outputs and the impact value of significant words (Wang, *et al.*, 2021). Its applicability to different datasets and tasks is yet unknown, despite its thorough examination in sentiment analysis tasks. Fast Gradient Projection Method (FGPM) for effective adversarial assaults and its defense counterpart, ATFL, which enhanced model resilience and inhibited transferability of adversarial cases, were developed in another work (Wang, *et al.*, 2021). Nevertheless, the study lacks hyperparameter ablation tests, which are essential for evaluating the robustness of the approach.

The method developed by Sakaguchi et al. used grammatical inconsistencies to detect character-level assaults, but it was unable to handle complicated attacks that match the syntax and semantics of typical samples (Sakaguchi, *et al.*, 2017). Although adversarial training is a common defensive strategy, it frequently results in a decline in performance on clean samples (Jin, *et al.*, 2020; Jones, *et al.*, 2020). Studies on detection methods, including the use of Frequency Guided Word Substitution (FGWS) by Mozes et al., have demonstrated potential in detecting adversarial assaults using word frequency characteristics (Mozes, *et al.*, 2020). However, their evaluation was limited to the F1 score, lacking other performance metrics and generalizability discussions for different linguistic tasks.

3. METHODOLOGY

This section describes our approach, starting with a summary of the two main NLP tasks of interest: sentence categorization and sentiment analysis. A thorough description of the Local Outlier Factor (LOF) and the metrics used to assess the effectiveness of our suggested method for adversarial instance identification are provided, along with the definitions and notations of adversarial training and adversarial examples.

3.1. Classification of sentences and sentiment analysis

Although adversarial example detection may be used for a wide variety of NLP tasks, we focus on sentiment analysis and phrase classification due to their widespread usage.

Formal Definition: Think about an input text instance $x \in X$, where X is the input text space and y is the task's goal label (for example, 0 or 1 in sentiment classification, positive vs. negative). As a classification issue, we tackle the challenge of detecting adversarial cases, which may be formally expressed as a function $f(x)$: $x \rightarrow y$. Many learning algorithms, such as sophisticated neural networks like BERT and RoBERTa, may be used to carry out this monitoring.

3.2. Formal definition of local outliers

The Local Outlier Factor (LOF) is a method that assesses the degree of outlierness in data points by using local

DOI: https://doi.org/10.34010/injiiscom.v6i2.15232
p-ISSN 2810-0670 e-ISSN 2775-5584

densities. An outlierness score is calculated by comparing the density of a point to that of its neighbors (Alshawabkeh, *et al*., 2010; Bai, *et al*., 2016; Lozano & Acufia, 2005). When it comes to NLP, LOF plays a crucial role in locating irregularities in text databases.

A data point A's k-distance (A) is the distance to its k-th nearest neighbor. All items within this distance are included in the neighborhood Nk(A). The definition of the reachability distance in Nk(A) between any point A and any point B is:

$$\text{Reachability-distance}_k(A, B) = \max k - \text{distance}(B), d(A,B) \quad (1)$$

Where $d(A, B)$ is the distance between points A and B.

The mean reachability distance from a place A's k-nearest neighbors is inversely proportional to its local reachability density (LRD):

$$\text{lrd}\ k(A) = \left[\frac{\sum B \in N_k(A) \text{reach-dist} k(A, B)}{|Nk(A)|}\right] \quad (2)$$

The LOF of a data point A is then given by the average of the LRD ratios between A and its neighbors:

$$\text{LOF}k(A) = \frac{\sum B \in N_k(A) \frac{\text{lrd}k(B)}{\text{lrd}k(A)}}{|N_k(A)|} \quad (3)$$

A data point's status as an outlier in relation to its immediate neighborhood is determined by this ratio (Cheng, *et al*., 2019).

3.2.1 Algorithm

Outlier identification based on LOF: In our situation, the LOF method for outlier detection works as follows:

Input: A dataset $D = x_1, x_2, \ldots, x_n$ and a threshold r(>0.1).

Output: Outliers in D. Method:

1) For each data point X in D, calculate $D^k(X)$ (distance to k-th neighbor) and define $L_k(X)$ (set of points within $D^k(X)$).
2) Compute reachability distance $R_k(X, Y) = \max(\text{dist}(X, Y), D^k(Y))$ for each pair of points X and Y in D.
3) Calculate the average reachability distance $AR_k(X)$ for each point X.
4) Determine LOF score for each point X as $LOF_k(X) = \text{MEAN}^{Y \in Lk(X)} \left(\frac{AR_k(X)}{AR_k(Y)}\right)$.
5) Identify data points with high LOF values as outliers.

3.2.2 Reducing dimensions with kPCA

Kernel Principal Component Analysis is a popular nonlinear dimensionality reduction technique in the text domain (KPCA) (Mika, *et al*., 1998), which is perfect for identifying significant characteristics and eliminating pointless material. Because KPCA is effective at modeling data using lower-dimensional manifolds, we selected it. KPCA transforms non-linearly separable datasets into linearly separable ones in a higher-dimensional feature space so that principal component analysis may be performed. The process involves no calculations in the high-dimensional space but projects new data points into this space to find their lowerdimensional representations.

3.2.3 Formal Definition

The original data is nonlinearly mapped into feature space F via KPCA:

DOI: https://doi.org/10.34010/injiiscom.vxix.xxxx
p-ISSN 2810-0670 e-ISSN 2775-5584

$$\phi: R^N \rightarrow F \quad (4)$$

PCA is implemented in F after the mapping, implicitly identifying nonlinear main components in the original space; for some mappings ϕ, PCA may still be effectively carried out in F using kernel functions.

3.2.4 Analysis of hyperparameters

To ensure computational efficiency and effectiveness in detecting adversarial examples with our LOF method, we conducted an ablation study. We used Scikitlearn's Grid Search to find optimum hyperparameters, observing that settings of r = 0.5 and k = 20 yield the best $F1$ scores and detection rates. A higher k value improves detection but increases training time. Our goal was to optimize LOF's hyperparameters to distinguish between adversarial and normal examples.

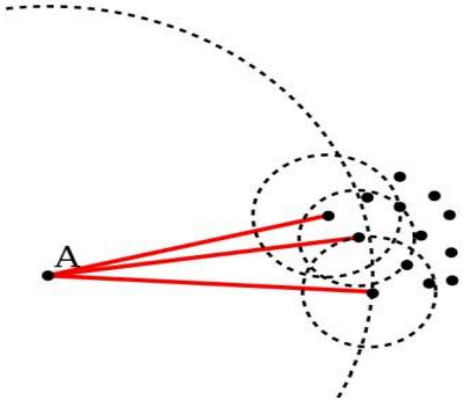

Fig. 1. Illustration of LOF. Data point (A)'s local density is higher than that of its K neighbors when compared to the neighbors.

3.2.5 Using LOF to identify hostile instances

To find outliers, or adversarial instances, in data vectors that have undergone dimensionality reduction, we employed LOF. Comparative tests with other outlier detection methods like K-means and Isolation Forest showed that LOF was the most accurate and robust in our experiments.

3.3. Datasets, models, and evaluation metrics

3.3.1. Datasets

YELP, MR, and AG NEWS were the three benchmark datasets that we employed in our investigations.

3.3.2. Models

Three deep learning models that have been shown to be successful in sentiment analysis and sentence categorization were used: BERT (Topal, *et al.*, 2021), WordCNN (Ma, *et al.*, 2018), and LSTM (Graves & Graves, 2012).

3.3.3. Datasets

Evaluation Metrics: Our evaluation metrics for the classifiers included:

- Accuracy: Correct classification rate.
- Precision: Ratio of correctly classified adversarial examples.
- Recall: Rate of incorrectly classified adversarial examples.

DOI: https://doi.org/10.34010/injiiscom.v6i2.15232
p-ISSN 2810-0670 e-ISSN 2775-5584

- F1 Score: Harmonic mean of precision and recall, calculated as:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (5)$$

- AUC: Area under the Curve, derived from TPR and Specificity:

$$TPR = \frac{TP}{TP + FN} \quad (6)$$

$$\text{Specificity} = \frac{TN}{FP + TN} \quad (7)$$

$$AUC = 1 - \text{Specificity} \quad (8)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

Fig. 1. A pipeline for producing hostile instances.

- Attack Success Rate (ASR): Proportion of successful adversarial attacks:

$$ASR = \frac{\text{number of successful attacks}}{\text{total number of attacks}} \quad (10)$$

4. RESULTS AND DISCUSSION

As explained in the Methodology section, our experimental results across a range of datasets and model architectures are detailed in this section.

4.1. Creation of adversarial examples

We created adversarial samples using the encoder-decoder model provided (Wang, *et al.*, 2020). We trained these components on a large text corpus, using grammar checks for grammaticality and following (Morris, *et al.*, 2020) for semantic consistency to ensure semantic preservation and linguistic fidelity. The adversarial example generation process, depicted in Figure 3, involves three key steps (Morris, *et al.*, 2020):

1) A way to look for effective disturbances.
2) A method of transformation (such as character replacements) to change input text from x to x′.
3) Linguistic restrictions to guarantee that the altered input x′ preserves the original input x's semantic and fluency integrity.

4.2. Implementation details and model architecture

The Hugging Face Transformers collection included a pre-trained BERT model that we used in accordance with (Mozes, *et al.*, 2020). The input sequence lengths for the YELP, MR, and AG NEWS datasets were 512, 256, and 128 correspondingly. During 10 epochs, the BERT model was trained with a batch size of 16 and a learning rate of 1e6. There are 125 million of its characteristics.

DOI: https://doi.org/10.34010/injiiscom.vxix.xxxx
p-ISSN 2810-0670 e-ISSN 2775-5584

Checkpoints with the highest performance were selected based on validation set performance.

In addition to 100 feature maps, the CNN architecture had three convolutional layers with kernel sizes of 2, 3, and 4. The LSTM model employed a Dropout rate of 0.5, contained 128 hidden states, and was initialized using GLOVE (Pennington, *et al.*, 2014) embeddings. Using the Adam optimizer, the CNN and LSTM models were trained for eight epochs with a batch size of twelve and a learning rate of 1e – 4 (Kingma, 2014). Each experiment employed a subset of 1,000 training samples. Scikit-learn was used to develop our LOF algorithm (Pedregosa, *et al.*, 2011).

4.3. Baseline: Model performance under adversarial attacks before lof implementation

We used the Deepwordbug attack recipe (Gao, *et al.*, 2016) and the TextAttack framework (Morris, *et al.*, 2020) to evaluate our NLP classifiers' performance under adversarial assaults. Table 2 demonstrates that adversarial assaults cause a considerable decline in categorization accuracy. It is noteworthy that on the YELP dataset, the accuracy of the LSTM model falls to 7.88%. WordCNN has a score of 9.64%, whereas BERT performs relatively better with 21.97% accuracy on the MR dataset.

Table 1. Datasets For our sentiment analysis and phrase classification tasks, we used three benchmark datasets: YELP, MR, and AG NEWS. Positive and negative evaluations are binarized in the YELP and MR datasets. and divided into sets for training, validation, and testing.

Dataset Name	Dataset Description	Atributes
YELP (Asghar, 2016)	Large Yelp Review Dataset	Set of 560,000 for training, and 38,000 for testing
MR (Poth, *et al.*, 2021)	Movie Review Dataset	Set of 5,331 for training, and 5,331 for testing
AG NEWS (Zhang, *et al.*, 2015)	news topic classification	Set of 12000 for training and 7600 for testing

Table 2. Displays our three classifiers' performance against the deepwordbug attack method before the LOF methodology was used.

Dataset	Model	Accuracy
AG NEWS	BERT	21.09
	WordCNN	13.68
	LSTM	11.56
MR	BERT	12.97
	WordCNN	20.59

DOI: https://doi.org/10.34010/injiiscom.v6i2.15232
p-ISSN 2810-0670 e-ISSN 2775-5584

Dataset	Model	Accuracy
	LSTM	19.29
Yelp	BERT	9.98
	WordCNN	9.64
	LSTM	7.88

Further experiments were conducted to assess the LOF technique's capacity to differentiate between normal and adversarial samples. As Table 3 shows, the attack success rate (ASR) dropped markedly across all models and datasets. When it came to preventing hostile samples, BERT consistently outperformed WordCNN and LSTM, with LSTM exhibiting the lowest detection rates. This decrease in ASR indicates the LOF method's success in identifying and rejecting adversarial examples, leading to fewer successful attacks.

4.4. Comparing and contrasting LOF with prior works

This study's fundamental purpose was to offer a cutting-edge strategy for recognizing and mitigating hostile cases in NLP. To align with existing research, our expanded experiments included three attack methods (Deepwordbug, Genetic Attack, and Textbugger) and two datasets (YELP and MR) to evaluate our three classifiers (BERT, WordCNN, and LSTM). We compared our LOF technique's performance with DISP and FGWS from existing literature, using AUC, F1 score, and TPR for a comprehensive analysis.

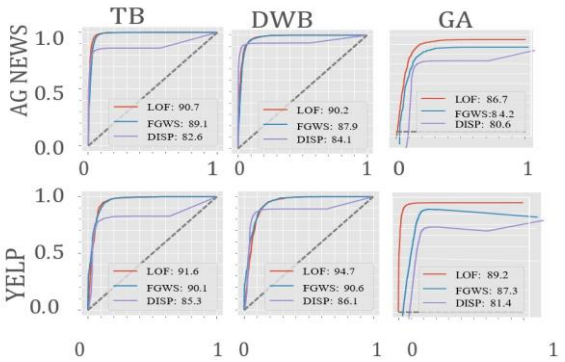

Fig. 3. BERT's ROC curves under Textbugger (TB) and Deepwordbug (DWB) assaults. The charts show the dataset (column) and attack method (row), with the y-axis denoting TPR and the x-axis FPR. The AUC for each detection technique is displayed in the legend.

4.4.1. DISP

DISP focuses on identifying and adjusting malicious perturbations in text classification models. According to our comparison research (Table 4), LOF routinely performs better than DISP on every indicator. Notably, LOF demonstrated excellent efficacy in detecting adversarial samples produced by this technique by achieving an F1 score of 92.4 on the YELP dataset versus BERT attacked using Textbugger. With an F1 score of 49.8, LOF's performance versus LSTM under the Deepwordbug

DOI: https://doi.org/10.34010/injiiscom.vxix.xxxx
p-ISSN 2810-0670 e-ISSN 2775-5584

assault was less spectacular, indicating limits in some situations.

4.4.2. FGWS

In order to identify adversarial assaults, FGWS uses frequency discrepancies between the original words and their replacements. LOF often outperforms FGWS in our comparison (Table 4). For instance, against BERT on YELP with Textbugger, LOF achieved an F1 score of 92.4, compared to FGWS's 89.1. Similarly, on the MR dataset with BERT and Deepwordbug, LOF scored 77.6 in F1, while FGWS scored 73.8. However, FGWS showed superior performance on MR with WordCNN under the Genetic attack, scoring 84.9 in F1 versus LOF's 74.8.

The results indicate that while LOF is highly effective in many scenarios, its performance can vary based on the classifier and attack method. This emphasizes the need for adaptable and versatile detection techniques in NLP adversarial defense strategies.

4.5. Limitations

Our investigation showed several sensitivity in our Local Outlier Factor (LOF) technique's performance. Its efficacy is greatly influenced by variables such as the threshold value, the attack recipe (e.g., Deepwordbug vs. Textfooler), and the quantity of adversarial cases. The lack of a standard LOF threshold value means outlier identification depends heavily on the specific context and domain of the problem. While our results demonstrate robustness across various adversarial attacks, datasets, and models, the generalizability of our technique beyond NLP tasks, such as in the vision domain, remains uncertain and untested.

5. INSIGHTS AND OPEN CHALLENGES

5.1. Dataset security

The increasing reliance on largescale training data in NLP models brings significant security risks. Outsourcing or automating dataset creation and curation exposes businesses to vulnerabilities, including potential manipulation or control of training data by adversaries. This can degrade model performance or result in incorrect predictions. There's a notable gap in research addressing security flaws in NLP datasets, highlighting an urgent need for studies on dataset vulnerabilities and defensive strategies.

5.2. Equitable comparison of detection methods

A possible avenue for future study is to build a baseline for evaluating protection and detection strategies in a standardized framework, as various studies have varied experimental settings.

5.3. Broader goals

There is still much to learn about developing detection methods that can recognize every hostile case in a given input class. Pursuing this line of research could yield significant advancements in the field.

6. CONCLUSION

There are serious safety concerns since NLP models are vulnerable to hostile assaults. Our research presents and tests a novel method for identifying

DOI: https://doi.org/10.34010/injiiscom.v6i2.15232
p-ISSN 2810-0670 e-ISSN 2775-5584

hostile cases in NLP that is based on the Local Outlier Factor (LOF). A range of model architectures, including transformer-based models, WordCNN, and LSTM, as well as real-world datasets, were employed to evaluate its effectiveness. Our results show that our LOF approach can identify hostile cases with up to 92.59 percent accuracy.

REFERENCES

Alshawabkeh, M., Jang, B., & Kaeli, D. (2010, March). Accelerating the local outlier factor algorithm on a GPU for intrusion detection systems. In *Proceedings of the 3rd workshop on general-purpose computation on graphics processing units* (pp. 104-110).

Asghar, N. (2016). Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

Bai, M., Wang, X., Xin, J., & Wang, G. (2016). An efficient algorithm for distributed density-based outlier detection on big data. *Neurocomputing*, *181*, 19-28.

Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems* (pp. 161-168).

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1915-1929.

Gao, J., Lanchantin, J., Soffa, M. L., & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)* (pp. 50-56). IEEE.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Graves, A., & Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37-45.

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 05, pp. 8018-8025).

Jones, E., Jia, R., Raghunathan, A., & Liang, P. (2020). Robust encodings: A framework for combating adversarial typos. *arXiv preprint arXiv:2005.01229*.

DOI: https://doi.org/10.34010/injiiscom.vxix.xxxx
p-ISSN 2810-0670 e-ISSN 2775-5584

Keller, Y., Mackensen, J., & Eger, S. (2021). BERT-defense: A probabilistic model based on BERT to combat cognitively inspired orthographic adversarial attacks. *arXiv preprint arXiv:2106.01452*.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, D., Zhang, Y., Peng, H., Chen, L., Brockett, C., Sun, M. T., & Dolan, B. (2020). Contextualized perturbation for textual adversarial attack. *arXiv preprint arXiv:2009.07502*.

Lozano, E., & Acufia, E. (2005). Parallel algorithms for distance-based and density-based outliers. In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (pp. 4-pp). IEEE.

Ma, X., Jin, R., Paik, J. Y., & Chung, T. S. (2018). Large scale text classification with efficient word embedding. In *Mobile and Wireless Technologies 2017: ICMWT 2017 4* (pp. 465-469). Springer Singapore.

Madry, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mika, S., Schölkopf, B., Smola, A., Müller, K. R., Scholz, M., & Rätsch, G. (1998). Kernel PCA and de-noising in feature spaces. *Advances in neural information processing systems*, *11*.

Morris, J. X., Lifland, E., Lanchantin, J., Ji, Y., & Qi, Y. (2020). Reevaluating adversarial examples in natural language. *arXiv preprint arXiv:2004.14174*.

Morris, J. X., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., & Qi, Y. (2020). Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*.

Mozes, M., Stenetorp, P., Kleinberg, B., & Griffin, L. D. (2020). Frequency-guided word substitutions for detecting textual adversarial examples. *arXiv preprint arXiv:2004.05887*.

Mrkšić, N., Séaghdha, D. O., Thomson, B., Gašić, M., Rojas-Barahona, L., Su, P. H., ... & Young, S. (2016). Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*.

Omar, M. (2022). *Machine learning for cybersecurity: Innovative deep learning solutions*. Springer Nature.

Omar, M. (2023). VulDefend: A Novel Technique based on Pattern-exploiting Training for Detecting Software Vulnerabilities Using Language Models. In *2023 IEEE*

DOI: https://doi.org/10.34010/injiiscom.v6i2.15232
p-ISSN 2810-0670 e-ISSN 2775-5584

Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 287-293). IEEE.

Omar, M., & Sukthankar, G. (2023). Text-defend: detecting adversarial examples using local outlier factor. In *2023 IEEE 17th international conference on semantic computing (ICSC)* (pp. 118-122). IEEE.

Omar, M., Choi, S., Nyang, D., & Mohaisen, D. (2022). Robust natural language processing: Recent advances, challenges, and future directions. *IEEE Access*, *10*, 86038-86056.

Omar, M., Jones, R., Burrell, D. N., Dawson, M., Nobles, C., Mohammed, D., & Bashir, A. K. (2023). Harnessing the power and simplicity of decision trees to detect IoT Malware. In *Transformational Interventions for Business, Technology, and Healthcare* (pp. 215-229). IGI Global.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Poth, C., Pfeiffer, J., Rücklé, A., & Gurevych, I. (2021). What to pre-train on? efficient intermediate task selection. *arXiv preprint arXiv:2104.08247*.

Pruthi, D., Dhingra, B., & Lipton, Z. C. (2019). Combating adversarial misspellings with robust word recognition. *arXiv preprint arXiv:1905.11268*.

Sakaguchi, K., Post, M., & Van Durme, B. (2017). Grammatical error correction with neural reinforcement learning. *arXiv preprint arXiv:1707.00299*.

Sun, G., Su, Y., Qin, C., Xu, W., Lu, X., & Ceglowski, A. (2020). Complete defense framework to protect deep neural networks against adversarial examples. *Mathematical Problems in Engineering*, *2020*(1), 8319249.

Topal, M. O., Bas, A., & van Heerden, I. (2021). Exploring transformers in natural language generation: Gpt, bert, and xlnet. *arXiv preprint arXiv:2102.08036*.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

Wang, T., Wang, X., Qin, Y., Packer, B., Li, K., Chen, J., ... & Chi, E. (2020). Cat-gen: Improving robustness in nlp models via controlled adversarial text generation. *arXiv preprint arXiv:2010.02338*.

DOI: https://doi.org/10.34010/injiiscom.vxix.xxxx
p-ISSN 2810-0670 e-ISSN 2775-5584

Wang, W., Wang, R., Ke, J., & Wang, L. (2021). Textfirewall: Omni-defending against adversarial texts in sentiment classification. *IEEE Access*, *9*, 27467-27475.

Wang, X., Yang, Y., Deng, Y., & He, K. (2021, May). Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 16, pp. 13997-14005).

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., & Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning* (pp. 7472-7482). PMLR.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, *28*.

Zhou, Y., Jiang, J. Y., Chang, K. W., & Wang, W. (2019). Learning to discriminate perturbations for blocking adversarial attacks in text classification. *arXiv preprint arXiv:1909.03084*.

DOI: https://doi.org/10.34010/injiiscom.v6i2.15232
p-ISSN 2810-0670 e-ISSN 2775-5584