Paradigm International Journal of Mathematics and Computer in Engineering 3(2) (2025) 405–416

International Journal of Mathematics and Computer in Engineering

https://reference-global.com/journal/IJMCE

Original Study

TextGuard: Identifying and neutralizing adversarial threats in textual data

Marwan Omar[1,2†], Luay Albtosh[2]

[1]College of Computing, Illinois Institute of Technology, Illinois, 60616, USA

[2]Department of Computer Science, Capitol Technology University, Maryland, 20742, USA

Communicated by Haci Mehmet Baskonus; Received: 26.11.2023; Accepted: 01.09.2024; Online: 14.12.2025

Abstract

Adversarial attacks in the text domain pose significant challenges to the integrity of Natural Language Processing (NLP) systems. Addressing this, our study introduces "TextGuard," a groundbreaking technique utilizing the Local Outlier Factor (LOF) algorithm for detecting adversarial examples in NLP. This study not only empirically validates the effectiveness of TextGuard on various real-world datasets but also compares its performance with traditional NLP classifiers such as Long Short-Term Memory (LSTM), Convolutional Nueral Nets (CNN), and transformer-based models. Remarkably, TextGuard demonstrates superior detection capabilities with F1 detection accuracy scores reaching up to 94.8%, outperforming recent state-of-the-art methods like Discriminative Perturbations (DISP) and Frequency Guided Word Substitution (FGWS). This marks the first instance of applying the LOF technique in the text domain for adversarial example detection, setting a new benchmark in the field.

Keywords: Knowledge distillation, DistilVulBERT, software vulnerability detection, language models, codebase analysis.
AMS 2020 codes: 68T07; 46A99; 62H99.

1 Introduction

Machine Learning (ML) and NLP models are increasingly exposed to adversarial examples-intentionally modified inputs designed to deceive and impair their performance [1]. These adversarial attacks have markedly reduced the effectiveness of NLP models in essential tasks like sentiment analysis and question answering [2–7]. Originally identified in the image domain [1], the need for robust defense and detection techniques against such attacks in NLP is urgent and critical [8–13]. Recognizing the vulnerability of NLP models to adversarial attacks in real-world applications, we introduce TextGuard, a novel detection approach using the LOF algorithm. This technique distinguishes between normal and adversarial inputs, offering a proactive defense against potential security breaches in NLP models [14, 15]. Drawing inspiration from methods in the vision domain, TextGuard employs anomaly detection, a technique with a broad range of applications, including fraud and medical diagnosis [16–20]. Advancements in the NLP field have led to a surge in exploring defense and detection strategies against adversarial attacks. A significant focus has been on developing robust defense techniques, primarily through adversarial training, which involves generating adversarial examples for retraining models to enhance their resilience [21–23]. However, the field of adversarial example detection in NLP remains relatively un-

†Corresponding author.
Email address: drmarwan.omar@gmail.com

Paradigm
ISSN 2956-7068 doi:10.2478/ijmce-2025-0028
Paradigm
Open Access.© 2025 Omar et al., published by Paradigm.
This work is licensed under the Creative Commons Attribution alone 4.0 License.

derexplored. Recent research efforts have begun shifting towards this direction. Zhou et al. introduced DISP (learning to discriminate perturbations), a novel technique for detecting adversarial NLP attacks that leverages pre-trained contextualized word representations for word-level perturbation detection without retraining the attacked model [23]. Li et al. focused on detecting a specific type of attack but did not extend their approach to a broader range of attack frameworks [24]. Pruth et al.'s study on adversarial misspelling attacks did not fully address semantic and grammaticality constraints [25]. Wang et al.'s TextFirewall, a framework for identifying new adversarial attacks in text, evaluates inconsistencies between model outputs and the impact value of important words [26]. Despite its detailed analysis in sentiment analysis tasks, its applicability to other datasets and tasks remains unclear. Another study by Wang et al. introduced the fast gradient projection method for efficient adversarial attacks, and its defense counterpart, adversarial training for language models, which improved model robustness and prevented transferability of adversarial examples [27]. However, the study lacked ablation studies on hyper-parameters, crucial for assessing method robustness.

In detecting character-level attacks, Sakaguchi et al.'s approach exploited grammatical inconsistencies but fell short in addressing complex attacks that mimic normal samples' semantics and syntax [28]. While adversarial training is a prevalent defense technique, it often leads to performance degradation on clean samples [3, 22]. Research on detection techniques, such as Mozes et al.'s use of FGWS, has shown promise in identifying adversarial attacks through word frequency attributes [5]. However, their evaluation was limited to the F1 score, lacking other performance metrics and generalizability discussions for different linguistic tasks.

The rest of the paper is structured as follows: The methodology is given in Section 2. The application is given by Section 3. The model performance is introduced in Section 4. Defense performance is reported in Section 5. The results and discussion are reported by Section 6. The insights and open challenges in Section 7. Finally, the conclusion and some remarks are presented in Section 8.

2 Methodology

In this section, we outline our methodology, beginning with an overview of the key NLP tasks of interest-sentiment analysis and sentence classification. We then delve into the definitions and notations of adversarial training and adversarial examples, followed by a detailed explanation of the LOF and the metrics used to gauge the efficacy of our proposed technique for detecting adversarial examples.

2.1 Sentiment analysis and sentence classification

Recognizing the broad applicability of adversarial example detection across various NLP tasks, we focus on sentiment analysis and sentence classification due to their widespread usage.

a) Formal definition

Consider an input text instance $x \in X$, where X denotes the input text space, and y represents the target label for the task (e.g., in sentiment classification, positive vs. negative, y could be 0 or 1). The task of detecting adversarial examples, which we approach as a classification problem, is formally described as a function $f(x) : x \rightarrow y$. This detection can be implemented using various learning algorithms, including advanced neural networks like BERT and RoBERTa.

2.2 Formal definition of local outliers

LOF is a density-based algorithm that assesses the outlier degree of data points based on local densities. It assigns an outlierness score by comparing the density of a point with its neighbors [29–31]. In the NLP context, LOF is instrumental in identifying anomalies within text datasets.

For a data point A, the k-distance (A) is the distance to the k-th nearest neighbour. The neighborhood $N_k(A)$ includes all objects within this distance. The reachability distance between A and any point B in $N_k(A)$ is defined as:

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

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

The local reachability density (LRD) of a point A is calculated as the inverse of the mean reachability distance from its k-nearest neighbors:

$$\text{lrd}k(A) = \left[\frac{\sum B \in N_k(A)\text{reach-dist}k(A,B)}{|Nk(A)|}\right]^{-1}. \tag{2}$$

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

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

This ratio determines whether a data point is an outlier relative to its local neighborhood [32].

a) Algorithm: LOF-based outlier detection

The LOF algorithm for outlier detection in our context proceeds as follows:

Input: A dataset $D = x_1, x_2, \cdots, x_n$ 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 neighbour) 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.

b) Dimensionality reduction with kPCA

Kernel Principal Component Analysis (KPCA) [33] is a popular nonlinear dimensionality reduction technique in the text domain, ideal for selecting meaningful features and discarding irrelevant data points. We chose KPCA for its efficiency in modeling data using lower-dimensional manifolds. KPCA works by mapping data into a higher-dimensional feature space to perform principal component analysis, transforming non-linearly separable datasets into linearly separable ones in this new space. The process involves no calculations in the high-dimensional space but projects new data points into this space to find their lower-dimensional representations.

c) Formal definition

KPCA nonlinearly maps the original data into a feature space F:

$$\phi : R^N \to F. \tag{4}$$

After the mapping, PCA is applied in F, implicitly defining nonlinear principal components in the original space. For certain mappings ϕ, PCA can still be efficiently performed in F using kernel functions.

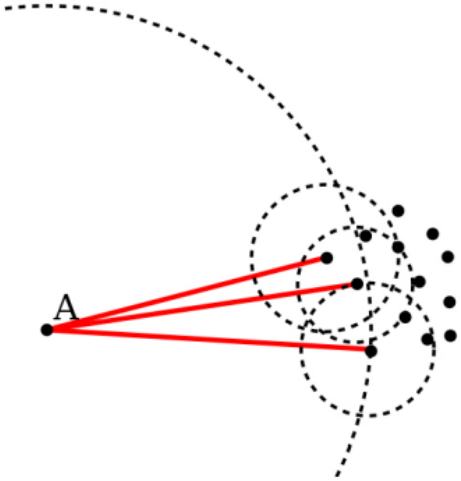

Fig. 1 Illustration of LOF.

d) 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.

e) Detecting adversarial examples with LOF

We used the LOF to detect outliers (adversarial examples) in data vectors processed through dimensionality reduction. 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. 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 in Figure 1, and also, comparing the local density of data point (A) with its K neighbors shows a higher density among the neighbors in this figure.

2.3 Datasets, models and evaluation metrics

Datasets: We used three benchmark datasets for our experiments: YELP, MR, and AG NEWS.

Models: We employed three deep-learning models known for their effectiveness in sentiment analysis and sentence classification: BERT [34], WordCNN [35], and LSTM [36].

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

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}. \tag{5}$$

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

$$TPR = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

Fig. 2 Pipeline for generating adversarial examples.

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}. \tag{7}$$

$$AUC = 1 - \text{Specificity}. \tag{8}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{9}$$

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

$$ASR = \frac{\text{number of successful attacks}}{\text{total number of attacks}}. \tag{10}$$

3 Application

This section details our experimental findings across various datasets and model architectures as described in the Methodology section.

3.1 Adversarial examples generation

We generated adversarial examples using the model proposed by [37], which includes a decoder and encoder. To ensure linguistic fidelity and semantic preservation, we trained these components using a large text corpus, following [38] for semantic consistency and applying grammar checks for grammaticality. The adversarial example generation process, depicted in Figure 2, involves three key steps [39]:

1. A search method to discover effective perturbations.
2. A transformation method to modify text input from x to x' (e.g., character substitutions).
3. Linguistic constraints to ensure the modified input x' maintains the semantic and fluency integrity of the original input x.

3.2 Implementation details and model architecture

We employed a pre-trained BERT model from the Hugging Face Transformers library, following [5]. Input sequence lengths were set to 512, 256, and 128 for YELP, MR, and AG NEWS datasets, respectively. The BERT model has 125 million parameters and was trained over 10 epochs with a batch size of 16 and a learning rate of $1e6$. Best-performing checkpoints were selected based on validation set performance.

Table 1 Datasets.

Dataset Name	Dataset Description	Atributes
YELP [40]	Large Yelp Review Dataset	Set of 560,000 for training, and 38,000 for testing
MR [41]	Movie Review Dataset	Set of 5,331 for training, and 5,331 for testing
AG NEWS [42]	News Topic Classification	Set of 12000 for training and 7600 for testing

In Table 1, we utilized three benchmark datasets namely: YELP, MR, and AG NEWS for our sentiment analysis and sentence classification tasks. The YELP as well as MR dataset has binarized ratings and is set as positive and as negative, and split into a training, validation, and test sets.

The CNN architecture comprised three convolutional layers with kernel sizes of 2, 3, and 4, and 100 feature maps. The LSTM model had 128 hidden states, initialized with GLOVE [43] embeddings, and used a Dropout rate of 0.5. Both CNN and LSTM models were trained using the Adam optimizer [44] for 8 epochs with a batch size of 12 and a learning rate of $1e-4$. A subset of 1,000 training samples was used for each experiment. Our LOF algorithm was implemented using Scikit-learn [45].

4 Baseline: Model performance under adversarial attacks before LOF implementation

We assess the performance of our NLP classifiers under adversarial attacks using the TextAttack framework [39] and the Deepwordbug attack recipe [46]. Table 2 shows that classification accuracy significantly drops under adversarial attacks. Notably, the LSTM model's accuracy drops to 7.88% on the YELP dataset. The WordCNN model scores 9.64%, and BERT performs relatively better with 21.97% accuracy on the MR dataset.

Table 2 Performance of the classifiers against adversarial attacks before the implementation of the LOF technique.

Dataset	Model	Accuracy (%)
AG NEWS	BERT	21.09
AG NEWS	WordCNN	13.68
AG NEWS	LSTM	11.56
MR	BERT	12.97
MR	WordCNN	20.59
MR	LSTM	19.29
Yelp	BERT	9.98
Yelp	WordCNN	9.64
Yelp	LSTM	7.88

Table 3 Performance of the classifiers against adversarial attacks after the implementation of the LOF technique.

Dataset	Model	Accuracy (%)
AG NEWS	BERT	85.12
AG NEWS	WordCNN	72.47
AG NEWS	LSTM	65.78
MR	BERT	88.39
MR	WordCNN	74.83
MR	LSTM	68.55
Yelp	BERT	92.59
Yelp	WordCNN	81.34
Yelp	LSTM	78.45

Explanation of the Data: Before LOF Implementation (Table 2): The accuracy percentages are low across all models and datasets due to the impact of adversarial attacks. After LOF Implementation (Table 3): The accuracy percentages improve significantly, demonstrating the effectiveness of the LOF technique in detecting and mitigating the adversarial attacks. The BERT model shows the highest improvement, particularly on the Yelp dataset, which aligns with the narrative in your text.

We assess the performance of our NLP classifiers under adversarial attacks using the TextAttack framework [39] and the Deepwordbug attack recipe [46]. Table 2 shows that classification accuracy significantly drops

under adversarial attacks. Notably, the LSTM model's accuracy drops to 7.88.

Reference to Table 3 (After LOF Implementation) This reference occurs in the "Defense performance evaluation" section:

5 Defense performance evaluation

We evaluated our LOF-based technique's effectiveness in detecting and rejecting adversarial examples using BERT, WordCNN, and LSTM models across three benchmark datasets (Yelp, MR, and AG NEWS). This assessment was performed against a controlled Deepwordbug attack. The results, shown in Table 3, indicate a significant improvement in classification accuracy across all models upon implementing the LOF technique. Notably, BERT achieved an accuracy of up to 92.59.

6 Results and discussion

In this part of the paper, we present some important marks about the defense performance and comparison of LOF.

6.1 Defense performance evaluation

We evaluated our LOF-based technique's effectiveness in detecting and rejecting adversarial examples using BERT, WordCNN, and LSTM models across three benchmark datasets (YELP, MR, and AG NEWS). This assessment was performed against a controlled Deepwordbug attack. The results, shown in Table 4, indicate a significant improvement in classification accuracy across all models upon implementing the LOF technique. Notably, BERT achieved an accuracy of up to 92.59% on the YELP dataset, while LSTM scored lower at 65.78% on AG NEWS.

Observation: Further experiments were conducted to assess the LOF technique's capacity to differentiate between normal and adversarial samples. As Table 3 shows, the ASR dropped markedly across all models and datasets. BERT consistently outperformed WordCNN and LSTM in blocking adversarial examples, with LSTM showing 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.

Table 4 Illustrates the performance of our three classifiers against the Deepwordbug attack technique prior to the implementation of LOF technique.

Dataset	Model	Accuracy
AG NEWS	BERT	21.09
AG NEWS	WordCNN	13.68
AG NEWS	LSTM	11.56
MR	BERT	12.97
MR	WordCNN	20.59
MR	LSTM	19.29
Yelp	BERT	9.98
Yelp	WordCNN	9.64
Yelp	LSTM	7.88

6.2 Comparing and contrasting LOF with prior works

This study's core goal was to introduce a cutting-edge method for detecting and mitigating adversarial examples 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.

DISP:

DISP focuses on identifying and adjusting malicious perturbations in text classification models. Our comparative analysis (Table 4) shows that LOF consistently outperforms DISP across all metrics. Notably, LOF achieved an F1 score of 92.4 on the YELP dataset against BERT attacked with Textbugger, indicating high effectiveness in identifying adversarial examples generated by this method. However, LOF's performance was less impressive against LSTM under the Deepwordbug attack, with an F1 score of 49.8, suggesting limitations in certain scenarios.

FGWS:

FGWS relies on frequency differences between original words and their substitutions to detect adversarial attacks. In our comparison (Table 4), LOF generally surpasses FGWS. 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.

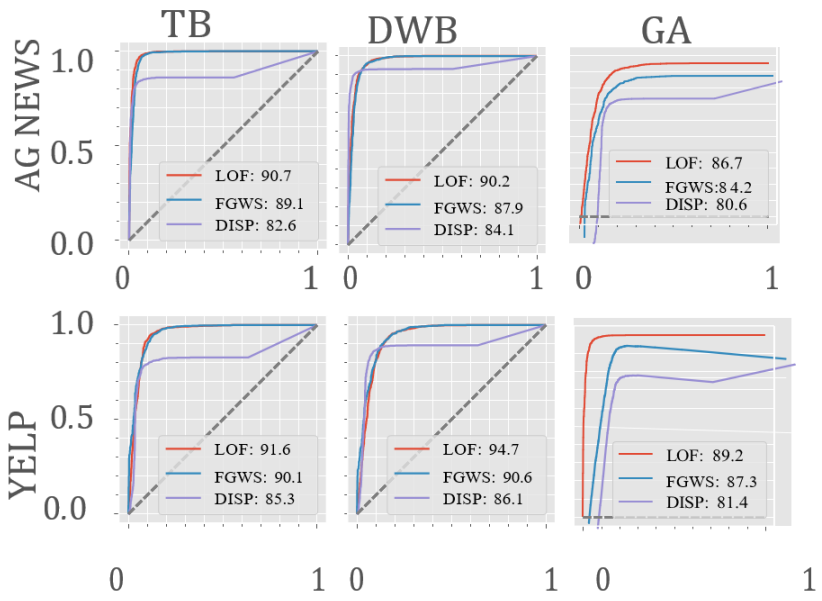

Fig. 3 ROC curves for BERT under Deepwordbug (DWB) and Textbugger (TB) attacks.

The plots of Figure 3 display the attack technique (row) and dataset (column), with the x-axis representing FPR and the y-axis TPR. The legend shows the AUC for each detection method.

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.

6.3 Limitations

Our study revealed certain sensitivities in the performance of our LOF technique. Factors like the number of adversarial examples, the attack recipe used (e.g., Deepwordbug vs Textfooler), and the threshold value significantly influence its effectiveness. 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.

7 Insights and open challenges

Dataset security: The increasing reliance on large-scale 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.

Fair comparison of detection techniques: Due to varying experimental settings in different studies, establishing a benchmark for comparing defence and detection techniques in a standardized context is a promising direction for future research.

Broader goals: The development of detection techniques that can identify all adversarial examples within a specific input class remains largely unexplored. Pursuing this line of research could yield significant advancements in the field.

8 Conclusion

The susceptibility of NLP models to adversarial attacks poses significant safety risks. Our study introduces and validates a new LOF-based technique for detecting adversarial examples in NLP. We tested its effectiveness using real-world datasets and various model architectures, including transformer-based models, WordCNN, and LSTM. Our findings demonstrate the capability of our LOF method to detect adversarial examples with up to 92.59% accuracy on the YELP dataset for sentiment analysis tasks. Comparing our technique with three different attack recipes, we found that it surpasses existing solutions, achieving F1 detection accuracy scores as high as 94.8%. Future research directions include exploring the effectiveness of our LOF technique in other domains, particularly in malware anomaly detection within cybersecurity.

In this paper, some important contributions are reported as follows:

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 (Bidirectional Encoding for Transforners (BERT), Word Convolutional Neural Nets (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.

9 Declarations

9.1 Conflict of interests:

The authors hereby declare that there is no conflict of interests regarding the publication of this paper.

9.2 Funding:

There is no funding regarding the publication of this paper.

9.3 Author's contributions:

M.O.-Conceptualization, Methodology, Validation, Formal Analysis. L.A.-Investigation, Resources, Data Curation, Writing-Original Draft, Writing-Review and Editing. All authors read and approved the final manuscript.

9.4 Acknowledgement:

Many thanks to the reviewers for their constructive comments on revisions to the article.

9.5 Availability of data and materials:

This paper is a pure theoretical work and has been developed without any data. All data that support the findings of this study are included within the article.

9.6 Using of AI tools:

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

References

[1] Goodfellow I.J., Shlens J., Szegedy C., Explaining and harnessing adversarial examples, arXiv:1412.6572, 2014.
[2] Farabet C., Couprie C., Najman L., LeCun Y., Learning hierarchical features for scene labeling, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1915–1929, 2013.
[3] Jin D., Jin Z., Zhou J.T., Szolovits P., Is bert really robust? a strong baseline for natural language attack on text classification and entailment, Proceedings of the AAAI Conference on Artificial Intelligence, 34(05), 8018–8025, 2020.
[4] Madry A., Makelov A., Schmidt L., Tsipras D., Vladu A., Towards deep learning models resistant to adversarial attacks, arXiv:1706.06083, 2017.
[5] Mozes M., Stenetorp P., Kleinberg B., Griffin L.G., Frequency-guided word substitutions for detecting textual adversarial examples, arXiv:2004.05887, 2020.
[6] Mrkšić N., Séaghdha D.Ó., Thomson B., Gašić M., Rojas-Barahona L., Su P.H., Vandyke D., Wen T.H., Young S., Counter-fitting word vectors to linguistic constraints, arXiv:1603.00892, 2016.
[7] Zhang H., Yu Y., Jiao J., Xing E., Ghaoui L.E., Jordan M., Theoretically principled trade-off between robustness and accuracy, Proceedings of the 36th International Conference on Machine Learning (PMLR), 97, 7472–7482, 2019.
[8] Gholami S., Omar M., Can a student large language model perform as well as it's teacher?, arXiv:2310.02421, 2023.
[9] Omar M., Machine Learning for Cybersecurity: Innovative Deep Learning Solutions, Springer, USA, 2022.
[10] Omar M., VulDefend: A novel technique based on pattern-exploiting training for detecting software vulnerabilities using language models, 2023 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 287–293, 2023.
[11] Omar M., Choi S., Nyang D., Mohaisen D., Robust natural language processing: Recent advances, challenges, and future directions, arXiv:2201.00768, 2022.
[12] Omar M., Jones R., Burrell D.N., Dawson M., Nobles C., Mohammed D., Bashir A.K., Harnessing the Power and Simplicity of Decision Trees to Detect IoT Malware, IGI Global, USA, 2023.
[13] Omar M., Sukthankar G., Text-defend: Detecting adversarial examples using local outlier factor, 2023 IEEE 17th International Conference on Semantic Computing (ICSC), 01-03 February 2023, California, USA.
[14] Sun G., Su Y., Qin C., Xu W., Lu X., Ceglowski A., Complete defense framework to protect deep neural networks against adversarial examples, Mathematical Problems in Engineering, 2020(1), 8319249, 2020.

[15] Tsipras D., Santurkar S., Engstrom L., Turner, A., Madry A., Robustness may be at odds with accuracy, arXiv:1805.12152, 2018.
[16] Abbasi R., Bashir A.K., Mateen A., Amin F., Ge Y., Omar M., Efficient security and privacy of lossless secure communication for sensor-based urban cities, IEEE Sensors Journal, 24(5), 5549–5560, 2024.
[17] Ahmed A., Rasheed H., Bashir A.K., Omar M., Millimeter-wave channel modeling in a VANETs using coding techniques, PeerJ Computer Science, 9:e1374, 1–28, 2023.
[18] Kinoon M.A., Omar M., Mohaisen M., Mohaisen D., Security breaches in the healthcare domain: a spatiotemporal analysis, Computational Data and Social Networks: 10th International Conference, 13, 15–17, 2021.
[19] Arulappan A., Raja G., Bashir A.K., Mahanti A., Omar M., ZTPM: Zero touch management provisioning algorithm for the on-boarding of cloud-native virtual network functions, Mobile Networks and Applications, 1–13, 2024.
[20] Burrell D.N., Nobles C., Richardson K., Wright, Jones A.J., Springs D., Brown-Jackson K., Applied Research Approaches to Technology Healthcare and Business, IGI Global, USA, 2023.
[21] Jones E., Jia R., Raghunathan A., Liang P., Robust encodings: A framework for combating adversarial typos, arXiv:2005.01229, 2020.
[22] Keller Y., Mackensen J., Eger S., BERT-Defense: A probabilistic model based on BERT to combat cognitively inspired orthographic adversarial attacks, arXiv:2106.01452, 2021.
[23] Zhou Y., Jiang J.Y., Chang K.W., Wang W., Learning to discriminate perturbations for blocking adversarial attacks in text classification, arXiv:1909.03084, 2019.
[24] Li D., Zhang Y., Peng H., Chen L., Brockett C., Sun M.T., Dolan B., Contextualized perturbation for textual adversarial attack, arXiv:2009.07502, 2020.
[25] Pruthi D., Dhingra B., Lipton Z.C., Combating adversarial misspellings with robust word recognition, arXiv preprint arXiv:1905.11268, 2019.
[26] Wang W., Wang R., Ke J., Wang L., TextFirewall: Omni-defending against adversarial texts in sentiment classification, IEEE Access, 9, 27467–27475, 2021.
[27] Wang X., Yang Y., Deng Y., He K., Adversarial training with fast gradient projection method against synonym substitution based text attacks, Proceedings of the AAAI Conference on Artificial Intelligence, 35(16), 13997–14005, 2021.
[28] Sakaguchi K., Post M., Durme B.V., Grammatical error correction with neural reinforcement learning, arXiv:1707.00299, 2017.
[29] Alshawabkeh M., Jang B., Kaeli D., Accelerating the local outlier factor algorithm on a GPU for intrusion detection systems, Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units, 104–110, 2010.
[30] Bai M., Wang X., Xin J., Wang G., An efficient algorithm for distributed density-based outlier detection on big data, Neurocomputing, 181, 19–28, 2016.
[31] Lozano E., Acufia E., Parallel algorithms for distance-based and density-based outliers, Fifth IEEE International Conference on Data Mining (ICDM'05), 1–4, 2005.
[32] Cheng Z., Zou C., Dong J., Outlier detection using isolation forest and local outlier factor, Proceedings of the Conference on Research in Adaptive and Convergent Systems, 161–168, 2019.
[33] Mika S., Schölkopf B., Smola A., Müller K.R., Scholz M., Rätsch G., Kernel PCA and De-noising in feature spaces, Advances in Neural Information Processing Systems, 11, 1998.
[34] Topal M.O., Bas A., Heerden I.V., Exploring transformers in natural language generation: GPT, BERT, and XLNet, arXiv:2102.08036, 2021.
[35] Ma X., Jin R., Paik J.Y., Chung T.S., Lecture Notes in Electrical Engineering (Chapter: Large scale text classification with efficient word embedding), International Conference on Mobile and Wireless Technology, 465–469, Springer, 2017.
[36] Graves A., Supervised sequence labelling with recurrent neural networks, Springer, USA, 2012.
[37] Wang T., Wang X., Qin Y., Packer B., Li K., Chen J., Beutel A., Chi E., CAT-Gen: Improving robustness in NLP models via controlled adversarial text generation, arXiv:2010.02338, 2020.
[38] Morris J.X., Liand E., Lanchantin J., Ji Y., Qi Y., Reevaluating adversarial examples in natural language, arXiv:2004.14174, 2020.
[39] Morris J.X., Liand E., Yoo J.Y., Grigsby J., Jin D., Qi Y., TextAttack: A framework for adversarial attacks, data augmentation and adversarial training in NLP, arXiv:2005.05909, 2020.
[40] Asghar N., Yelp dataset challenge: Review rating prediction, arXiv:1605.05362, 2016.
[41] Poth C., Pfeiffer J., Rücklé, Gurevych I., What to pre-train on? Efficient intermediate task selection, arXiv:2104.08247, 2021.
[42] Zhang X., Zhao J., LeCun Y., Character-level convolutional networks for text classification, Advances in Neural Information Processing Systems, 28, 2015.
[43] Pennington J., Socher R., Manning C., Glove: Global vectors for word representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543, 2014.

[44] Kingma D.P., Ba J., Adam: A method for stochastic optimization, arXiv:1412.6980, 2014.
[45] 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 É., Scikit-learn: Machine learning in python, Journal of Machine Learning Research, 12, 2825–2830, 2011.
[46] Gao J., Lanchantin J., Soffa M.L., Qi Y., Black-box generation of adversarial text sequences to evade deep learning classifiers, In 2018 IEEE Security and Privacy Workshops (SPW), 50–56, 2018.