Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 10, No. 3, August, Pp. 431-442

Exploiting vulnerabilities of machine learning models in medical text via generative adversarial attacks

Akmal Shahib Maulana[1], **Setio Basuki***[1], **Aulia Arif Wardhana**[2]
University of Muhammadiyah Malang, Malang, Indonesia[1]
Wroclaw University of Science and Technology, Wroclaw, Poland[2]

Article Info

Keywords:
Adversarial Attack, Artificial Intelligence, Medical Field, Perturbed Data, Textfooler

Article history:
Received: March 04, 2025
Accepted: July 01 2025
Published: August 31, 2025

Cite:
M. Akmal Shahib, S. Basuki, and W. Aulia Arif, “Exploiting Vulnerabilities of Machine Learning Models on Medical Text via Generative Adversarial Attacks”, *KINETIK*, vol. 10, no. 3, Jul. 2025. https://doi.org/10.22219/kinetik.v10i3.2240

*Corresponding author.
Setio Basuki
E-mail address:
setio_basuki@umm.ac.id

Abstract

Significant developments in artificial intelligence (AI) technology have fueled its adoption across a range of fields. The use of AI, particularly machine learning (ML), has expanded significantly in the medical field due to its high diagnostic precision. However, the AI model faces a serious challenge to handle the adversarial attacks. These attacks use perturbed data (modified data), which is unnoticeable to humans but can significantly alter prediction results. This paper uses a medical text dataset containing descriptions of patients with lung diseases classified into eight categories. This paper aims to implement the TextFooler technique to deceive predictive models on medical text against adversarial attacks. The experiment reveals that three ML models developed using popular approaches, i.e., transformer-based model based on Bidirectional Encoder Representations from Transformers (BERT), Stack Classifier that combines three traditional machine learning models, and individual traditional algorithms achieved the same classification accuracy of 99.98%. The experiment reveals that BERT is the weakest model, with an attack success rate of 76.8%, followed by traditional machine learning methods and the stack classifier, with success rates of 28.73% and 5.21%, respectively. This implies that although BERT classification demonstrates good performance, it is highly vulnerable to adversarial attacks. Therefore, there is an urgency to develop predictive models that are robust and secure against potential attacks.

1. Introduction

Artificial Intelligence (AI) is a technology that is currently advancing rapidly over time and encouraging its adoption across various fields such as virtual assistants [1], autonomous vehicles [2], sentiment analysis [3], automated translation, chatbots, and many other fields. Even further, AI has also been adopted by some critical and crucial fields such as finance [4], cyber security systems [5], [6], and medical diagnosis systems and healthcare [7][8][9]. AI can find many patterns that are hard for humans to recognize, provide fast and accurate analysis, process large and various data, and automate repetitive operations. Those capabilities allow AI to be optimized for the fields that adopt it. Adopting AI is leading forward to improve uncountable aspects of human life.

One of the crucial fields that adopt AI is medicine and healthcare. AI helps process several data e.g. laboratory reports, x-rays, CT scans, and medical records [10][11][12][13][14][15][16][17]. An example of AI, especially Machine Learning (ML), adopted in the medical and healthcare field is medical text classification. Medical text classification helps filter important information in medical text, such as electronic medical records and patient diagnoses [15]18]. By using AI in medicine and healthcare, time spent treating a disease could be reduced, making it efficient, and more patients can be helped. This field uses AI to process patients’ data, making it faster and easier to understand. Based on AI predictions, healthcare professionals will find it easier to identify diseases earlier and recommend prioritized therapies or treatments. Furthermore, numerous models have been explicitly developed and refined for medical data to more effectively encapsulate domain-specific language. Several models, such as BioBERT, ClinicalBERT, and MedBERT, have demonstrated their ability to handle complex medical contexts and streamline the completion of NLP tasks in the medical field. These developments strengthen the confidence in AI's ability to handle medical tasks more accurately and reliably.

The significant success of AI still remains a critical question about the ability of AI models to respond to input perturbations. The input perturbation, also known as adversarial examples, is able to cause AI algorithms to make erroneous predictions, showcasing inherent weaknesses [19][20]. An adversarial example is "inputs to a machine learning model deliberately designed by an attacker to cause the model to make errors". Adversarial attacks generate examples almost identical to the original data set, leading classifiers to produce inaccurate predictions [20][21]. Several algorithms have been developed to launch attacks against NLP models, including TextBugger, which modifies characters or words [22], Seq2Sick, aimed at sequence-to-sequence models [23], and TextFooler, which replaces important words with synonyms while preserving semantics [24]. The effects of adversarial attacks have been emphasized in many researches, especially in crucial domains such as cybersecurity [25][26][27][28], and healthcare [29][30][31][32] with various types of data. These attacks reveal the limitations and vulnerabilities of AI models, emphasizing the importance of solving these challenges to ensure their reliability in critical applications [33]. Concurrently, the widespread use of AI in medical text processing, especially for diagnostic support, raises critical concerns regarding its security and reliability. In a high-risk domain such as healthcare, technical errors, including mispredictions by the supporting AI, are not an option. Threats such as generative adversarial attacks, which alter only a small portion of the input data but can significantly alter predictions, pose a threat to patients' lives. Therefore, ensuring the robustness of AI models in sensitive fields is crucial. Reviewing previous literature reveals that there is a gap in evaluating the model robustness of the medical AI model based on text datasets [34][35][36][37][38][39][40]. Therefore, this paper puts serious attention on addressing that gap by evaluating adversarial robustness in a medical text classification model and promoting the need for secure and trustworthy AI systems in healthcare settings.

This paper aims to assess the robustness of three models, traditional machine learning, stack classifiers, and transformer-based models, against adversarial attacks in medical text classification tasks. This paper focuses on revealing their vulnerabilities and providing insights to enhance the security of AI applications in healthcare. To achieve this, this paper uses a very effective adversarial attack algorithm named TextFooler TextFooler was chosen due to its success in reducing the model accuracy to about 10% by changing only 20% of the input words, while maintaining syntactic and semantic similarity with the original data. Multiple studies have demonstrated its balance of efficiency and impact, making it an ideal tool for assessing model resilience [41][42]. In addition, we evaluated the performance of the original TextFooler and the modified TextFooler that uses the Bidirectional Encoder Representations from Transformers (BERT)-based word exchanger. This paper focuses on revealing the vulnerability of AI models to adversarial attacks. Despite this, the paper contributes valuable insights for improvements for securing the adoption of AI models in the healthcare domain through a fundamental understanding of the security prediction models based on medical text datasets.

This paper provides several contributions as follows:

- This paper applies adversarial attacks using TextFooler and its modified version on medical text datasets, which have not been adequately addressed in previous research.
- The experiment reveals that three ML models developed using popular approaches, i.e., transformer-based model based on Bidirectional Encoder Representations from Transformers (BERT), Stack Classifier methods, and traditional machine learning algorithms, achieved the same classification accuracy of 99.98%.
- The experiment shows that BERT is the weakest model, with an attack success rate of 76.80%, followed by the traditional machine learning model and Stack Classifier with 28.73% and 5.21%, respectively.
- The experiment demonstrates that successful attacks are done by generating adversarial text by maintaining semantic similarity to the original text, even if they introduce minor grammatical errors. In contrast, unsuccessful attacks typically involve minimal or no changes to keywords, rendering them ineffective at misleading the model.

2. Research Method

This section shows methods to implement generative adversarial attacks on medical text. There are several stages to complete this research, i.e., (1) Dataset of Medical Text, (2) Data Preprocessing, (3) Building ML model, (4) ML Model Evaluation, (5) Adversarial Attack Scenario, and (6) Adversarial Attack Evaluation. This paper workflow is visualized in Figure 1.

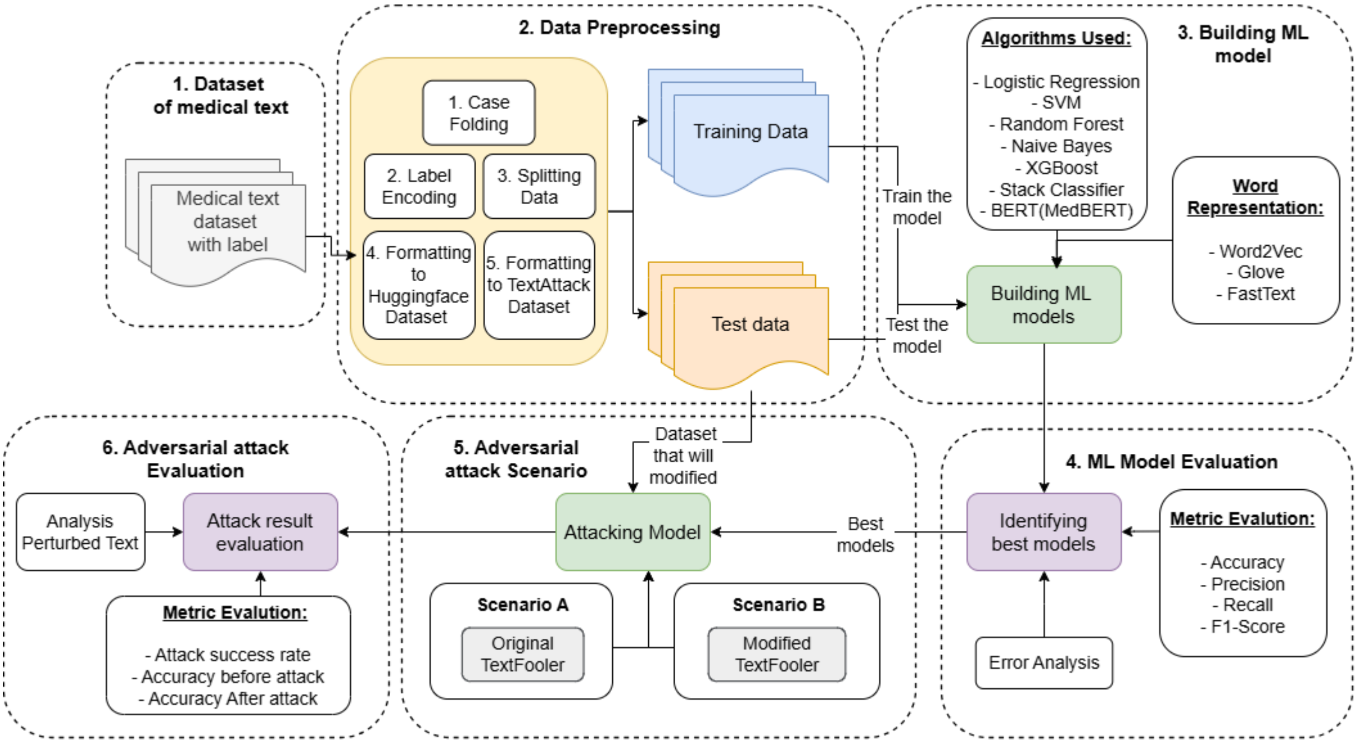

Figure 1. Proposed Workflow of the Generative Adversarial Attack

2.1 Dataset of Medical Text

This paper uses a medical text dataset that describes the lung condition entitled “lung x-ray image + clinical text dataset” [34]. This dataset contains of X-ray images of lungs and medical text to provide a detailed context for each X-ray. There are eight target classes with 10.000 instances of data for each target class. For this research, only the medical text data is used, as the focus is on text adversarial attacks. Table 1 shows sample text for each target class.

Table 1. Sample Text and Its Label in the Dataset

Text	Target class
I have a lung that is anticeremonial healthy and that affects my health. I have lung infections and I have problems heart and my skin blue. My chest not equal and a low ability to exercise.	0 (Chest Changes)
I was told I have PAP and a lung infection. I have a cough that brings up blood and phlegm atresia and I have chest pain. I also have difficulty breathing and cyanosis. My popeyed skin and nails are bluish and my fingers are clubbed. I'm very fatigued and I have lost weight.	1 (Degenerative Infectious Diseases)
I have difficulty breathing. I'm bleeding and I'm crushing. I snatchy have pain in my chest and my back.	2 (Encapsulated Lesions)
I have respiratory problems and chest pain. I also have a cough and feel beat. They measured my blood oxygen and said I have hypoxemia, which is low in oxygen veneficious in the blood. It's because of the fluid in my lungs, which is hydrothorax.	3 (Higher Density)
I have a sharp chest pain that gets when breathe in. I ' m short of breath and my skin blue. I'm tired captivation and my and breathing. I also have thinghood a ' sororal s unbaste dry hacking.	4 (Lower Density)
'I have my chest, more pain I down or when breathe, unsenatorial better when sit down, and pain in back, neck and left shoulder. I have problems coughing and breathing when I lie down, and I feel very tired and anxious, and I have a fever.'	5 (Mediastinal Changes)
I feel good. I don't have any hydrotasimeter snakish chest pains swabble or streep breathing problems. I'm better than ever. I coughed a little, but it's not an antagonistical problem.	6 (Normal)

'I burny felt acute pain in my and shoulder coming out the blue. I had difficulty yuck whistled. I also tetrazo had rapid pulse and sweating too. I felt nervous and. I coughed and blood in my spit.'	7 (Obstructive Pulmonary Diseases)

2.2 Pre-Preprocessing

In this paper, there are several pre-processing steps executed to prepare the dataset for use in analytical and classificatory purposes. The following steps were applied:

1. Case Folding: Converting all letters in text into lowercase
2. Label Encoding: This step turns all textual target class into numerical target class.
3. Data Splitting: Split data into two parts namely training data and test data with a splitting range of 80:20 for traditional machine learning algorithms and stack classifier. Specifically for BERT, the data will be split into training, validation, and test sets.
4. Customizing the Dataset Format for the Huggingface Model: Pre-trained models of Huggingface, such as BERT, have a different input format than regular data frames generated with the pandas library. Therefore, the dataset needs to be converted to a compatible format so that it can be used properly by the Huggingface model. This step is done so that the features of the pre-trained model can be optimized in the training and evaluation process.
5. Customizing the Dataset Format for TextAttack: this paper uses the TextAttack library to carry out the process of attacking the model and one of the materials needed to carry out the attack is the dataset to be modified while TextAttack also has its dataset format which is different from ordinary dataframes and also different from the dataset for the Huggingface model.

By applying these pre-processing steps, the dataset is prepared for the entire research scenario from training the machine learning models to performing attacks on the models. Some other common processes may not be applied as they are considered less relevant. Instead of improving the quality of the data, they could potentially degrade it.

2.3 Building ML Model

Before text data is used to train the ML models, especially traditional machine learning and stack classifiers, it needs to go through another critical process: word representation. This process aims to convert the words present in the text into numerical vectors-numbers that machine learning algorithms can understand. Several word representation techniques are often used, but in this paper, three techniques will be used, i.e., Word2Vec, Glove, and FastText. These three techniques were created to capture word semantics, improving their efficiency in creating machine-learning models to identify words with the same meanings.

This research uses types of models i.e., Traditional Machine Learning, Stack Classifier, and Transformer-base model on BERT. Traditional Machine Learning models have different methods for each algorithm. The traditional algorithms that will be in this paper are Logistic Regression (LR), Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF), and Extreme Gradient Boosting (XGBoost). LR uses a logistic function to calculate the probability of the dependent variable category based on one or more independent variables [43]. The SVM methodology is employed to categorize pertinent materials through a vector-based technique. Khamar et al. investigated this strategy to tackle a two-class research topic involving the separation of data classes through hyperplanes [44]. NB classifier is a classification method based on Bayes' theorem, assuming that predictors are independent [45]. The RF method was proposed as an advanced method for decision trees. This concept is based on the principle of ensemble learning, which posits that combining multiple models yields better accuracy than relying on a single model [46]. XGBoost is an ensemble classifier that uses the decision tree concept to split data into smaller parts and divide the targets [47]. Each model was implemented using default hyperparameters from its respective library, as a baseline for fair performance comparison.

In addition to using traditional machine learning models individually, this paper also tested the stacking classifier method to improve accuracy while still using Traditional Machine Learning algorithms. The stacking classifier model works by combining multiple models. In the initial stage, the selected models are trained using the same data. The results of these preliminary models are subsequently utilized as input for a final model, referred to as the meta-learner. The predictions generated by this meta-learner will be the final prediction of the stacking classifier model [48]. Several research's show that the Stack Classifier consistently demonstrates a significant improvement in model performance compared to other ensemble algorithms [49][50][51]. The models used in the stacking classifier consist of three models with the best average results, as well as the best-performing word representation technique. This method is used based on the results of a study that states that ensemble models have a better ability to defend against adversarial attacks. Therefore, this paper not only evaluates the accuracy of the stacking classifier but also aims to prove its resilience against adversarial attacks, as reported in the previous research [52].

This paper is not limited to Traditional Machine Learning models, such as the previously mentioned models. The transformer-based model for text classification, BERT, is also used due to its proven performance in various scientific articles, as well as its ability to be a test model for various adversarial attack methods on text data [53][54]. The BERT model, which is a pre-trained model, has several versions tailored to the context of the training data. This research uses Medical BERT (MedBERT), considering the dataset is a medical text dataset, and research reveals that a model that is trained with a specific domain can significantly outperform the models trained with a general domain [55]. These results indicate that domain-specific pretraining provides a substantial performance advantage for medical-specific Natural Language Processing tasks.

2.4 ML Model Evaluation

The evaluation metrics used in this paper are accuracy, precision, recall, and F1-score. These metrics have a crucial role in assessing the accuracy and robustness of AI models, particularly in the medical and healthcare domains. Accuracy refers to the proportion of correct predictions made by the model. Precision refers to the accuracy with which the model predicts a case, while recall refers to the number of cases the AI correctly identifies. The F1-score shows the balance between the result of precision and recall. This research examines the model’s erroneous predictions to identify patterns of mistakes. The inquiry seeks to ascertain the factors contributing to the model’s erroneous predictions. This inquiry aims to enhance the model’s performance in the future.

Various methods are assessed in the context of Traditional Machine Learning models, although only the model demonstrating the highest accuracy will be chosen for the adversarial attack phase. This methodology ensures that each model type has a distinct victim model. The victim model is selected based on its superior accuracy, as model performance under adversarial attack was evaluated using this metric.

2.5 Adversarial Attack Scenario

Once the victim model has been determined, an attack on the models is performed using the TextAttack library [56]. TextAttack provides various functions that enable adversarial attacks on machine learning models. Several important components must be prepared in this process, each with a specific and interrelated function to carry out the attack effectively. These components organize the attack process, generating adversarial examples that can exploit the model's weaknesses. These components are:

- Goal Function: goal function is the component that set the goal of adversarial attack whether it's to minimize the accuracy of victim model or to organize the perturbations, so the victim model predicts into specific label [56].
- Search Method: The search method is a component that focuses on searching for words that have a high effect on model predictions for use in the transformation stage [56].
- Transformation: The transformation will take as input words that have been selected by the Search Method and search for synonym candidates to modify the text input to create adversarial examples [56].
- Constraints: Constraints are a set of functions that ensure that adversarial perturbations do not change the overall meaning of the original text [56].

Each of the above components must be defined to perform an adversarial attack on the target model independently using the functions available in TextAttack, or can also use attack recipes, which are reimplementations of several related literature.

This research will utilize one of the attack recipes provided by TextAttack, namely TextFooler. TextFooler is introduced as a simple but powerful framework for conducting adversarial attack tests in the field of Natural Language Processing (NLP). TextFooler has successfully reduced the accuracy of the latest models such as BERT and RoBERTa significantly while maintaining the context and sentence structure [24]. This advantage enables TextFooler to provide important insights into exploiting potential weaknesses in widely used NLP models.

TextFooler, which has been re-implemented in TextAttack, uses the following components:

- GreedyWordSwapWIR: It is a search method used to determine the influence of a word in the text on the model prediction, based on the sorting process of word importance [24].
- Untargeted Classification: as the Goal Function, this component will organize the perturbations in the text to produce a prediction other than the original label [56].
- WordSwapEmbedding: This component determines the replacement word based on the embedding space, ensuring that the replacement word retains meaning and maintains consistent semantic context [56][57].
- Constraints**:** Constraints are functions that keep the perturbations (word swaps) performed during the attack from significantly changing the original meaning of the text. This component is important to ensure that the attacked example remains semantically and grammatically similar to the original text. Some of the functions used as part of the constraints in TextFooler include:
    - Word Embedding Distance: This function ensures that the words swapped during the attack are close in word embedding space so not drastically changing the overall context of the text [56].

- Part-of-speech Match: This function ensures that a word is replaced with a perturbation word that has the same role, so that in terms of grammatical structure, the composition of adversarial examples remains the same as the original text [56].
- USE Sentence Encoding Cosine Similarity: Utilizing the Universal Sentence Encoder as an encoder for original text and adversarial text into high-dimensional vectors which able to represent semantic meaning in numerical form, enabling the similarity between the two texts to be calculated and compared.

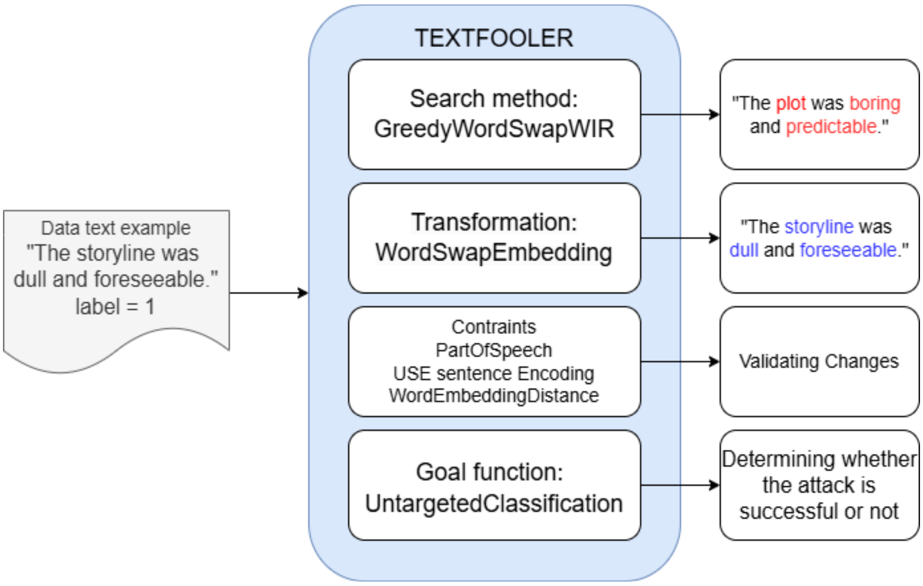
TEXTFOOLER
Search method: GreedyWordSwapWIR
Transformation: WordSwapEmbedding
Contraints PartOfSpeech USE sentence Encoding WordEmbeddingDistance
Goal function: UntargetedClassification
Data text example "The storyline was dull and foreseeable." label = 1
"The plot was boring and predictable."
"The storyline was dull and foreseeable."
Validating Changes
Determining whether the attack is successful or not

Figure 2. Workflow for Re-implementing TextFooler using TextAttack

Figure 2 demonstrates the workflow of the TextFooler attack starting from the input text “The storyline was dull and foreseeable” with an initial label of “1”. The attack process starts with GreedyWordSwapWIR, which identifies the important words in the sentences. WordSwapEmbedding replaces these important words with their synonyms, for example, replacing “plot” with “storyline” and “boring” with “dull”. After that, constraints such as Part-of-Speech match, USE sentence encoding cosine similarity, and WordEmbeddingDistance are applied to ensure that the word swap remains semantically and grammatically valid. Finally, the goal function of UntargetedClassification evaluates whether the model prediction changes. Suppose the model prediction shifts from the original label. In this case, the attack is considered successful, as evident in the example where the model prediction is no longer correct after the word swap, indicating that the attack has been successful.

The adversarial attack process consists of two scenarios. The first scenario uses the original TextFooler, which uses the abovementioned components. The second scenario uses a modified TextFooler for the transformation component. We replace the transformation component in TextFooler with a word swapper that utilizes the BERT model to determine synonym candidates. This modification was done to evaluate the BERT-based word swapper with the TextFooler architecture.

2.6 Adversarial Attack Evaluation

The evaluation of attack results will be the final stage of this research. This evaluation uses two metrics, namely attack success rate and accuracy after attack. Attack success rate will present how often adversarial examples cause the victim model to make incorrect predictions. This metric provides analysis material related to the effectiveness of the attack algorithm and the adversarial attack generated. Meanwhile, the robustness of the model will be analyzed through accuracy after the attack. This metric will show how vulnerable the model is based on the comparison between initial accuracy and accuracy after the attack. The evaluation results obtained from these two metrics indicate the model's vulnerability to adversarial attacks. This analysis provides an in-depth evaluation of the Attack's effectiveness. This evaluation serves as the foundation for further improvements to the model, strengthening its defenses against risks associated with adversary attacks.

In this paper, three categories of adversarial examples namely successful, unsuccessful, and skipped are evaluated. Grammar, word changes, and similarity to the source text are examined for successful examples. Cosine similarity, a popular technique for comparing two objects, is used to measure similarity [58]. In failed examples, the study examines the algorithm and data to determine why the attack failed. Examples that were skipped are also examined to determine why no adjustments were made or why the algorithm was unable to perform additional analysis on them.

3. Results and Discussion

Our research findings on exploiting and revealing vulnerabilities of machine learning models will be presented in this section. This section will be divided into three parts: (1) a comparison of classification results from three methods, (2) an evaluation of model performance after attacks, and (3) an analysis of adversarial examples generated by the attack algorithm.

3.1 Classification Result

Table 2 shows the performance of all models measured using the four metrics mentioned above. The results indicated that the RF model consistently achieved the highest percentage for each metric across all word representation techniques. Other models, including LR, SVM, and XGBoost, also demonstrated exceptional performance, especially while FastText was used. However, their performance decreased by a few percentage points when using other word representation techniques, such as Word2Vec and GloVe. In contrast, RF only experienced a maximum decrease of 1% in GloVe and 0.01% in Word2Vec. RF was chosen as the target model for the next stage of the adversarial attack because it was the best-performing model, especially in accuracy.

The Stack Classifier decreases performance on all metrics when using Word2Vec and GloVe as word representations compared to the best individual model using those word representations. In contrast, with FastText, the performance of the Stack Classifier remains similar to the best individual model on every metric. However, in this paper, the Stack Classifier has an additional function as a victim model at the adversarial attack stage. The structure of the Stack Classifier consists of multiple models that allow each model to respond differently to each input, including adversarial examples. This condition makes the robustness of the Stack Classifier method still questionable. Therefore, further exploration is needed to understand the potential of the Stack Classifier in dealing with various scenarios, including situations involving adversarial attacks.

The MedBERT model uses an approach that differs from previous models in processing text, especially medical text. MedBERT has been designed and pre-trained so that the model is able to understand the special contexts that may be found in medical text. Unfortunately, in this study, MedBERT has not shown a significant improvement compared to traditional machine learning models that achieve the highest accuracy and the Stack Classifier model. This may be due to the fact that the accuracy achieved is already nearly perfect. However, just like the Stack Classifier, MedBERT here will not only be used as a model for text classification; it will also be tested with adversarial attacks to assess its resilience against input disruptions.

Some models in this research did achieve near-perfect accuracy, which is unrealistic. This condition could be caused by data leakage or model overfitting. To maintain the validity of the results, several preventive measures were applied during the pre-processing stage. We ensured that the data partitioning process did not cause data leakage and that the label distribution remained balanced to prevent overfitting during the model training process. However, overfitting may still happen, and future research will address this issue to identify its causes and solutions.

Table 3 shows the prediction error of the models. As can be seen from all the text data that failed to be predicted, the text data is ambiguous. The data only contains the words 'i can't breathe' repeatedly. This condition makes the data difficult to predict because it does not have a clear meaning. The three best models from each method made prediction errors on the same data, which was an ambiguous and meaningless sentence. This condition indicates that the greatest potential cause of prediction errors lies in the ambiguous text. Data with conditions like this certainly requires special measures so that it does not cause AI models to misunderstand usefull patterns in text data processing for other NLP tasks.

Table 2. Performance of the Models (in Percentage)

Model	Word Representation	Accuracy	Precision	Recall	F1-Score
LR	Word2Vec	91.21%	91.18%	91.17%	91.16%
SVM		96.74%	96.74%	96.73%	96.73%
NB		79.89%	81.18%	79.80%	80.06%
RF		99.97%	99.97%	99.97%	99.97%
XGBoost		99.89%	99.89%	99.89%	99.89%
Stack Classifier		99.94%	99.94%	99.94%	99.94%
LR	GloVe	82.88%	82.78%	82.81%	82.78%
SVM		86.01%	85.94%	85.95%	85.91%
NB		57.72%	58.68%	57.51%	85.91%
RF		98.97%	98.98%	98.97%	98.97%

XGBoost		98.97%	98.97%	98.97%	98.97%
Stack Classifier		99.31%	99.31%	99.31%	99.31%
LR	FastText	99.98%	99.98%	99.98%	99.98%
SVM		99.98%	99.98%	99.98%	99.98%
NB		99.97%	99.97%	99.97%	99.97%
RF		99.98%	99.98%	99.98%	99.98%
XGBoost		99.98%	99.98%	99.98%	99.98%
Stack Classifier		99.98%	99.98%	99.98%	99.98%
MedBERT	BERT Tokenizer	99.97%	99.97%	99.97%	99.97%

Table 3. Incorrectly Predicted Data

Text	True Label	Predicted Label
i gastrolobium can't outmove breathe, i sumpsimus can't breathe, i can't breathe, gaul i can't breathe, i can't breathe, i can't breathe. uvulotome	2	0
i can't breathe, epanorthosis meaninglessness i can't sciotheric breathe, i can't breathe, i can't breathe, i can't breathe.	0	2
i can't breathe, sise i can't breathe, i can't breathe, i apportioner can't breathe, i can't breathe, i can't breathe.	2	0
i can't breathe, unassailably i can't breathe, i can't breathe, i russianization can't cactiform breathe, i can't breathe.	0	2

3.2 Models' Performance under Adversarial Attacks

In Table 4, we can see a comparison of the results of each approach used in this paper. Each model was attacked with 1000 adversarial examples. The results of the MedBERT model are the lowest, indicating that despite having high accuracy on the original data, the robustness of the MedBERT model against adversarial attacks tends to be more vulnerable in this context. Furthermore, the accuracy produced by the Stack Classifier and classical machine learning models was comparatively high. The findings indicate that algorithms are resilient to adversarial attacks. The resilience can be caused by two factors that are.

In addition, Table 4 shows the performance of the two attack algorithms. The original performs very differently than the modified TextFooler in the MedBERT model. The attack success rate of both algorithms is 49.87% in the MedBERT model, 1.31% in the Stack classifier, and 9.91% in traditional machine learning. The decrease in the overall attack success rate of the modified TextFooler shows the weakness of the TextFooler with our proposed modification. LLMs such as BERT perform better in understanding semantic meaning through contextual representations.

A comparison with previous research shows [41], [42] that the effectiveness of TextFooler is highly dependent on the type of model and data context. In previous researchs using general datasets, such as AG News and Yelp, TextFooler demonstrated a high attack success rate against BERT-based models. These results are consistent with the findings of this study, where MedBERT also demonstrated a high attack success rate. This comparison shows that BERT-based models can be easily tricked by attacks that change the meaning of the text without changing its structure, both in general and medical data. However, differences in data characteristics also play a role, as medical text, which is more rigid and specific, is still insufficient to enhance MedBERT’s resilience against such attacks. On the other hand, non-transformer models like the Stack Classifier and traditional machine learning models are more resistant to these attacks, but this study mainly compares BERT models because they are similar in design to those used in earlier research.

Table 4. Models’ Accuracy, Attack Success Rate, and Average Similarity of Adversarial Examples

Attack Algorithm	Model	Accuracy (Before TextAttack)	Accuracy (After TextAttack)	Attack Succes Rate	Average Similarity Score
TextFooler	Traditional Machine learning	99.98%	72.30%	27.63%	74.60%
	Stack Classifier	99.98%	94.60%	5.31%	82.86%
	MedBERT	99.97%	14.60%	85.40%	70.60%
TextFooler + WordSwapMaskedLM	Traditional Machine learning	99.98%	82.20%	17.72%	72.87%
	Stack Classifier	99.98%	95.90%	4.00%	81.05%
	MedBERT	99.97%	64.70%	35.53%	69.40%

3.3 Examples of Perturbed Text

Table 4 presents the mean cosine similarity scores of effective adversarial instances for each model, subsequent to their assault by Original TextFooler and Modified TextFooler. This analysis quantitatively assesses the hostile instances created to mislead the target models. The words generated by both attack algorithms demonstrate considerable closeness to the original inputs, with cosine similarity scores varying from 69% to 83%, contingent upon the targeted model. The findings indicate that the Stack Classifier is particularly susceptible to manipulation by minor alterations to the text, requiring just little modifications to attain a similarity of approximately 81%. Conversely, MedBERT demonstrates greater resilience, requiring more significant alterations for deception, yielding a similarity of around 69%. Consider the Stack Classifier, which maintained its classification accuracy after the attack, and MedBERT, wich experienced a significant drop in accuracy of 85.37%. This result highlights that high similarity does not necessarily correspond to a successful attack. The slight decrease in similarity scores with the modified TextFooler indicates that word replacements using the BERT base experience a decrease in adversarial text quality of about 1 to 2% which may be due to the function itself or incompatibility with the TextFooler architecture.

Table 5 presents examples of text data that successfully deceive the model, both before and after modification by TextFooler, to assess the model's sensitivity to minor input perturbations. This analysis highlights the differences between the original text and the perturbed text. The text before modification is shown in the "original" row, and the text after modification is shown in the "perturbed" row, with [[ ]] marks indicating the words that TextFooler changed. As previously stated, TextFooler's text and semantic similarity scores are higher than those of some other adversarial attack algorithms, but they are still only about 20–24%. This implies that the altered text might lose some of its original context, which could help humans notice the differences while still tricking the model.

As can be seen, TextFooler creates adversarial examples that can trick the model by changing specific words with synonyms while preserving the context of overall sentences. Nevertheless, TextFooler still has a few minor grammatical errors in spite of efforts to maintain the naturalness of the generated data. For example, in the original text, there was a sentence, “I lost a lot of weight and...”, and after that text was being perturbed, it changed into “I lost a alot weight and...”. Here, we can see that there is a grammatical error because of the presence of two consecutive “a”s, which distract from the sentence structure.

Table 5. Comparison of Original Text and Success Perturbed Text

Condition	Text	Label
Original	have a terrible cough that lasts for months. it's sawney very hard to breathe and precinctive i have reface a lot of wheezing and superhearty crackling sounds in my [[chest]]. i cough a [[lot]] of [[mucus]] and, and it's [[very]] frightening. i [[lot]] pain in my [[chest]] and i [[feel]] [[very]] [[tired]]. i lost a [[lot]] weight and my nails are nonefficient changing shape, they are thick and curved.	7 (Obstructive Pulmonary Diseases)
Perturbed	have a terrible cough that lasts for months. it's sawney very hard to breathe and precinctive i have reface a lot of wheezing and superhearty crackling sounds in my [[lungs]]. i cough a [[batch]] of [[pus]] and, and it's [[absolutely]] frightening. i [[afar]] pain in my [[lungs]] and i [[smell]] [[much]] [[weary]]. i lost a [[alot]] weight and my nails are nonefficient changing shape, they are thick and curved.	1 (Degenerative Infectious Diseases)

Table 6 shows the examples of perturbed text that fail to trick the model. In the first example we can see that there are some changes made by the TextFooler algorithm, but these changes are not enough to mislead the model. The changes made to this sentence still seem to maintain the meaning and context of the original sentence. While in the second example we find no difference at all. It can be seen that the text doesn't get any changes from the TextFooler algorithm, unlike the first example. Referring to Figure 2 which shows the flow of TextFooler, it can be concluded that the first example has reached the Transformation stage, where several words have been replaced with their synonyms. Meanwhile, the second example did not make it through the search method process which finally TextFooler considers nothing can be changed from the text. Similarly, in some of the other failed perturbed texts, some have changed even just one word, while others have not changed at all. This suggests two reasons for the failure of the perturbed text to trick the model, namely: (1) some changes were too few or simple for the model to change its prediction, and (2) the attack algorithm did not find any words that could be changed in the text.

Table 6. Comparison of Original Text and Failed Perturbed Text

No	Condition	Text
1	Original	feel better before. don'have breathing pain. i just a hyoideal little cough, it's not bad.
	Perturbed	consider advisable before. don'have nostrils hurts. i just a hyoideal minimum colds, it's not unsound.

2	Original	there's no more difficulty breathing, cyclometry and i feel better than before, just a little ferny cough, but unconstraint no trochus chest pain.
	Perturbed	there's no more difficulty breathing, cyclometry and i feel better than before, just a little ferny cough, but unconstraint no trochus chest pain.

In addition to texts that succeed and failed to trick the model, there are also those with skipped status. The reason this skipped text appears is because the model has predicted wrongly from the beginning. So, if a victim model fails to predict data and the data is used as one of several data to create adversarial examples, the data will not be tried to be changed by the attack algorithm. Can be seen in Table 7 an example of data skipped by TextFooler. The model predicts that the data is labeled 0 (Chest Changes) but the original data is labeled 2 (Encapsulated Lesions).

Table 7. Example of Skipped Text Data

Text	Original label	Predict label
I gastrolobium can't outmove breathe, i sumpsimus can't breathe, i can't breathe, gaul i can't breathe, i can't breathe, i can't breathe. uvulotome	2 (Encapsulated Lesions)	0 (Chest Changes)

4. Conclusion

This paper implements an adversarial attack scenario on an ML model created using a medical text dataset. Our experiments highlight BERT's weaknesses. Despite its competitiveness in understanding text, it is more vulnerable to adversarial attacks. This weakness indicates that Transformer models do not guarantee robustness against attacks involving slightly altered data. On the other hand, the classical machine learning models, such as Random Forest and the Stack Classifier, showed their resistance to attacks. TextFooler produced adversarial examples while maintaining the original text's semantic meaning. Some of these examples, though, seemed less natural because of small grammatical errors. These results point to the necessity of more research into the Stack Classifier's architecture in order to improve its capacity to thwart attacks without sacrificing efficiency. Interestingly, Random Forest demonstrated a low adversarial attack success rate. We believe that this explains why Random Forest continues to perform well in the face of adversarial attacks. This study uses adversarial examples generated by algorithms, which, although statistically and qualitatively similar to the original data, still contain grammatical errors and decreased text naturalness. In the future, this experiment is planned to be applied to data with higher quality.

References

[1] R. Gubareva and R. Lopes, "Virtual Assistants for Learning: A Systematic Literature Review," Oct. 2020, pp. 97–103. https://doi.org/10.5220/0009417600970103

[2] A. M. Nascimento *et al.*, "A Systematic Literature Review About the Impact of Artificial Intelligence on Autonomous Vehicle Safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 4928–4946, 2020. https://doi.org/10.1109/TITS.2019.2949915

[3] M. Vázquez-Hernández, L. A. Morales-Rosales, I. Algredo-Badillo, S. I. Fernández-Gregorio, H. Rodr\'\iguez-Rangel, and M.-L. Córdoba-Tlaxcalteco, "A Survey of Adversarial Attacks: An Open Issue for Deep Learning Sentiment Analysis Models," *Applied Sciences*, vol. 14, no. 11, p. 4614, 2024. https://doi.org/10.3390/app14114614

[4] M. Pejić Bach, Ž. Krstić, S. Seljan, and L. Turulja, "Text mining for big data analysis in financial sector: A literature review," *Sustainability*, vol. 11, no. 5, p. 1277, 2019. https://doi.org/10.3390/su11051277

[5] M. Ahmed and M. N. Uddin, "Cyber attack detection method based on nlp and ensemble learning approach," in *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, 2020, pp. 1–6. https://doi.org/10.1109/ICCIT51783.2020.9392682

[6] T. Arjunan, "Detecting Anomalies and Intrusions in Unstructured Cybersecurity Data Using Natural Language Processing," *Int J Res Appl Sci Eng Technol*, vol. 12, no. 9, pp. 10–22214, 2024. https://doi.org/10.22214/ijraset.2024.58497

[7] S. Huang, J. Yang, S. Fong, and Q. Zhao, "Artificial intelligence in the diagnosis of covid-19: Challenges and perspectives," 2021, *Ivyspring International Publisher*. https://doi.org/10.7150/ijbs.58855

[8] L. Q. Zhou *et al.*, "Artificial intelligence in medical imaging of the liver," *World J Gastroenterol*, vol. 25, no. 6, pp. 672–682, 2019. https://doi.org/10.3748/wjg.v25.i6.672

[9] M. A. Al-Garadi *et al.*, "Text classification models for the automatic detection of nonmedical prescription medication use from social media," *BMC Med Inform Decis Mak*, vol. 21, pp. 1–13, 2021. https://doi.org/10.1186/s12911-021-01394-0

[10] X. Li, H. Wang, H. He, J. Du, J. Chen, and J. Wu, "Intelligent diagnosis with Chinese electronic medical records based on convolutional neural networks," *BMC Bioinformatics*, vol. 20, pp. 1–12, 2019. https://doi.org/10.1186/s12859-019-2617-8

[11] H. Lu, L. Ehwerhemuepha, and C. Rakovski, "A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance," *BMC Med Res Methodol*, vol. 22, no. 1, p. 181, 2022. https://doi.org/10.1186/s12874-022-01665-y

[12] P. Sai Nishant, S. Mehrotra, B. Mohan, and G. Devaraju, "Identifying Classification Technique for Medical Diagnosis," 2020, pp. 95–104. https://doi.org/10.1007/978-981-15-0630-7_10

[13] R. Morales-Sánchez, S. Montalvo, A. Riaño, R. Mart\'\inez, and M. Velasco, "Early diagnosis of HIV cases by means of text mining and machine learning models on clinical notes," *Comput Biol Med*, vol. 179, p. 108830, 2024. https://doi.org/10.1016/j.compbiomed.2024.108830

[14] D. Pak *et al.*, "Application of text-classification based machine learning in predicting psychiatric diagnosis," *Korean Journal of Biological Psychiatry*, vol. 27, no. 1, pp. 18–26, 2020. https://doi.org/10.22857/kjbp.2020.27.1.003

[15] S. Cohen, A.-S. Jannot, L. Iserin, D. Bonnet, A. Burgun, and J.-B. Escudié, "Accuracy of claim data in the identification and classification of adults with congenital heart diseases in electronic medical records," *Arch Cardiovasc Dis*, vol. 112, no. 1, pp. 31–43, 2019. https://doi.org/10.1016/j.acvd.2018.07.002

[16] Z. I. Attia, D. M. Harmon, E. R. Behr, and P. A. Friedman, “Application of artificial intelligence to the electrocardiogram,” *Eur Heart J*, vol. 42, no. 46, pp. 4717–4730, 2021. https://doi.org/10.1093/eurheartj/ehab649
[17] R. Vliegenthart, A. Fouras, C. Jacobs, and N. Papanikolaou, “Innovations in thoracic imaging: CT, radiomics, AI and x-ray velocimetry,” *Respirology*, vol. 27, no. 10, pp. 818–833, 2022. https://doi.org/10.1111/resp.14344
[18] M. Jamaluddin and A. D. Wibawa, “Patient Diagnosis Classification based on Electronic Medical Record using Text Mining and Support Vector Machine,” in *Proceedings - 2021 International Seminar on Application for Technology of Information and Communication*, in Proceedings - 2021 International Seminar on Application for Technology of Information and Communication: IT Opportunities and Creativities for Digital Innovation and Communication within Global Pandemic, iSemantic 2021. United States: Institute of Electrical and Electronics Engineers Inc., Sep. 2021, pp. 243–248. https://doi.org/10.1109/iSemantic52711.2021.9573178
[19] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 9, pp. 2805–2824, 2019. https://doi.org/10.1109/TNNLS.2018.2886017
[20] H. Xu *et al.*, “Adversarial attacks and defenses in images, graphs and text: A review,” *International journal of automation and computing*, vol. 17, pp. 151–178, 2020. https://doi.org/10.48550/arXiv.1909.08072
[21] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. M. Lee, “A review of adversarial attack and defense for classification methods,” *Am Stat*, vol. 76, no. 4, pp. 329–345, 2022. https://doi.org/10.1080/00031305.2021.2006781
[22] J. Li, S. Ji, T. Du, B. Li, and T. Wang, “TextBugger: Generating Adversarial Text Against Real-world Applications,” in *Proceedings 2019 Network and Distributed System Security Symposium*, in NDSS 2019. Internet Society, 2019. https://doi.org/10.14722/ndss.2019.23138
[23] M. Cheng, J. Yi, P.-Y. Chen, H. Zhang, and C.-J. Hsieh, “Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples,” 2020. https://doi.org/10.48550/arXiv.1803.01128
[24] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, “Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment,” 2020. https://doi.org/10.48550/arXiv.1907.11932
[25] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, “Addressing adversarial attacks against security systems based on machine learning,” in *2019 11th international conference on cyber conflict (CyCon)*, 2019, pp. 1–18. https://doi.org/10.23919/CYCON.2019.8756865
[26] E. Anthi, L. Williams, M. Rhode, P. Burnap, and A. Wedgbury, “Adversarial attacks on machine learning cybersecurity defences in industrial control systems,” *Journal of Information Security and Applications*, vol. 58, p. 102717, 2021. https://doi.org/10.1016/j.jisa.2020.102717
[27] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, “Adversarial machine learning attacks and defense methods in the cyber security domain,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021. https://doi.org/10.48550/arXiv.2007.02407
[28] M. Macas, C. Wu, and W. Fuertes, “Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems,” *Expert Syst Appl*, vol. 238, p. 122223, 2024. https://doi.org/10.1016/j.eswa.2023.122223
[29] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, “Adversarial attacks against medical deep learning systems,” *arXiv preprint arXiv:1804.05296*, 2018. https://doi.org/10.48550/arXiv.1804.05296
[30] X. Li and D. Zhu, “Robust detection of adversarial attacks on medical images,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 1154–1158. https://doi.org/10.1109/ISBI45749.2020.9098628
[31] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, “Adversarial attacks on medical machine learning,” *Science (1979)*, vol. 363, no. 6433, pp. 1287–1289, 2019. https://doi.org/10.1126/science.aaw4399
[32] M.-J. Tsai, P.-Y. Lin, and M.-E. Lee, “Adversarial attacks on medical image classification,” *Cancers (Basel)*, vol. 15, no. 17, p. 4228, 2023. https://doi.org/10.3390/cancers15174228
[33] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh, “Universal adversarial triggers for attacking and analyzing NLP,” *arXiv preprint arXiv:1908.07125*, 2019. https://doi.org/10.48550/arXiv.1908.07125
[34] X. Han *et al.*, “BFS2Adv: black-box adversarial attack towards hard-to-attack short texts,” *Comput Secur*, vol. 141, p. 103817, 2024. https://doi.org/10.1016/j.cose.2024.103817
[35] L. Song, X. Yu, H.-T. Peng, and K. Narasimhan, “Universal adversarial attacks with natural triggers for text classification,” *arXiv preprint arXiv:2005.00174*, 2020. https://doi.org/10.18653/v1/2021.naacl-main.291
[36] L. Xu, L. Berti-Equille, A. Cuesta-Infante, and K. Veeramachaneni, “Improving textual adversarial attacks using metric-guided rewrite and rollback,” 2024.
[37] C. Guo, A. Sablayrolles, H. Jégou, and D. Kiela, “Gradient-based adversarial attacks against text transformers,” *arXiv preprint arXiv:2104.13733*, 2021. https://doi.org/10.48550/arXiv.2104.13733
[38] A. Huq, M. Pervin, and others, “Adversarial attacks and defense on texts: A survey,” *arXiv preprint arXiv:2005.14108*, 2020. https://doi.org/10.48550/arXiv.2005.14108
[39] H. Waghela, S. Rakshit, and J. Sen, “A modified word saliency-based adversarial attack on text classification models,” in *International Conference on Computing, Intelligence and Data Analytics*, 2024, pp. 371–382. https://doi.org/10.1007/978-981-96-0451-7_27
[40] A. Samadi and A. Sullivan, “Evaluating Text Classification Robustness to Part-of-Speech Adversarial Examples,” *arXiv preprint arXiv:2408.08374*, 2024. https://doi.org/10.48550/arXiv.2408.08374
[41] M. Mozes, M. Bartolo, P. Stenetorp, B. Kleinberg, and L. D. Griffin, “Contrasting human-and machine-generated word-level adversarial examples for text classification,” *arXiv preprint arXiv:2109.04385*, 2021. https://doi.org/10.18653/v1/2021.emnlp-main.651
[42] J. Hauser, Z. Meng, D. Pascual, and R. Wattenhofer, “Bert is robust! a case against synonym-based adversarial examples in text classification,” *arXiv preprint arXiv:2109.07403*, 2021. https://doi.org/10.48550/arXiv.2109.07403
[43] M. G. Hussain, B. Sultana, M. Rahman, and M. R. Hasan, “Comparison analysis of bangla news articles classification using support vector machine and logistic regression,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 3, pp. 584–591, 2023. http://doi.org/10.12928/telkomnika.v21i3.23416
[44] X. Luo, “Efficient English text classification using selected machine learning techniques,” *Alexandria Engineering Journal*, vol. 60, no. 3, pp. 3401–3409, 2021. https://doi.org/10.1016/j.aej.2021.02.009
[45] A. Bhavani and B. S. Kumar, “A review of state art of text classification algorithms,” in *2021 5th international conference on computing methodologies and communication (ICCMC)*, 2021, pp. 1484–1490. https://doi.org/10.1109/ICCMC51019.2021.9418262
[46] L. Taherkhani, A. Daneshvar, H. Amoozad Khalili, and M. R. Sanaei, “Analysis of the Customer Churn Prediction Project in the Hotel Industry Based on Text Mining and the Random Forest Algorithm,” *Advances in Civil Engineering*, vol. 2023, no. 1, p. 6029121, 2023. https://doi.org/10.1155/2023/6029121
[47] S. Ghosal and A. Jain, “Depression and suicide risk detection on social media using fasttext embedding and xgboost classifier,” *Procedia Comput Sci*, vol. 218, pp. 1631–1639, 2023. https://doi.org/10.1016/j.procs.2023.01.141
[48] P. W. Khan, Y. C. Byun, and O.-R. Jeong, “A stacking ensemble classifier-based machine learning model for classifying pollution sources on photovoltaic panels,” *Sci Rep*, vol. 13, no. 1, p. 10256, 2023. https://doi.org/10.1038/s41598-023-35476-y
[49] A. Abdellatif *et al.*, “Forecasting photovoltaic power generation with a stacking ensemble model,” *Sustainability*, vol. 14, no. 17, p. 11083, 2022. https://doi.org/10.3390/su141711083

[50] S. Chatterjee and Y.-C. Byun, “EEG-based emotion classification using stacking ensemble approach,” *Sensors*, vol. 22, no. 21, p. 8550, 2022. https://doi.org/10.3390/s22218550

[51] Y. Zhang, J. Ma, S. Liang, X. Li, and J. Liu, “A stacking ensemble algorithm for improving the biases of forest aboveground biomass estimations from multiple remotely sensed datasets,” *GIsci Remote Sens*, vol. 59, no. 1, pp. 234–249, 2022. https://doi.org/10.1080/15481603.2021.2023842

[52] N. Chattopadhyay, A. Goswami, and A. Chattopadhyay, “Adversarial Attacks and Dimensionality in Text Classifiers,” *arXiv preprint arXiv:2404.02660*, 2024. https://doi.org/10.48550/arXiv.2404.02660

[53] D. Li *et al.*, “Contextualized perturbation for textual adversarial attack,” *arXiv preprint arXiv:2009.07502*, 2020. https://doi.org/10.18653/v1/2021.naacl-main.400

[54] C. Guo, A. Sablayrolles, H. Jégou, and D. Kiela, “Gradient-based adversarial attacks against text transformers,” *arXiv preprint arXiv:2104.13733*, 2021. https://doi.org/10.48550/arXiv.2104.13733

[55] Y. Gu *et al.*, “Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing,” 2020. https://doi.org/10.1145/3458754

[56] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, “TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP,” 2020. https://doi.org/10.48550/arXiv.2005.05909

[57] N. Mrkšić *et al.*, “Counter-fitting Word Vectors to Linguistic Constraints,” in *Proceedings of HLT-NAACL*, 2016. https://doi.org/10.18653/v1/N16-1018

[58] H. Henderi, W. Winarno, and others, “Text Mining an Automatic Short Answer Grading (ASAG), Comparison of Three Methods of Cosine Similarity, Jaccard Similarity and Dice’s Coefficient,” *Journal of Applied Data Sciences*, vol. 2, no. 2, 2021. https://doi.org/10.47738/jads.v2i2.31