DESIGN OF FRAMEWORK FOR ADVERSARIAL ATTACKS & DEFENCES IN CLASSIFICATION MODELS

A Thesis Submitted
in the Partial Fulfilment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

by

ASHISH BAJAJ

(2K21/PHDIT/01)

Under the Supervision of
Prof. DINESH KUMAR VISHWAKARMA
Delhi Technological University

Department of Information Technology

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

August, 2024

ACKNOWLEDGEMENT

I am profoundly grateful to my esteemed PhD supervisor, **Prof. Dinesh Kumar Vishwakarma**, whose exceptional guidance and unwavering support have been instrumental in completing this thesis. His exemplary discipline, unyielding focus, and relentless work ethic have inspired me and set a high standard for academic excellence. I am fortunate to have benefited from his expertise and commendable commitment throughout this challenging yet rewarding journey. I thank **Mrs. Sushma Vishwakarma**, **Diya** and **Advika** for ensuring I always had a home away from home.

No man is complete without his family, who silently toil behind the scenes to help him fight for his dreams. I thank my parents, **Mr. Anil Kumar Bajaj** and **Mrs. Seema Bajaj,** for being the best parents one could hope for. I thank my sibling **Madhav Bajaj**, for doing what siblings do best, i.e., be loving in their own fun way.

Finishing PhD is a highly challenging journey, and the seniors who helped navigate this path need a special mention. To this end, I am grateful for the support of my PhD seniors, **Dr. Ankit Yadav.**

As I went through the most challenging phase of my PhD, my juniors ensured that I never gave up and always came back stronger. I am grateful to **Ananya Pandey, Abhishek Verma** and **Bhavana Verma** for all the light-hearted conversations.

I extend my heartfelt appreciation to the state-of-the-art research lab established by my supervisor. Equipped with cutting-edge NVIDIA GPUs, it was pivotal in facilitating the success of the computationally expensive deep learning-based research experiments throughout my PhD.

Last but not least, I thank God for giving me the persistence and strength to show up at my lab each day and work through the ups and downs of this PhD journey.

Ashish Bajaj
2K21/PHDIT/01

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I Ashish Bajaj hereby certify that the work which is being presented in the thesis entitled "Design of Framework for Adversarial Attacks & Defences in Classification Models" in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy, submitted in the Department of Information Technology , Delhi Technological University is an authentic record of my own work carried out during the period from 02/08/2021 to 30/07/2024 under the supervision of Prof. Dinesh Kumar Vishwakarma.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that Ashish Bajaj (2K21/PHDIT/01) has carried out their research work presented in this thesis entitled "Design of Framework for Adversarial Attacks & Defences in Classification Models" for the award of Doctor of Philosophy from Department of Information Technology, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature:

Name of the Supervisor: Prof. Dinesh Kumar Vishwakarma

Designation: Professor**;** Head of Department (Information Technology)

Address: Rohini, Delhi

Date: 30/07/2024

ABSTRACT

The advent of Deep Learning has enabled us to effectively train neural networks to handle intricate datasets with exceptional efficiency. Nevertheless, as research progresses, numerous weaknesses in neural networks have been revealed. Adversarial Machine Learning is a specific area of research that focuses on identifying and exploiting vulnerabilities in neural networks that lead to misclassification of input data that is very similar to the original data. Adversarial assaults refer to a category of methods designed to intentionally cause neural networks to misclassify data across multiple domains and tasks. Our comprehensive review of the extensive and growing research on adversarial attacks has revealed a notable dearth of research in the domain of NLP. This research presents a comprehensive examination of current textual adversarial attacks and their comprehension from many angles in the field of Natural Language Processing. We have created three innovative techniques for adversarial attacks on text, as well as a strategy for defending against such attacks. Additionally, we will end by examining the potential areas for future research in the domain of adversarial machine learning specifically in the textual realm.

The investigation illustrates that linguistic frameworks have an inherent vulnerability to adversarial texts, where a few words or characters are altered to create perturbed text that misleads the machine into making incorrect predictions while preserving its intended meaning among human viewers. The present study introduces *HOMOCHAR, Non-Alpha-Num* & *Inflect-Text,* novel approaches for attacking text that works at character and word level granularity in a situation where the inner workings of the system are unknown. The objective is to deceive a specific neural text classifier while following specified language limitations in a manner that makes the changes undetectable to humans. Extensive investigations are carried out to evaluate the viability of the proposed attack methodologies on various often utilized frameworks, inclusive of Word-CNN, Bi-LSTM, and various advanced transformer models across different benchmark text datasets: AG news, MR, IMDb, Yelp, etc which are commonly employed for text classification tasks. Experimental proof demonstrates that the suggested attack architectures regularly outperform conventional methods by achieving much higher attack success rates (ASR) & generating better adversarial examples. The findings suggest that neural text classifiers can be bypassed, which could have substantial ramifications for existing policy approaches.

For our subsequent strategy, we conducted a comprehensive assessment and examination of the performance of several models across a range of attack scenarios to identify their relative levels of vulnerability, identifying the most and least susceptible ones. Furthermore, we ascertain the perturbation strategy that has the greatest impact on these classifiers. Lastly, we introduced a novel system called *Adversarial Robust Generalized Network (ARG-Net)* that aims to protect against word-level adversarial assaults. ARG-Net improves the model's performance by using both adversarial training and data perturbation techniques during the training process. The results of our tests on two datasets demonstrate that the model, which is built upon our framework, successfully mitigates word-level adversarial assaults. Furthermore, our model exhibits superior accuracy on the standard testing set compared to current defense techniques. The accuracy is comparable to, or even surpasses, that of the conventional model.

In conclusion, this thesis presents substantial discoveries and identifies potential areas for future research on the subject of adversarial machine learning in the text domain.

LIST OF PUBLICATIONS

Publications Arising from Research Work in this Thesis

SCIE Journal Papers

- **A. Bajaj** and D. K. Vishwakarma, "HOMOCHAR: A novel adversarial attack framework for exposing the vulnerability of text based neural sentiment classifiers," ***Engineering Applications of Artificial Intelligence***, vol. 126, Nov. 2023, doi: 10.1016/j.engappai.2023.106815.
- **A. Bajaj** and D. K. Vishwakarma, "A state-of-the-art review on adversarial machine learning in image classification," ***Multimedia Tools & Applications***, 2023, doi: 10.1007/s11042-023-15883-z.
- **A. Bajaj** and D. Kumar Vishwakarma, "Evading text-based emotion detection mechanism via adversarial attacks," ***Neurocomputing***, vol. 558, p. 126787, Nov. 2023, doi: 10.1016/j.neucom.2023.126787.
- **A. Bajaj** and D. K. Vishwakarma, "Non-Alpha-Num: a novel architecture for generating adversarial examples for bypassing NLP-based clickbait detection mechanisms," ***International Journal of Information Security***, 2024, doi: 10.1007/s10207-024-00861-9.
- **A. Bajaj** and D. K. Vishwakarma, " Bypassing Neural Text Classification Mechanism by Perturbing Inflectional Morphology of Words." *Under Minor Revision* in ***Neural Networks,*** June 2024.

Conference Papers

- **A. Bajaj** and D. K. Vishwakarma, "ARG-Net: Adversarial Robust Generalized Network to Defend Against Word-Level Textual Adversarial Attacks," in *2024 IEEE **9th International Conference for Convergence in Technology** (I2CT)*, Institute of Electrical and Electronics Engineers (IEEE), Jun. 2024, pp. 1–7. doi: 10.1109/i2ct61223.2024.10543623.
- **A. Bajaj** and D. K. Vishwakarma, "Deceiving Deep Learning-based Fraud SMS Detection Models through Adversarial Attacks," in *Proceedings - **17th International Conference on Signal-Image Technology and Internet-Based Systems**, SITIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 327–332. doi: 10.1109/SITIS61268.2023.00059.
- **A. Bajaj** and D. K. Vishwakarma, "Bypassing Deep Learning based Sentiment Analysis from Business Reviews," in *2023 **2nd International Conference on Vision Towards***

Emerging Trends in Communication and Networking Technologies *(ViTECoN)*, IEEE, May 2023, pp. 1–6. doi: 10.1109/ViTECoN58111.2023.10157098.

- **A. Bajaj** and D. K. Vishwakarma, "Exposing the Vulnerabilities of Deep Learning Models in News Classification," in *2023 4th* ***International Conference on Innovative Trends in Information Technology*** *(ICITIIT)*, IEEE, Feb. 2023, pp. 1–5. doi: 10.1109/ICITIIT57246.2023.10068577.

Publications Arising from Research Work Outside this Thesis

SCIE Journal Papers

- **A. Bajaj**, A. K. Ghosh, D. K. Vishwakarma. " Text-Muddler: A Novel Adversarial Attack Framework for Fooling Sentiment Analysis from App Reviews" Under Review in ***International Journal of Information Security***, May 2024.

- S. Aggarwal, **A. Bajaj** and D. K. Vishwakarma, " HOMOGRAPH: a novel textual adversarial attack architecture to unmask the susceptibility of linguistic acceptability classifiers" ***Accepted*** in ***International Journal of Information Security,*** August 2024.

- N. Gautam, A. Gupta, **A. Bajaj**, D. K. Vishwakarma. " A Robust Framework for Person Re-Identification against Adversarial Attacks" Under Review in ***Intelligent Systems***, April 2024.

Conference Papers

- R. Sharma, A. Ghosh, **A. Bajaj**, and D. K. Vishwakarma, "BEACOMP: A Novel Textual Adversarial Attack Architecture for Unveiling the Fragility of Neural Text Classifiers," in *4 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, Institute of Electrical and Electronics Engineers (IEEE), Jun. 2024, pp. 161–166. doi: 10.1109/incacct61598.2024.10550990.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

In recent years, deep neural networks have been increasingly popular in several Artificial Intelligence (AI) fields, including Computer Vision, Natural Language Processing, Web Mining, and Game Theory, owing to the advancements in high processing devices. Nevertheless, the comprehensibility of deep neural networks remains inadequate due to their functioning as black-box systems, making it challenging to derive insights into the specific knowledge acquired by each neuron[1]. An issue associated with poor interpretability is the assessment of the resilience of deep neural networks. A recent study has utilized subtle, imperceptible perturbations to assess the resilience of deep neural networks and has discovered that these networks are not resilient to such perturbations. It is initially assessed the cutting-edge deep neural networks employed for classification by subjecting the input data to minor produced perturbations. It was discovered that the neural classifier was easily deceived, but human judgment remained unaffected. The altered, nearly undetectable inputs were labelled as adversarial instances, and this terminology is subsequently employed to encompass all types of modified samples in a comprehensive manner. The emergence of adversarial examples has prompted extensive research on assessing the resilience of neural classifiers. This research can be categorised into three main areas: **i)** assessing deep neural networks by deceiving them with imperceptible perturbations, **ii)** deliberately altering the output of deep neural networks, and **iii)** identifying the vulnerabilities and excessive stability of deep neural networks and developing defensive measures against attacks.

1.1 Growing Applicability vs Robustness of Machine/ Deep Learning Models

Deep learning, a subset of machine learning and artificial intelligence, is widely recognised as a fundamental technology in the current era of the Fourth Industrial Revolution (4IR or Industry 4.0). DL technology, derived from artificial neural network (ANN), has gained significant attention in the field of computers due to its ability to learn from data. It is extensively utilised in diverse domains such as healthcare, visual identification, text analytics, cybersecurity, and more. In **Figure 1.1**, we have compiled a summary of various potential practical domains where deep learning can be applied. In summary, based on the information shown in **Figure 1.1**, it can be inferred that deep learning modelling has significant potential for future applications in real-world settings, offering several opportunities for further exploration and development.

Therefore, DL techniques have the potential to be crucial in constructing intelligent data-driven systems that align with current requirements. This is due to their exceptional ability to learn from past data[2], [3]. However, we still lack complete confidence in deploying these models in the real world due to the presence of adversarial methodologies.

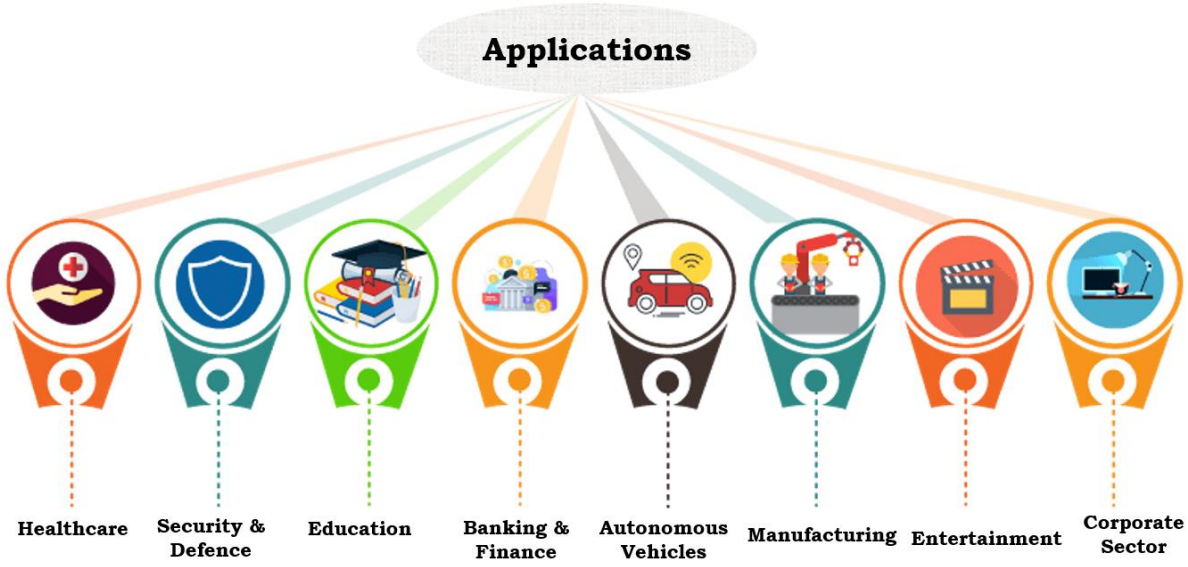


Figure 1.1 Applications of Machine/ Deep Learning Models in Different Domains

This research seeks to answer the essential issue, "Are our models sufficiently accurate to be employed in the real world?" Whether the dataset is skewed or the models are flawed. Do naturally generated adversarial instances also exist? And how can we adapt our models to make them resistant to samples that have been perturbed? How can model-stealing attacks be prevented? These are all the questions we evaluated, to which we will react in distinct sections. Our research found that neural network layers and hyperparameters may influence the model's training accuracy. They do not reveal the network's robustness to adversarial examples. This discovery proved to be the fundamental motivation for our investigation of adversarial attacks and their countermeasures as an effort to comprehend the weaknesses of deep network models. **Figure 1.2** illustrates the fragility in various stages of a supervised model and the corresponding defensive frameworks, which is to provide a broader understanding of the operations of DNN in an adversarial framework. This problem was selected to provide researchers with a thorough knowledge of technical formalities and crucial concepts related to adversarial machine learning[4]. In this survey[5], all pertinent and significant works relevant to the research subject are cited. The uniqueness of this survey lies in the thorough and systematic combination of the existing knowledge in this field of study[5], [6], [7].

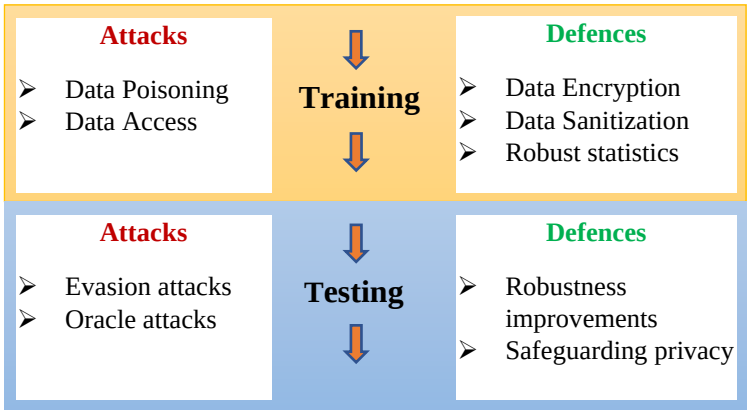


Figure 1.2 A glimpse of attacks and defences in pipeline

1.2 Brittleness in Different Phases of ML Pipeline

This section will analyse the likelihood of intentional alterations during the training and testing phases of the entire system, which serve as the foundation for carrying out different types of attacks. The research topic known as Adversarial Machine Learning focuses on identifying weaknesses and addressing them to strengthen the overall resilience of the system. The following will highlight the susceptibilities in ML pipeline.

There are two dominant stages in the supervised learning framework, i.e., training & testing. After that, the model gets deployed. Malicious activities can be conducted in each phase, as shown in **Figure 1.3**, which can deceive predictions by disturbing the learning procedure. Evasion attacks exist during the testing phase, data poisoning & data access is an issue in the training phase[8], and after deployment, oracle attacks can be conducted. All these types of attacks are described in depth in subsequent sections.

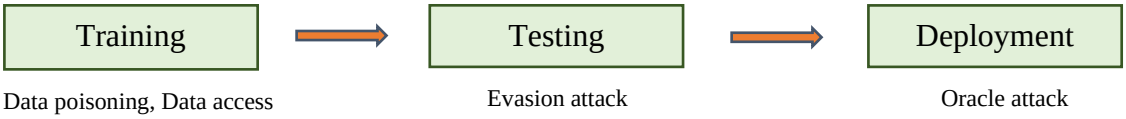


Figure 1.3 Stages on which attacks can be performed

1.2.1 Vulnerability in the Training Phase

With advancements in machine learning to attain more accurate and precise results, more and more data is needed; basically, deep learning is known as “infinite data-hungry.” But the source

from where the dataset is taken cannot be trusted blindly, which results in introducing the regime of data poisoning.

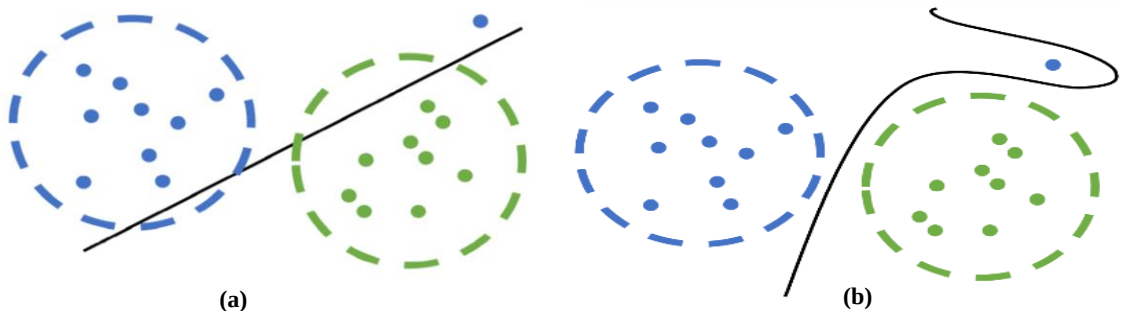


Figure 1.4 (a) Hampers generalization for ML classifier and **(b)** learns unnatural (outliers) by deep learning classifier[107]

The primary objective of data poisoning in machine learning is to hinder a training set by just varying a small fraction (by adding outliers), which hampers generalization. It is considered a fundamental problem in classical ML models. Still, the same problem does not occur in deep learning classifiers because it "memorizes the unnatural examples," which can be inferred from **Figure 1.4**. The problems confronted in deep learning are much worse[9]. Data poisoning in deep learning affects a specific class of input, i.e., just by manipulating a single class entity, whole classes of prediction are altered, which can be seen in **Figure 1.5**.

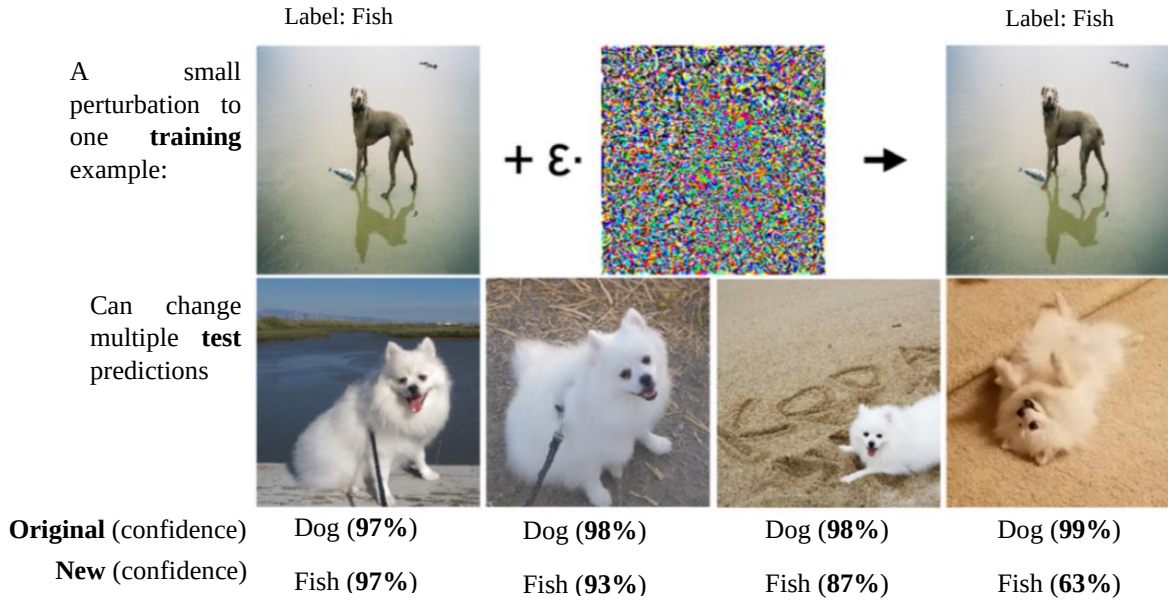


Figure 1.5 A single "poisoned" input can manipulate many predictions[108]

By inserting an innocuous or unnoticeable figure within an image, as depicted in **Figure 1.6**, the machine will predict the image according to the goals of the attacker irrespective of true

class prediction. It can also be considered an undetectable back door entry by an attacker with complete access to a confidential dataset.

Figure 1.6 Can plant an undetectable backdoor that gives an almost total control over the model[10]

1.2.2 Vulnerability in the Testing phase

In the supervised learning framework, it is usual practise to randomly partition the available data into two groups, training and test. The training set is used to help the model choose a decision rule, or to construct a decision boundary for each class, while the test set is used to evaluate the model's performance on different dataset/unseen samples. Mathematically, the notion of similarity between the training and test set is the assumption that the samples in both sets are drawn from the same available data.

However, when the test set is taken apart from available data, i.e., out-of-distribution data[11], the two distributions will not be the same, as visible in **Figure 1.7** (Actual distribution)[12]. This is because many covariant shifts occur continuously until their deployment. The test samples perceived by the model are not always from the same distribution on which it is trained. From this limitation in the supervised learning framework, it can be inferred that predictions made by the machine are accurate but brittle. From this limitation in the supervised learning framework, it can be inferred that predictions made by the machine are accurate but

brittle. This issue forms the basis of the attacks in which minute manipulation of an image at the test time can cause a shift in the decision boundary of a particular class.

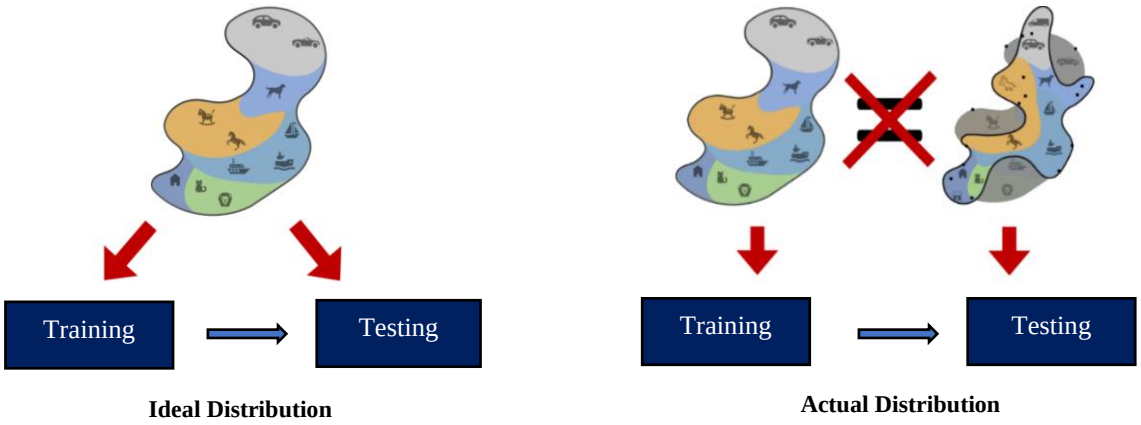


Figure 1.7 Difference between ideal and actual distribution

1.2.3 Vulnerability after Deployment

Securing the model only during the training and inference phase is insufficient for protection against attacks. Limited access to internal model metrics such as its confidential training set, weights, bias, and other parameters, depicted in **Figure 1.8**[13], does not assure security. Discrepancies can also occur when the model gets deployed[14]. Access to the input-output pairings & class probabilities only is enough for copying a model (substitute model), known as an oracle (model stealing or query-based) attack.

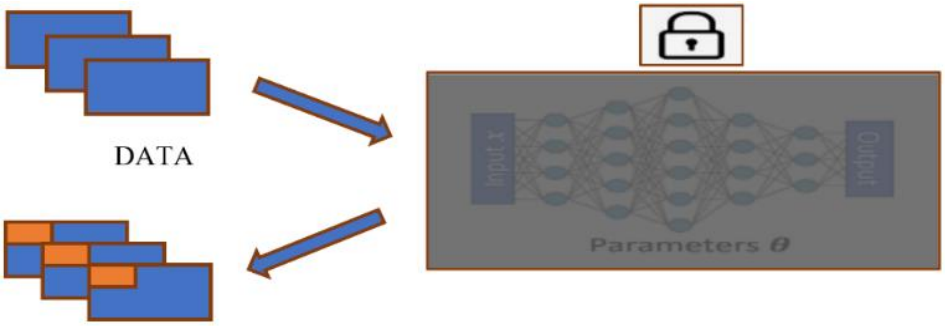


Figure 1.8 Query-based Black-Box Attack when all parameters of the model are Encrypted[109]

The attack technique entails training a local model to substitute for the target DNN using adversary-generated synthetic inputs, labelled by the target DNN, papernot et al.[15] use the local substitute to craft adversarial examples and find that ty misclassify the targeted DNN.

1.3 Adversarial Attacks in Different Phases of ML Pipeline

1.3.1 Attacks during the Training phase:

The attack during training is divided into two categories, i.e., **Data Poisoning**: It includes all the methodologies influencing the training data or model. **Data access**: Access to the confidential training set and inputs with their corresponding outputs may lead to model stealing as illustrated from **Figure 1.9**.

Data poisoning: Data poisoning includes *Inadequate data injection*, *Logic corruption*, *Backdoor attacks* & *Data manipulation*. All divisions provide a crisp idea of the manipulations made, which can disrupt the learning process of the model.

- *Inadequate data injection*: To make any decision by the model, it has to grasp only valid inputs. This will provide improvisation in each step for precise results. But if an attacker feeds deceitful inputs (inadequate inputs) that are not relevant to the corresponding output & trains on it, it emanates degradation of accuracy resulting in misclassification. Inadequate data can be injected before pre-processing, known as direct poisoning, and it can be after pre-processing of the input data, known as indirect poisoning.
- *Logic corruption*: When an attacker has the ability to alter both the algorithm and how it learns, logic corruption occurs. The machine learning phase becomes irrelevant at this stage as an attacker can encode any logic. This can disturb the learning process resulting in absurd predictions.
- *Backdoor attacks*: Backdoor attacks can be conducted by inserting a perceptible but unobjectionable pattern or watermark in an input image, or it can be imperceptible (random-looking noise) to be embedded as a backdoor pattern. These poisoned backdoor samples are then given target (backdoor) class labels resulting in accomplishing targeted attacks, which can be inferred from **Figure 1.6**.
- *Data manipulation*: The dataset's source cannot be trusted blindly as there may be manipulations in the inputs & their corresponding labels, which can be inferred from **Figure 1.5**. *Input manipulations* disturb the class distributions in a way that there is an interference of different norm bounds sharing a particular boundary. *Label manipulation* is subdivided into two *typical* and *atypical*. For example, in the classification of bird type, if any label is poisoned to be a vehicle as it doesn't have the same feature density as that particular class of bird, known to be *atypical label*

manipulation, if the samples have the same feature density, i.e., falling under the category of the similar domain but mislabeled are *typical label* manipulations.

Data access: This attack does not seek to change the classifier's decision-making. Instead, it deals with losing confidential information, which an attacker can further utilize to perform malicious activity by having access to a set of inputs & outputs. An adversary attempts to get the confidential data used to train a supervised neural network to access data. A successful attack should produce realistic samples with a wide range of characteristics representing each class in the private dataset. In these attacks, a classifier is questioned to determine its decision rule or to discover information regarding the training set. Tramèr et al. emphasize the contradiction between model privacy and public access; an attacker with black-box access and no previous knowledge of the settings or training samples of a machine learning model attempts to mimic (i.e., "steal") the model's functionality. This attack type generally leads to the loss of sensitive data from the dataset or the models. The malevolent party can use the stolen information for malicious purposes.

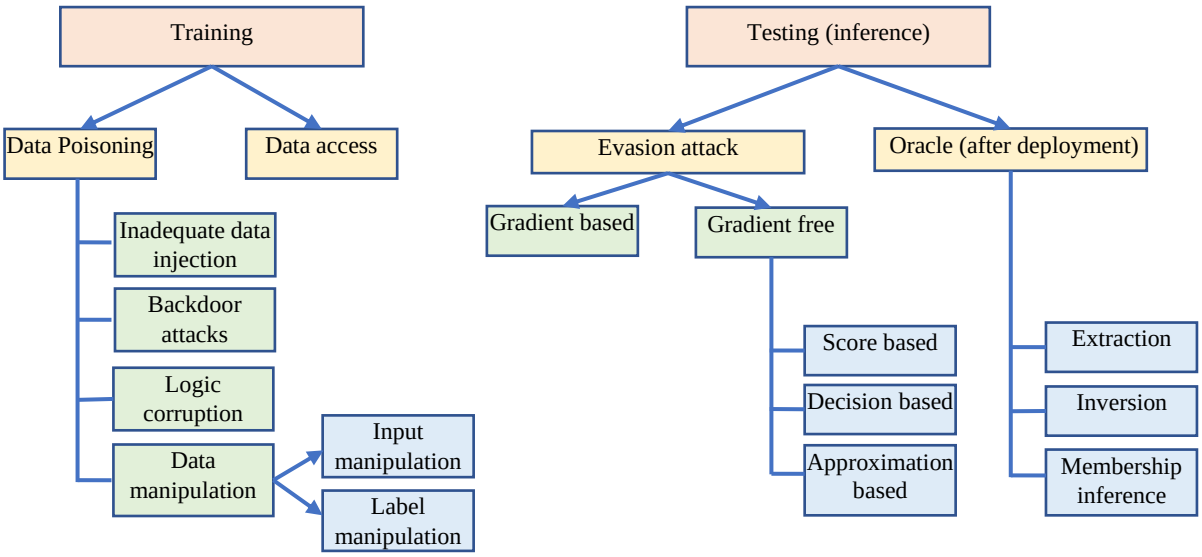


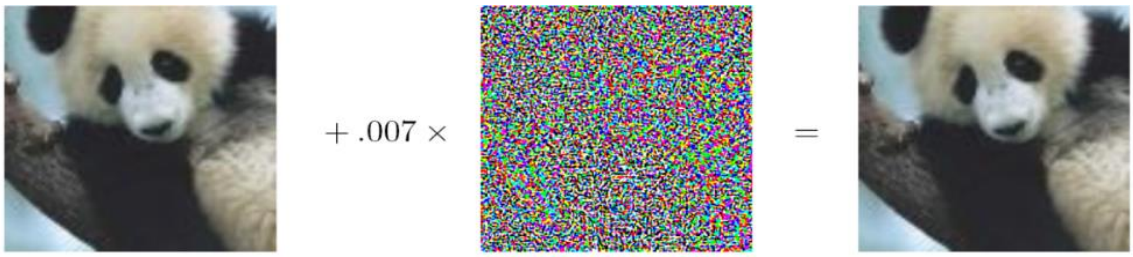
Figure 1.9 Taxonomy of adversarial capabilities during training and testing stage

1.3.2 Attacks during the Testing phase:

During the testing phase, attacks are separated into two categories., **Evasion attacks** & **Oracle attacks**.

Evasion: In adversarial learning, evasion attack is recognized to be the most popular attack method. The percept is to evade the classifier from its true class prediction. Evasion attack may alternatively be Gradient-based or Gradient-free.

- *Gradient-based:* In the gradient-based strategy, the adversary tackles a constrained optimization problem to identify a minor input perturbation that results in a significant shift in the loss function. These "*perturbations are then inserted into clean test samples to form* ***adversarial images (adversarial examples)*** *which, when fed to classifier at test time, result in output misclassification.****"*** In order to generate the perturbations[16], the



x
"**Panda**"
57.7% Confidence

$sign(\nabla_x J(\Theta, x, y))$
"**nematode**"
8.2% Confidence

$x + \epsilon\ sign(\nabla_x J(\Theta, x, y))$
"**gibbon**"
99.3% Confidence

Figure 1.10 Adversarial image created by gradient based the Fast Gradient Sign Method (FGSM)[110]

gradient of the loss with regard to the input data is calculated. The gradient-based method needs access to full model information. This helps in evading its actual class as shown in **Figure 1.10**.

- *Gradient-free*: In this attack, the adversary does not require direct access to the gradient or the entire model's data. These are conducted even by gaining access to a limited set of parameters[17]. It includes *Score-based, Decision-based* & *Approximation-based* procedures.

- Score-based*:* This strategy generally performs in a black-box adversarial scenario. The intruder only requires information of the training set and the scores (class probabilities or logits) acquired by examining the actual model. This substitute model will construct perturbations injected into clean samples to generate adversarial examples. These

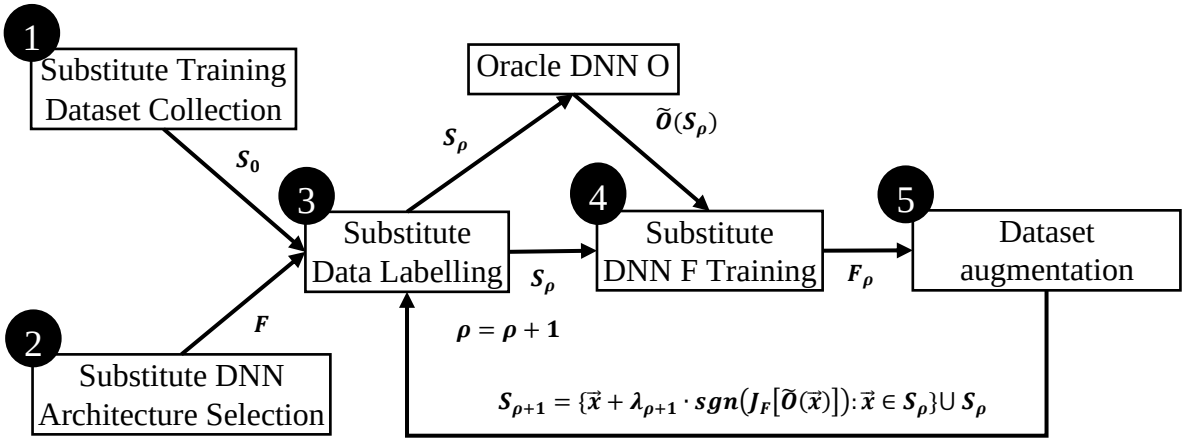


Figure 1.11 Framework for training of substitute model[111]

obtained adversarial instances are then utilized to deceive the real model by leveraging its transferability attribute.

The framework for training a substitute classifier is shown in **Figure 1.11**. Training of the substitute DNN F: the attacker (1) collects an initial substitute training set S_0 and (2) selects an architecture F. Using querying $\tilde{O}$, the attacker (3) labels S_0 and (4) trains substitute F. After (5) Jacobian-based dataset augmentation, steps (3) through (5) are repeated for several substitute epochs ρ.

- Decision-based*:* The technique is meant to iteratively modify the pixels of test images in such a manner that it prevents the image from reaching the properly categorized border by rejecting those images that sit on and inside the class boundary of the initial image samples, **Figure 1.12**. The technique is known as rejection sampling. It is considered a simple and efficient approach compared to the gradient-based method because it requires little manipulation of parameters.

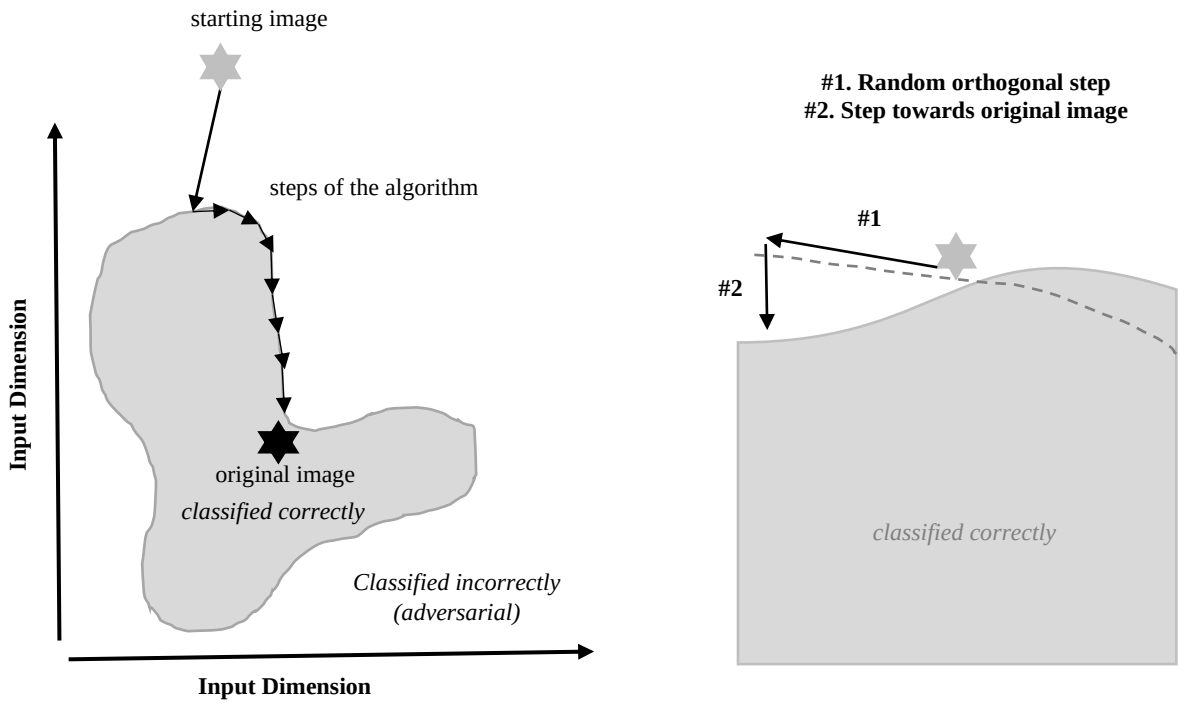


Figure 1.12 Randomly initializing a point already present in adversarial region which is always rejected upon reaching the boundary between original and adversarial region, such that it stays in adversarial region

- Approximation-based*:* Based on this method, algorithms like BPDA and EOT, respectively, utilize a differentiable function that, either in a model or defense, substantially simulates the outputs provided by a non-differentiable layer. The gradient-based attacks can then use this approximated output to carry out the evasion.

Oracle attacks*:* Oracle attacks are mainly conducted under a black-box adversarial setting, the attacker doesn't have information regarding internal model metrics, but by just having input-output pairings & class probabilities, attacks are conducted. Oracle attacks are sub-categorized into three, i.e., *Extraction*, *Model inversion* & *Membership inference.*

- *Extraction:* In extraction, the adversary can extract model information by observing its prediction from the set of inputs. The attackers aim to build a surrogate model with the reverse engineering approach. The surrogate model closely approximates the target model. Using gradient information of the surrogate model, the attacker can generate adversarial samples from the surrogate model. These adversarial samples are then used to fool the actual model using the property of transferability of adversarial examples.
- *Model inversion:* In a model inversion, the objective is to rebuild input data used in model training by only having access to its limited parameters and output labels under black-box adversarial settings.
- *Membership inference:* This attack approach checks whether an input sample is present in the training set. The technique used here is brute force, i.e., by feeding sample input to check its presence. If it gives satisfying output according to the attacker, it confirms its presence in the training set. This attack methodology is known as Membership inference.

1.4 Defence Techniques against Adversarial Attacks

Defences can be classified based on whether they are applicable to attacks targeting the training or testing (inference) stages of system functioning. Defensive techniques in both scenarios can frequently result in performance overhead and negatively impact model accuracy. **Figure 1.13** is a processing flow chart that showcases various attack and defence methods in a machine learning pipeline.

1.4.1 Defences against Attacks during Training phase:

Defences during the training phase are divided into three categories, i.e.***, Data encryption***, ***Data Sanitization*** *&* ***Robust Statistics***. Defences during the training help to resist the attacks that cause availability violations, i.e., against data poisoning and data access.

Data encryption*:* Various ML service providers are available online, e.g., Google Cloud AI, AWS, BigML, Microsoft Azure, Clarifai, Face++, and IBM Bluemix, in which users provide their confidential data to avail ML-based predictions, but such services entail serious privacy

issues as an eavesdropper or intermediary can steal and misuse their data which may lead to fatal consequences. Data encryption was introduced to preserve data privacy. It works on the principle of converting original data into cipher data to be accessible only to the user and service provider. Various data encryption techniques are raised in literature DeepSecure, CryptoNets [99], and CryptoNN[100] that support training a neural network model over encrypted data and processing it to make decisions.

***Data sanitization**:* In data sanitization, all malicious input samples (poisoned inputs) that create a negative impact during class distribution are removed immediately. Malicious input samples are identified by evaluating the influence of such examples on classification performance [18]. The inputs that cause a high error rate are removed from the training set, known as Reject on negative impact. But this defence can be easily broken as the attacker may produce "inliers," or poisoned points, that closely resemble the genuine data distribution and trick the model[19].

***Robust statistics**:* Robust statistics enhance generalization [20], which mitigates the impact of poisoned examples using constraints and regularization techniques instead of attempting to detect poisoned data (data sanitization). This suppresses potential distortions of the learning model caused due to poisoned samples [21],[22]. This estimation method is insensitive to minor deviations from the idealized assumptions used to improve the algorithm. In[23] a defence algorithm TRIM is introduced, which provides high resilience and robustness against a large class of poisoning attacks.

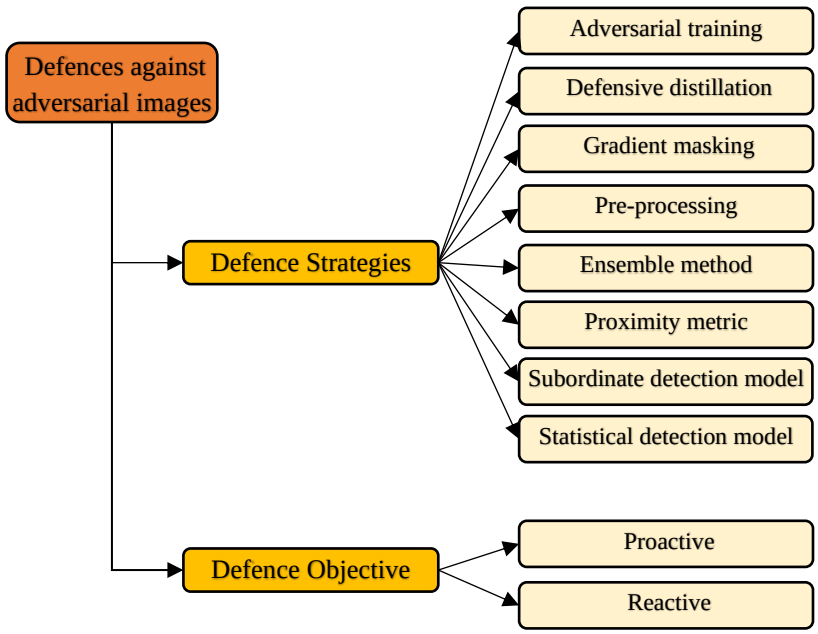


Figure 1.13 Defence strategies against adversarial attacks

1.4.2 Defences against Testing (Inference) Attacks:

Defence strategy includes two main objectives, i.e., **(a)** *Proactive* and **(b)** *Reactive***.** Proactive defences aim to "*correctly classify input samples even if they are perturbed*." On the other hand, reactive defences "*detect legitimate or adversarial images before it reaches the classifier*." Afterward, malicious images are either discarded or sent to the recovery phase. The defences against adversarial attacks are categorized in this part using a novel taxonomy, as shown in **Figure 1.13**. The defences against adversarial images are categorized into two, namely **(i)** defence objective and **(ii)** defensive strategy. On systematic analysis of defence strategies, the most relevant robustness improvement methods against adversarial examples/adversarial images include Adversarial training, Defensive distillation, Gradient masking, Pre-processing techniques, Ensemble method, Proximity metric, Subordinate detection models.

Adversarial training*:* Adversarial training is considered the first defensive technique against adversarial attacks [24], [25]. This defence method incorporates training on a hybrid dataset containing legitimate and as well as adversarial images in training and trains it on the corresponding true label, as shown in **Figure 1.14**. This provides robust classification against the particular attack type on which it is trained. Adversarial training is also known as the brute-force defence method. However, this technique is not comprehensive against all adversarial

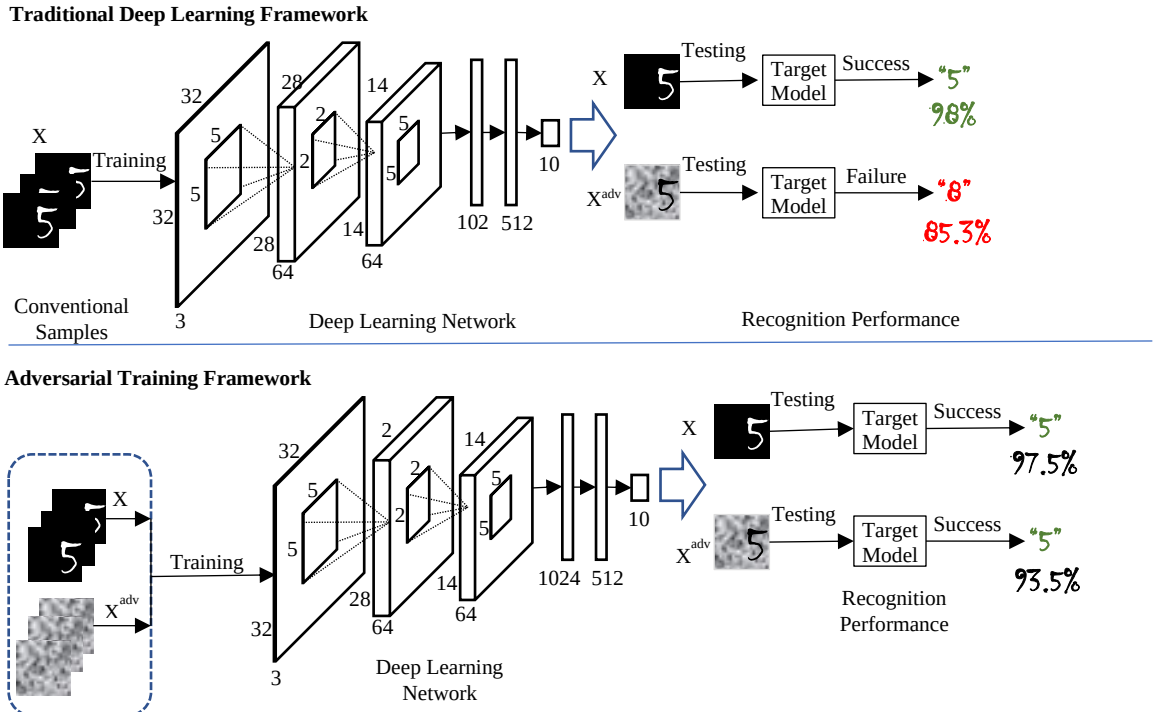


Figure 1.14. Framework for Adversarial training methodology for building a robust classifier[112]

attack algorithms, as it has to be trained on different adversarial images obtained through different attack algorithms, which is not feasible.

***Gradient masking**:* Adversarial attack forms need to calculate the gradient with respect to inputs in order to devise a perturbation vector to generate the adversarial image. Gradient masking is a defence strategy that hides or masks the gradients with respect to the inputs. Gradient masking (also known as obfuscated gradient [26]) results in models with smoother gradients, which prevents optimization-based attack algorithms from finding the wrong directions in space [27], that is, without useful gradients for producing adversarial examples. Defences based on gradient masking can be divided into (i) *Shattered gradients:* non-differentiable defences lead to shattered gradients, which introduce false or non-existent gradients, (ii) *Stochastic gradients:* Stochastic gradients are produced by randomised defences, in which the network is either randomly generated or the input is randomly altered before being fed to the classifier. This leads to inaccurate estimation of the true gradient by methods that employ a single sample of the randomness, and (iii) *Exploding/vanishing gradients:* Gradients that are exploding or vanishing are a result of defences built by very deep architectures, which typically include several iterations of neural network evaluation in which the output of one layer is provided as an input to the next layer.

These techniques facilitate updating model parameters by altering the gradient of input samples and activation functions, which tend to un-reveal the true gradient. This technique is hampered by gradient masking since it results in sharper decision boundaries, as shown in **Figure 1.15**.

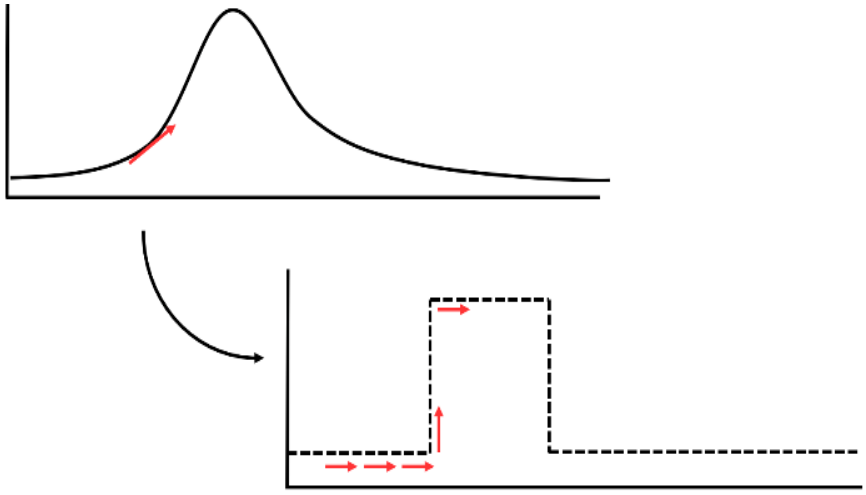


Figure 1.15. Sharper decision boundaries are made possible by using gradient masking, which obfuscates adversarial cases.

***Defensive distillation**:* Defensive distillation is a proactive defensive technique developed by Papernot et al. [28]. Motivated by gradient masking, a target model is used in defensive distillation to train a smaller model that exhibits a smoother output surface. In [28], the model contains a dataset $\boldsymbol{x}$ as input samples in the training set with their corresponding labels as, generally, in one hot encoding format with specific temperature t. After training the model on input, samples produced a probabilistic vector set $\boldsymbol{f(x)}$. A model $\boldsymbol{f^d}$ have the same architecture created and trained with the same input samples $\boldsymbol{x}$ but now using the label set as $\boldsymbol{f(x)}$ and at the end of the training, the distilled output is produced $\boldsymbol{f^d(x)}$ .It reduces the model sensitivity to smaller perturbations. Hence, this approach induces to feed output model to retrain on the smaller model to distill extensive features to foremost crucial ones exhibiting much smoother output surfaces. The framework of this defensive mechanism is shown in **Figure 1.16**.

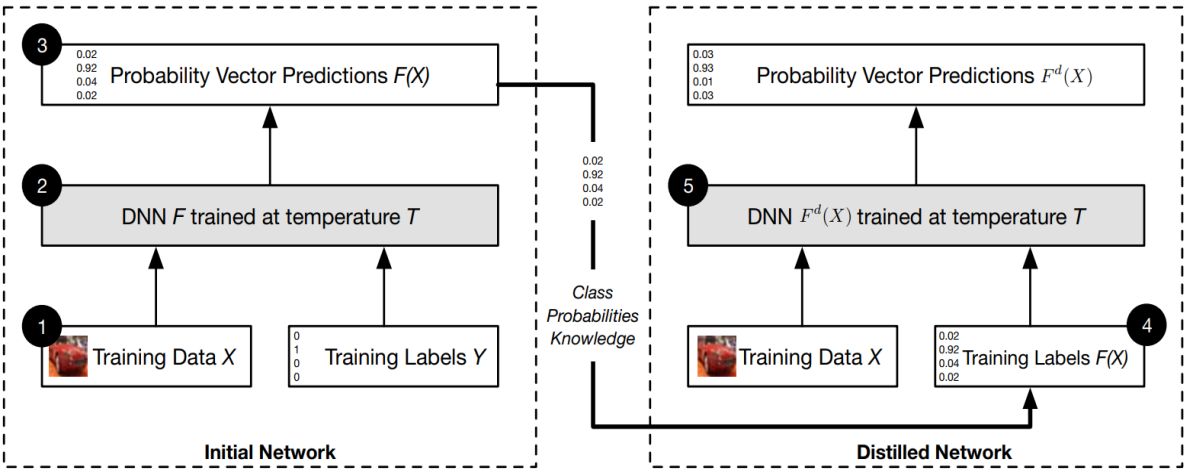


Figure 1.16. Architecture for defensive distillation technique[28]

Proximity metric: Papernot et al. [29] introduced the proximity technique by creating DkNN (Deep k-Nearest Neighbour). Here, the hybrid classifier uses the k-nearest neighbors' algorithm to aggregate the data with similar representations learned by each layer of the DNN. The group of similarly represented data in the data manifold is assigned with the same ground truth. This technique helps to enhance the generalization of the inputs outside the training data manifold. It includes adversarial examples as well. Adversarial samples are misclassified for conventional DNN architectures because one of its layers alters the input representation, which was initially in the correct class. The framework of the proximity metric is shown in **Figure 1.18.**

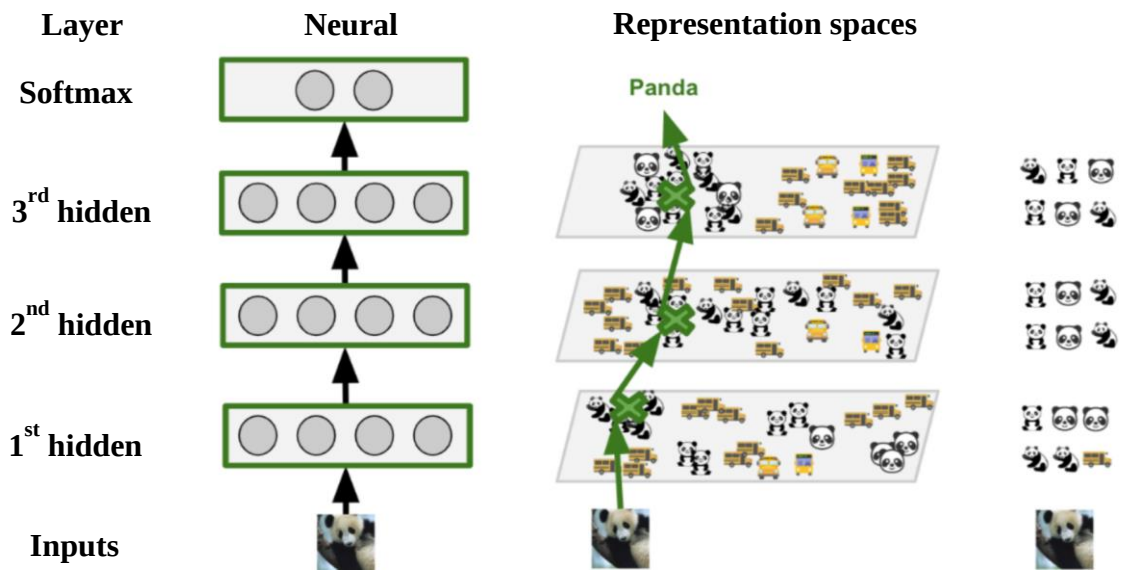


Figure 1.18. Proximity metric technique, using Deep k Nearest Neighbours method in order to compute most proximal class from training samples over internal representation spaces [29]

Pre-processing techniques: Methods in which the defender has to make use of the pre-processing techniques. It includes GANs and autoencoders, which inhibit an input sample and move it toward the closest legitimate sample in the training set can be seen from **Figure 1.19.** Similarly, techniques of dimensionality reduction using feature squeezing for smoothing input features as shown in **Figure 1.17**. Also, by adding noise layers and various image transformations to enhance the generalization of the model as illustrated from **Figure 1.20.**

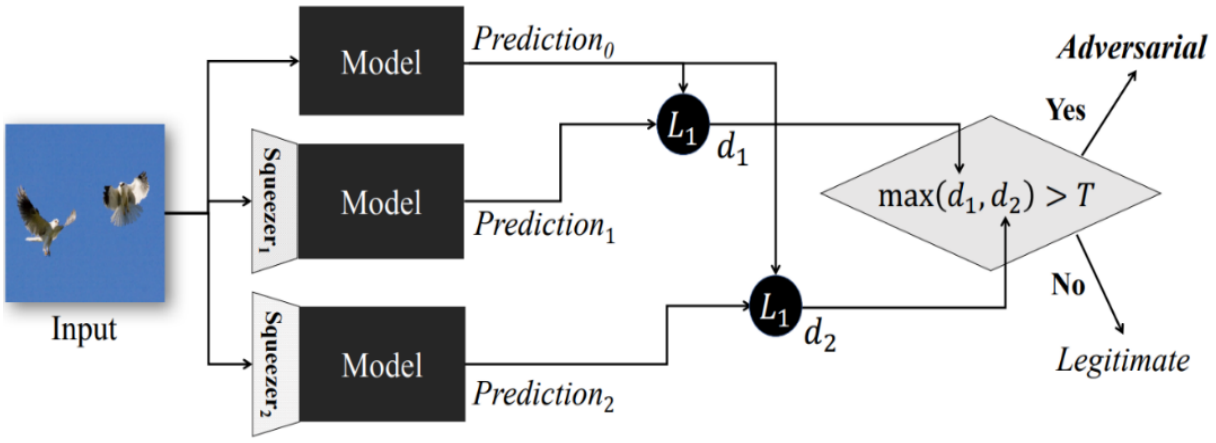


Figure 1.17. Pre-processing technique using feature squeezing for detection of adversarial images [113]

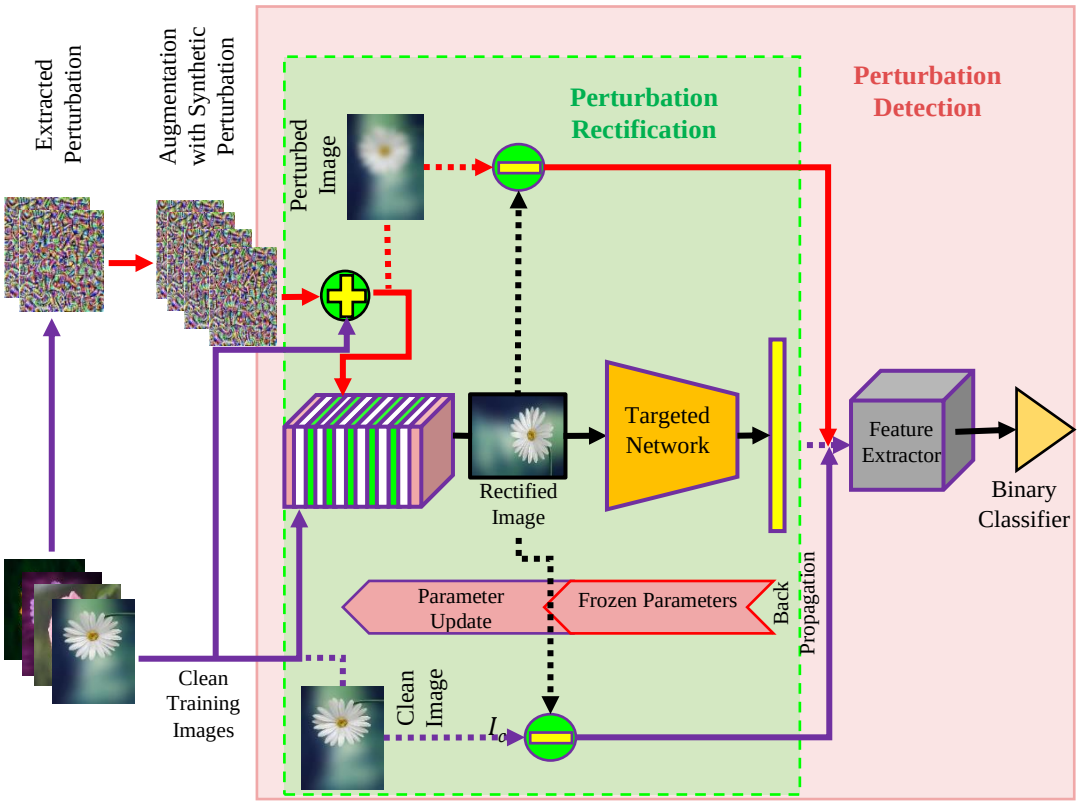


Figure 1.20. Defence methodology for Universal adversarial perturbation [30]

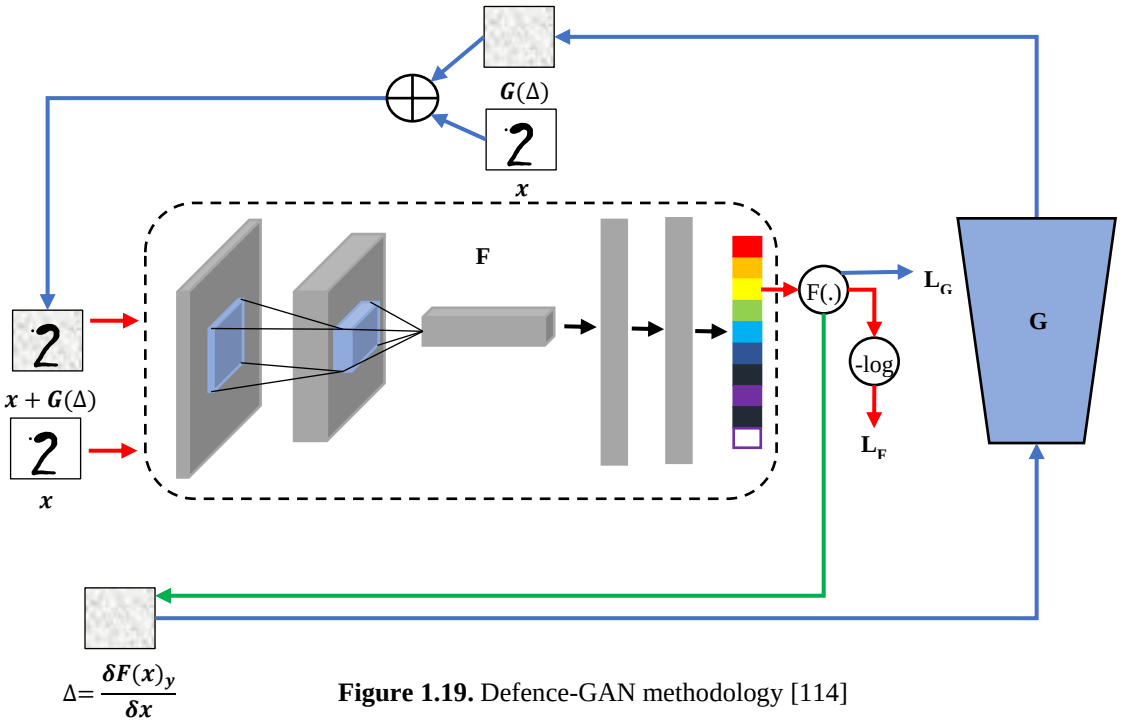


Figure 1.19. Defence-GAN methodology [114]

1.5 Adversarial Attacks in Natural Language Processing

The domain of adversarial machine learning has witnessed significant growth in recent years. The current body exists of a substantial body of scholarly work about the adversary interpretation of textual modelling but is still limited, indicating a scarcity of research in this area. In contemporary literature, **Figure 1.21** provides a visual representation of the prevalence of adversarial examples.

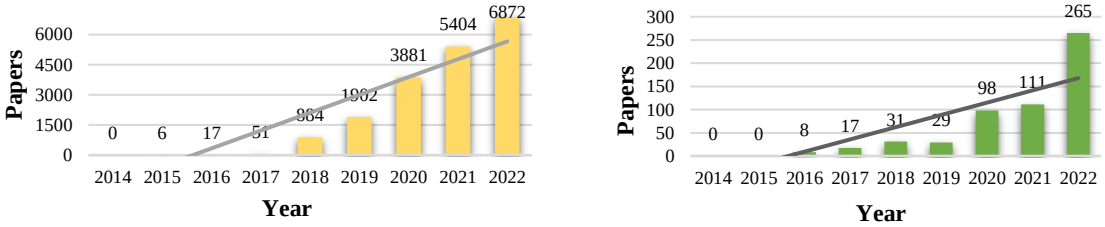


Figure 1.21 (a) Represents the number of publications in the field of adversarial example, as compiled by Carlini, including image, audio, and text across a broad spectrum. **Figure 21 (b)** depicts the number of publications in the adversarial text-domain.

Threats are being extensively investigated in numerous studies, specifically focusing on the field of computer vision. Nevertheless, there's a shortage of scholarly articles about the domain of textual analysis. Adversarial attacks have garnered significant attention in the realm of research, particularly in the domain of image manipulation. Therefore, in this study, we analyse prominent research papers in the field of NLP to ascertain the datasets employed within the textual context. **Table 1.1** showcases the specifications of the extensively employed datasets that have been utilised in various studies pertaining to adversarial attacks[31].

Table 1.1 The prior research utilizes twelve widely used text datasets in the examination of adversarial assaults, with a focus on the domains of Neural Machine Translation (**NMT**), Question and Answer (**QA**), and Natural Language Inference (**NLI**).

Problem	Title	Size	Specification	Dataset Link
Classification	SST	240T	The standard sentiment dataset from Stanford	https://github.com/stanfordnlp/sentiment-treebank
	MR	10T	Information extracted from movie review	http://www.cs.cornell.edu/people/pabo/movie-review-data/
	Yelp	140T	Customer testimonials of business reviews	https://huggingface.co/docs/datasets/index
	Amazon	2M	Amazon merchandise evaluation	
	Yahoo	1.4M	Yahoo! responds to detailed questions	
	IMDB	50T	different opinions about films	
	AG news	144T	More than 2,000 sources of news	
	DBPedia	45T	The organised information of Wikimedia projects	
NMT	WMT14	---	Texts that are next to each other (for example,	http://www.statmt.org/wmt14/translation-task.html

Problem	Title	Size	Specification	Dataset Link
			translation models) German/English)	
NLI	MultiNLI	433*T*	Crowdsourced collection of sentence pairs	https://cims.nyu.edu/~sbowman/multinli/
	SNLI	570*T*	Authors compose sentence pairs in the English language.	https://nlp.stanford.edu/projects/snli/
QA	SQuAD	100*T*	Data collection from Wikipedia for question answering and reading comprehension	https://datarepository.wolframcloud.com/resources/SQuAD-v1.1
T(Thousand), ****M***(Million)				

Figure 1.22 illustrates the distribution of percentages for the NLP assignments. A significant majority, over 50%, of databases are specifically allocated for the purpose of categorising textual data. This activity has considerable importance within the realm of NLP[31]. For almost a decade, IT giants and startups have invested in deep NLP. Predictive algorithms study human emotions in textual reviews to evaluate their services or products. In light of an urgent requirement, this study used various text classification datasets to demonstrate the fragility of text classification under adversarial situations. This motivates the authors to provide a comprehensive examination of the vulnerability of deep learning models for text classification to adversarial attacks through rigorous experimentation. To date, there has been a lack of comparative analysis among different deep learning models in terms of their ability to withstand adversarial attacks. This article is a novel contribution to the field, as it challenges the robustness of established cutting-edge deep learning models frequently used in NLP tasks. It offers valuable insights for readers who rely on these models and seek to enhance their understanding of their limitations.

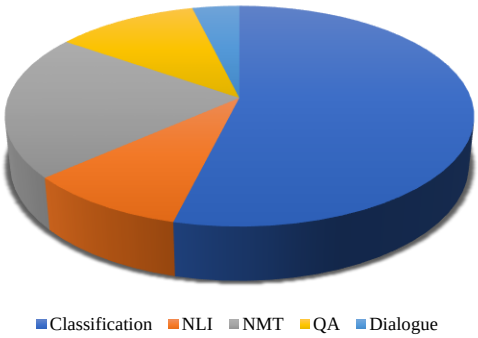


Figure 1.22 Statistics pertaining to datasets utilized in the study of adversarial attacks

Apart from their capacity to deceive the targeted models, the adversarial example must also fulfil three essential attributes that preserve their utility: 1.) semantic similarity—based on human interpretation, the generated instances should mean the same thing as the real one, 2.) Created adversarial examples should sound grammatical and natural. 3.) Human predictions should be consistent and remain constant. In natural language processing, adversarial instances can be produced by perturbing characters, words, and sentences, often referred to as sentence-level, word-level & character-level attacks [32],[33].

An adversarial attack involves intentionally modifying the input data of a neural network to

assess its ability to maintain its output under such conditions as shown in **Figure 1.23**. The present study involves a collection of n sentences, denoted as $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$, along with a corresponding set of n labels, $\boldsymbol{Y} = \{y_1, y_2, \ldots, y_n\}$. The textual input field $\boldsymbol{X}$ is linked to the label space $\boldsymbol{Y}$ through a pre-existing model known as $\boldsymbol{f}$**:** $\boldsymbol{X} \rightarrow \boldsymbol{Y}$**.** An authentic adversarial instance, denoted as $\boldsymbol{x_{adv}}$, pertaining to the expression $\boldsymbol{x} \in \boldsymbol{X}$, must satisfy the criteria outlined in **Eqn. (1.1)** and **Eqn. (1.2).**

$$f(\boldsymbol{x_{adv}}) \neq f(\boldsymbol{x}) \quad \textbf{(1.1)}$$

$$Sim(\boldsymbol{x_{adv}}, \boldsymbol{x}) \geq \in, \quad \textbf{(1.2)}$$

The symbol "$\in$" denotes the minimum degree of similarity between the adversarial and genuine samples, while the function Sim: $\boldsymbol{X}\times\boldsymbol{X} \rightarrow$ **(0, 1)** represents an analogy function. Sim **(.)** is a function which is frequently employed for the purpose of detecting similarities in both semantics and syntax within the realm of textual data. The intuition behind this is to consider $\boldsymbol{f}$ as a classification model which is trained with clean inputs and supposing x to be a valid input. Then it is modified from $\boldsymbol{x}$ to $\boldsymbol{x}$**'**, such that $\boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is the perturbation required for $\boldsymbol{x}$ to cross the decision boundary of true class entity, resulting in $f(\boldsymbol{x}) \neq f(\boldsymbol{x'})$.

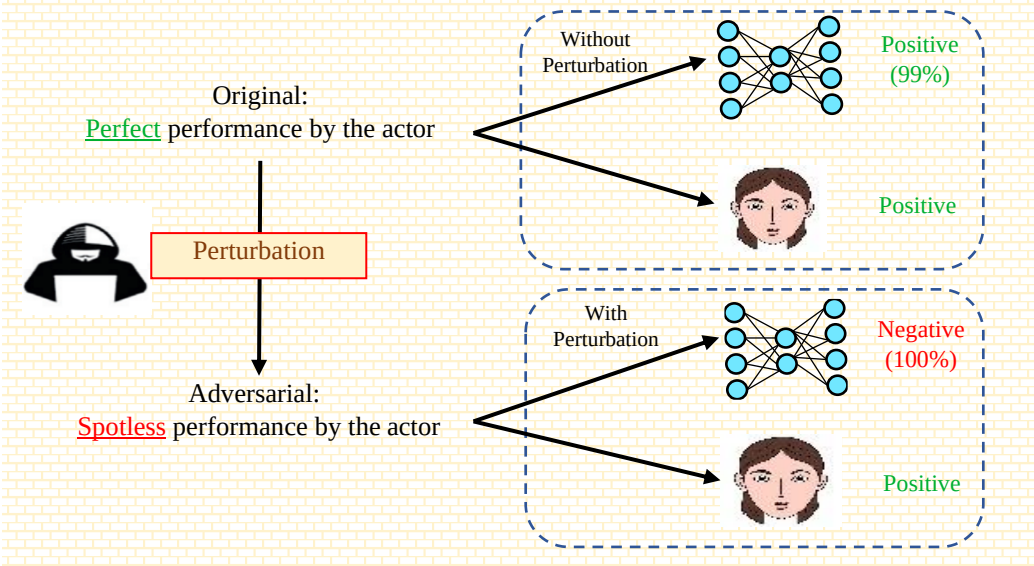


Figure 1.23 Despite maintaining semantic similarity for human readers, the adversarial example produced by word perturbation tricks the Bert-based sentiment classifier into producing the incorrect results [34].

The following research works form the basis of this chapter:

- **A. Bajaj** and D. K. Vishwakarma, "A state-of-the-art review on adversarial machine learning in image classification," ***Multimedia Tools & Applications***, 2023, doi: 10.1007/s11042-023-15883-z.

1.6 Overview of Chapters

The remaining section of the document is structured in the following manner.

- **Chapter 2** The literature review examines the current cutting-edge techniques for textual adversarial attacks and their effects, as well as a few defensive strategies.
- **Chapter 3** Describes the most potent textual adversarial attack methodologies for deceiving text classification mechanisms.
- **Chapter 4** Comparing deep learning approaches to textual adversarial attacks to determine which models are robust and which are fragile, and which perturbation method poses the most threat.
- **Chapter 5** Discusses a crucial defensive mechanism for obfuscating textual adversarial attacks.
- **Chapter 6** Conclusion & Future scope.

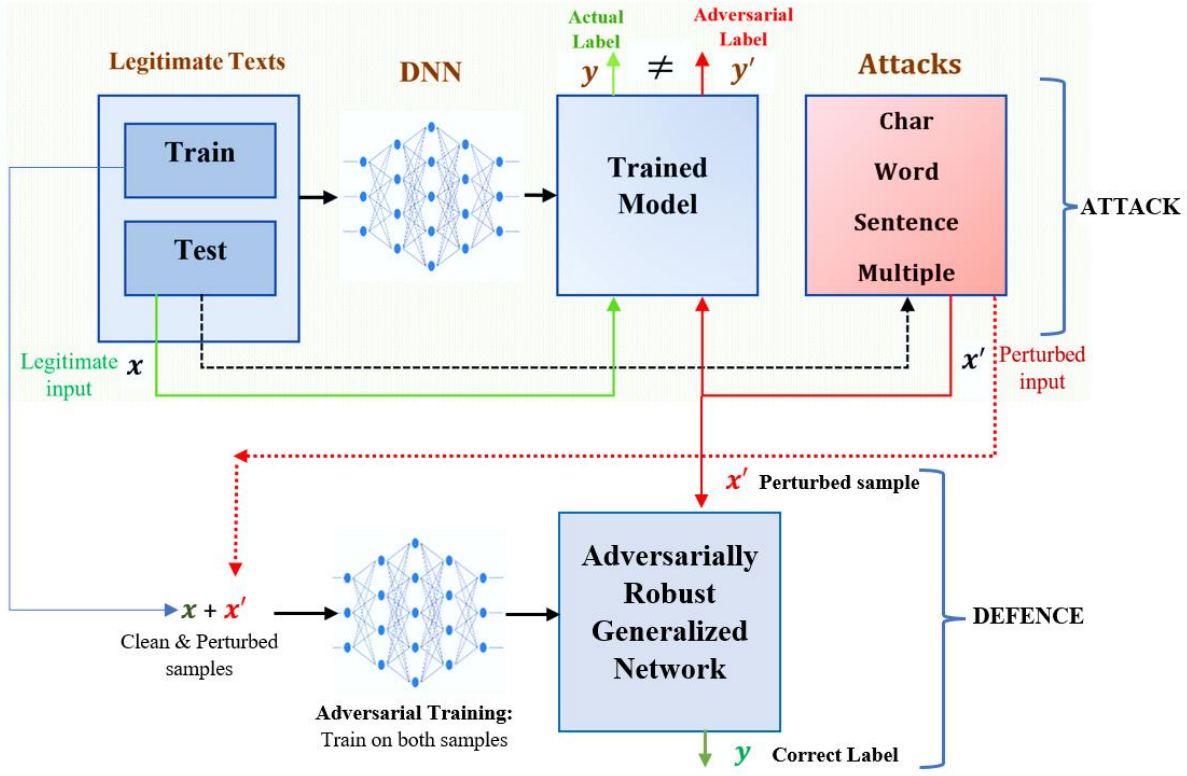


Figure 1.24 Overview of the methodology used in building different frameworks in various chapters of this thesis and their alignment with the central research title.

This research investigation focuses on textual adversarial attacks and defences in classification models. **Figure 1.24** outlines the fundamental methodology for developing both attack and defence strategies, which is extensively discussed throughout the thesis. Various chapters introduce and examine several novel frameworks for both attacks and defences.

Chapter 2: Literature Review

While there has been significant progress in computer vision on adversarial assaults and defences, there is a lack of research on adversarial machine learning in the textual domain. We have surveyed only a limited number of adversarial assaults and defences in the text domain and have chosen to contribute to this area.

This part explores the underlying context of adversarial manipulations in classification of text, with a special emphasis on manipulating text through basic changes to mislead machine algorithms. This section also includes a collection of already verified sophisticated assault strategies in language processing.

2.1 Fundamentals of Adversarial Machine Learning in Natural Language Processing

The following section offers a basic comprehension of adversarial situations, including structured explanations, clarifications, and the classification of such occurrences.

Common Terminologies

- *Perturbation:* Perturbations are deliberately crafted little disturbances that are introduced into genuine samples with the intention of deceiving the target model.
- *Adversarial Example:* The adversarial instances are generated by a strategy model by the addition of tiny modifications to authentic instances, causing the target models to generate erroneous predictions. Simultaneously, adversarial instances must be indiscernible to beings, implying that **1**) individuals cannot differentiate between adversarial instances and genuine instances, and **2**) persons should nevertheless accurately anticipate outcomes on adversarial instances.
- *Attack Model:* The attack mechanism pertains to the framework responsible for generating adversarial samples.
- *Victim Model*: The victim model is the model that has been subjected to an adversarial assault in order to assess its susceptibility to adversarial instances.
- *Robustness*: A framework is considered resilient if it is capable of making accurate predictions even when presented with undetectable disturbances. Adversarial defenses aim to improve the resilience of models.

Basics of Adversarial Attack

Deep Neural Networks (DNNs): Deep Neural architectures are a specific kind of neural network made up of neurons. They consist of a mathematical equation called $\boldsymbol{F}$, which is a depiction of typical Deep neural network algorithm (DNN). This equation may be expressed as $\boldsymbol{F}: \mathbf{X} \rightarrow \mathbf{Y}$. The purpose of this operation is to establish a mapping between the components of the set of inputs $\mathbf{X}$ and their associated labels in the corresponding label set $\mathbf{Y}$. The set $\mathbf{Y}$ comprises $\boldsymbol{k}$ categories, represented as $\{\mathbf{1}, \mathbf{2} \ldots, \boldsymbol{k}\}$. For the selected specimen $\boldsymbol{x}$ from collection $\mathbf{X}$, it can be seen that the categorization operation $\boldsymbol{F}$ correctly allocates the actual label $\boldsymbol{y}$ to $\boldsymbol{x}$, represented as $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{y}$.

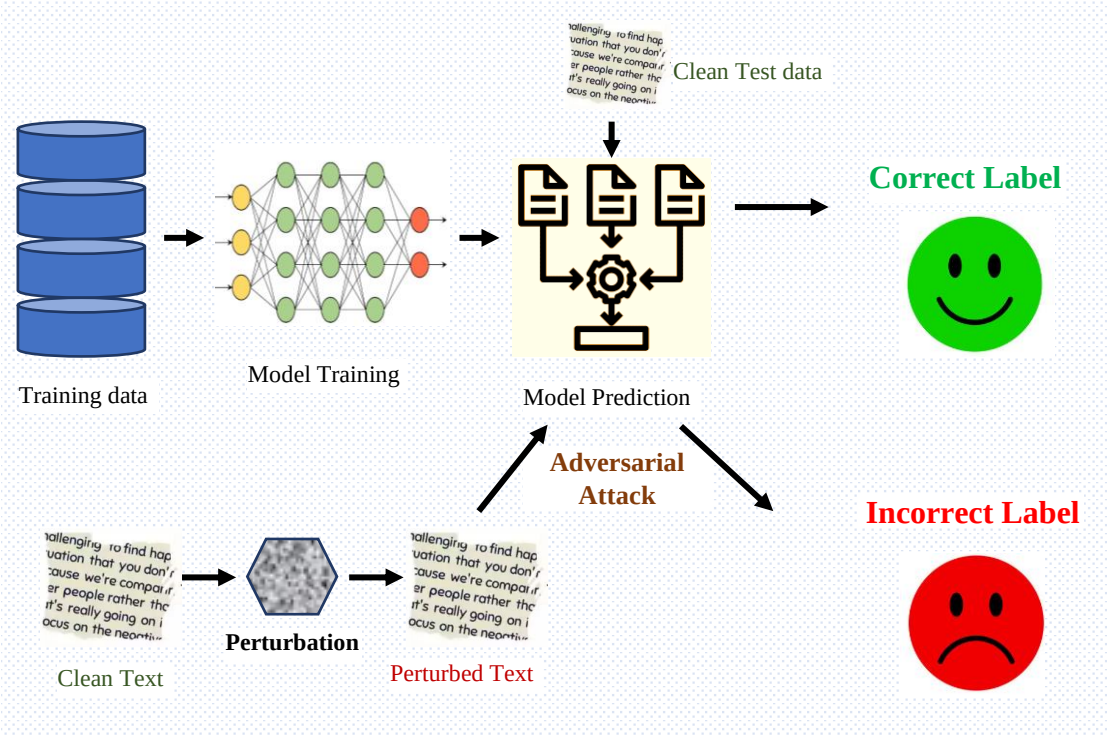


Figure 2.1 The methodology for executing a textual adversarial attack

Adversarial Attack: With an adversarial assault, an adversary aims to make a small disruption $\boldsymbol{\varepsilon}$ to an input variable called $\boldsymbol{x}$ in order to create an adversarial instance $\boldsymbol{x}'$. An adversarial instance is specifically crafted to produce a distinct outcome label $\boldsymbol{y}'$ (where $\boldsymbol{y}$ is not equivalent to $\boldsymbol{y}'$) when assessed by the target classifier $\boldsymbol{F}$. It is crucial for $\boldsymbol{x}'$ to both fool $\boldsymbol{F}$ and remain undetectable to the human eye at the same time[34]. In order to maintain the imperceptibility of the produced $\boldsymbol{x}'$, other metrics, such as semantically resemblance are employed to achieve this goal, especially by ensuring that the magnitude of $\boldsymbol{\varepsilon}$ is smaller than $\boldsymbol{\delta}$. $\boldsymbol{\delta}$ is used as a threshold to limit the number of disturbances. The process of carrying out an adversarial assault on the model is depicted in the **Figure 2.1**.

2.2 Taxonomy of Adversarial Examples

This section offers an elaborate elucidation of the hierarchy of Adversarial occurrences within the textual domain, as illustrated in **Figure 2.2**.

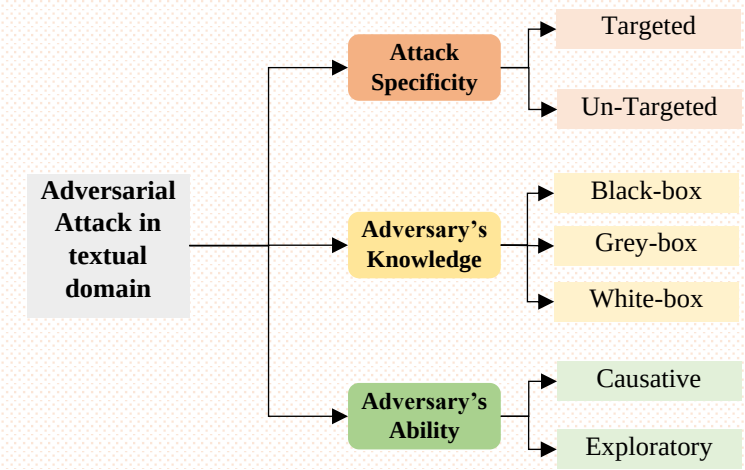


Figure 2.2 Taxonomy of Adversarial Examples

Adversary's Ability: Various assault strategies, such as causative and exploratory assaults, were suggested to investigate possible weaknesses within these frameworks. **"***Exploratory***"** assaults, also known as test time evasion assaults, consist of creating adversarial instances to evade a particular classifier. These malevolent testing examples are specifically created to take advantage of weaknesses in the framework's decision-making process. Conversely, **"***Causative***"** assaults specifically aim to manipulate the training information to fool the machine learning model[35]. Such assaults seek to manipulate the classification algorithm by modifying the training data during the process of learning. This paper exclusively centres around the subject of evading test time, with an emphasis on analysing assaults that take advantage of algorithms for classification as a possible susceptibility in terms of assurance.

Attack specificity: Adversarial invasions may be designated as either Targeted or Un-targeted assault, based on the objectives of the adversary. Within framework of a targeted attack, the hostile instance $\boldsymbol{x'}$ is deliberately assigned to a certain specified category $\boldsymbol{t}$, which serves as the adversary's chosen target[36]. The main mechanism of this strategy focuses on enhancing the accuracy linked to class $\boldsymbol{t}$. In the case of a Un-targeted assault, the adversary's main goal is to mislead the framework without specifically targeting a particular intended outcome. The result $\boldsymbol{y'}$ has the capacity to be assigned to any class, except $\boldsymbol{y}$. Unlike a focused assault, a non-targeted assault works by decreasing the accuracy linked to valid outcome $\boldsymbol{y}$.

Adversary's Knowledge: Attacks by adversaries can be executed by attackers who possess different levels of understanding about the intended architectures, ranging from complete cognizance (white-box) to no knowledge (black-box)[37], or partial understanding (grey-box).

Within the context of white-box assessment, adversary have unrestricted acquisition to their intended architecture. Adversaries can generate optimal adversarial examples by using their understanding of the intended framework, including its layouts, settings, and learning data. In the black-box scenario[38], adversaries are unable to ascertain the specific victim classifiers. Black-box attacks use the capabilities of malicious samples or repeated queries to be transferred for optimization purposes. Grey-box opponents have limited comprehension of the model as they can only access its settings[39]. Grey-box attacks presuppose that the intended architecture remains available during the whole learning process, in contrast to the other two types.

Perturbation-level: Based on text, adversarial strategies may be categorized into several levels depends upon the granularity of disruption used to create samples as illustrated from **Figure 2.3**. These levels include character-level, sentence-level, word-level, or multi-level[40], [41]. Char granularity assaults include modifying multiple letters within terms to create instances that trick detectors[42].

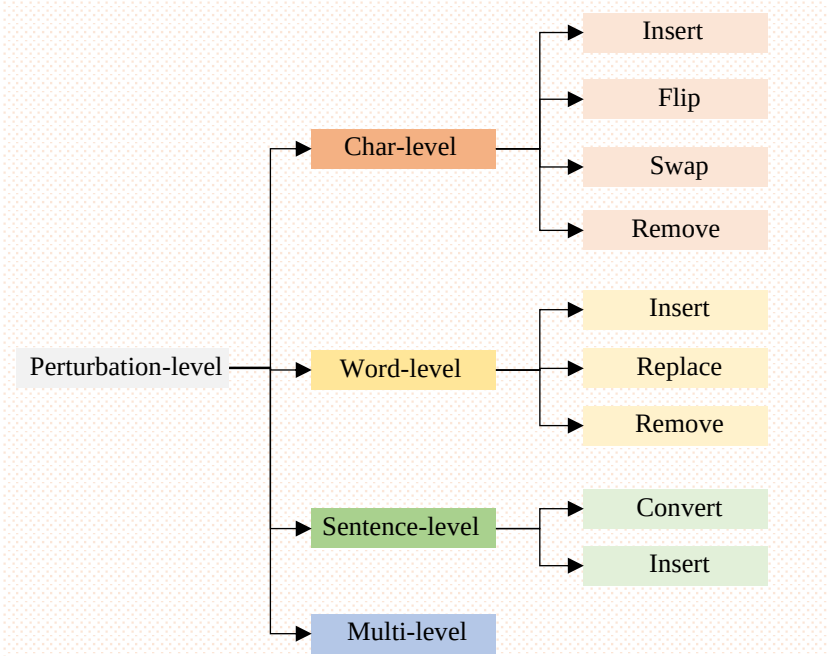


Figure 2.3 Taxonomy of Perturbation Granularity

2.3 Conventional Textual Adversarial Attack mechanisms & their Components

Morris et al.[43] deliver a concise illustration of the four elements implicated in the process of creating adversarial text instances. **(1)** The sequential search approach conducts a thorough study to identify the most effective changes. **(2)** A modification component is employed to convert an initial data, represented as $\boldsymbol{x}$, into a changed version, marked as $\boldsymbol{x}'$. To achieve this interruption, several approaches, such as substituting equivalents and randomly inserting

characters, are employed. These tactics are undetectable under individual inspection. **(3)** A collection of limitations or restrictions is employed to prevent undesirable alterations to x', guaranteeing that the changed x' maintains the meaning and smoothness of the genuine x. The **(4)** goal function is to find an adversarial instance that produces a label that is different from the real label.

Table 2.1 adheres to the criteria outlined in ***Section 2.1.3 & 2.2*** *by* showcasing prominent textual adversarial attack techniques in six different fields. The distinctive nature of each attack technique detailed in the current investigation which is illustrated by the fusion of perturbation level, Attack specificity, Adversary's knowledge, searching strategy, transformation applied, and set of limitations chosen for their attack approach.

Table 2.1 Evaluating conventional adversarial assault tactics in comparison to the suggested methodology (**BB***-Black Box, **WB***-White Box)

Attack	Transformation	Search Method	Constraints	Attack Specificity	Adversary's Knowledge	Perturbation Granularity	Limitation of the work
TextFooler [34]	Word replaced with strongest matching GLOVE embedding of words	Greedy – WIR (Random)	Similarity of word embeddings, BERT Score, POS coherence	Un-targeted	BB	Word	Robust adversarial training effectively defends against this form of attack.
TextBugger [44]	Character switching, deletion, substitution, and insertion	Greedy – WIR	USE similarity, POS consistency	Targeted, Un-targeted	BB	Character	Sometimes, however, the alterations made to the input sequence become apparent.
PWWS [45]	WordNet-based similar word swapping	Greedy – WIR (saliency)	USE cosine coherence, POS consistency	Un-targeted	WB	Word	To implement a white box technique, an adversary must have access to the algorithm's parameters.
PSO[46]	How – Net Word swapping	PSO	Leven-shtein edit-distance	Un-targeted	WB	Word	The search methodology employs a procedure that necessitates an extensive period to identify potential interruptions.
Pruthi et al. [47]	Adding, switching keypad characters, deletion, and exchanging nearby characters.	Greedy Search	word embedding similarity, POS consistency, USE similarity	Targeted, Un-targeted	BB	Character	By employing an alphabetizer and morphological checker, it is highly feasible to prevent any alterations made to a valid input.
Kuleshov et al. [48]	Strongest match ing term in GLOVE embedding replaces word.	Greedy – WIR (Random)	word embedding similarity, POS consistency, USE similarity	Un-targeted	WB	Word	To implement a white box technique, the adversary is required to possess accessibility to the framework, its

Attack	Transformation	Search Method	Constraints	Attack Specificity	Adversary's Knowledge	Perturbation Granularity	Limitation of the work
							characteristics for input, and the model's settings.
IGA[49]	The term has been substituted with the most closely related GLOVE embedding of words.	Genetic Algorithm	POS consistency, BERT Score	Targeted, Un-targeted	WB	Word	The alterations that are implemented on the input are frequently detectable
DWB [50]	Adjacent Char Swapping, Eliminating, Inserting, Substituting	Greedy-Word Importance Ranking (Random)	Leven-shtein edit-distance	Targeted, Un-targeted	BB	Character	It is straightforward to prohibit the modifications performed to genuine input by utilising a spell and grammar checker.
Checklist[51]	Changing Name, Changing Number, Word Swap, Insertion, Changing Location of word, Swapping Contracting	Greedy Search	Repeat word Modification, POS consistency	Un-targeted	BB	Word	There are numerous additional changes to input sequences and the assault technique is less successful.
BAE [52]	Prediction of Masked Tokens with BERT	Greedy – WIR (Random)	USE similarity, POS consistency	Un-targeted	BB	Word	Reliability is lacking since several detection techniques let one readily find word-level disturbances.
A2T[53]	The term has been substituted with the most closely related GLOVE embedding of words.	Greedy-Word Importance Ranking	word embedding similarity, POS consistency, USE similarity	Un-targeted	WB	Word	Token substitutions can be complicated and out of context in some instances, but humans can easily recognize them.
Iyyer et al. [54]	Create several variants of the input that satisfy the required standards without sacrificing the general quality.	---------- ----------	POS verification, grammatical restraints	Un-targeted	BB	Sentence	Sometimes the semantic meaning suffers when one paraphrases the whole input phrase to create adversarial input.
Liang et al. [**43**]	Modification, removal, and inclusion. The term has been substituted with the most closely related GLOVE embedding of words.	Greedy Search	Highest percentage of letters changed , minimum word length	Un-targeted	WB, BB	Mixed-Perturbations character & word	When changes made to the input sequence have an impact on both the word and char granularity , the idea of imperceptibility is undermined.
Inflect – Text (Proposed Approach**)**	Explores the use of word inflections as deliberate attacks against NLP algorithms.	Beam Search (b=8)	Ensuring part-of-speech integrity and minimizing cosine resemblance.	Un-targeted	BB	Word – level	----------

This investigation presents three novel robust adversarial NLP attacks, considering the existing limitations on attacks. These assaults create adversarial examples by modifying the input text at the granularity of individual characters and words. The objective of the assault strategies is to manipulate a targeted classifier by exploiting certain linguistic criteria. (for example, limitations related to similarity in grammar and meaning). The approach of manipulating text at different levels to trick the classifier is shown in **Figure 2.4.**

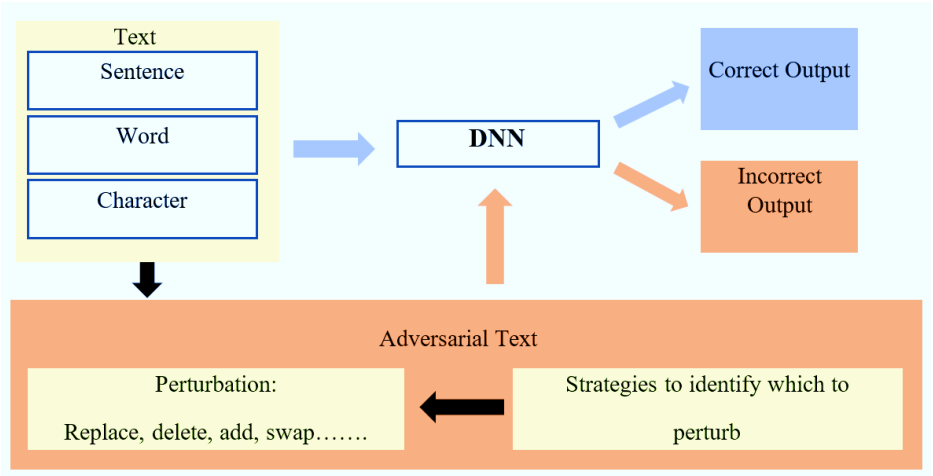


Figure 2.4 Framework for Deceiving Classifiers by means of Adversarial Text

2.4 Defences Against Textual Adversarial Attacks

The objective of adversarial defences is to train a model that can attain a high level of accuracy on both benign and adversarial cases. Adversarial defences should not just protect against static adversarial examples, but also guard against reiterated attacks. In the context of defence, it is assumed that attackers have knowledge of the defence model and can repeatedly assault it to create adversarial examples.

The studies in the field of textual adversarial defences can be categorized into **three** main areas:

- ***Adversarial training*:** Adversarial training refers to a technique used in machine learning where a model is trained to improve its performance by exposing it to adversarial examples or scenarios. Adversarial training is commonly used to enhance the resilience of victim models by utilizing adversarial cases for data augmentation. Nevertheless, these adversarial training strategies are susceptible to a restricted quantity of adversarial samples.
- ***Adversarial Restoration:*** Adversarial restoration refers to the process of repairing or recovering something that has been damaged or compromised as a result of adversarial actions or attacks. The concept of adversarial restoration involves identifying and subsequently reconstructing altered tokens. ScRNN [45] utilizes an RNN semi-

character architecture to detect and recover words that have been distorted through character-level attacks.

- ***Certified Robustness:*** Certified Robustness presents an alternative approved and robust approach that utilizes the randomized smoothing methodology. Nevertheless, achieving certified resilience necessitates imposing a stringent limitation on the attack space, which poses challenges in scaling up to huge datasets and neural networks due to their inherent high complexity.

Based on our current understanding, the majority of defence mechanisms either focus only on a specific sort of attack or necessitate knowledge of the targeted attacks, hence restricting their efficacy in practical application scenarios.

*The following research works form the basis of this chapter**:***

- **A. Bajaj** and D. K. Vishwakarma, "Deceiving Deep Learning-based Fraud SMS Detection Models through Adversarial Attacks," in *Proceedings* - ***17th International Conference on Signal-Image Technology and Internet-Based Systems****, SITIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 327–332. doi: 10.1109/SITIS61268.2023.00059.
- **A. Bajaj** and D. K. Vishwakarma, "Bypassing Deep Learning based Sentiment Analysis from Business Reviews," in *2023* ***2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies*** *(ViTECoN)*, IEEE, May 2023, pp. 1–6. doi: 10.1109/ViTECoN58111.2023.10157098.
- **A. Bajaj** and D. K. Vishwakarma, "Exposing the Vulnerabilities of Deep Learning Models in News Classification," in *2023 4th* ***International Conference on Innovative Trends in Information Technology*** *(ICITIIT)*, IEEE, Feb. 2023, pp. 1–5. doi: 10.1109/ICITIIT57246.2023.10068577.

2.5 Research Gaps

- **RG1:** There is a limited research investigation on developing adversarial examples using character-level perturbation granularity.

- **RG2:** Fewer investigations are carried out on building adversarial cases in a black box environment.

- **RG3:** Analysis of adversarial texts on the basis of a variety of factors such as scalability, sensitivity, runtime analysis, transferability, and applicability are missing.

- **RG4:** Investigation of the relative vulnerability of modern deep learning frameworks to adversarial perturbations is under explored.

- **RG5:** Certain conventional techniques for generating adversarial examples are readily detectable and violate grammatical and semantic constraints.

- **RG6:** There is a dearth of research on the development of appropriate defensive systems that can effectively mitigate textual adversarial attacks.

2.6 Research Objectives

The proposed objectives are based on identified research needs:

- **RO1:** To introduce a novel architecture for developing adversarial examples in text with character-level perturbations under black-box settings.

- **RO2:** To analyse the effect of adversarial examples on deep learning models on account of scalability, sensitivity, transferability and runtime.

- **RO3:** To assess the relative vulnerability and resilience of deep learning classifiers against adversarial examples.

- **RO4:** To build a defensive mechanism that can effectively mitigate textual adversarial attacks.

2.7 Research Contributions

The main objective of the thesis is to design and develop novel adversarial attacks and defence architectures capable of identifying vulnerabilities in neural text classifiers and with the possible solution to have adversarial robust generalization. Therefore, the following architectures and frameworks are proposed to achieve this task, with RC referring to the Research Contributions and RO corresponds to Research Objectives.

- **RC1: HOMOCHAR** is a novel textual adversarial attack operating within a black box setting. The proposed method generates more resilient adversarial examples by considering the task of perturbing a text input with transformations at the character level by replacing normal characters with imperceptible homoglyph characters. The objective is to deceive a target NLP model while adhering to specific linguistic constraints in a way such that the perturbations are unnoticeable under human

observation. This research contribution aligns with Research Objective **RO1**.

- **RC2: Non-Alpha-Num**: A novel architecture for generating adversarial examples using Punctuations and Non-alphanumeric character insertion as perturbations for bypassing NLP-based clickbait detection mechanisms. This contribution is related to Research Objectives **RO1** & **RO2**.

- **RC3: Inflect-Text:** Training on only perfect Standard English corpora predisposes pre-trained neural networks to discriminate against minorities from nonstandard linguistic backgrounds (e.g., African American Vernacular English, Colloquial Singapore English, etc.). We propose an **Inflect-Text** word-level attack that perturbs the inflectional morphology of words to craft plausible and semantically similar adversarial examples that expose these biases in popular NLP models. This work addresses Research Objectives **RO1** & **RO2**.

- **RC4: ARG-Net:** Due to the recent increase in textual adversarial attack methods, neural text classifiers are facing a more significant risk. In response to this, we have suggested a strategy to enhance the generalization capability of these classifiers by implementing adversarial training (defensive strategy). In the proposed **ARG-Net** model, it utilizes Augmented text to generate adversarial examples. The model undergoes training using both clean and adversarial cases to develop robust classification capabilities against word-level synonym replacement assaults. This study focuses on Research Objective **RO4**.

- **RC5: Comparative Analysis:** Trained the most popular models on the emotion dataset and applied conventional adversarial attacks on the pre-trained models to have a **comparative analysis** among models to find out which model is more vulnerable and which attack method is a greater threat to state-of-the-art Classifiers. This study is centred to Research Objective **RO3**.

Chapter 3: Textual Adversarial Attacks

3.1 Scope of this Chapter

This chapter focuses on the issue of tricking neural text categorization systems using adversarial attack techniques. Three novel architectures for textual adversarial attacks are proposed. The initial assault model is called ***HOMOCHAR***. In the HOMOCHAR adversarial attack, the individual characters of the important words in an input text are modified. During the process of transformation, regular characters are substituted by homoglyph characters. The second approach is a ***Non-Alpha-Num*** adversarial attack. This attack operates in black box scenarios and allows for perturbations at both the character and word level. Specifically, it involves replacing regular characters in crucial words with non-alphanumeric characters. The third attack mechanism, known as ***Inflect-Text***, utilises the inflectional morphology of words to generate perturbed words. The adversarial examples created using these three new attack strategies surpass standard methods by generating significantly higher attack success rates (ASR). The results indicate that neural text classifiers can be circumvented, potentially leading to significant consequences for current policy strategies.

3.2 HOMOCHAR: A Novel Adversarial Attack Framework for Exposing the Vulnerability of Text-based Neural Sentiment Classifiers

3.2.1 Abstract

State-of-the-art deep learning algorithms have demonstrated remarkable proficiency in the task of text classification. Even though deep learning-based language models are very common, not much is known about their security flaws. This is particularly concerning for their growing use in sensitive applications, such as sentiment analysis. This study demonstrates that language models possess inherent susceptibility to textual adversarial attacks, wherein a small number of words or characters are modified to produce an adversarial text that deceives the machine into producing erroneous predictions while maintaining the overall semantic coherence for a human reader. The current study offers HOMOCHAR, a novel textual adversarial attack that operates within a black box setting. The proposed method generates more robust adversarial examples by considering the task of perturbing a text input with transformations at the character level as a combinatorial search problem. The objective is to deceive a target NLP model while adhering to specific linguistic constraints in a way such that the perturbations are imperceptible

to humans. Comprehensive experiments are performed to assess the effectiveness of the proposed attack method against several popular models, including Word-CNN, Word-LSTM along with five powerful transformer models on two benchmark datasets, i.e., MR & IMDB utilized for sentiment analysis task. Empirical findings indicate that the proposed attack model consistently attains significantly greater attack success rates (ASR) and generates high-quality adversarial examples when compared to conventional methods. Additional experiments are being conducted to analyse the attack methodology across various parameters. The results indicate that text-based sentiment prediction techniques can be circumvented, leading to potential consequences for existing policy measures.

3.2.2 Proposed Methodology

3.2.2.1 Problem definition

The objective of a proficient DNN classifier **F** is to accurately predict the label $\boldsymbol{Y_{true}} \in \boldsymbol{y}$ for any given input $\mathbf{X} \in \boldsymbol{x}$, i.e., $\mathbf{F(X)} = \boldsymbol{Y_{true}}$, This is achieved by maximising the posterior probability, as demonstrated in **Eqn. (3.1**).

$$\text{argmax}_{\boldsymbol{Y_i} \in \boldsymbol{y}} \mathrm{P}(\mathbf{Y_i}/\mathbf{X}) = \mathbf{Y_{true}} \quad \textbf{(3.1)}$$

The objective of a rational text attack is to introduce a perturbation $\Delta\mathbf{X}$ that is imperceptible to humans, but has the ability to deceive the classifier **F** when it is incorporated into the original **X**. The modified input $\mathbf{X}^* = \mathbf{X} + \Delta\mathbf{X}$ is referred to as the adversarial example in the literature. In general, an adversarial example that is successful has the ability to deceive a well-trained classifier into assigning an incorrect label that is different from the true label or a pre-specified label $\mathbf{Y_{target}}$, where $\mathbf{Y_{target}} \neq \mathbf{Y_{true}}$. Several techniques are employed to achieve the objective of rendering the generated $\mathbf{X}^*$ indiscernible, including measures such as similarity in meaning. This is done to ensure that the standard deviation of the distinction between the original and modified text, must be less than a certain threshold value, denoted by $\boldsymbol{\delta}$. The symbol $\boldsymbol{\delta}$ represents a threshold that limits the number of manipulations.

$$\text{argmax}_{\boldsymbol{Y_i} \in \boldsymbol{y}} \mathrm{P}(\mathbf{Y_i}/\mathbf{X}^*) \neq \mathbf{Y_{true}} \quad \textbf{(3.2)}$$

$$\text{argmax}_{\boldsymbol{Y_i} \in \boldsymbol{y}} \mathrm{P}(\mathbf{Y_i}/\mathbf{X}^*) = \mathbf{Y_{target}} \quad \textbf{(3.3)}$$

Eqn. (3.2) and **Eqn. (3.3)** represent the attack strategies commonly referred to as untargeted and targeted attacks, respectively. A text perturbation that is considered valid must adhere to semantic, grammatical and lexical and constraints. This paper presents a novel framework for

adversarial attack, which exhibits the ability to produce adversarial text. According to the research findings, the proposed method for generating adversarial instances yields imperceptible alterations that present a formidable challenge for human observers, as they are unable to discern the perturbations.

3.2.2.2 Attack Design

An effective textual adversarial approach **HOMOCHAR** is developed under black-box environment that formulates stronger adversarial examples as a combinatorial search task with the goal (untargeted attack) for deceiving neural text classifier by perturbing at character-level which adheres to specific linguistic constraints. The attack is built using four essential components, which include *transformation* (that generates a list of potential $\mathbf{X}_{adv}$(adversarial samples), *search method* (that applies transformation until a successful $\mathbf{X}_{adv}$ is found), a *set of constraints* (that filter out $\mathbf{X}_{adv}$ that does not satisfy lexical, grammatical, and semantic

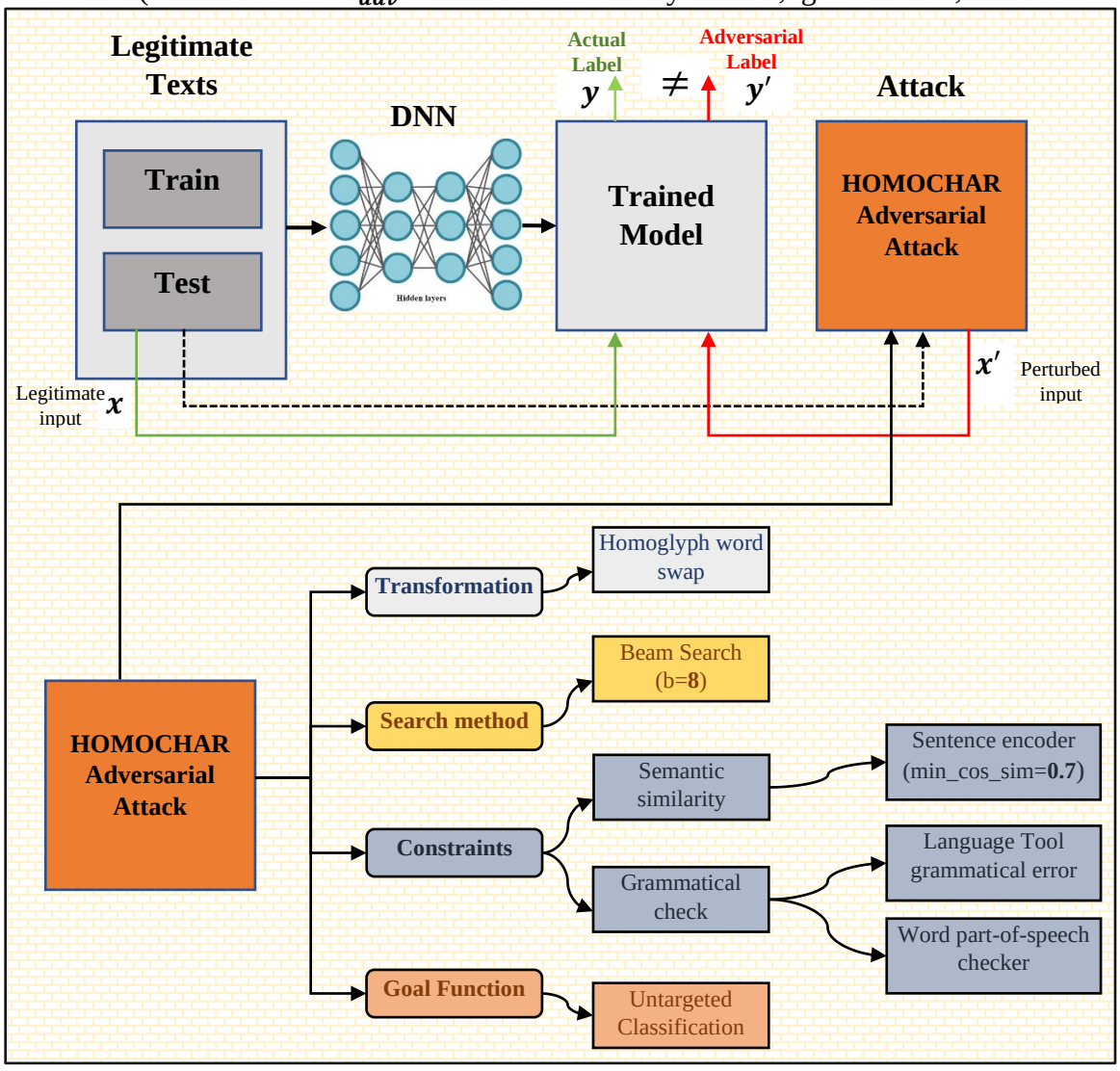


Figure 3.1 Design of proposed HOMOCHAR adversarial attack method using four essential set of components

constraints), and a *goal function* (which assesses the effectiveness of the method such that it always misclassify the true prediction), the design of proposed methodology is shown in **Figure 3.1**. The algorithm looks for potential changes that could lead to a successful perturbation.

This section delineated the methodology for creating adversarial examples through a framework consisting of four distinct elements: transformations, a set of constraints, a search algorithm and a goal function. The aforementioned system is intended to detect a change from **X** to **X′** that deceives a predictive NLP model. This is accomplished by satisfying specific constraints while simultaneously achieving a particular objective, such as misleading the model into producing an incorrect classification label. The search algorithm aims to identify a sequence of alterations that result in a positive perturbation outcome. The following discourse provides a comprehensive elucidation of the four fundamental sets of components utilised in the construction of the proposed methodology.

Figure 3.2 Adversarial Examples generated by replacing normal English characters with visually similar homoglyphs

Transformations:

From an input, a transformation generates a number of prospective perturbations. If $\boldsymbol{x} = (\mathbf{X_1},...,\mathbf{X_i},...,\mathbf{X_n})$, then replacing $\mathbf{X_i}$ with a changed version of $\mathbf{X_i'}$ will result in a perturbed text. Depending on the granularity of $\mathbf{X_i}$, the alteration may take place at the word, character, or sentence level. Since word substitution is a common literary technique, in this research, it is decided to concentrate the investigation on swapping the important words with character-level perturbed words. In HOMOCHAR adversarial attack, the individual characters of the significant words in an input text are transformed. In transformation, normal characters are replaced with homoglyphs[1] characters (for instance, changing all English "**a**" in a neural text sample to Cyrillic "**а**"). These are chosen because, while they look visually similar to their counterparts, neural text classifiers tokenize them differently[7]. Therefore, normal characters of the most important words in an input sentence are substituted with homoglyph characters. Homoglyph characters are also named as homographs. In the past, homoglyphs have been used to substitute similar-looking characters in a trusted URL to transfer users to malicious websites. This research includes an experiment to explore if this technique can also be utilized to develop efficient black-box adversarial attacks against neural sentiment classifiers.

Homoglyph characters represent the same glyph or an identical-looking glyph. Typically, this occurs when the same written script is used in various language families. These characters are different according to the Unicode specification. A character set called Unicode[2] was created to standardize how text is represented electronically. At the current time, the character capacity of the system is 143,859 and it can accommodate diverse languages and symbol sets. Traditional Chinese characters, mathematical symbols Latin letters and emojis are just a few of the characters that can be represented by Unicode. Each character is assigned a code point, which is a numeric representation. There are other ways to encrypt these numerical code points, which are commonly identifiable by the prefix U+, but UTF-8 is the most widely used. The Unicode specification poses a significant security threat due to the diverse encoding options available for homoglyphs, which are distinct characters that exhibit identical or comparable glyphs. For instance, the look-alike digit zero 0 (U+0030) is used in place of the Latin minuscule letter O (U+006F), which will cause the computer to tokenize it differently while classifying the data. This issue is not unique to Unicode. As an illustration, within the ASCII range, the lowercase Latin character "l" frequently bears a resemblance to the uppercase Latin

[1] IDN homograph attack - Wikipedia
[2] https://www.unicode.org/versions/Unicode13.0.0/

character "I". Certain character combinations can function as pseudo-homoglyphs, as exemplified by the "rn" and "m" pairing in most sans serif fonts.

The fundamental idea behind this transformation is to add noise by exchanging English characters for corresponding international characters, as seen in **Figure 3.2**. Such noise may lead to classification errors that differ from the actual results[56]. Text appears the same to humans but yields different results for deep-learning sentiment classifiers. The findings of the research indicate that the transformed adversarial instances exhibit a high degree of imperceptibility to human visual perception, which poses a challenge for users to accurately discern the attack as being adversarial in nature. It demonstrates that HOMOCHAR attack methodology is far better compared to the standard attack techniques, which include misspellings, insertions, switching, synonym replacements, etc.

Searching algorithm:

The search algorithm attempts to identify, from the transformations, the set of most potent perturbed words in an input sequence that will result in the most efficient attack. The text $\boldsymbol{x}$ is subjected to various perturbations by substituting each word $\mathbf{X}_{\boldsymbol{i}}$, resulting in multiple perturbed texts $\mathbf{X}'$**.**The beam search algorithm is utilized in order to find out the best set of perturbed sequence which uses the scoring function mentioned in previous section of this article**.** The words that achieved the highest scores adhering to perturbations are selected. In beam search, the top **b** most potent perturbed texts are kept (**b** is known to be the "beam width") with **b=8**. Subsequently, the aforementioned procedure is reiterated through perturbation of each of the highest-ranking **b** texts, resulting in the production of the subsequent group of candidates. This process requires $\mathbf{O(b*}\boldsymbol{W}^{\mathbf{2}}\mathbf{*T)}$ queries, where $\boldsymbol{W}$ denotes the quantity of words present in the input. The variable **T** represents the upper limit of available transformation choices for a specific input.

Heuristic scoring function

In the event of an untargeted attack on a classifier, the perpetrator's objective is to identify instances that cause the classifier to inaccurately predict the class (label) for $\mathbf{X}'$. The underlying premise is that the veritable classification of $\mathbf{X}'$ corresponds to that of the initial **X**. The heuristic scoring function computes the score of every element $\mathbf{X}_{\boldsymbol{i}}$ that belongs to the set $\boldsymbol{x}$. To identify the optimal group of prospective candidates for perturbation. The candidates who receive the highest scores are selected over other candidates.

Typically, a heuristic scoring function is employed, where the score is defined as shown in **Eqn. (3.4)**:

$$Score(\mathbf{X}') = 1 - F_y(\mathbf{X}') \quad \textbf{(3.4)}$$

Semantic similarity:

A 'fine-grained metric' is required that quantifies the extent to regulate the quality of generated adversarial texts, such that it can be contended that the produced adversarial texts will preserve semantic similarity with the original texts. The HOMOCHAR framework utilises the Universal Sentence Encoder (USE) to evaluate the semantic similarity among textual instances [57]. The USE model utilises a process of encoding distinct input sentences into embedding vectors of 512 dimensions, thereby facilitating the computation of their cosine similarity score. **Eqn. (3.5)** defines the cosine similarity between two n-dimensional vectors, denoted as $\boldsymbol{a}$ and $\boldsymbol{b}$.

$$\boldsymbol{S\ (a,\ b)} = \boldsymbol{\delta} = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}|| \cdot ||\boldsymbol{b}||} = \frac{\boldsymbol{\Sigma_{i=1}^{n} a_i \times b_i}}{\sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{a_i})^2} \times \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{b_i})^2}} \quad \textbf{(3.5)}$$

The USE encoder employed in this study has been trained on a diverse range of web-based textual data with a broad scope, including but not limited to Wikipedia, web-based news articles, web-based question-and-answer pages, and online discussion forums. Thus, it possesses the ability to provide input for numerous subsequent tasks. Formally, denote USE encoder by Encoder, then the USE score between an example $\boldsymbol{X}$ and its adversarial variation $\boldsymbol{X'}$ is defined in **Eqn. (3.6)** below.

$$USE_{\text{score}} = \text{Cosine (Encoder } (\boldsymbol{X}), \text{ Encoder } (\boldsymbol{X'})) \quad \textbf{(3.6)}$$

Given that the primary objective is to effectively produce adversarial texts, it suffices to regulate the semantic similarity to a predetermined threshold ($\boldsymbol{\delta}$)**,** and a threshold of $\boldsymbol{\delta = 0.7}$ is selected. The Universal Sentence Encoders (USE) is utilised to conduct a comparison between the sentence encodings of the original text denoted as $\boldsymbol{X}$ and the perturbed text denoted as $\boldsymbol{X'}$. In the event that the cosine similarity of two encodings decreases to a specific threshold, the value of $\boldsymbol{X'}$ is disregarded. The utilisation of large encoders such as Universal Sentence Encoder (USE) poses a challenge due to the considerable GPU memory consumption, which can reach up to 9GB in the case of USE [53]. Also, Language Tool[58] is used to induce the minimum number of grammatical errors along with Part-of-speech consistency (The substitute word should share the same part of speech as the original one.). Support taggers provided by flair, SpaCy, and NLTK attempt to preserve semantics between $\boldsymbol{X}$ and $\boldsymbol{X'}$.

Goal function:

The efficacy of an attack is evaluated in relation to model outputs through the utilisation of a goal function. It probes the search method along with transformations and a particular set of constraints until it leads to the misclassification of the actual output. The generated Adversarial Examples resulting from the proposed algorithm are depicted in **Figure 3.2**. **Table 3.1** illustrates the algorithm utilised in the proposed framework.

Table 3.1 Algorithm of the proposed framework

Algorithm 1: HOMOCHAR Adversarial Attack
Aim: Adversarial attack framework to fool neural text classifier **Input:** legitimate input $\boldsymbol{X}$ and its ground truth label $\boldsymbol{Y}$, classifier $\boldsymbol{F}(.)$, threshold $\boldsymbol{\delta}$, semantic similarity $\boldsymbol{S}(.)$. **Output:** Generated an adversarial sequence $\boldsymbol{X_{adv}}$
1. Initialization**:** $\boldsymbol{X^* \Leftarrow X}$ **2.** **for** $\boldsymbol{X_i}$ in $\boldsymbol{x}$ **do** **3.** Compute score ($\boldsymbol{X_i^*}$) **4.** **end for** **5.** $\boldsymbol{W_{Ordered} \Leftarrow Sort(X_1, X_2, X_3 \ldots\ldots X_n)}$ in descending order **6.** Remove the stop words in $\boldsymbol{W_{Ordered}}$ **7.** **for** $\boldsymbol{X_i}$ in $\boldsymbol{W_{Ordered}}$ **do** **8.** $\boldsymbol{X^* \Leftarrow replacing\ X\ with}$ ($\boldsymbol{words\ containing\ homoglyphs}$) **9.** **if** S ($\boldsymbol{X, X^*}$) <= $\boldsymbol{\delta}$ **then** **10.** return None: **11.** **else if** $\boldsymbol{F(X^*) \neq Y_{true}}$, **then** **12.** Solution found. return $\boldsymbol{X^*}$. **13.** **end if** **14.** **end for** **15.** **return None**

Assuming the given input document $\boldsymbol{x} = (\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_n})$, where each $\boldsymbol{X_i}$ denotes input sequence located at the i-th position. Initially, the spaCy library is employed to segment each document into distinct sentences. It is imperative to conduct further research and analysis before assuming that the investigation of this study can be generalised to the entire population. The process involves eliminating sentences that have predicted labels that differ from the original document label, specifically by filtering out $\boldsymbol{F(X_i) \neq Y_{true}}$. To accomplish this, the first step is to identify the most significant words that have the maximum influence on the original prediction outcomes, using a heuristic score as outlined in **Equation (6)**. These words are then subject to slight modifications while ensuring that their semantic similarity is maintained. The heuristic scoring function possesses three key characteristics. Firstly, it has the ability to accurately reflect the significance of words in relation to the prediction. Second, it can compute word scores without any prior understanding of the classification model's framework and settings. Lastly, it is a highly efficient method of calculation. In the development of adversarial instances, a preference is given to making small modifications to the original words. This is

due to the requirement that the resulting adversarial sentence must maintain visual and semantic similarity to the original sentence, in order to facilitate comprehension by human observers.

The design decisions are made so as to generate adversarial examples with higher quality and less disturbance. In the transformation function, the process involves substituting standard English characters with homoglyph characters for the dominant terms within a given input sentence. The beam search algorithm is utilised to determine the optimal set of perturbed candidates for a successful attack. Furthermore, the semantic similarity between $\boldsymbol{X_{adv}}$ & $\boldsymbol{X}$ is measured using the Universal Sentence Encoder [57]. In addition, Language Tool [58] is utilised to generate minimal grammatical errors while maintaining Part-of-Speech consistency. All of these design decisions result in constructing a robust adversarial attack method relative to the baselines which in turn results in a potent untargeted attack that misclassifies the actual output. The proposed technique involves perturbing text through the substitution of normal English letters with homoglyphs at the character level. An observation of significance is that words possess a symbolic quality, and language models conventionally depend on a lexicon to depict a finite range of possible words. The magnitude of the standard vocabulary is considerably lesser than the potential permutations of characters at a commensurate extent (e.g., approximately 26^n for the English language, wherein n denotes the word's length). This suggests that purposeful manipulation of significant terms can lead to their conversion into "unknown" words, which are not listed in the vocabulary. In deep learning modelling, any unfamiliar words will be assigned to the "unknown" embedding vector. The outcomes of this investigation offer persuasive proof that the adoption of a simple methodology can effectively prompt text categorization models to display flawed conduct.

3.2.3 Experimental Settings

A detailed description of the datasets, victim models, attack techniques, evaluation metric, and experimental settings were all explained in this part. After that, in the following section, we'll assess the findings and go over several likely causes of the observed performance.

3.2.3.1 Dataset Description

This study investigates adversarial text samples on two publicly available benchmark datasets that are extensively used for sentiment analysis tasks. On the test set, the final adversarial examples are generated and evaluated. The **Table 3.2** presents a summary of the datasets.

Rotten Tomatoes Movie Reviews[3] (**MR**) [59]: The movie reviews in this dataset were gathered by Lee & Pang [59]. It has 5,331 negative & 5,331 positive processed sentences/snippets with an average length of 32 words. The dataset is split into three sections for the experiment, using 80% and 20% for training and testing respectively. The models were trained to perform binary classification on movie reviews, with the aim of categorizing them as either having a positive or negative sentiment.

Internet Movie Database[4] **(IMDB)**[60]**:** The dataset of movie reviews from IMDB comprises 50,000 reviews that exhibit a high degree of polarity, with 25,000 reviews allocated for training and 25000 reviews for testing. The dataset contains an average sample length of 215.63 words. The models underwent training to execute binary classification of movie reviews, with the objective of categorizing them into either a positive or negative sentiment.

Table 3.2 Overview of the datasets

Task	Dataset	Classes	Train	Test	Avg Len
Sentiment Classification	**MR**	2	8.5K	2K	32
Sentiment Classification	**IMDB**	2	25K	25K	215.63

3.2.3.2 Victim Models

Experiments are performed on the following models to demonstrate the efficacy of the proposed framework. The descriptions of the models and their hyperparameters are provided below.

Word-LSTM: In sequence modelling, long-short term memory (LSTM [61]) is frequently utilised. A 150 hidden state LSTM with bidirectional operation was designed. The input is initially converted to 200-dimensional GLoVE embeddings before being sent to the LSTM. then the final logistic regression is utilized to predict the sentiment, averaging the LSTM outputs at each timestep to produce a feature vector with a dropout set to 0.3 and achieving an *0.8070* & *0.8830* testing accuracy on MR and IMDB dataset respectively.

Word-CNN: Convolutional neural networks constitute potential strategy for text classification tasks. For the investigation, Kim's architecture of convolutional neural network model [62] is chosen. Word-CNN with 100 filters and 3 window sizes (3, 4, and 5) is utilized. Model dropout is set at 0.3, and a base of the 200-dimensional GLoVE embeddings is used, followed by a fully

[3] https://huggingface.co/datasets/rotten_tomatoes
[4] https://huggingface.co/datasets/imdb

connected, max-pooling over time layer for classification. The model is achieving 0.*7940* & *0.8630* accuracy on the test set for MR and IMDB dataset respectively.

BERT: BERT (Bidirectional Encoder Representations from Transformers) [63] employs a Masked Language Model (MLM) & Next Sentence Prediction (NSP); it uses a stack of self-attention and fully connected layers to encode a sentence. The BookCoorpus and English Wikipedia datasets served as the first training grounds for the BERT. In this study, for the sentiment classification task, the "bert-base-uncased" model underwent five iterations of training, utilising a batch size of 16, a learning rate of 2e-05, and a maximum sequence length of 128. The aim of this optimisation was to enhance its performance in sequence classification on the given dataset. Given that this was a classification problem, a cross-entropy loss function was used to train the model. The model's highest score on this job, as determined by the eval set accuracy, was *0.875234 & 0.89088* for MR & IMDB respectively, which was discovered after 4 epochs for both the datasets.

DistilBERT: DistilBERT [64] is a transformer model that was derived from the BERT base. It is characterised by its compactness, speed, efficiency, and lightweight nature. In comparison to bert-base-uncased, the aforementioned model exhibits a 60% increase in speed and a 40% reduction in parameters. Despite these optimisations, it sustains a performance level of over 95% in relation to BERT, as assessed by the GLUE language comprehension benchmark. The model known as "distilbert-base-uncased" underwent training for sequence classification, lasting three epochs. The training process employed a batch size of 128, a learning rate of 2e-05, and a maximum sequence length of 16. The model was trained using a cross-entropy loss function because this task involved classification. The evaluation set accuracy, determined after two epochs, revealed that the model achieved a maximum score of *0.839587* and *0.88* on the MR and IMDB datasets, respectively.

ALBERT: BERT base has 110 million parameters, which makes it computationally expensive, a light version with fewer parameters was needed. The ALBERT [65] model comprises 128 embedding layers, 768 hidden layers, and 12 million parameters. The lighter model lowered the training and inference times as expected. Cross-layer parameter sharing and factorised embedding layer parameterization are the 2 strategies used to achieve a smaller set of parameters. The "albert-base-v2" model was improved for sequence classification. Running it for 5 epochs with a 32-batch size, a 2e-05 learning rate, and a 128-bit maximum sequence length. A cross-entropy loss function was used to train the model. The model's highest score

on this job, as determined by the eval set accuracy, was *0.880863* on MR which was discovered after one epoch & *0.89236* for IMDB following three epochs.

RoBERTa: Robustly Optimized BERT pre-training Approach is called RoBERTa [66]. This is, in many ways, an improved form of the BERT model as it incorporates the idea of dynamic masking, which strengthens the model. In addition, RoBERTa had also been trained on datasets which include CC-News (Common Crawl-News), Open WebText, and others. These datasets have a combined size of about 160 GB. RoBERTa used a batch size of 8,000 with 300,000 steps to increase the model's speed and efficiency. BERT, in contrast, use a 256-batch size with 1 million steps. The "Roberta-base" model has been fine-tuned for sequence classification. Running it with a maximum sequence length of 128 and a batch size of 32 for 10 epochs with a 5e-05 learning rate. Given that this was a classification problem, a cross-entropy loss function was used to train the model. The model's highest score on this job, as determined by the eval set accuracy, was *0.903377*, which was discovered after 9 iterations for MR dataset & *0.91436* was attained within 2 epochs for IMDB.

XLNet: Transformer-XL, the most advanced autoregressive model, is incorporated into XLNet's pretraining. Empirically, on 20 tasks, XLNet[67] outperforms BERT in similar experimental conditions. The design of BERT is comparable to that of XLNet. The way pre-training is handled where it differs the most, though. In contrast to BERT, which is based on autoencoding (AE), XLNet is an auto-regressive model (AR). The MLM challenge makes this disparity clear by requiring the model to predict language tokens that have been randomly disguised. The "Xlnet-base-cased" model was improved for sequence classification. Using a cross-entropy loss function, it was run for 5 epochs with a batch size of 16, a learning rate of 2e-05, and a maximum sequence length of 128. The evaluation set accuracy, which was discovered after two epochs, indicated that the model's best performance was *0.907129* on MR and *0.95352* for IMDB after running it for 2 epochs.

The accuracy score is the metric employed to assess binary sentiment classification models. The accuracy of the target models on the standard test set is presented in **Table 3.3**.

Table 3.3 Testing accuracy of the Targeted Models

	Word-CNN	**Word-LSTM**	**BERT**	**DistilBERT**	**ALBERT**	**RoBERTa**	**XLNet**
MR	79.40%	80.70%	87.50%	83.90%	88.00%	90.30%	90.70%
IMDB	86.30%	88.30%	89.0%	88.00%	89.20%	91.40%	95.30%

3.2.3.3 Baseline Attack Methods

The attack approaches were applied to the dataset to formulate adversarial examples. Such adversarial samples are then used to manipulate the seven introduced models to classify the positive sentiment of a review as negative which results in obfuscating sentiment detection. A brief explanation of all adversarial attack approaches used in conjunction with proposed method is provided in **Table 3.4**.

Table 3.4 Adversarial Attack Algorithms in NLP

Attacks	Perturbation	Description
TextFooler[34]	Word-level	Word swapping is used in this attacking strategy with the victims' 50 nearest embedding neighbors. optimized on BERT.
TextBugger[44]	Char-level	This attack strategy's potency has been increased for use in realistic circumstances. They use character switching, space insertions, and character deletions. In context-aware word vector space, they also swap out words with their top nearest neighbours and characters with letters that seem similar (for example, o with 0).
PWWS[45]	Word-level	These attacks aim to retain lexical accuracy, grammatical correctness, and semantic closeness by leveraging synonym swap. A combination of a word's saliency score and its maximum word-swap efficacy determines its priority.
PSO[46]	Word-level	A sememe-based word replacement technique combined with particle swarm optimization for word-level attacks.
Pruthi[47]	Char-level	Simulates typical typos, focusing on the QWERTY keyboard. This approach uses character switching, deletion, and insertion.
Kuleshov[48]	Word-level	Replaces the key words in an input sequence from counter-fitted word embedding space under a set of essential constraints.
IGA[68]	Word-level	This attack approach ranks the most crucial words in an input sequence using a scoring function and then replaces them with counter-fitted word embeddings. Along with grammatical and natural checks, it leverages word embedding distance and sentence encoding cosine similarity to maintain the validity of the perturbed sample.
Liang[69]	Word-level	Swapping words with the synonyms in the nearest word embedding space by utilising a genetic algorithm, the aforementioned task can be accomplished under a set of potential constraints for a valid adversarial sample**.**
DWB[50]	Char-level	Produces minor text alterations in a black-box environment. With greedy replace-1 scoring, it employs a variety of character-swapping techniques, including swapping, substituting, deleting, and insertion.
BAE[52]	Char-level	This attack methodology uses a language model transformation with a BERT mask. To better fit the entire context, it replaces tokens using the language model.
A2T[53]	Word-level	This attack approach uses gradient-based synonym word swap under white-box adversarial settings. It uses sentence encoding cosine similarity for retaining semantic similarity along with grammatical checks.
HOMOCHAR (Proposed approach)	Char-level	Word replaced with the character-level agitated word (containing homoglyph characters). In this study, beam search was employed to identify the optimal group of potentially perturbed words. Keeping the key modifications under particular grammatical and semantic similarity constraints can deceive the model into making accurate predictions.

3.2.3.4 Attack Evaluation Metric

The ultimate goal of attack algorithms is to alter the input in a way that leads to the model making inaccurate predictions. For accessing the effectiveness of the attack models, 500 correctly classified cases are randomly chosen from the test set, so that the accuracy of the

classifiers does not affect the evaluation. The attack algorithms are then run on these source texts to produce adversarial instances. The deep learning-based sentiment classifiers are then given the adversarial instances to produce the final prediction. The percentage of incorrect predictions made by these classifiers is used to define the attack algorithm's success rate. A higher success rate indicates that the attack algorithm can produce stronger adversaries that can make these sentiment classifiers act inappropriately. The Attack Success Rate (ASR) *(ratio of successful attack samples to the sum of successful and failed samples* $\frac{\boldsymbol{successful\ samples}}{\boldsymbol{successful+failed\ samples}}$*)* metric is utilised to evaluate the efficacy of individual attack algorithms in compromising a victim model. The ASR indicates the degree to which an adversary can deceive a victim model. Formally, an attack is successful if the classifier $\boldsymbol{F}$ can accurately classify the original legitimate input $\boldsymbol{F(X) = Y_{true}}$, but makes an incorrect prediction for the attacked input $\boldsymbol{F(X + \Delta X) = Y^*}$**.** Consequently, the ASR is described in **Eqn. (3.7)**.

$$\frac{\boldsymbol{F(X+\Delta X)=Y^*}}{(\boldsymbol{F(X+\Delta X)=Y^*})+(\boldsymbol{F(X+\Delta X)=Y_{true}})} \quad \textbf{(3.7)}$$

where $\boldsymbol{Y^*}$ is any label other than $\boldsymbol{Y_{true}}$ (an untargeted attack). The $\boldsymbol{\Delta X}$ indicates modifications to the legitimate text sample. A successful attack in this context means that the adversarial sample can incorrectly predict with high accuracy score. In the case of a failed attack, the adversarial sample is incapable of misclassifying the actual prediction. In addition, there are statements that were omitted from the calculation. The statements that the model initially incorrectly classified during its training. The investigation focuses on the success rates of attacks and their efficacy in misclassifying outputs.

3.2.4 Results & Discussion

To understand the vulnerability of a sentiment classifier. The first step of the analysis entails the utilisation of the provided dataset to train advanced deep-learning models. *Section 4.2* presents the hyperparameters of the models for both datasets with their corresponding descriptions along with their testing accuracy. The trained models are subjected to manipulation through the utilisation of the HOMOCHAR adversarial attack technique. Following perturbation of the test samples by the proposed algorithm, **Table 3.5** depicts the reduction in accuracy scores.

Table 3.5 The study reports on the outcomes of an automated evaluation of an attack system on datasets for text classification. The evaluation includes metrics such as the accuracy of the original model's predictions prior to the attack, referred to as "**OA**" or "**Original Accuracy**," as well as the accuracy of the model following the adversarial attack, referred

to as "**AAA**" or "**After-Attack Accuracy**." Additionally, the study reports on the percentage of perturbed words in relation to the original sentence length, referred to as "**PR**" or "**Perturbation Rate**."

	Word-CNN		**Word-LSTM**		**BERT**		**DistilBERT**		**ALBERT**		**RoBERTa**		**XLNet**	
	MR	**IMDB**	**MR**	**IMDB**	**MR**	**IMDB**	**MR**	**IMDB**	**MR**	**IMDB**	**MR**	**IMDB**	**MR**	**IMDB**
OA	79.40%	86.30%	80.70%	88.30%	87.50%	89.00%	83.90%	88.00%	88.00%	89.20%	90.30%	91.40%	90.70%	95.30%
AAA	04.10%	03.90%	01.24%	02.30%	08.40%	06.60%	0.40%	02.70%	0.2%	01.30%	03.40%	05.70%	05.40%	04.50%
PR	15.40%	11.30%	16.40%	13.20%	12.40%	11.70%	15.60%	10.30%	15.8%	12.30%	12.50%	11.80%	16.40%	09.80.%

A total of 500 samples that were accurately classified were extracted from the test set. Subsequently, the adversarial examples are generated by employing various attack algorithms. The present study involved subjecting a set of adversarial cases to seven cutting edge sentiment classifiers. The performance of various adversarial attacks was then compared with the proposed attack in this study, using ASR as the metric. This metric can show how effective the attack strategy is. A higher ASR value indicates that a particular attack type is more effective at deceiving the model. **Table 3.6** and **Table 3.7** present a summary of the primary outcomes of the HOMOCHAR attack method on the MR and IMDB datasets, alongside a performance comparison with previous attack techniques. **Table 3.5** illustrates that the models under consideration exhibit commendable performance in non-adversarial scenarios. Furthermore, it has been observed that non-transformer-based models exhibit a higher vulnerability to adversarial texts in comparison to transformer-based sentiment classifiers.

Table 3.6 Attack Results on models trained on **MR** dataset (****ASR***=*Attack Success Rate* & ****APR***=*Average Perturbed rate*)

Attacks	**BERT**		**DistilBERT**		**RoBERTa**		**ALBERT**		**WordCNN**		**WordLSTM**		**XLNet**	
	ASR %	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**
A2T[53]	29.29	11.67	33.71	14.41	30.43	12.94	54.50	14.47	46.23	13.00	54.50	11.33	31.95	14.26
BAE[52]	54.55	18.79	56.18	16.07	63.04	14.26	73.54	14.19	62.06	15.15	73.54	12.78	56.70	16.61
DWB[50]	91.92	22.63	98.88	18.43	97.83	14.97	95.21	19.98	97.49	19.90	99.21	17.37	95.88	22.39
Liang[69]	56.56	17.96	61.80	17.92	62.92	18.77	75.57	17.97	77.22	16.77	78.57	14.59	59.79	19.86
IGA[68]	90.91	14.36	93.26	16.96	92.13	16.55	96.56	17.32	96.98	15.13	96.56	12.68	95.88	17.15
Kuleshov[48]	90.91	13.94	**100**	13.05	98.75	13.22	98.37	12.71	97.40	12.03	98.37	11.31	95.88	14.13
Pruthi[47]	52.53	08.80	42.70	09.28	70.65	08.27	46.83	07.56	34.42	08.20	46.83	07.80	65.98	08.55
PSO[46]	93.94	19.99	92.13	17.80	95.65	14.09	92.41	14.94	96.40	12.03	98.41	13.04	93.81	14.73
PWWS[45]	88.89	15.12	91.01	13.76	92.39	12.37	96.30	13.19	94.97	13.00	96.30	11.74	88.66	12.59
Textbugger[44]	61.62	16.86	79.78	17.92	80.43	12.80	81.08	19.61	85.07	10.55	81.08	14.12	68.04	18.29
TextFooler[34]	96.97	20.48	97.75	15.09	**98.91**	14.21	99.47	15.83	97.75	13.75	99.47	11.66	95.88	16.64
HOMOCHAR	**98.98**	15.40	**100**	14.32	**98.91**	12.67	**99.64**	16.60	**97.78**	18.45	**99.64**	17.11	**96.91**	16.23

Table 3.7 Attack Results on models trained on **IMDB** dataset (****ASR***=Attack Success Rate & ****APR***=Average Perturbed rate)

Attacks	**BERT**		**DistilBERT**		**RoBERTa**		**ALBERT**		**WordCNN**		**WordLSTM**		**XLNet**	
	ASR %	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**
A2T[53]	25.20	10.44	38.52	15.41	38.88	11.38	50.34	14.02	42.34	16.08	59.96	12.45	38.67	13.09
BAE[52]	48.48	16.08	58.82	18.03	70.03	16.06	77.08	13.76	70.31	13.04	70.22	14.42	49.69	15.44
DWB[50]	95.62	18.43	97.72	16.66	95.55	16.22	94.72	18.02	94.88	15.28	98.46	16.02	92.12	19.89
Liang[69]	50.04	14.26	64.44	15.78	58.58	14.32	77.21	13.88	80.81	09.44	75.78	13.00	62.86	14.32
IGA[68]	89.87	11.28	90.12	12.44	90.21	13.84	96.02	13.04	92.55	12.78	97.87	10.34	89.85	14.00
Kuleshov[48]	88.62	13.42	**94.92**	11.28	94.66	11.58	97.72	12.78	97.46	13.68	97.02	11.67	96.44	13.54
Pruthi[47]	55.43	10.02	48.80	09.12	68.72	10.44	49.49	11.42	38.96	10.76	48.98	07.02	70.31	11.04
PSO[46]	94.76	21.38	90.10	16.66	91.92	19.00	92.68	14.48	92.12	19.49	95.84	17.78	87.82	16.68
PWWS[45]	85.52	14.44	95.62	11.64	97.77	12.28	92.21	14.43	96.64	13.42	98.20	14.00	88.42	11.24
Textbugger[44]	71.84	15.88	97.92	19.42	89.92	18.22	85.07	18.07	88.87	16.67	84.33	16.69	71.12	18.68
TextFooler[34]	94.78	19.46	94.94	13.32	94.48	17.68	98.71	12.64	96.66	18.28	97.92	13.44	92.56	13.07
HOMOCHAR	**95.91**	11.28	**97.94**	09.28	**98.29**	11.44	**98.77**	11.02	**97.82**	09.33	**98.96**	13.08	**97.47**	08.88

rate)

The HOMOCHAR method exhibits a notable ability to perturb a limited number of words, thereby achieving a considerably high rate of attack success. This approach outperforms baseline algorithms across all model. The perturbation of only a limited number of words in samples resulted in a success rate of 97.82% on the IMDB dataset and 97.78% on the MR dataset against the Word-CNN model. For the Word-LSTM model, the proposed method achieved an ASR value of 98.96% on the IMDB and 99.64% on MR dataset with a perturbance rate of less than 18% on both datasets. In comparison, all baselines failed to surpass this success rate. The ALBERT model demonstrates the highest ASR of 99.64% & 98.77% on MR and IMDB datasets respectively compared to other attack methods. The attack methodology employed in this study results in a DistilBERT model prediction accuracy of 0%, achieved through a perturbation rate of merely 14.32% on MR dataset, For IMDB dataset HOMOCHAR attains 97.94% on DistilBERT model. Despite the reputation as the top-performing model for various natural language processing tasks, BERT - a complex model with 110 million parameters - is still susceptible to HOMOCHAR adversarial attack. The findings indicate that with a perturbance rate of less than 16%, the proposed approach able to achieve an ASR of 98.98% & 95.91% on MR and IMDB respectively. Also, for RoBERTa and XLNet, HOMOCHAR outperforms prior cutting-edge attack techniques on both datasets. This indicates that the proposed attack system is capable of manipulating classifiers to generate faulty predictions.

For the purpose of evaluating the proposed attack model, the code is made available on GitHub[5] repository.

The additional goal of the research is to compare the susceptibility of different sentiment models to different types of adversarial perturbations. The objective is to determine the comparative vulnerability and resilience of various models to adversarial perturbations. On each targeted model, the attack's success rate of each attack is evaluated to determine which model is the most and least vulnerable. The calculation of the mean attack success rate for each model is determined through the utilisation of **Eqn. (3.8),** as illustrated below.

$$\boldsymbol{S_r} = \frac{\sum_{i=1}^{a} \frac{S_i}{S_i + F_i}}{a} \quad \textbf{(3.8)}$$

[5]https://github.com/Ashish250996/HOMOCHAR-adversarial-attack

$\boldsymbol{S_r} = Attack\ Success\ rate;\ \ \boldsymbol{a} = attack;\ \ \boldsymbol{S_i} = successful\ attack;\ \ \boldsymbol{F_i} = Failed\ attack$

(The Attack Success Rate is $\boldsymbol{S_r}$, no. of successful attacks is $\boldsymbol{S_i}$, the no. of unsuccessful attacks is $\boldsymbol{F_i}$, and the no. of attack recipes is $\boldsymbol{a}$. The statements the model initially incorrectly anticipated during its training are skipped statements. They were not included in the calculation)

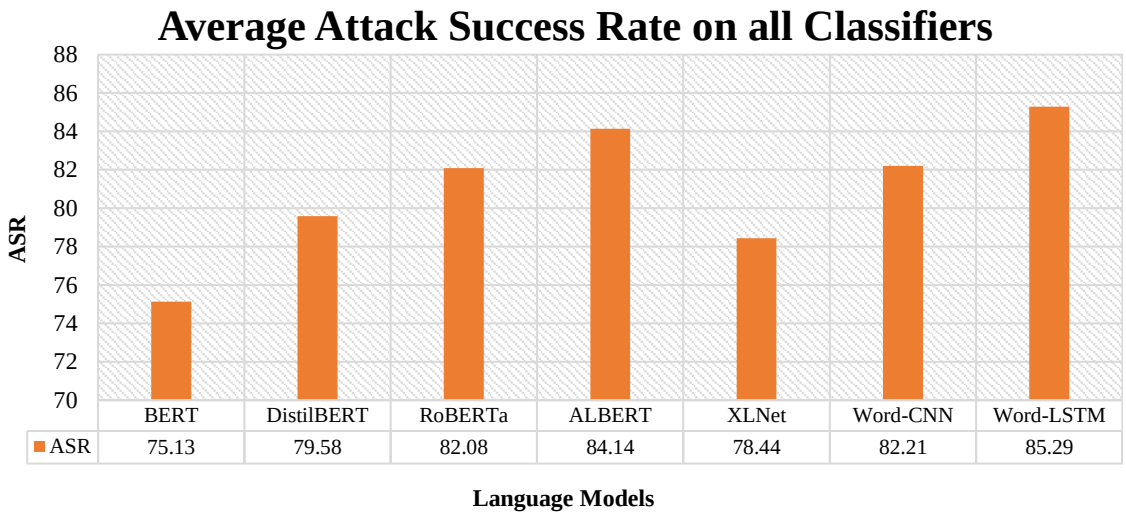


Figure 3.3 Mean success rate of sentiment classifiers on all adversarial perturbations

As illustrated in **Figure 3.3**, Upon evaluation, it has been determined that the Word-LSTM model exhibits the highest susceptibility to adversarial attacks. Among transformer models, model BERT is the least & ALBERT is the most vulnerable; the observations conclude that lighter models are more susceptible to these attacks as the ALBERT model comprises 128 embedding layers, 768 hidden layers, and 12 million parameters but on the other hand BERT base has 110 million parameters, which is a heavy model with a very high computational complexity which makes it least vulnerable as compared to all models.

Among the transformer models, the BERT model is the least vulnerable, while the ALBERT model is the most vulnerable. The observations lead to the conclusion that lighter models are more susceptible to these attacks. This is due to the fact that the ALBERT model has 128 embedding layers, 768 hidden layers, and 12 million parameters. On the other hand, the BERT base model has 110 million parameters, which is a heavy model with a very high computational complexity. This makes it the least vulnerable among all models. According to the findings of the study, it has been established that the Word-LSTM model demonstrates the greatest extent of vulnerability to adversarial attacks.

In order to evaluate the effectiveness and efficacy of various attack types in deceiving the model, based on their respective average perturbance rates. The mean success rate and perturbation rate for each attack type across all models are presented in **Figure 3.4**.

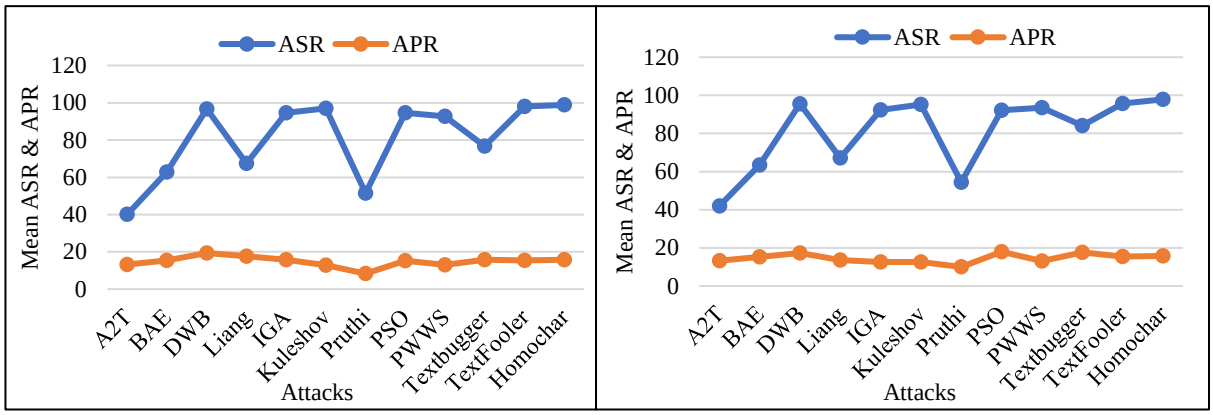


Figure 3.4 Mean ASR & APR score of each attack type on all models trained on **(a)** MR & **(b)** IMDB dataset

3.2.5 Further Analysis

An additional study is conducted to evaluate the effectiveness of the HOMOCHAR attack approach in various scenarios, including its overall execution time in producing an adversarial example. What is the requisite memory capacity for the generation of adversarial sequences? Additionally, the effectiveness and efficiency of the proposed methodology in enabling the scalability of sample size are assessed. Furthermore, its responsiveness to alterations in the semantic similarity score is noteworthy. The evaluation of scalability and sensitivity involved an analysis of the three models that were introduced in the study, namely Word-CNN, Word-LSTM, and BERT. Furthermore, the property pertaining to the transferability of adversarial examples generated through the proposed algorithm is also evaluated. The evaluation of its efficacy in the presence of random word perturbation is also conducted. The subsequent section presents a comprehensive evaluation of the HOMOCHAR methodology, encompassing various parameters.

Runtime Considerations: An investigation was carried out to assess the computational time consequences of the proposed framework. From the perspective of an attacker, the primary objective is to deceive a model through the execution of the proposed attack. The potential results of the average runtime to generate a single adversarial sequence using HOMOCHAR framework for each particular model is presented in **Figure 3.5**. Empirical evidence suggests that the process of producing adversarial samples for the IMDB dataset is time-consuming. The average length of the input review is 215.63 and 32 for the IMDB and MR datasets

respectively. Specifically, there exists a positive correlation between the time required to generate a single adversarial text and the average length of the input. As the input's size increases, the duration for producing a single adversarial text experiences a slight rise due to the increased time required to identify significant terms for perturbations.

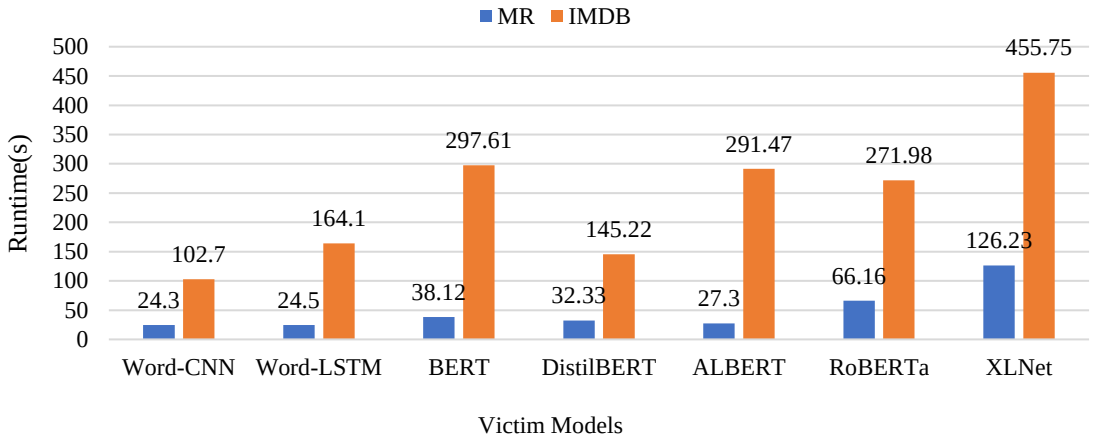


Figure 3.5 Average time required for generating an adversarial sample on each model.

Sensitivity: The norm constraint on image perturbations ($\|\boldsymbol{X} - \boldsymbol{X'}\|_\infty < \boldsymbol{\varepsilon}$) is a crucial determinant of an attack's effectiveness in computer vision. Elevated values of the variable $\boldsymbol{\varepsilon}$ result in an increased probability of misclassification for $\boldsymbol{X'}$. In the field of natural language processing, it is common to obtain invariance without much effort. For instance, when using a model at the word-level, the majority of perturbations generated by a character-level adversary result in an "unknown" token at the model's input. The cosine similarity function is employed in attack class for the purpose of limiting the word error rate (WER). A decrease in the cosine similarity score ($\boldsymbol{\delta}$) corresponds to an increase in the word error rate (WER), indicating that the model becomes more susceptible to perturbations. However, it can inevitably invalidate the constraint of human imperceptibility. The characteristic of a model that pertains to its responsiveness is commonly referred to as sensitivity. Therefore, in order to minimise the quantity of disturbances. In HOMOCHAR, the $\boldsymbol{\delta}$ is assigned a value of **0.7** which will maintain WER such that it is sensitive to perturbations while preserving the semantic coherence of the sentence. To assess the sensitivity of a model, one must observe the fluctuations in the ASR scores in relation to variations in $\boldsymbol{\delta}$, which is defined within the range of [0.1,1] as shown in **Figure 3.6**(b) & **Figure 3.7**(b).

Scalability: To assess the efficacy of the suggested algorithm, a collection of test specimens has been established within a predetermined range of values that have been subjected to HOMOCHAR-induced perturbations. To evaluate the variability of ASR values in relation to

changes in the scale of the test samples, the success rates of three models were examined during the process. As the scope of the test samples was progressively increased throughout this procedure, there was a corresponding increase in the mean runtime required to produce adversarial samples. As illustrated in **Figure 3.6** (a) and **Figure 3.7** (b). The observation drawn from the illustration suggests an unfavourable relationship between the population sample size and the ASR scores. This correlation is characterised by a discernible downward trend as the sample size increases. However, it is noteworthy that the ASR scores do not exhibit a huge substantial variation with changes in the sample size.

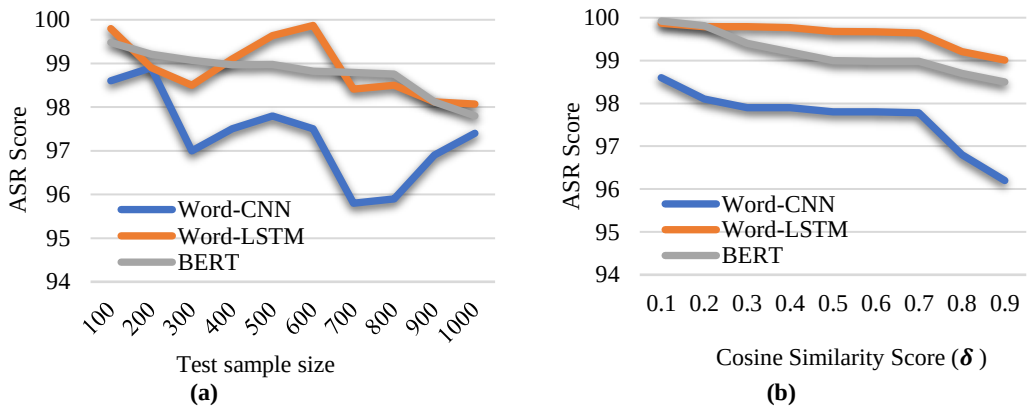


Figure 3.6 The fluctuations in the ASR values with the variations in **(a)** test sample size and **(b)** cosine similarity score for the models trained on **MR** dataset

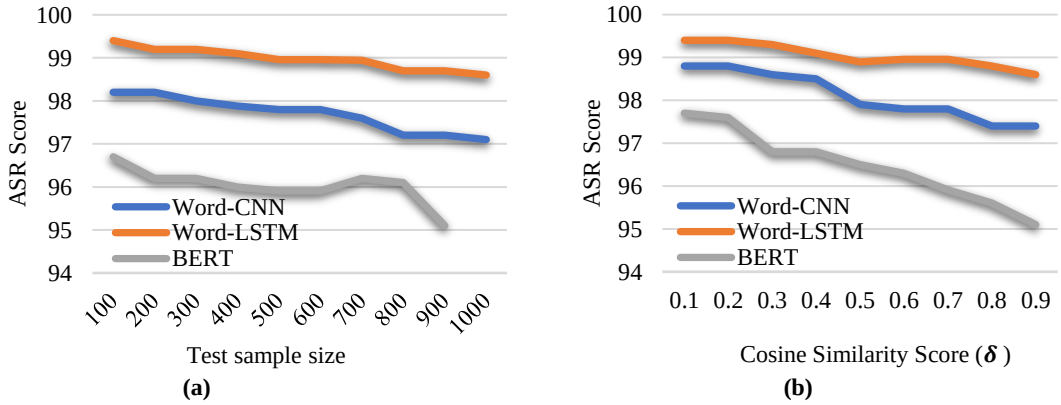


Figure 3.7 The fluctuations in the ASR values with the variations in **(a)** test sample size and **(b)** cosine similarity score for the models trained on **IMDB** dataset

Utility Analysis: It is evident that the adversarial texts produced by HOMOCHAR exhibit a greater degree of similarity to the original texts in comparison to those generated by conventional baseline attack algorithms. From **Figure 3.2**, it can be concluded that the adversarial examples generated through proposed methodology are more effective in preserving utility. The rationale behind this is that the baseline methods encompass a range of

linguistic errors such as misspellings, insertions, transpositions, and synonym substitutions etc., The HOMOCHAR algorithm operates by substituting characters in the original review with homoglyph characters, resulting in a perturbed review where each character appears visually identical. This technique is designed to deceive neural text classifiers. Sometimes, the perturbed review does not appear adversarial even to the adversary, this can be observed in the very last example presented in **Figure 3.2**. Homographs exhibit distinct tokenization and are considered "out of vocabulary" in the word embedding domain. Therefore, it can be asserted that perturbation generated from HOMOCHAR can be most crucial technique for conducting malicious manipulation in text classification.

Transferability: This investigation examines the transferability of adversarial text, specifically, the extent to which adversarial samples generated from one model can deceive a different model. The present investigation involved the collection of adversarial examples from the MR test set that exhibited misclassification by the Word-CNN model. Subsequently, the predictive ASR of the aforementioned examples was assessed in comparison to two additional target models. The findings presented in **Table 3.8** indicate that there exists a moderate level of transferability among the models. Furthermore, the adversarial samples that were produced using the BERT model, which exhibited greater prediction accuracy, demonstrated a higher degree of transferability.

Table 3.8 Transferability of adversarial examples on MR dataset. Row ***i*** and column ***j*** is the accuracy of adversaries generated for model ***i*** evaluated on model ***j***.

		Word-CNN	**Word-LSTM**	**BERT**
MR	**Word-CNN**	----	79.10%	89.70%
	Word-LSTM	64.70%	----	82.40%
	BERT	81.20%	81.80%	----

Random word perturbation: The data presented in **Table 3.9** indicates that the act of randomly selecting words to modify, as denoted by "Random," exhibits minimal impact on the ultimate outcome. This suggests that indiscriminately altering words would not deceive classifiers, and it is imperative to select significant words to modify for a successful attack.

Table 3.9 Comparison of the ASR scores via random selection of words or via words selected by computing the importance score for perturbation.

Model	**Dataset**	**Accuracy**	**Random**		**HOMOCHAR (use heuristic score to find important words)**	
			Success Rate	**Perturbed word**	**Success Rate**	**Perturbed word**
Word-CNN	MR	79.40%	18.60%	15%	97.70%	18.45%
	IMDB	86.30%	32.50%	15%	97.82%	09.33%

Model	Dataset	Accuracy	Random		HOMOCHAR (use heuristic score to find important words)	
			Success Rate	Perturbed word	Success Rate	Perturbed word
Word-LSTM	MR	80.70%	21.70%	15%	99.64%	17.11%
	IMDB	88.30%	24.20%	15%	98.96%	13.08%
BERT	MR	87.50%	41.30%	15%	98.98%	15.40%
	IMDB	89.00%	37.60%	15%	95.91%	11.28%

Implementation and memory details*:* The experimentation process was carried out utilising NVIDIA RTX A5000 GPUs, with a system memory of 128GB. Equipped with a total of 48GB of graphics memory, driver version 460.32.03, CUDA version 11.2, and 10TB GB of hard disc space. The experiments were conducted in a repeated manner, with each experiment being replicated 5 times. The reported value is the mean of the obtained results. The significance of this replication lies in the stochastic nature of training, which results in variability in performance. Stop-words are commonly filtered out during feature extraction in various NLP tasks, including this experiment. This is due to the observation that the presence or absence of stop-words has minimal influence on the outcome of the prediction results. The experiments conducted in this study employ the 200-dimensional GloVe embeddings[6], which were trained on a corpus of 840 billion tokens sourced from Common Crawl. Moreover, a semantic similarity threshold of 0.7 is established to ensure an optimal balance between the calibre and potency of the produced adversarial text. ***Memory utilization***: During the development of 500 adversarial samples on both datasets, an average of 8.3 GB of RAM, 3.9 GB of graphics memory, and 26.2 GB of disc space were utilised in this investigation.

3.2.6 Discussion & Future Work

NLP models that rely on text as input are vulnerable to a diverse range of subtle perturbations that have the potential to modify model output and prolong inference runtime while leaving the visual appearance of the input unchanged. These attacks consist of arbitrary character substitutions, insertion, deletion, and substitution of essential words with synonyms. Using homoglyphs, this article presents a novel method for fooling neural sentiment classifiers. Although they have occasionally been observed in obfuscating spam and phishing scams detection mechanisms in the past, it seems that the designers of the various NLP systems that are presently being implemented on a large scale have disregarded these concerns entirely. This article explores the phenomenon of adversarial attacks on natural language sentiment

[6] http://nlp.stanford.edu/projects/glove/

classification applications through the utilisation of homoglyphs. The experimental findings reveal that HOMOCHAR attack is superior to other conventional methodologies in terms of both success rates and average perturbation. In addition, the user study demonstrated that it was challenging for users to recognise homoglyph adversarial examples as perturbed text.

3.2.6.1 Future Scope

Extension to Targeted Attack: This paper solely focuses on conducting untargeted attacks, which involve altering the output of the model. However, it is worth noting that HOMOCHAR exhibits the potential for facile customization in the context of targeted attacks, wherein the model can be compelled to produce a predetermined label. The deliberate modification of a model to generate a predetermined outcome, commonly referred to as a targeted attack., involves a modification of the goal function component in the proposed method.

Extension to wider NLP applications: As future work or research opportunities, we will expand the scope of HOMOCHAR attack to distort a variety of NLP applications such as toxic content detection, rumour detection, smishing & phishing detection etc., in addition to sentiment analysis.

Apply HOMOCHAR attack on API platforms: The advent of machine learning has spurred a proliferation of companies offering their own Machine-Learning-as-a-Service (MLaaS) platforms, which are tailored to Deep Learning Text Understanding tasks, such as text classification. The models are deployed on cloud-based servers and user access is limited to utilising an application programming interface exclusively. In situations where such a setting is present, a perpetrator is devoid of information regarding the structure of the model, its parameters, or the data used for training. Their sole capability lies in querying the target model, with the output being in the form of prediction or probability scores. The HOMOCHAR model, as developed in the present study, has demonstrated efficacy in operating within black-box scenarios. In future research, it may be feasible to carry out a HOMOCHAR attack on these digital platforms.

Adversarially Robust Generalization: Using an adversarial training technique[53],[70] (defensive technique), we will also attempt to propose a model that is resistant to all adversarial perturbations in conjunction with HOMOCHAR. The current investigation delves more extensively into the phenomenon of adversarial examples within the framework of text classification, for future work our aim is to strengthen these systems and enhance the accuracy

of classification algorithms through the implementation of adversarial training techniques. To ensure the resilience of applications against malevolent manipulations, it is advisable to suggest that all enterprises involved in the creation and distribution of NLP systems incorporate these protective measures.

3.2.6.2 Limitation

The findings of this study indicate the presence of adversarial perturbations in natural language. However, the effectiveness of the perturbations could be enhanced by utilising a more advanced algorithm, such as particle swarm optimisation, to identify and modify significant words. This approach has the potential to further improve the outcomes of the proposed attack.

3.2.7 Conclusion

The present study aims to investigate the vulnerability of automatic sentiment classification to adversarial attacks. The results unambiguously indicate that sentiment analysis can be disrupted by altering the vocabulary and syntax for machine learning algorithms while preserving semantic equivalence for human evaluators. The present study examines a deficiency in deep learning models utilised for the purpose of sentiment analysis. By taking advantage of this weakness, in this paper, HOMOCHAR, a novel framework designed to generate adversarial text sequences capable of misleading deep learning networks. Furthermore, the study also demonstrated a comparative analysis of various sentiment classifiers to determine which model is more susceptible to adversarial perturbations and which is more robust against them. In general, empirical evidence has demonstrated the feasibility of perturbing automatic sentiment prediction models through adversarial modifications. Therefore, it is imperative to prioritise the development of adversarial robust generalisations over standard generalisations in order to advance societal progress. Furthermore, this study advocates for the exploration of models that exhibit higher resilience against adversarial attacks, as opposed to solely relying on higher accuracy scores.

3.3 Non-Alpha-Num: a novel architecture for generating adversarial examples for bypassing NLP-based clickbait detection mechanisms

3.3.1 Abstract

The vast majority of online media rely heavily on the revenues generated by their readers' views, and due to the abundance of such outlets, they must compete for reader attention. It is a

common practise for publishers to employ attention-grabbing headlines as a means to entice users to visit their websites. These headlines, commonly referred to as clickbaits, strategically leverage the curiosity gap experienced by users, enticing them to click on hyperlinks that frequently fail to meet their expectations. Therefore, the identification of clickbaits is a significant NLP application. Previous studies have demonstrated that language models can effectively detect clickbaits. Deep learning models have attained great success in text-based assignments, but these are vulnerable to adversarial modifications. These attacks involve making undetectable alterations to a small number of words or characters in order to create a deceptive text that misleads the machine into making incorrect predictions. The present work introduces "*Non-Alpha-Num"*, a newly proposed textual adversarial assault that functions in a black box setting, operating at the character level. The primary goal is to manipulate a certain NLP model in a manner that the alterations made to the input data are undetectable by human observers. A series of comprehensive tests were conducted to evaluate the efficacy of the suggested attack approach on several widely-used models, including Word-CNN, BERT, DistilBERT, ALBERTA, RoBERTa, and XLNet. These models were fine-tuned using the clickbait dataset, which is commonly employed for clickbait detection purposes. The empirical evidence suggests that the attack model being offered routinely achieves much higher attack success rates (ASR) and produces high-quality adversarial instances in comparison to traditional adversarial manipulations. The findings suggest that the clickbait detection system has the potential to be circumvented, which might have significant implications for current policy efforts.

3.3.2 Proposed Architecture

The process of generating textual adversarial examples is structured as a system consisting of four key components: an objective function, a set of restraints, a modification mechanism, and a searching technique which are discussed in depth in this section[43]. The aim of this system is to search for a perturbation from $\boldsymbol{x}$ to $\boldsymbol{x_{adv}}$ that can deceive a predictive NLP model. This perturbation should be able to achieve a specific objective, such as causing the model to predict an incorrect classification label. Additionally, it must adhere to a predefined set of limitations. The searching technique aims to identify a series of transformations that lead to a successful perturbation. The Proposed adversarial attack architecture is depicted in **Figure 3.8**.

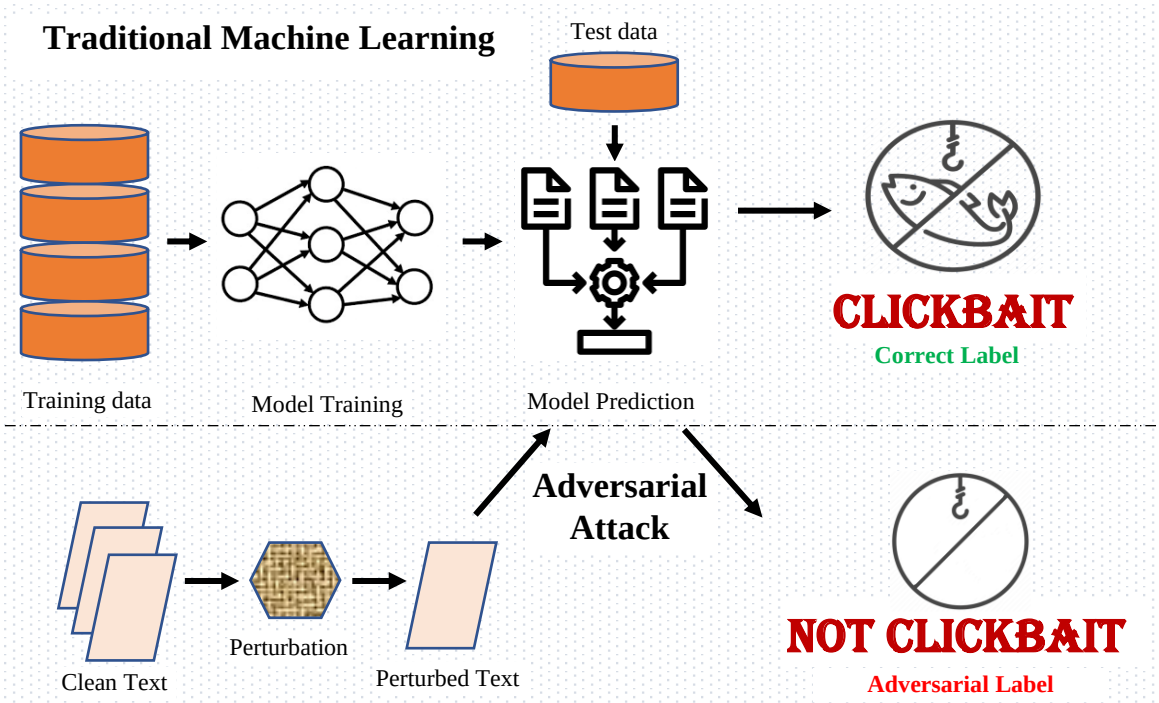


Figure 3.8 Proposed architecture of obfuscating clickbait detection mechanism.

Modification Function

The input undergoes a transformation process, resulting in the generation of several potential perturbations. If **X** is represented as a vector $(x_1,...,x_i,...,x_n)$, substituting x_i with a modified version of x_i' will lead to a perturbed text. The focus of this work is to examine the substitution of significant words with character-level perturbed terms. The Non-Alpha-Num adversarial attack involves the random insertion of non-alphanumeric characters from a list that contain **(!"#$%&'() *+, -. /: ;<=>? @ [\] ^` {|})** into the most significant words. Subsequently, the words that have been perturbed are substituted with their original counterparts. The aforementioned selections are made based on their ability to maintain the semantic closeness of the phrase as seen by human observers, despite the fact that neural text classifiers tokenize them in a different manner. Non-Alpha-numeric characters are also referred to as punctuation. Punctuation insertion might be a viable attack vector since grammar checkers struggle to identify punctuation while also not significantly impacting the content of the statement. Deep learning models perform poorly when punctuation is removed because punctuation includes crucial information that models require to function properly. Furthermore, punctuation can include antagonistic downstream information that undesirable users might use.

One crucial point in these perturbations is that words are symbolic entities, and Deep Learning frameworks that depends on learning-based approaches often employ a dictionary to represent

a finite collection of potential words. The size of the average word dictionary is considerably less compared to the potential combinations of characters of a comparable length. In the context of English words, it may be seen that the total number of potential combinations is around $\mathbf{26^n}$, where n signifies the word length. This implies that premeditatedly perturbed important tokens allow for their effortless conversion into "unknown" words, which are not recognized by the dictionary. In deep learning modelling, any words that are not recognized or known will be allocated to the "unknown" embedding vector[44]. The investigation from the study provides compelling evidence that the use of random punctuation insertion is a straightforward method can significantly manipulate the behaviour of text categorization models, leading to erroneous outcomes.

Searching Technique

The search mechanism is responsible for identifying the most optimal perturbations based on the modification function. The score is allocated to the optimal collection of altered words. The task entails the utilization of the beam search technique[71]. This algorithm employs a heuristic scoring function, as described in **Eqn. (3.9)**. In this function, for a given text $\boldsymbol{x}$, all potential perturbed texts $\boldsymbol{x'}$ are formed by substituting each word $\boldsymbol{x_i}$, and then scored.

$$Score\ (\boldsymbol{x'}) = \mathbf{1} - \boldsymbol{F_y}(\boldsymbol{x'}) \qquad \textbf{(3.9)}$$

The function $F_y(\boldsymbol{x})$ represents the projected probability of class $\boldsymbol{y}$ by the model, whereas $\boldsymbol{y}$ represents the actual output of the original text $\boldsymbol{x}$. The highest-ranking $\boldsymbol{b}$ texts are retained, with $\boldsymbol{b}$ being referred to as the "beam width." The iterative process continues by applying additional perturbations to each of the top $\boldsymbol{b} = \mathbf{8}$ perturbed texts, resulting in the generation of the subsequent set of candidate texts. The computational complexity of this operation is $\boldsymbol{O}$ ($\boldsymbol{b} * \boldsymbol{W^2} * \boldsymbol{T}$), where $\boldsymbol{W}$ is the number of words in the input. The variable $\boldsymbol{T}$ denotes the maximum number of modification options that are accessible for a given input.

Set of Restraints

A collection of linguistic restrictions is utilized to ensure that $\boldsymbol{x}$ and perturbed $\boldsymbol{x'}$ exhibit similarity in terms of both meaning and fluency, rendering $\boldsymbol{x'}$ a viable prospective adversarial example. The search space should be designed in a way that ensures the proximity of $\boldsymbol{x}$ and $\boldsymbol{x'}$ in the semantic embedding space. We employed the Universal Sentence Encoder ($\boldsymbol{USE}$) in this work to assess the semantic similarity of textual occurrences by utilizing cosine similarity[57]. The cosine similarity between two $\boldsymbol{n}$-dimensional vectors, represented as $\boldsymbol{m}$ and $\boldsymbol{n}$, is defined

by **Eqn. (3.10)**. the word embedding vectors $\boldsymbol{e_{x_i}}$ and $\boldsymbol{e_{x_i'}}$ must satisfy a specified minimal threshold, in Non-Alpha-Num attack, the value of the threshold is taken as $\boldsymbol{\tau = 0.6}$.

$$\boldsymbol{\tau = Cosine\ Similarity(m,n) = \frac{m.n}{||m||.||n||} = \frac{\sum_{i=1}^{k} m_i \times n_i}{\sqrt{\sum_{i=1}^{k} m_i^2} \times \sqrt{\sum_{i=1}^{k} n_i^2}}} \quad \textbf{(3.10)}$$

The notation $\boldsymbol{T(x) = x'}$ is employed to represent transformations that perturb $\boldsymbol{x}$ to $\boldsymbol{x'}$. Additionally, it is assumed that the $\boldsymbol{j-th}$ constraints are represented as Boolean functions $\boldsymbol{C_j(x, x')}$, which indicate whether $\boldsymbol{x'}$ satisfies the constraint $\boldsymbol{C_j}$. Next, the search space $\boldsymbol{S}$ can be formally defined using mathematical notation as shown in **Eqn. (3.11)**.

$$\boldsymbol{Sim(x) = \{T(x) | C_j(x, T(x)) \forall j \in [m]\}} \quad \textbf{(3.11)}$$

The objective of the algorithm for searching is to identify an element $\boldsymbol{x'}$ that belongs to the set $\boldsymbol{Sim(x)}$ and is capable of deceiving the victim model. Additionally, in the set of restraints, the Language Tool[58] is employed to minimize the occurrence of grammatical mistakes while ensuring uniformity in part-of-speech usage. Specifically, the alternative term selected should possess the identical part of speech as the original word. The support taggers offered by SpaCy, NLTK and flair aim to maintain semantic consistency between $\boldsymbol{x}$ and $\boldsymbol{x'}$.

Table 3.10 Algorithm of the Proposed Attack Framework

Algorithm 1: Non − Alpha − Num Adversarial Attack

Aim: **Adversarial attack framework to fool neural clickbait classifier**
Input: **legitimate input** x **and its ground truth label** y, **classifier** $F(.)$, **threshold** τ, **semantic similarity** $Sim(.)$.
Output: **Generated an adversarial sequence** x'

1. **Initialization:** $x' \Leftarrow x$
2. **for** x_i in **X do**
3. Compute score (x_i')
4. **end for**
5. W_{Ranked} ⇐Ranking($x_1, x_2, x_3 \ldots\ldots\ldots x_n$) in descending order
6. Remove the stop words in W_{Ranked}
7. **for** x_i **in** W_{Ranked} **do**
8. x' ⇐Substituting x with (words with Punctuations)
9. **if** $Sim\ (x, x') <= \tau$ **then**
10. **return None**:
11. **else if** $F(x') \neq y_{true}$, **then**
12. **Solution found. return** x'.
13. **end if**
14. **end for**
15. **return None**

Searching Technique

Modification

Restraints

Objective Function

Objective Function

The effectiveness of an assault is assessed by considering the model outcomes and employing an objective function. The search strategy is explored, coupled with modifications and a

specific set of restrictions, until it results in the misinterpretation of the true output[43]. **Table 3.10** presents the algorithm employed in the suggested architecture.

Comparison of Clickbait detector

In order to get a deeper comprehension of the challenges associated with clickbait detection approaches, the experiment has been bifurcated into two distinct phases. The first training of cutting-edge deep learning-based linguistic models is conducted utilizing the clickbait dataset. The baseline textual assaults and Non-Alpha-Num adversarial attacks are employed to alter the trained models in order to misclassify clickbait as non-clickbait and vice versa. **Figure 3.9** illustrates the methodology for conducting an assault on the clickbait detector. The objective of this approach is to ascertain the relative susceptibility of different models to adversarial manipulations. The authors assert that this study represents the inaugural endeavour within the literature to juxtapose clickbait detectors in order to assess their susceptibility to hostile scenarios.

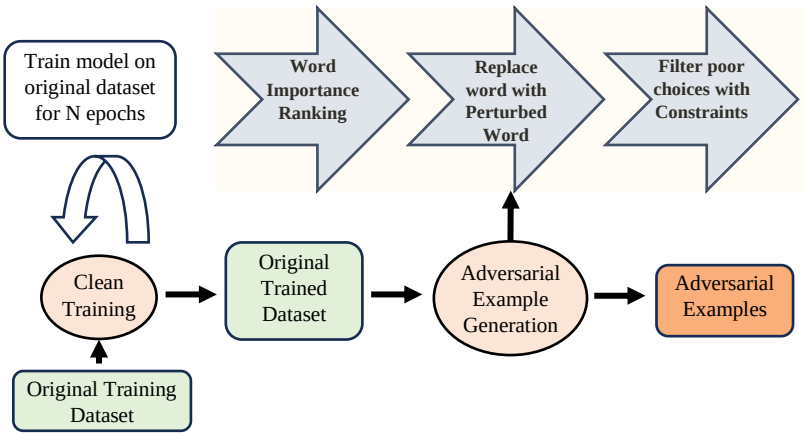


Figure 3.9 Comparison of different clickbait classifiers

3.3.3 Experimental Design & Approach

This section provides an introduction to the dataset, victim models, attack tactics, assessment metrics, and execution details. Subsequently, in the subsequent part, we will evaluate the data and examine numerous probable factors contributing to the measured outcome.

3.3.3.1 Description of Dataset

This study aims to examine the impact of malicious textual examples on a widely used benchmark dataset within the domain of clickbait detection. The test set is employed for the production & assessment of the adversarial instances. **Table 3.11** offers a concise summary of the dataset.

Clickbait[72]**:** The current investigation employs the clickbait dataset in order to conduct a clickbait identification task. The clickbait dataset originates from Chakraborty et al.'s paper[72]. The corpus comprises news titles extracted from a corpus of news articles annotated with two labels, namely: **0**: "*Not-Clickbait*" & **1**: "*Clickbait*". For non-clickbait, the headlines were taken from a repository of 18,513 Wikinews articles compiled by NewsReader, whereas for clickbait, the headlines were retrieved from 8,069 online articles of 'BuzzFeed', 'Upworthy', 'ViralNova', 'Scoopwhoop', and 'ViralStories' news portals. The average sample length of the dataset is 8.50 words, while the average length of clickbait headlines is 10 and the average length of non-clickbait headlines is 7.

Figure 3.10 illustrates the percentage of distribution of both clickbait and non-clickbait headlines, also it depicts the percentage of word contractions, hyperbolic words, determiners, and stop words in both clickbait and non-clickbait headlines. The data has already been pre-processed based on the methodology outlined in their paper, and it is accessible to the public via the Hugging face library. The hugging face datasets library offers an API to facilitate the acquisition of public datasets. There are a total of 16000 samples in the dataset. In our experimental configuration, we divided the dataset into three, namely 0.80, 0.10, and 0.10, for training, testing, and validation purposes, respectively.

Table 3.11 Concise Description of the Dataset

Task	Application domain	Granularity	Classification	Dataset	Labels	Train	Validation	Test	Average Length
Clickbait Detection	News Media	News Headlines	Binary	Clickbait Dataset	2	12.8K	1.6K	1.6K	08.50

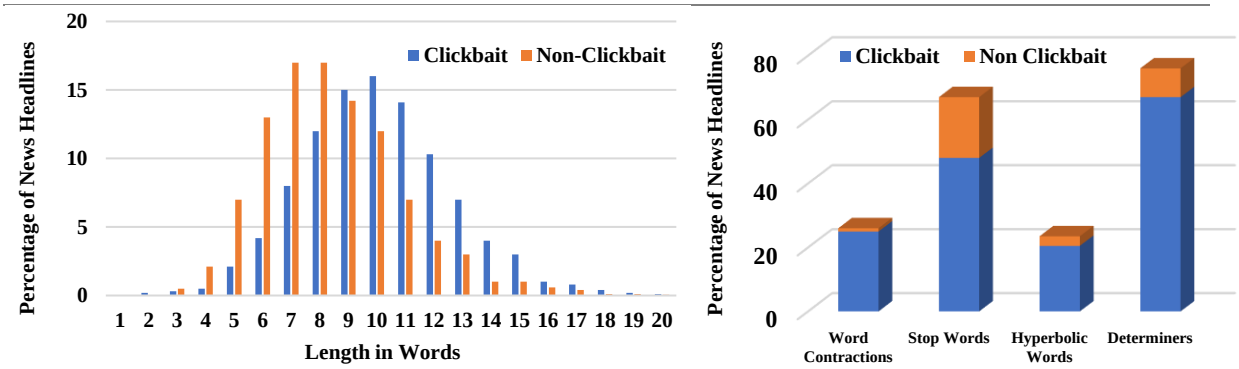


Figure 3.10 (a) Percentage distribution of Clickbait & Non-Clickbait News Headlines **(b)** Percentage distribution of word contractions, hyperbolic words, determiners, and stop words in News titles

3.3.3.2 Models Utilized

This part encompasses the depiction of the models that underwent training on the clickbait dataset, along with their respective parameter sets. The metric employed for assessing models

in the context of binary classification for clickbait detection is the accuracy score. The next section contains empirical evidence that showcases the vulnerability of each clickbait model to adversarial situations. This vulnerability is measured by a decrease in accuracy resulting from various adversarial assault strategies.

Description of the Models

Deep neural network models demonstrate the capacity to independently acquire knowledge and identify relevant characteristics, resulting in enhanced precision and efficacy. The study employed a variety of well-known deep-learning classification methods, such as basic CNN & various transformer networks. **Figure 3.11** presents a comprehensive representation of the models employed in this study.

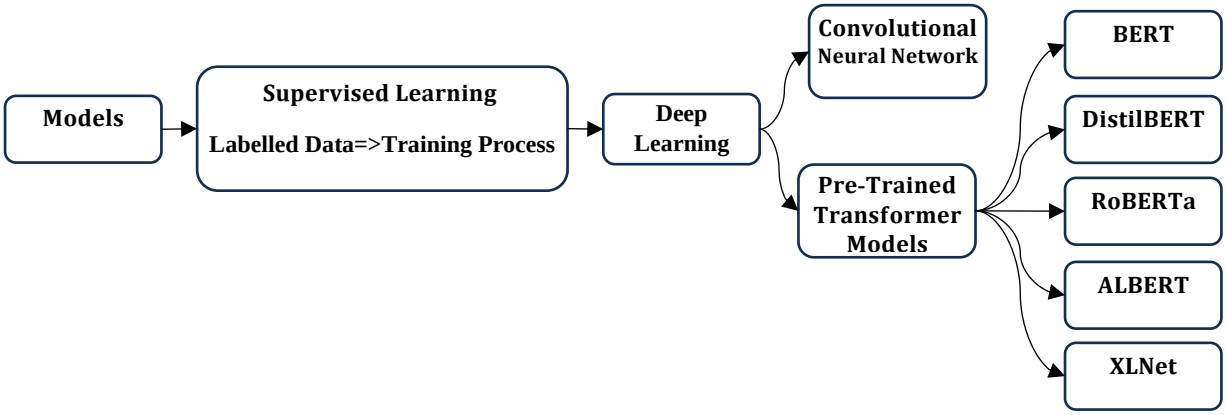


Figure 3.11 Synopsis of the models employed in the investigation

Convolutional Neural Network (***CNN***): In Convolutional Neural Networks, Layers of convolution extract features through the process of screening the input information, wherein the combination of numerous filters results in the generation of outputs. subsampling or pooling is an approach utilized in CNNs to decrease the granularity of feature maps in various tiers. This process aims to improve the network's robustness against distortions and disturbances. Pooling is a technique used to decrease the dimensionality of the output produced by a certain layer in order to pass it on to the succeeding layer. The execution of categorization operations is carried out by the use of completely linked layers[62]. Convolutional Neural Networks (CNNs) have a notable capability in effectively identifying local patterns as well as patterns that are invariant to changes in position. The utilization of Convolutional Neural Networks (CNNs) has been found to be highly beneficial in the domain of text classification.

Transformer Models*:* Deep neural networks that employ transformers incorporate a self-attention mechanism to allocate various levels of importance to distinct regions of the input data. **Table 3.12** displays a comprehensive descriptive study of many pre-trained classifiers.

Table 3.12 A thorough examination of the transformer variants.

Model	Trained on Datasets	Tuning	Description	Advantages	challenges
BERT [63]	English Wikipedia & the BookCorpus	Fine-tuned on target dataset, pre-trained on specific parameters	The model under consideration is a bidirectional transformer architecture that incorporates both Masked Language Modelling (MLM) methods and Next Sentence Prediction (NSP).	1.) The ability to efficiently handle and analyse context-specific data. 2.) Training at a Faster Pace	1.) The process of categorizing is limited to a certain vocabulary. 2.) The total length of the input phrases is predetermined and remains constant. 3.) Exhibits issues pertaining to reasoning that is logical. 4.) The processing expense is substantial.
DistilBERT [**37**]	English Wikipedia & the BookCorpus	Fine-tuned on target dataset, pre-trained on specific parameters	The approach adopted in this study utilizes an early iteration of BERT, with a reduction in the total amount of tiers by a factor of 2. Furthermore, the implementation of adaptive masking has been carried out.	1.) The implementation of preliminary training has been employed to enhance the effectiveness of linguistic simulation proficiency. 2.) When comparing the freshly designed model to BERT, it demonstrates improved acceleration and decreased weight.	1.) The set-length restriction is a limitation imposed on a system or process that requires a specific length to be adhered to. This constraint ensures that the system or process operates within
ALBERT[**38**]	English Wikipedia & the BookCorpus	Fine-tuned on target dataset, pre-trained on specific parameters	A modified version of BERT has undergone feature reduction, leading to a more lightweight version. The diminution of factors is easily accomplished by employing factorized embedding layer parameterization & cross-layer parameter allocation	1.) Reduced memory utilization 2.) One potential area of improvement for BERT is the enhancement of its training pace.	1.) framework exhibits incompatibility for problems that involve the production of textual content. 2.) The text demonstrates deficiencies in logical reasoning.
RoBERTa [66]	English Wikipedia & the BookCorpus, CC − News, Stories, OpenWebText	Fine-tuned on target dataset, pre-trained on specific parameters	The present study aims to replicate the BERT model by employing an enlarged training dataset and fine-tuning the hyper-parameters. Additionally, the	1.) The utilization of a larger volume of preliminary training data has been shown to enhance effectiveness.	1.) The attribute requires a significant number of resources. 2.) The current undertaking necessitates substantial

Model	Trained on Datasets	Tuning	Description	Advantages	challenges
			utilization of dynamic masking is incorporated in this replication effort.	2.) The performance of the model surpasses that of both XLNet and BERT.	computer resources and involves a lengthy processing time.
XLNet[**40**]	English Wikipedia & the BookCorpus, Giga5, ClueWeb 2012-B, Common Crawl	Fine-tuned on target dataset, pre-trained on specific parameters	It blends auto-regressive models with bi-directional context modelling to overcome BERT's drawbacks and surpass BERT on 20 tasks, frequently by a wide margin in sentiment analysis, question answering, document rating, natural language inference,	Using greater amounts of initial training data improves efficacy.	The absence of fictitious symbols, such as [MASK], utilized by BERT during the pretraining phase, in actual data during the finetuning phase leads to a disparity between the training and finetuning processes.

Parameter Configurations

Table 3.13 displays the settings of the parameters for every framework that underwent training on the clickbait dataset, aiming to evaluate the effectiveness of adversarial attack approaches.

Table 3.13 Configuration of parameters for the intended classifiers

Models	Parameter configurations
Word − CNN	For the aim of this investigation, the CNN model developed by Kim et al. [62] was used. The Word-CNN model utilizes a total of 100 filters and incorporates three different window sizes, namely $3, 4, and\ 5$. The selected dropout rate of the framework is 0.3, and it utilizes a basis of $200-$ dimensional GLoVE embeddings. Afterward, a fully linked layer is utilized, which is then followed by a time-dependent max-pooling layer in order to facilitate the process of categorization. The algorithm used has achieved a test set for accuracy of 89.47% on the clickbait dataset.
BERT	The "bert − base − uncased" model is subjected to a training process consisting of 10 iterations. Each iteration involves a batch size of 64, a learning rate of $2e-05$, and a maximum sequence length of 128. The objective of this training is to enhance the model's performance in sequence classification specifically for the clickbait dataset. The framework underwent training by employing a cross − entropy loss mechanism. The framework attained its maximum efficacy in this assignment, as determined by the accuracy of the test set, which reached 91.55% after eight epochs.
DistilBERT	The "distilbert − base − uncased" network was trained for 10 epochs using a batch size of 64, an average rate of learning of $2e-05$, and an optimal sequence length of 128. This optimization was performed specifically for sequence classification on the clickbait dataset. The cross-entropy loss function was employed during the training of the model. The accuracy of the evaluation set, determined after 5 epochs, revealed that the model achieved a maximum accuracy of 90.83% on this particular job.
ALBERT	We enhanced the performance of the "albert − base − v2" model for text categorization on the clickbait dataset. The model was trained for 10 epochs using a batch size of 64, a learning rate of $2e-05$, and a maximum sequence length of 128 bits. The model was trained using a cross − entropy loss function. The greatest accuracy score achieved by the model in this task, as evaluated by the test set accuracy, was 91.72%.
RoBERTa	The performance of the "Roberta − base" model has demonstrated enhancement in sequence classification when applied to the specific dataset utilized in our experiment. The model was

Models	Parameter configurations
	executed using a maximum sequence length of 128 and a batch size of 64 for a total of 10 epochs, with a learning rate of $2e-05$. The model was trained using a $cross-entropy$ loss function, as it was a problem with categorization. The assessment of the model's performance on this particular task resulted in the highest score of 92.12% for accuracy, which was attained after 8 epochs.
XLNet	The efficiency of the "$xlnet-base-cased$" classifier on the clickbait dataset was improved. The training process involved training the model for a total of 10 epochs. During each epoch, a batch size of 64 was used to process the data. The learning rate, which determines the step size at each iteration of the training process, was set at $2e-05$. Additionally, a maximum sequence that measured 128 bits was specified to handle the input data. The training of the model was conducted with a $cross-entropy$ loss function. The model earned a maximum accuracy score of 91.96% in this task, as determined by evaluating the test set accuracy.

Table 3.14 displays the testing scores for the accuracy of all of the models that completed training using the clickbait dataset.

Table 3.14 Testing Accuracy of the Targeted Models

	Word − CNN	BERT	DistilBERT	ALBERT	RoBERTa	XLNet
ACC	89.47%	91.55%	90.83%	91.72%	92.12%	91.96%

3.3.3.3 Attack Assessment criteria

The effectiveness of strategies for attack has been empirically demonstrated by employing three evaluation parameters: **Post Attack Accuracy**, **Attack Success Rate**, and **Average Perturbed Rate**. The approach employed to evaluate and clarify the three indicators is outlined below.

- **Post Attack Accuracy:** The primary objective of adversarial assaults is to undermine the effectiveness of the classifiers. Performance measures, such as accuracy, are commonly employed for the evaluation of classification work. The accuracy scores have been presented both before and after the attack. The utilization of potent adversarial attacks has been found to result in a notable reduction in accuracy scores due to the efficacy of their attack techniques.

- **Attack Success Rate:** To analyze the effectiveness of the attack techniques, a random sample of five hundred correctly classified instances is chosen from the test set. This ensures that the evaluation is not influenced by the classifiers' classification accuracy. The source texts are subsequently subjected to attack algorithms in order to produce adversarial examples. Subsequently, the adversarial instances are forwarded to neural clickbait classifiers in order to provide the ultimate prediction. The success rate of the attack algorithm is determined by utilizing the percentage of erroneous predictions generated by these classifiers. A higher success rate indicates that the assault algorithm possesses the

ability to generate more powerful adversaries, perhaps leading to the malfunctioning of these clickbait classifiers. We use attack success rate **ASR** (ratio of successful attack samples to the total of successful and failed samples$\frac{(\boldsymbol{successful\ samples})}{\boldsymbol{successful+failed\ samples}}$) to determine the effectiveness of an attack technique against the victim classifier. The successful samples refer to those that can misclassify the true prediction, whereas failed samples are unable to incorrectly categorize the genuine outcome. In an analytical context, an assault is considered effective when the algorithm $\boldsymbol{F}$ accurately classifies the initial legal input $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{y}$, but predicts incorrectly the attacked input $\boldsymbol{F}(\boldsymbol{x} + \boldsymbol{\Delta x}) = \boldsymbol{y}'$. Therefore, the ASR may be mathematically represented as depicted in **Eqn. (3.12).**

$$\textbf{ASR} = \frac{\boldsymbol{F}(\boldsymbol{x}+\boldsymbol{\Delta x})=\boldsymbol{y}'}{(\boldsymbol{F}(\boldsymbol{x}+\boldsymbol{\Delta x})=\boldsymbol{y}')+(\boldsymbol{F}(\boldsymbol{x}+\boldsymbol{\Delta x})=\boldsymbol{y})} \quad \textbf{(3.12)}$$

In the setting of untargeted attacks, the symbol $\boldsymbol{y}'$ represents any label that differs from $\boldsymbol{y}$. The sign $\boldsymbol{\Delta x}$ is used to represent modifications made to the initial specimen. In the present context, a successful assault is defined as the ability of an adversarial sample to make inaccurate predictions with a significantly high level of confidence. In the event of an unsuccessful assault, the adversarial sample lacks the ability to cause misclassification of the true prediction. The missing statements refer to those that the model wrongly categorized throughout the training process. Our research focuses on analysing the success rates of assaults and evaluating the effectiveness of those assaults in incorrectly identifying outcomes.

3.3.4 Experimental Results & Analysis

To comprehend the susceptibility of clickbait classifiers. The initial stage of the investigation involves employing the offered clickbait dataset to train sophisticated deep-learning models. The settings for the parameters of each model are shown in prior section. The models that have been trained can potentially be manipulated by employing the *Non-Alpha-Num* adversarial attack strategy. After subjecting the test samples to perturbation using the technique described, the resulting drop in accuracy scores is presented in **Table 3.15**. The first measurement and recording of the accuracy of the intended models on the original test specimens is referred to as the Before-Attack Accuracy (**BAA**). Following this, the effectiveness of the target models is assessed by exposing them to adversarial samples created using the offered attack technique. Post-attack accuracy (**PAA**) refers to the score of accuracy obtained after the proposed attack has been executed. In addition, the study presents findings about the proportion of altered words in relation to the initial sentence length, denoted as Perturbation Rate (**PR**).

Table 3.15 comparison of the accuracy of each model before and after the proposed adversarial attack algorithm is conducted. (***BAA**=Before Attack Accuracy, ***PAA** =Post Attack Accuracy, ***PR**=Perturbed Rate)

	Word − CNN	**BERT**	**DistlBERT**	**ALBERT**	**RoBERTa**	**XLNet**
BAA	89.47%	91.55%	90.83%	91.72%	92.12%	91.96%
PAA	08.48%	19.30%	13.70%	15.80%	24.80%	21.9%
PR	12.52%	13.81%	14.07%	14.75%	12.09%	15.44%

In order to assess the efficacy of the suggested attack technique in comparison to traditional attack methods against clickbait classifiers, the (ASR) metric is employed. For this, a set of 500 samples, which were accurately classified were taken from the test set. Following this, the generation of adversarial cases is achieved by the utilization of different attack methods. The current investigation entailed exposing a collection of adversarial instances to six state-of-the-art clickbait detectors. In this study, the performance of several adversarial approaches was evaluated and compared with the proposed attack, with the metric of ASR being used. This statistic can demonstrate the level of effectiveness of the assault approach. A greater ASR value signifies that a particular assault type has a higher level of effectiveness in misleading the model. **Table 3.16** provides a concise overview of the main results obtained from the Non-Alpha-Num attack method when applied to the clickbait dataset. Additionally, it includes a comparative analysis of the effectiveness of this attack strategy with earlier attack methods.

Table 3.16 Attack Outcomes on different models (∗ **ASR** = Attack Success Rate & ∗ **APR** = Average Perturbed rate)

Attacks	**Word-CNN**		**BERT**		**DistilBERT**		**ALBERT**		**RoBERTa**		**XLNet**	
	ASR %	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**	**ASR %**	**APR %**
A2T [53]	15.8	14.7	12.3	16.3	13.8	15.5	14.7	14.4	07.2	13.6	09.2	06.7
BAE [73]	64.4	18.6	56.5	20.2	58.4	14.6	56.2	11.8	49.6	14.9	51.7	10.6
Checklist [51]	16.8	13.4	13.4	19.7	18.8	20.9	13.8	18.1	09.2	17.1	19.9	08.7
DWB [50]	68.7	11.8	53.9	19.2	59.6	10.3	55.2	13.1	48.5	11.4	50.6	12.8
HotFlip [74]	76.6	14.9	68.8	11.6	70.6	13.7	69.1	14.1	60.2	12.1	63.7	11.1
IGA [49]	69.4	15.2	59.0	12.9	63.7	15.5	60.1	14.9	50.6	13.9	54.4	12.8
InputReduction [75]	34.9	11.0	21.6	10.8	29.1	13.8	25.4	11.6	15.3	14.7	19.2	13.6
Kuleshov et al. [48]	76.5	17.5	65.4	18.3	69.9	12.6	68.4	13.7	49.6	14.5	55.5	14.2
Pruthi et al. [47]	32.8	08.7	21.3	09.2	28.4	08.2	26.6	09.9	19.7	09.2	26.6	16.3
PSO [**22**]	75.3	16.8	63.9	14.7	69.7	13.2	63.0	15.8	52.9	15.1	56.7	13.6
PWWS [**21**]	55.8	18.2	47.1	17.4	55.8	11.7	50.6	16.5	42.5	15.6	45.8	14.3
Textbugger [44]	95.6	13.4	90.1	12.8	93.8	13.6	90.6	13.8	86.2	14.8	89.6	13.8
TextFooler [34]	72.7	15.9	65.3	13.4	68.9	14.7	66.7	12.6	59.7	13.7	61.6	14.5
NonAlphaNum	**95.8**	10.6	**91.6**	12.4	**94.0**	11.1	**92.2**	11.2	**86.5**	12.6	**91.4**	12.5

The Non-Alpha-Num technique exhibited a notable impact on a restricted set of words, resulting in a significantly elevated rate of success in attacks. The aforementioned technique

demonstrates superior performance compared to the benchmark perturbation techniques throughout all of the models. The manipulation of a restricted set of words within the data yielded a success rate of 95.8% when tested against a Word-CNN method, with an alteration rate of just under 11%. In contrast, none of the baselines were able to exceed this level of success. The mean length of the clickbait dataset sample ranges from $8\ to\ 9$ words. To execute effective assaults, the Non-Alpha-Num adversarial attack method disturbed a mere percentage, namely less than 14%, of the words inside a singular sample. Research investigations have indicated that transformer representations can be deceived by the proposed attack approach, as long as the interference ratio remains under 20%. The attack approach under consideration introduced little perturbations, affecting just a small fraction of words, typically about 2-3, inside a single sample. The deception of transformer models through attacks has been seen, as long as the tampering rate stays under 20%. The ALBERT model has the highest (ASR) of 92.2% when compared to alternative attack strategies on the given dataset. The assault mechanism utilized in this work effectively deceives the DistilBERT model's predictions to a minimal extent, accomplished by introducing an average word perturbation rate of just 13.10%. Whilst being widely regarded as the leading model for a range of tasks relating to natural language processing, BERT, a sophisticated model with 110 M parameters, is vulnerable to Non-Alpha-Num adversarial attack. The results suggest that by maintaining a perturbance rate below 14.4%, the suggested methodology may successfully get an ASR of 91.6% on the clickbait dataset. Moreover, in the case of RoBERTa and XLNet, the suggested methodology demonstrates superior performance compared to previous state-of-the-art attack methodologies. This observation suggests that the suggested attack system possesses the ability to manipulate classifiers in order to provide inaccurate predictions.

In addition to the aforementioned decline in accuracy scores, we propose the concept of attack Success Rates (ASR) as a method for evaluating the efficacy of each assault. Furthermore, the Average perturbed Rates (APR) are provided, whereby the calculation entails dividing the number of disturbed words by the whole length of the text. The application of these 2 criteria enables the assessment of different adversarial assault strategies across many models. The objective of this research is to illustrate the relative level of danger that certain attack methods pose to distinct models. Furthermore, the employment of average ASR scores assists in illustrating the classifier that has the most susceptibility to adversarial manipulations. The work also aims to analyse the vulnerability of various clickbait classifiers to various forms of adversarial interference. This study aims to assess the relative susceptibility & robustness of

different models when subjected to adversarial perturbations. The success rate of each assault is assessed on each targeted paradigm in order to ascertain the relative vulnerability of each of the models as shown in **Figure 3.12**. The mean attack success rate for each model is calculated using **Eqn. (3.13)**.

$$\boldsymbol{Success_{avg}} = \frac{\sum_{i=1}^{n} \frac{\boldsymbol{Successful_i}}{\boldsymbol{Successful_i + Failed_i}}}{\boldsymbol{n_{attacks}}} \tag{3.13}$$

The Attack Success Rate will be denoted as $\boldsymbol{Success_{avg}}$, where $\boldsymbol{Successful_i}$ represents the number of successful attacks, $\boldsymbol{Failed_i}$ represents the number of unsuccessful assaults, and $\boldsymbol{n_{attacks}}$ represents the number of attack recipes. The skipped sentences are the assertions that the machine originally mis predicted during its training. They were excluded from the computation.

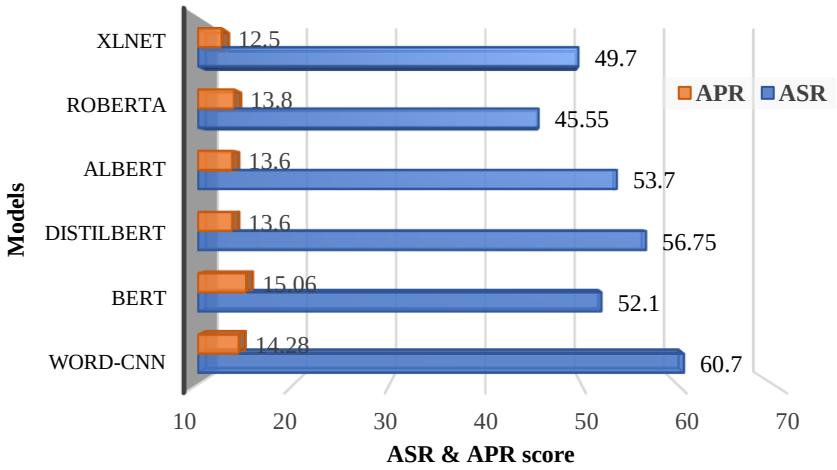


Figure 3.12 Average ASR and APR scores of all classifiers

As seen in **Figure 3.12**, subsequent examination has revealed that the Word-CNN model demonstrates the greatest vulnerability to adversarial assaults. Among transformer models, it has been observed that the RoBERTa model exhibits the least vulnerability, while the DistilBERT model is found to be the most susceptible. This observation leads to the conclusion that lighter models, such as DistilBERT[64] and ALBERT models are more prone to these attacks. This vulnerability can be attributed to the fact that the ALBERT[65] model possesses an architecture, consisting of 128 embedding layers, 768 hidden layers, and with only 12 million parameters, whereas the DistilBERT model is also a condensed variant of the BERT model. In the preliminary training stage, the BERT model experienced a process known as knowledge distillation, which led to a decrease in its overall size by 40%. Significantly, the aforementioned

reduction in size was accomplished while maintaining 97% of the model's language understanding skills. This characteristic renders RoBERTa the least susceptible among all the models. The results might be important for those who frequently utilize well – established, cutting – edge algorithms in their efforts to spot clickbait. The reader will possess the ability to determine the most appropriate model that corresponds to their particular problem. Moreover, this phenomenon acts as a motivating factor for academics to develop models that exhibit adversarial robust generalizations rather than traditional generalizations.

To ascertain the comparative efficacy of assault approaches in deceiving the model with a reduced average perturbance rate. The average rate of success and alteration rate for each assault type across all models have been computed and are presented in **Figure 3.13**. Next, we proceed to assign rankings to different assault techniques, as presented in **Table 3.17**.

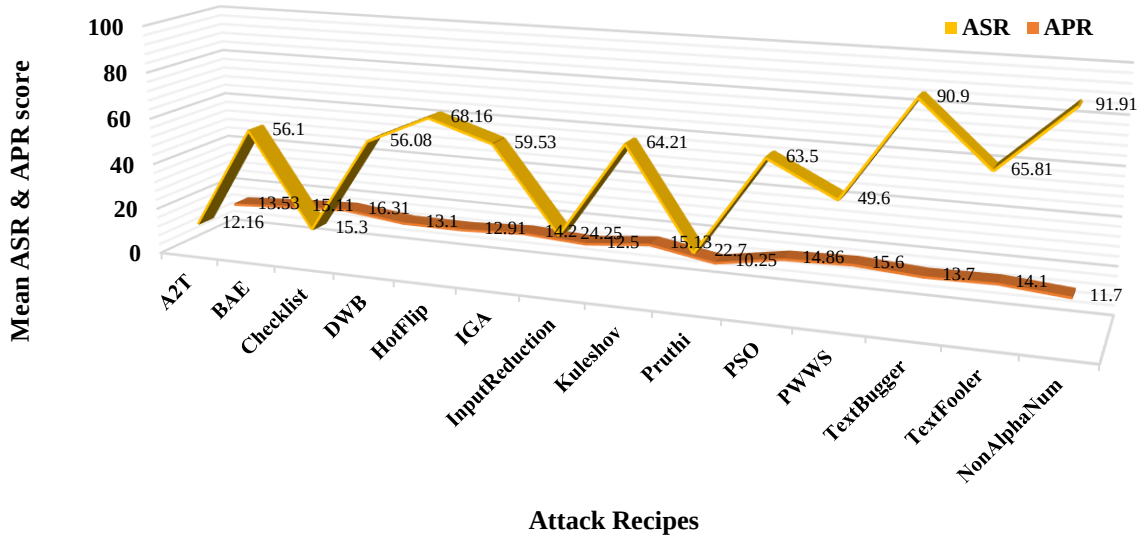


Figure 3.13 Mean ASR & APR score of each attack type on all models

The results depicted in **Figure 3.13** clearly demonstrate that the Non-Alpha-Num attack algorithm, which operates at the character level, exhibits the highest level of effectiveness in terms of perturbation. Following closely behind is TextBugger, another character-level perturbation attack. On the other hand, Hot-Flip emerges as the most effective word – level perturbation technique. Conversely, the attack method A2T, which employs gradient – based synonym word swap within the white-box adversarial setting, is found to be the least effective across all models, as indicated by a comprehensive evaluation of assault tactics.

Table 3.17 The average ASR for each assault recipe on each classifier

Attack Methodology	Mean ASR%	Mean APR%	Perturbation level
NonAlphaNum	91.91	11.7	Char – level
Textbugger [44]	90.95	13.7	Char – level
HotFlip [74]	68.16	12.91	Word – level
TextFooler [34]	65.81	14.1	Word – level
Kuleshov et al. [48]	64.21	15.13	Word – level
PSO [46]	63.5	14.86	Word – level
IGA [49]	59.53	14.2	Word – level
BAE [73]	56.10	15.11	Word – level
DWB [50]	56.08	13.1	Char – level
PWWS [45]	49.6	15.6	Word – level
InputReduction [75]	24.25	12.5	Word – level
Pruthi et al. [47]	22.7	10.25	Char – level
Checklist [51]	15.3	16.31	Char – level
A2T [53]	12.16	13.53	Word – level

The most effective attacks shown in **Table 3.17** are char-level perturbation assaults, specifically the Non-Alpha-Num suggested attack technique and TextBugger, which is a prominent conventional attack tactic. While the majority of attacks operate at the granularity of word-level perturbations, it is worth noting that the top two assaults specifically target character-level perturbations.

3.3.5 Further Investigation & Analysis

A further investigation is undertaken to assess the efficacy of the Non-Alpha-Num assault methodology across several circumstances, encompassing its entire execution duration in generating an adversarial instance. The evaluation also includes an assessment of the transferability property of adversarial instances created by the proposed technique. Moreover, the utility evaluation of the adversarial words generated by Non-Alpha-Num demonstrates a certain level of resemblance to the genuine text.

Execution time: A study was conducted to evaluate the computing time implications of the proposed framework. The fundamental aim of an attacker is to manipulate a model by implementing the intended attack. **Figure 3.14** displays the probable outcomes of the average runtime required to produce a solitary adversarial example utilizing the Non-Alpha-Num assault architecture for each specific model. Analysing the research findings, it is evident from

the Figure that BERT requires the longest duration in developing an adversarial example. Conversely, lighter models like ALBERT and DistilBERT exhibit less time. Furthermore, the non-transformer-based model, namely Word-CNN, demonstrates the least amount of time needed for producing an adversarial sequence.

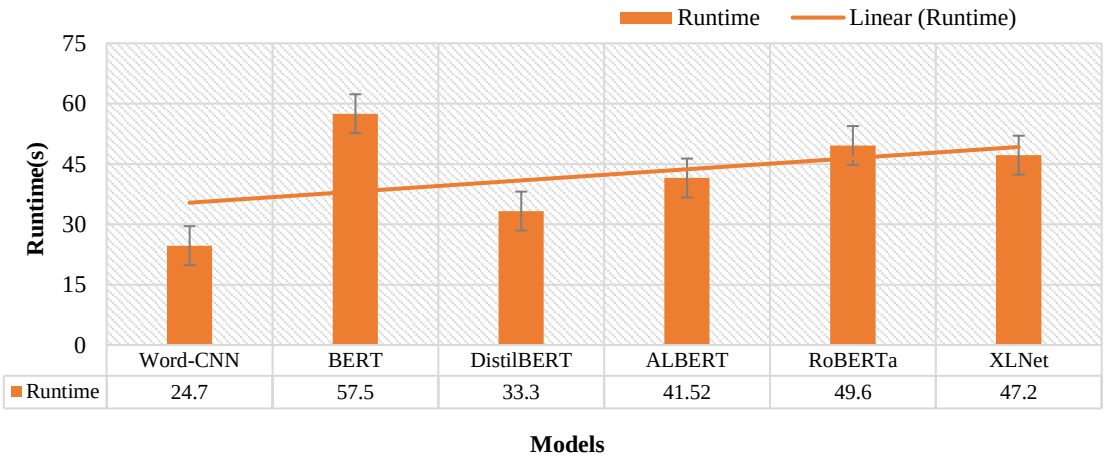


Figure 3.14 Mean time to produce an adversarial sample for each model.

Accountability and Applicability: The attack method presented in this study generates hostile texts that exhibit a greater degree of resemblance to the source texts. Based on the findings shown in **Figure 3.15**, it can be inferred that the viability preservation of adversarial instances created by Non-Alpha-Num is comparatively higher in terms of its accountability and applicability. Char-level attacks refer to a range of language blunders, such as misspellings, transpositions, and random character swaps[44]. Considering the proposed solution, the adversary can create changes through the insertion of punctuation marks that cannot be concealed by detection techniques.

Adversarial Transferability: The present study aimed to investigate the property of adversarial transferability in the text by assessing the effectiveness of adversarial instances developed by one classifier in deceiving a various classifier[76],[77]. This research study involved the collection of 100 adversarial examples generated through the utilization of Non-Alpha-Num Adversarial Attack. These examples were specifically chosen as they were incorrectly classified by a designated target model. The ASR scores of these examples were subsequently evaluated against alternative target models. The results displayed in **Table 3.18**. demonstrate that the Word-CNN model exhibits a moderate level of transferability.

Table 3.18 The Transferability of Adversarial Examples on a Clickbait Dataset. The ASR of adversaries developed for model p, when assessed on model q, is represented by row p and column q.

	Word – CNN	BERT	DistilBERT	ALBERT	RoBERTa	XLNet
Word – CNN	-----	65.57	63.21	68.82	59.44	58.91
BERT	42.14	-----	39.63	34.44	32.16	40.86
DistilBERT	48.76	45.54	-----	49.75	48.91	45.09
ALBERT	52.12	49.96	51.16	-----	47.72	41.66
RoBERTa	43.37	39.28	50.63	38.56	-----	40.63
XLNet	42.90	42.00	48.87	49.27	48.84	-----

Figure 3.15 Adversarial example generation using punctuation marks (non-alpha numeric characters) to evade clickbait detection mechanisms

Explainability & Interpretability: The LIME approach, as introduced by Ribeiro et al. [78], is utilized to provide localized interpretations regarding our algorithms. The LIME methodology employs a linear framework to estimate the local decision boundary for each example by fitting it to the associated data. The example was perturbed in order to get the acquired information. To evaluate the accuracy of the regional interpretations obtained from LIME, the area over perturbation curve (AOPC) is utilized as a quantitative measure. The mathematical expressions for this metric are provided in **Eqn. (3.14)**.

$$\mathbf{AOPC}=\frac{1}{M+1}\sum_{m=1}^{M}\frac{1}{N}\sum_{1=1}^{N}(X_{(0)}^{(j)})-F(X_{(m)}^{(j)}) \quad \textbf{(3.14)}$$

Within the framework of this research, the notation $\boldsymbol{X}_{(\boldsymbol{0})}^{(\boldsymbol{i})}$ refers to a specific occurrence where $\boldsymbol{X}^{(\boldsymbol{j})}$ is denoted as $(\mathbf{0})$ without any words being removed. On the other hand, $\boldsymbol{X}_{(\boldsymbol{m})}^{(\boldsymbol{j})}$ represents an instance where the $\boldsymbol{m}$ most significant words are excluded and denoted as $(\boldsymbol{m})$. The function $\boldsymbol{F}(\boldsymbol{X})$ is employed to represent the degree of confidence of the model in relation to the predicted target label $\boldsymbol{Y}^{(\boldsymbol{j})}$. The function $\mathbf{F}(\mathbf{X})$ is employed to represent the degree of confidence of the model in relation to the predicted target label $\mathbf{Y}^{(\mathbf{j})}$. The AOPC idea refers to the average change in the model's confidence towards the target label when the top $-$ **m** most significant words, as recognized by LIME, are removed. This concept is drawn from intuitive understanding. For assessment, an arbitrary number of 500 occurrences was picked from the test set. During the explanation generation process utilizing LIME, a set of 500 altered samples is created for each instance, using the proposed attack algorithm to evaluate the explanations. The selection of M=10 was made for the AOPC metric. Based on the data shown in **Table 3.19**, it is evident that the Word-CNN model achieves the greatest (AOPC) score. The analysis reveals that the AOPC scores of RoBERTa models are notably lower than those of other models, suggesting that RoBERTa may exhibit a comparatively decreased degree of explainability in comparison to all models.

Table 3.19 The AOPC ratings were calculated for the LIME explanations of each model. A model that has a higher AOPC score possesses greater interpretability.

Models	Word-CNN	BERT	DistilBERT	ALBERT	RoBERTa
AOPC Scores	0.68	0.36	0.48	0.47	0.23

3.3.6 Conclusion

The primary objective of the research is to examine the susceptibility of clickbait detection algorithms to adversarial assaults. The findings unequivocally demonstrate that the identification of clickbait may be impeded by modifying the lexicon and sentence structure in algorithms used for machine learning yet retaining semantic correspondence for human assessors. The current investigation focuses on a limitation seen in deep learning models employed for the purpose of clickbait detection tasks. This study introduces Non-Alpha-Num,

a unique framework that exploits a vulnerability to produce hostile text sequences with the intention of deceiving deep learning networks. Additionally, the research also conducted a comparative examination of several neural clickbait detection systems in order to ascertain the relative vulnerability of each model to adversarial perturbations, as well as their respective levels of resilience against such perturbations. Empirical data, in a broad sense, has substantiated the viability of perturbing strategies designed for clickbait detection through the implementation of adversarial alterations. Hence, it is crucial to give precedence to the advancement of antagonistic strong generalizations to foster societal growth.

3.4 Bypassing Neural Text Classification Mechanism by Perturbing Inflectional Morphology of Words

3.4.1 Abstract

Advanced neural text classifiers have shown remarkable ability in the task of classification. The investigation illustrates that linguistic frameworks have an inherent vulnerability to adversarial texts, where a few words or characters are altered to create perturbed text that misleads the machine into making incorrect predictions while preserving its intended meaning among human viewers. The present study introduces Inflect-Text, a novel approach for attacking text that works at the level of individual words in a situation where the inner workings of the system are unknown. The objective is to deceive a specific neural text classifier while following specified language limitations in a manner that makes the changes undetectable to humans. Extensive investigations are carried out to evaluate the viability of the proposed attack methodology on various often utilized frameworks, inclusive of Word-CNN, Bi-LSTM and three advanced transformer models, across two benchmark datasets: AG news and MR, which are commonly employed for text classification tasks. Experimental proof demonstrates that the suggested attack architecture regularly outperforms conventional methods by achieving much higher attack success rates (ASR) & generating better adversarial examples. The findings suggest that neural text classifiers can be bypassed, which could have substantial ramifications for existing policy approaches.

3.4.2 Motivation & Importance of the Investigation

Prior research on social bias in NLP predominantly concentrates on diverse characteristics. We explore a distinct feature in the field of NLP that has received little attention: Language

proficiency and knowledge in linguistics[79]. Modern NLP algorithms were developed under an unconscious presumption that all individuals understand proficient English, which is frequently of U.S. origin[80]. However, it is important to note that more than one billion English speakers, which accounts for 2/3 of the global English-speaking population, use English as a second language (L2)[81] as shown in **Table 3.20**. The data has been extracted from a source on Wikipedia[7]. Despite those who are native speakers, a considerable proportion communicate using a dialect such as African American Vernacular English (AAVE) instead of Standard English.

Table 3.20 Over one billion individuals speak English as their second language.

Language	Family	Branch	First Language (L1) Speakers	Second Language (L2) Speakers	Total Speakers (L1 + L2)
English	Indo-European	Germanic	380 million	1.077 billion	1.456 billion

Employing these algorithms in production without mitigating this inherent bias exposes them to the possibility of engaging in linguistic discrimination, resulting in subpar performance for various speech groups such as AAVE[82] and L2 speakers[83]. This may manifest as either a lack of comprehension of these individuals or a misinterpretation of their words. For instance, the recent misinterpretation of a social media message made by a minority speaker led to his unjustifiable apprehension[79]. The McArthur circle[84] of world English illustrated in **Figure 3.16** unequivocally demonstrates that not all individuals communicate in mainstream U.S. English.

Within the realm of natural language processing (NLP), we investigate a distinct aspect that has received a limited amount of attention: language ability and knowledge in linguistics. The development of modern natural language processing algorithms was based on the unintentional assumption that all people are capable of comprehending competent English, which is typically of American extraction [80]. On the other hand, it is essential to take into consideration the fact that more than one billion people who speak English, which constitutes two thirds of the total population of people who speak English worldwide, use English as a second language (L2). There is a sizeable population that communicates using a dialect other than Standard English, such as African American Vernacular English (AAVE), despite the fact that there are native speakers of the language.

[7] https://en.wikipedia.org/wiki/List_of_languages_by_total_number_of_speakers

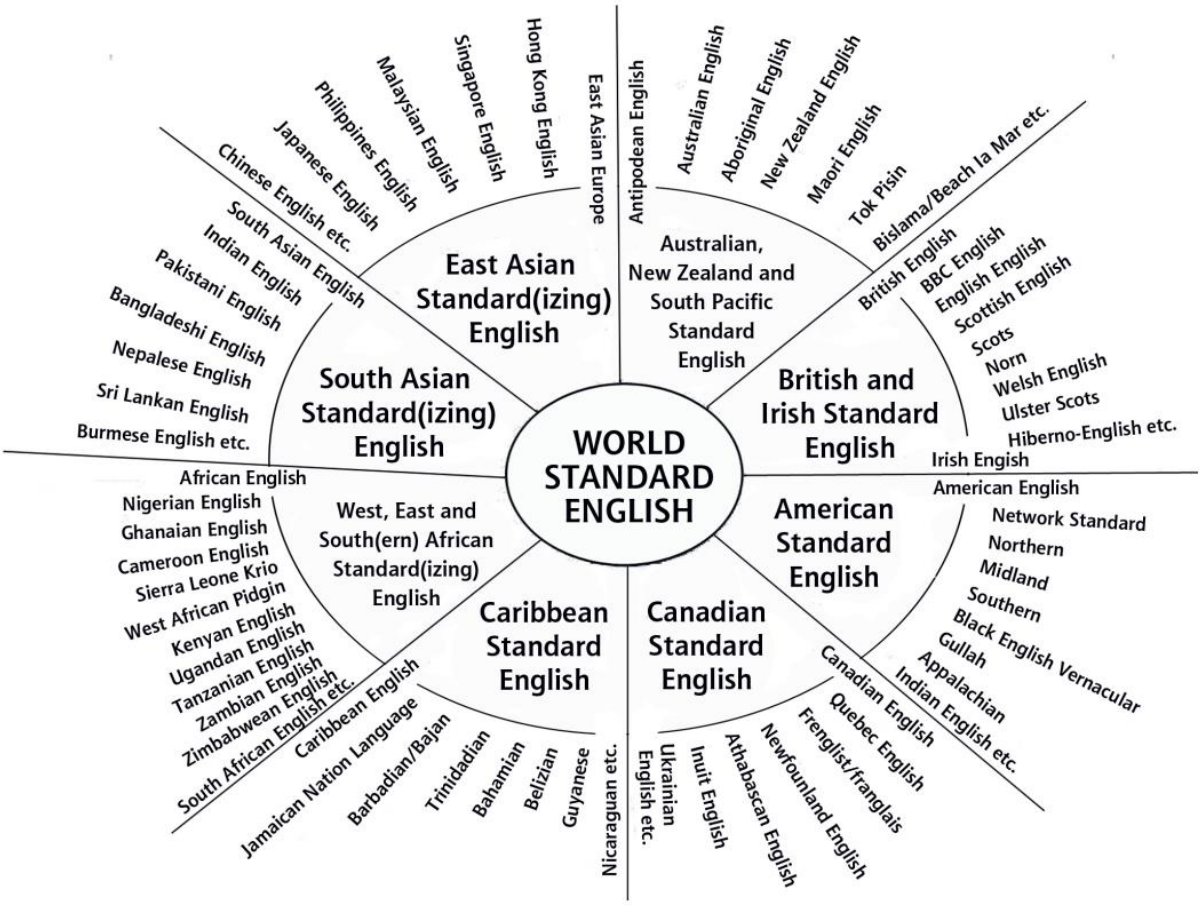


Figure 3.16 McArthur's typology of English language variations[84]

Considering the observed diversity in the production of inflectional morphology across L2 and many L1 dialect speakers[83], We suggest that Linguistic architectures ought to become capable of managing inflected instabilities[85] well to reduce the risk of perpetuating linguistic prejudice hence this article highlights the brittleness of neural text classifiers to inflectional perturbations. In this study, we provide a new approach called Inflect-Text, which generates convincing and semantically equivalent adversarial instances by modifying the inflections in the clean examples. Unlike prior research on adversarial manipulations in textual domain, we utilize morphology to generate our adversarial instances.

3.4.3 Proposed Architecture

This section provides a detailed discussion of the proposed "**Inflect-Text**" adversarial attack framework. The architecture of the attack is presented, consisting of four modules that are described in greater detail

3.4.3.1 Attack Methodology

The generation of textual adversarial instances involves a framework including four crucial elements: an objective function, a collection of limitations or restrictions, an alteration mechanism, & a searching approach. These aspects are thoroughly explored in this section. The objective of this framework is to find an imperceptible perturbation, denoted as x_{adv}, that can manipulate a predictive NLP framework. The alteration ought to have the capacity to accomplish a certain aim, such as inducing the model to make an inaccurate classification prediction. Furthermore, it is imperative that it strictly conforms to a predetermined set of constraints. The objective of the searching strategy is to discover a sequence of changes that

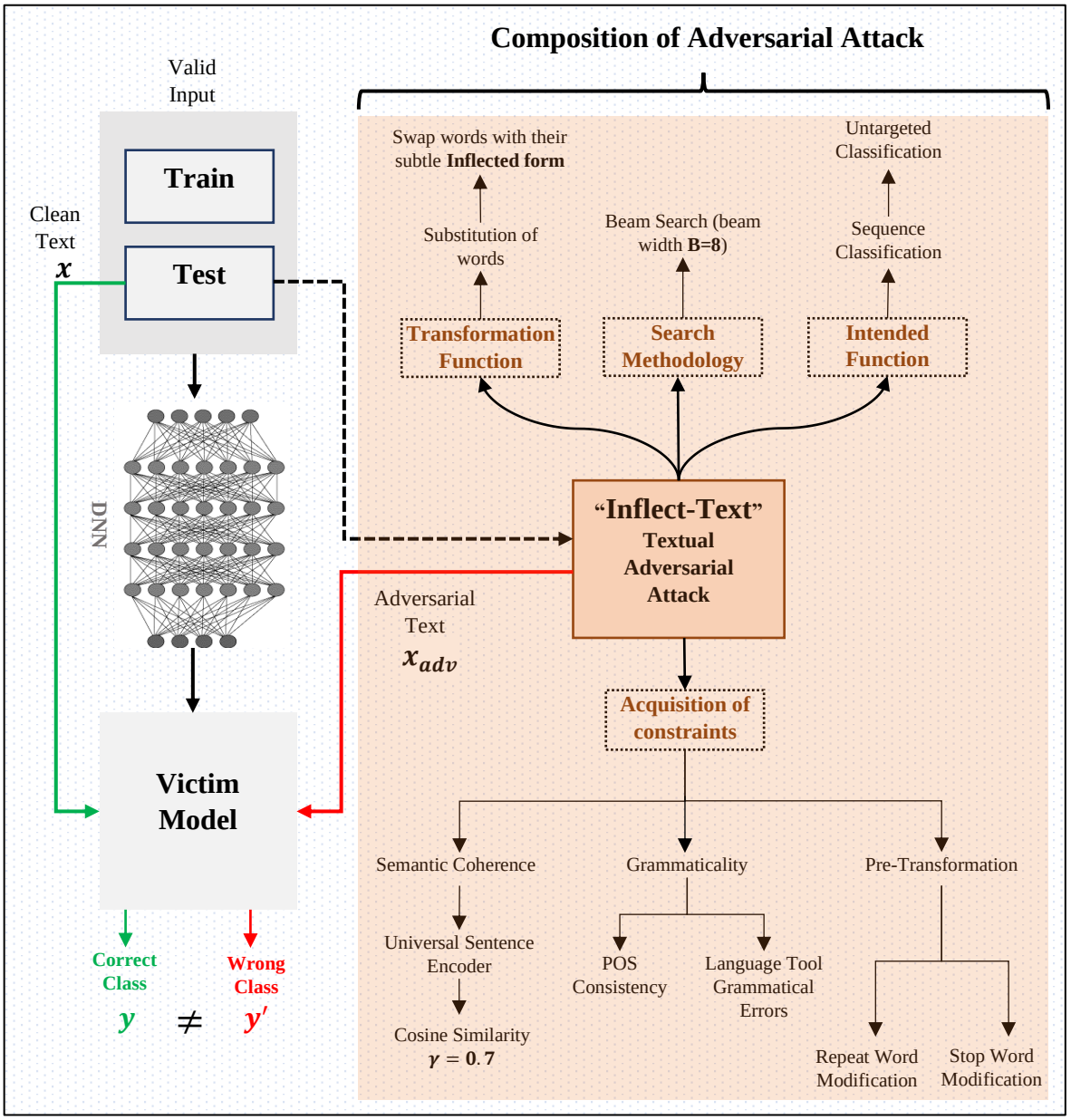


Figure 3.17 Architecture of Proposed "Inflect-Text" Adversarial Attack

result in a successful manipulation. The design of the offered adversarial assault is demonstrated in **Figure 3.17**.

The input is subjected to an alteration procedure, which leads to the creation of several possible perturbations. By replacing the $\boldsymbol{i-th}$ element, $\boldsymbol{x_i}$, of vector $\boldsymbol{X = \{x_1, x_2, x_3 \ldots\ldots x_i\}}$ with a modified version, $\boldsymbol{x_i'}$, the resulting text will be modified. This study aims to investigate the replacement of important words with perturbed keywords at the word level. In ***"Inflect-Text"*** adversarial attack, we suggest employing a transformation function. This function will convert each noun, verb, or adjective in $\boldsymbol{x}$ into its inflectional form[83], [85] resulting in the highest possible increase in $\boldsymbol{F's}$ loss. For each token in the variable $\boldsymbol{x}$, the attack function invokes the transformation module to identify the inflected version that resulted in the greatest increase in the loss function $\boldsymbol{F}$. **Table 3.25** displays adversarial instances that were generated by applying *Inflect-Text* to cutting-edge text categorization models. Present text classifiers are commonly trained with the underlying fundamental presumption that individuals possess a high level of proficiency in (frequently U.S.) Standard English[86]. **Table 3.20** demonstrates the heterogeneity in the development of inflectional morphology among $\boldsymbol{L2}$ speakers (and many $\boldsymbol{L1}$ dialect speakers)[82]. We utilise perturbations in inflectional morphology to emphasise the linguistic bias inherent in models such as BERT and Transformer models. Inflectional disturbances fundamentally maintain the overall meaning of a word as the root remains unaltered. When a word's part of speech ($\boldsymbol{POS}$) depends on the context, limiting changes to the original $\boldsymbol{POS}$ helps maintain its original meaning. The presumption that all users speak perfect standard English is unreasonable. Various types of English are shown in the McArthur Circle of World English as illustrated in **Figure 3.16**. Inflection refers to the act of appending additional components to the fundamental structure of a word to convey its grammatical meaning. The English word "inflection" has its origins in the Latin root inflectere, which translates to "to bend." A model trained solely on standard American English may be influenced by the inflectional morphological errors made by $\boldsymbol{L2}$ speakers. English words exhibit varying inflectional patterns depending on their grammatical classification and the syntactic context in which they are employed[11]. Below **Table 3.21** shows the most prevalent stipulations.

Table 3.21 Predominant rule for inflections

Part of Speech	Grammatical Category	Inflection	Examples
Adjective	Degree of Comparison (Comparative)	-er	Smart → Smarter
Adjective	Degree of Comparison (Superlative)	-est	Smart → Smartest
Noun	Number	-s, -es	Flower→ Flowers**;** Glass → Glasses
Noun, **Pronoun**	Case (Genitive)	-'s, -', *-s*	Paul → Paul's**;** Francis → Francis'**;** It → Its
Pronoun	Case (Reflexive)	-self, -selves	Him → Himself**;** Them → Themselves
Verb	Aspect (Progressive)	- ing	Run → Running
Verb	Aspect (Perfect)	en, ed	Fall→(Has) fallen**;** Finish→(Has) finished

Part of Speech	Grammatical Category	Inflection	Examples
Verb	Tense (Past)	-ed	Open → Opened
Verb	Tense (Present)	-s	Open → Opens

Linguistic morphology is the process of inflection, also known as inflexion, which modifies a word to indicate several grammatical categories, such as tense, case, voice, aspect, person, number, gender, mood, animacy, and definiteness[85]. English inflection serves to communicate several grammatical features such as noun pluralization (*e.g., cat, cats*), noun case (*e.g., girl, girl's, girls'*), third person singular present tense (*e.g., I, you, we, they buy; he buys*), past tense (e.g., *we walk, we walked*), aspect (e.g., *I have called, I am calling*), and comparatives (e.g., *big, bigger, biggest*). Inflections in English grammar encompass several elements such as the genitive's, the plural -s, the third-person singular -s, the past tense -d, -ed, or -t, the negative particle 'nt, -ing forms of verbs, the comparative -er, and the superlative -est. The arbitrary inclusion of linguistic inflections into the most essential words. Consequently, the words that have been perturbed are replaced with their original equivalents. The aforementioned picks are chosen based on their capacity to preserve the semantic proximity of the phrase using a set of constraints function, while being tokenized differently by neural text classifiers. The introduction of word inflections might potentially be an effective attack method, as second-language speakers often have difficulty recognizing these inflections without drastically altering the meaning of the statement.

The search procedure is designed to identify the optimal collection of possible perturbations derived from the transformation function[71]. Our emphasis is on black-box search algorithms because of their practicality and widespread use in the NLP attack literature[87]. The objective is to determine the significance of search algorithms in producing text adversarial instances and to evaluate the performance of different search algorithms under consistent search space or standardized search cost. In order to get best outcomes, we have compared and examined different families of search algorithms. We have chosen the search algorithms listed below for the purpose of producing adversarial cases. These methods are summarised in **Table 3.22**. Every search method has a constraint that restricts the modification of each word to a maximum of one time[71].

Table 3.22 Various search methods have been suggested for NLP attacks, each with its respective parameter settings. Here, ' $\boldsymbol{\omega}$ ' represents the number of words in the input, ' $\boldsymbol{\tau}$ ' denotes the maximum number of transformations, ' $\boldsymbol{s}$ ' indicates the population size, ' $\boldsymbol{n}$ ' represents the number of iterations, and ' $\boldsymbol{B}$ ' stands for beam width.

Searching Technique	Deterministic	Hyperparameters	Number of Queries
Genetic Algorithm	✗	$\boldsymbol{s, n}$	$\boldsymbol{O(s * n * \tau)}$
Particle Swarm optimization	✗	$\boldsymbol{s, n}$	$\boldsymbol{O(s * n * \omega * \tau)}$
Greedy Search	✓	$\boldsymbol{B}$	$\boldsymbol{O(\omega^2 * \tau)}$

Searching Technique	Deterministic	Hyperparameters	Number of Queries
Beam Search	✓	$\boldsymbol{B}$	$\boldsymbol{O(B * \omega^2 * \tau)}$
Greedy (WIR)	✓	-------	$\boldsymbol{O(\omega * \tau)}$

After conducting our evaluation, we have determined that the Beam Search is the most effective search mechanism for our combinatorial adversarial attack methodology. This mechanism allows us to identify the most promising perturbations resulting from inflection transformations. We reached this conclusion based on the highest ASR scores achieved using the beam search mechanism, as discussed in the relevant section. The beam search approach entails assigning scores to all possible disturbed texts $\boldsymbol{x'}$ that are created by replacing each word $\boldsymbol{x_i}$ in a given text $\boldsymbol{x}$. The scoring process utilises a heuristic scoring function, as depicted**.** Within this function, the process involves generating all possible modified versions (inflected texts) of a given text $\boldsymbol{x}$ by replacing each word $\boldsymbol{x_i}$, and subsequently evaluating their respective scores**.**

The formulae $F_y(\boldsymbol{x})$ denotes the estimated likelihood of class $\boldsymbol{y}$ as determined by the algorithm, whereas $\boldsymbol{y}$ provides the true result of the original sentence $\boldsymbol{x}$. The top-ranked $\boldsymbol{B}$ texts are kept, where $\boldsymbol{B}$ is known as the "beam width." The iterative method proceeds by introducing more modifications for each of the top $\boldsymbol{B = 8}$ altered texts, leading to the creation of the succeeding set of candidate texts. The most notable and influential $\boldsymbol{B}$ perturbed texts are thereafter replaced with clean ones. The computational complexity of this process is expressed as $\boldsymbol{O}$ ($\boldsymbol{b * W^2 * T}$), where $\boldsymbol{W}$ is the number of words in the input. $\boldsymbol{T}$ represents the upper limit of modification choices available for a particular input.

An effective assault must preserve the semantic meaning of the created adversarial writings, ensuring they remain identical to the source texts, while also being undetectable to humans. Hence, undetectable adversarial samples must have the following fundamental criteria. (**1**) No discernible mistakes were readily apparent to the human eye. (**2**) The adversary texts that have been carefully created should communicate with the exact same semantics as the source texts. (**3**) the model's sensitivity to the hostile text and the real input should be different, indicating the occurrence of an incorrect output. Therefore, most metrics used to measure texts are based on the symbolic representations of changes in input. These metrics, including as Euclidean distance, edit distance, Cosine similarity, & Jaccard similarity Coefficient, are used to measure the imperceptibility of content. For our attack strategy, we have used the cosine similarity function to create subtle hostile sample texts. In general, it outperforms other distance metrics due to the correlation between the vector's norm and the total frequency of word occurrences in the training corpus. The orientation of a vector and the cosine distance remain unchanged by

this; therefore, a shared word will still exhibit similarity to its inflected form. Our The main goal is to efficiently generate hostile texts; hence we just require the ability to regulate the semantic similarity to exceed a particular threshold. We have suggested a collection of linguistic limitations to ensure that $\boldsymbol{x}$ and perturbed $\boldsymbol{x'}$ exhibit similarity in terms of both interpretation and proficiency, thereby rendering $\boldsymbol{x'}$ a legitimate prospective adversarial instance. This implies that the search space should guarantee that $\boldsymbol{x}$ and $\boldsymbol{x'}$ are proximate in the semantic embedding space. Several automated methods for guaranteeing constraints have been suggested in academic literature. In this study, we utilised the Universal Sentence Encoder ($\boldsymbol{USE}$)[57] to evaluate the semantic similarity of textual occurrences by employing cosine similarity while substituting the word $\boldsymbol{x_i}$ with $\boldsymbol{x_i'}$. The cosine similarity between two $\boldsymbol{n}$-dimensional vectors, denoted as $\boldsymbol{\alpha}$ and $\boldsymbol{B}$, is mathematically described by **Eqn. (3.14).** The word embedding vectors $\boldsymbol{e_{x_i}}$ and $\boldsymbol{e_{x_i'}}$ must reach a defined minimum threshold. In an Inflect-Text attack, the threshold value is set as $\boldsymbol{\gamma} = \boldsymbol{0.7}$.

We define modifications modifying $\boldsymbol{x}$ to $\boldsymbol{x'}$ using the expression $\boldsymbol{Trans}$ ($\boldsymbol{x}$) = $\boldsymbol{x'}$. Additionally, we presume that the $\boldsymbol{k-th}$ restraints are represented as Boolean operators $\boldsymbol{Cons_k(x, x')}$ which indicate if $\boldsymbol{x'}$ meets the requirements, $\boldsymbol{Cons_k}$. Next, we may formally describe the criteria for searching the space $\boldsymbol{Search}$ using scientific notation as illustrated in **Eqn. (3.15):**

$$\boldsymbol{Searh(x) = \{Trans(x) | Cons_k\big(x, Trans(x)\big) \forall k \in [m]\}} \quad \textbf{(3.15)}$$

The objective of a searching technique is to locate an element $\boldsymbol{x'}$ in the set $\boldsymbol{Searh(x)}$ that successfully deceives the target framework. **Table 3.23** provides an overview of all the modules used to evaluate our attack algorithm. Furthermore, the $Language\ Tool$ is utilized in the acquisition of constraints to reduce grammatical errors and ensure consistent usage of parts of speech. Specifically, the chosen alternative inflection should possess the identical grammatical category as the original word. The assistance taggers offered by $SpaCy$, $NLTK$, & $flair$ are intended to preserve linguistic coherence among $\boldsymbol{x}$ & $\boldsymbol{x'}$.

Table 3.23 The Four modules in our attack benchmarking

Transformations	Search Methodology	Acquisition of constraints	Goal Function
Replacing the word with it's **inflected form** as a perturbation	**Beam Search** Technique with **beam width = 8**	**USE** similarity, **POS** consistency	**Untargeted** Classification

A particular task function that evaluates the efficacy of the assault based on the model's results. The objective is to achieve untargeted categorization, which involves creating an adversarial

instance that, when presented to the classifier, would provide a label that is intentionally incorrect[43]. This is referred to as an untargeted attack.

Table 3.24 Algorithm of the Proposed **Inflect-Text** Adversarial Attack Framework

Algorithm 1: "Inflect-Text*"* Textual Adversarial Attack
Aim: Generating Adversarial Example x' to Fool Neural Text Classifiers
Input: Input Text Sequence $X = x_1, x_2, \ldots\ldots\ldots x_n$, Model Function $F(.)$, Scoring Function $Score(.)$, Transformation Function $Trans(.)$, Cosine Similarity Function $Cons(.)$, Perturbation Constraint γ.
Output: Adversarial Example x'

1. Initialize $x' \Leftarrow x$
2. **for** each word x_i in X **do**
3. Evaluate *Score* (x_i)
4. **end for**
5. $O_{0rder} \Leftarrow Sorting\ (x_1, x_2, \ldots\ldots x_n)$ in Descending Order
6. Delete Input sentences **in** O_{0rder} **if** $F(x_i) \neq y$
7. Eliminate Stop words in O_{0rder}
8. **for** x_i **in** O_{0rder} **do**
9. $Trans\ (x_i) = x_i'$ (*replacing significant words in* x_i *with their inflected form* x')
10. **if** $(Cons(x, x') <= \gamma)$ **then**
11. **Return None**
12. **else if** $F(x_i') \neq y$ **then**
13. **Return** x'
14. **end if**
15. **end for**
16. **Return None**

An important aspect of these alterations is that words are representational units, and neural network architectures that rely on algorithms based on learning frequently employ thesaurus to depict a limited set of possible words. The standard word lexicon is considerably smaller in size in comparison to the potential permutations of letters of the same length. Regarding English words, it can be seen that the aggregate number of possible permutations is approximately 26^n, where n represents the length of the word[44]. This suggests that deliberately disturbed significant elements can easily be transformed into "unfamiliar" phrases that are not acknowledged by the lexicon. In the process of neural network simulation, any unrecognizable or unidentified word will be assigned the "unknown" embedding vector. The study's examination presents conclusive proof that utilizing word inflections is a direct approach that may greatly influence the decision-making process of text classification frameworks, resulting in inaccurate outputs. It is essential to ensure that NLP techniques are designed to be accessible and efficient for individuals with diverse linguistic backgrounds, including speakers of different English dialects like (L2) second language speakers. It is crucial because natural language user interfaces are more common[83]. We demonstrate the presence of linguistic bias in contemporary English NLP frameworks, includes BERT & Transformer by employing inflectional adversaries. We provide Inflect-Text, a method for generating adversarial examples that are both plausible and semantically identical by making deliberate

changes to inflectional morphology in an example, without the need to access the gradients of the model.

Table 3.25 The Inflect-Text adversarial approach examines every adjective, verbs, or adverb in the phrase and chooses the inflected form (highlighted in **red**) that increases the intended algorithm's loss the most. Inflect-Text restricts itself to inflections that are a component of the same universal part of speech as the original word to maximize lexical retention.

Dataset	Model	Original Prediction	Adversarial Prediction	Perturbed Texts
MR	Bi-LSTM	**Positive** Confidence=88.04%	**Negative** Confidence=45.27%	if there's a way to effectively teach kids about the dangers (danger) of drugs, i think it's in projects like the (these) (unfortunately r-rated) paid
MR	BERT	**Positive** Confidence=72.21%	**Negative** Confidence=37.00%	though everything might be literate and smart, it never took (takes) off and always seemed (seems) static
AG News	Word-CNN	**Sci/Tech** Confidence=78.49%	**World** Confidence=83.06%	Seoul allies calm on nuclear shock (shocks). south korea's (korea) key allies play down a shock admission its scientists experimented (experiment) to enrich uranium
MR	DistilBERT	**Positive** Confidence=81.42%	**Negative** Confidence=48.71%	cantet perfectly captures (captured) the hotel lobbies, two-lane highways, and roadside cafes that permeate (permeated) vincent′s days
AG News	RoBERTa	**Business** Confidence=84.45%	**Sci/Tech** Confidence=34.92%	site security gets a recount at rock the vote. grassroots movement to register younger voters leaves publishing (publication) tools accessible to outsiders

Adversarial examples were discovered for BERT, DistilBERT, and Bi-LSTM, as shown in the **Table 3.25**. Although not grammatically flawless, it is feasible for Speakers of English dialects and individuals who speak English as a second language (**L2**) create these kinds of phrases. Inflectional fluctuations maintain the broad semantic information of an expression by keeping its foundation unaltered. When the component part of speech is contextually dependent, limiting changes to its primary part of speech helps maintain its original significance.

3.4.4 Experimental Settings

The following part offers an overview of the dataset, intended architectures, assault methodologies, assessment criteria, and experimental specifications. Next, we will analyze the information & investigate other potential causes that may have influenced the observed result.

3.4.4.1 Description of the Dataset

This investigation intends to explore the influence of linguistic adversarial instances on a commonly used two standard datasets across the discipline of text categorization. The test set is applied for the development & evaluation of the adversarial cases. **Table 3.26** presents a quick synopsis of the dataset.

Rotten Tomatoes Movie Reviews (MR[8]): The dataset[59] contains 5331 negative and 5331 positive processed sentences/snippets, with a mean length of 22 words. The dataset is divided into three components for the study, with 80% allocated to training purposes and 20% for evaluation purpose. The algorithms underwent training to conduct binary categorization on critiques of movies, classifying them as either exhibiting positive or negative sentiment.

AG News[9]: The AG database[88] contains over 1 million news segments. A portion of AG's corpora of headlines consists of the names and summaries of publications from each of the 4 most significant genres (Sci/Tech, Sports, World and Business). For this study, the dataset consists of 1,900 test examples & 30,000 training samples in each class.

Table 3.26 Synopsis of the Dataset Utilized

Task	Granularity	Classification	Dataset	Labels	Train	Test	Average length
Sentiment Analysis	Movie Reviews	*Binary Classification*	MR	2	8.5K	2K	21.6
News Topic Classification	News Headlines	*Multiclass Classification*	AG news	4	120K	7.6K	44.1

3.4.4.2 Target Models

This section includes the representation of the mathematical models that were trained on two well-known NLP classification datasets, together with their corresponding parameter settings. The accuracy score is the statistic used to evaluate architectures for text categorization.

Model Description & Parameter Configurations

Algorithms that use deep learning exhibit self-sustaining capability to gain information and discern pertinent features, leading to improved effectiveness. The analysis employed various well-established neural frameworks, such as Recurrent Neural Networks, Convolutional Neural Networks, and several transformer-based frameworks. **Figure 3.18** displays a thorough depiction of the architectures implemented in this investigation.

The mentioned designs below are utilized to assess the suggested attack methodology, in addition to the usual adversarial attack methodologies. These architectures are highly effective for the task of text classification methods. Discovering vulnerabilities in these frameworks can result in significant engagement in this subject on a wide scale. During the analysis, a number of well-established neural frameworks were utilized. These frameworks included Recurrent Neural Networks, Convolutional Neural Networks, and numerous transformer-based

[8] https://huggingface.co/datasets/rotten_tomatoes
[9] https://huggingface.co/datasets/ag_news

frameworks. There is a comprehensive representation of the architectures that were utilized in this inquiry displayed in **Figure 3.18**.

Bi-LSTM: LSTM is commonly employed in sequential modelling. An LSTM model with 150 hidden states and bidirectional operation was created. While being delivered to the LSTM, the input is first transformed into 200 dimensional GLoVE embeddings. Subsequently, the text label is predicted using logistic regression. This is achieved by aggregating the LSTM outputs at each timestep, resulting in a feature vector. A dropout of 0.3 is applied throughout this process.

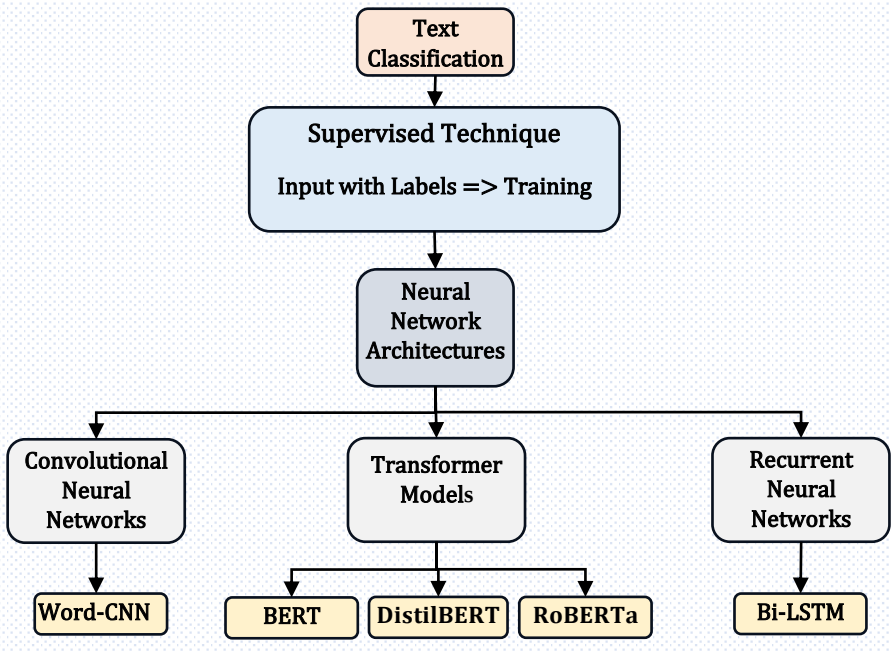


Figure 3.18 Description of Neural Text Classifiers

Word-CNN: Word-CNN offers a promising approach for the categorization of text applications. Kim′s structure[62] is selected for the examination. The Word-CNN model employs 100 filters & utilizes 3 window sizes (3, 4, & 5). The system's dropout rate is set to 0.3. It utilizes a baseline of 200-dimensional GLoVE embeddings. The classification process involves a fully connected layer followed by max-pooling over time.

Transformer Models*:* Transformers exhibit superior efficiency for training and inference in comparison to CNNs and RNNs because of their concurrent processing of input sequences, facilitated by positional encoding and self-attention mechanisms.

BERT: Google launched the Bidirectional Encoder Representations from Transformers (*BERT*) pre-trained linguistic framework[63]. It is regarded as a significant achievement in the field of natural language processing (NLP) for enhancing performance in various activities using

human language. Various models have been presented to overcome certain constraints of BERT since its release. In light of this, we will examine the efficacy of two contemporary transformer-based language models, namely DistilBERT and RoBERTa, in the context of text classification.

DistilBERT: DistilBERT[64] is a condensed iteration of BERT, characterised by reduced size, improved speed, lower cost, and decreased weight. This model is based on the Knowledge distillation methodology. It is a decompression strategy that involves training a smaller model to replicate the behaviour of a larger model. By employing this method, the dimensions of a BERT model are lowered by 40%, while retaining 97% of its linguistic skills. The model exhibits a 60% increase in speed.

RoBERTa: RoBERTa[66] is a model developed to improve the utilisation of BERT. Researchers utilised a larger dataset for their study. BERT was trained using a merged dataset consisting of BookCorpus and English Wikipedia text, amounting to a substantial 16GB of textual data. RoBERTa was trained using a blend of the aforementioned corpora, together with three supplementary corpora from various domains: CC-News, Open-Web Text, and Stories. The corpus used for training RoBERTa is 160 gigabytes in size. Furthermore, they proposed improvements to the design of the model. Throughout the training procedure, the authors substituted BERT's pre-training Next Sentence Prediction (NSP) task with dynamic masking. This approach entails modifying the concealed token at different intervals throughout the training epochs.

The bert-base-uncased, Distilbert-base-uncased, and Roberta-base models were obtained from the open-source hugging face library and underwent a training phase comprising 10 iterations. Every cycle consists of a batch size of 64, a learning rate of 2e-05, and a maximum sequence length of 128. The aim of this training is to improve the model's accuracy in classifying sequences, specifically for the MR and AG News datasets. The framework was trained using a cross-entropy loss mechanism. The framework achieved its highest level of effectiveness in this assignment, as measured by the accuracy of the test set, as displayed in **Table 3.27** after eight epochs for BERT, for DistilBERT after 7 epochs & achieved maximum testing accuracy score for RoBERTa after 4 epochs.

Table 3.27 Testing Accuracy of the Targeted Models

	Word-CNN	Bi-LSTM	BERT	DistilBERT	RoBERTa
MR	79.4%	80.7%	87.6%	88.7%	90.3%
AG News	91.0%	91.4%	94.2%	94.4%	94.7%

3.4.4.3 Conventional Adversarial Techniques

Attack techniques were used on the testing samples of the dataset to generate adversarial examples. The adversarial cases are then used to erroneously classify the genuine output of the input text when given to the pre-trained algorithms. The subsequent section offers a concise overview of the benchmarks that have been selected to showcase the effectiveness of the proposed method in misleading text classification tasks. A quick summary of the respective attack model with their perturbation granularity is presented in **Table 3.28**.

Table 3.28 Baseline Attack Methodologies and their perturbation granularities

Textattack	Methodology Elucidation	Modification Granularity
TextFooler [34]	This assault approach employs word swapping with the 50 closest embedding neighbours of the victims. Enhanced with BERT.	Word
TextBugger[**32**]	The effectiveness of this attack tactic has been enhanced for practical application. They employ character substitution, insertion of spaces, and deletion of characters. Within the framework of context-aware word vector space, they additionally replace words with their closest neighbouring characters and phrases with letters that appear comparable (such as replacing "o" with "0").	Character
PWWS [**33**]	Such assaults try to maintain linguistic precision, grammar uniformity, & contextual proximity by employing synonymous substitution. The order of importance of a word is determined by both its saliency rank and its maximal sentence-swap efficiency.	Word
PSO[**34**]	This approach combines a word-by-word replacement method that utilizes sememes combined particle swarm optimization to carry out assaults at the word's degree.	Word
Pruthi [47]	Simulation of common errors made while typing, with particular emphasis on the QWERTY key's layout. This methodology employs letter substitution, removal, and addition.	Character
Kuleshov[**36**]	Substitutes the important terms in given sequences using counterfeited embeddings of words while adhering to a specific set of necessary limitations.	Word
IGA[**37**]	The proposed method involves prioritizing the significant terms in a given sequence by utilizing a rating operation, and subsequently substituting them with counter-fitted word incorporation. In addition to conducting syntactic and logical examinations.	Word
Liang[**55**]	Using a genetic algorithm, we can replace words with their equivalents in the closest word embedding space. This process is performed while adhering to a set of restrictions to ensure the resulting sample is a genuine adversarial example.	Word
DWB[**38**]	Generates subtle textual modifications within an enclosed system with little visibility. Using the greedy substitute-1 scoring method, this approach utilizes many ways for switching symbols, including substitution, replacing, eliminating, and inserting.	Character
BAE[**56**]	This kind of assault strategy employs a BERT masking in conjunction with a linguistic model alteration. To more accurately align with the whole setting, the linguistic model changes words.	Word
A2T [53]	This assault strategy employs the substitution of words with synonyms using gradient-based methods, within the context of a white-box antagonistic situation. The method uses cosine similarity for encoding sentences to preserve semantic similarity, while also incorporating syntactic tests.	Word
Inflect-Text (Our approach)	Substituting the prospective words in a given sequence with their inflected form while ensuring that the semantic meaning remains unchanged for the human observer.	Word

3.4.4.4 Attack Effectiveness Evaluation

The efficacy of textual adversarial attacks is assessed based on three factors: (i) After Attack Accuracy (**AAA**), (ii) Attack Success Rate (**ASR**), and (iii) Average Perturbed Percentage

(**APR**). The effectiveness of each attack strategy has been empirically validated through the use of this set of assessments.

- $After\ Attack\ Accuracy$ $(\boldsymbol{AAA})$

The main goal of adversarial attacks is to weaken the efficacy of the algorithms. The categorization job is often evaluated using indicators of accomplishment, such as accuracy. The accuracy scores have been given both prior to and subsequent to the attack. The application of powerful adversarial attacks has been observed to lead to a significant decrease in accuracy scores as a consequence of the effectiveness of their tactics.

- $Attack\ Success\ Rate(\boldsymbol{ASR})$

In order to evaluate the efficiency of the attack methods, a random sample of five hundred accurately categorised instances is selected from the test set. The original texts are then processed with attack algorithms to generate adversarial samples. Afterwards, the adversarial examples are passed on to neural text classifiers to generate the final prediction. The efficacy of the attack algorithm is measured by utilising the percentage of inaccurate predictions made by these classifiers. A larger success rate implies that the assault algorithm has the capacity to create more formidable adversaries, potentially resulting in the failure of these algorithms. To assess the efficacy of an attack technique against the victim classifier, we utilise the attack success rate (ASR), which is the proportion of successful attack samples to the combined total of successful and failed samples. Successful samples are defined as those that are capable of misclassifying the accurate prediction, whereas failing samples are unable to erroneously categorize an actual result. Within a theoretical framework, an attack is deemed successful if an algorithm $\boldsymbol{f}$ successfully categories the original valid input $\boldsymbol{f(X) = Y}$, but erroneously predicts the manipulated data $\boldsymbol{f(X + \delta) = Y'}$. Thus, $(\boldsymbol{ASR})$ can be statistically expressed as shown in **Eqn. (3.16)**. In the context of untargeted attacks, the sign $\boldsymbol{Y'}$ denotes any label that is distinct from $\boldsymbol{Y}$. The symbol $\boldsymbol{\delta}$ is employed to denote alterations made to the original test.

$$\mathbf{ASR} = \frac{\boldsymbol{f(X+\delta)=Y'}}{\boldsymbol{(f(X+\delta)=Y')+(f(X+\delta)=Y)}} \qquad \textbf{(3.16)}$$

- $Average\ Perturbed\ Percentage$ $(\boldsymbol{APR})$

This statistic, which is referred to as the Average Perturbed Rate, illustrates the percentage of words that have been changed in comparison to the length of the sentence in its initial phase.

3.4.5 Evaluation Outcome & Analysis

To understand the vulnerability of classifiers that are based on text. The first phase of the research is utilizing the provided text classification dataset to train advanced deep-learning models. The trained models can be modified by utilizing the Inflect-Text adversarial attack approach. **Table 3.29** displays the decrease in accuracy scores of the test samples after using the specified perturbation technique. The initial assessment and documentation of the precision of the intended models on the original test specimens is known as the Before-Attack Accuracy (BAA). Subsequently, the efficacy of the target models is evaluated by subjecting them to adversarial samples generated using the provided attack technique. After-attack accuracy (AAA) is the measurement of accuracy achieved after the intended attack has been carried out. Furthermore, the study provides information regarding the ratio of modified words about the original phrase length, referred to as the Average Perturbed Rate (APR).

Table 3.29 comparison of the accuracy of each model before and after the proposed adversarial attack algorithm is conducted. (***BAA**=Before Attack Accuracy, ***AAA** =After Attack Accuracy, ***APR=** Average Perturbed Rate)

	Word-CNN		**Bi-LSTM**		**BERT**		**DistilBERT**		**RoBERTa**	
	MR	**AG News**	**MR**	**AG News**	**MR**	**AG News**	**MR**	**AG News**	**MR**	**AG News**
BAA	79.4%	91.0%	80.7%	91.4%	87.6%	94.2%	88.7%	94.4%	90.3%	94.7%
AAA	05.0%	02.9%	03.7%	02.3%	09.4%	10.1%	08.2%	06.9%	11.4%	09.8%
APR	15.6%	15.9%	13.7%	14.1%	16.8%	12.5%	18.2%	14.4%	13.6%	14.7%

To evaluate the effectiveness of the proposed approach in contrast to conventional assault methodologies on neural text classifiers, the ASR (Attack Success Rate) measure is utilized. To do this, a collection of 500 test instances, which were precisely categorized, were extracted from the test set. Subsequently, the production of adversarial instances is accomplished by the application of various assault techniques. The present experiment involved subjecting a set of adversarial examples to five cutting-edge neural text classifiers. The present investigation assessed and contrasted the effectiveness of several adversarial methods against the suggested assault, using the ASR metric as a measure of performance. This figure can serve as an indicator of the efficacy of the attack strategy. A higher ASR value indicates that a specific type of assault is more successful in deceiving the model. **Table 3.30** & **Table 3.31** presents a succinct summary of the primary outcomes achieved by the implementation of the Inflect-Text assault technique on the text classification datasets. Furthermore, it involves an unbiased assessment of the efficacy of this offensive approach concerning previous ways of attack.

Table 3.30 Results of the Adversarial Attacks on **MR** Dataset
(∗ **ASR** = Attack Success Rate & ∗ **APR** = Average Perturbed rate)

Attacks	Word-CNN		Bi-LSTM		BERT		DistilBERT		RoBERTa	
	ASR %	APR %	ASR %	APR %	ASR %	APR %	ASR %	APR %	ASR %	APR %
A2T [53]	15.8	13.4	16.6	18.2	12.1	18.4	14.2	16.9	15.3	12.5
BAE [73]	63.7	18.2	68.5	16.9	55.4	17.5	62.7	14.4	50.6	13.8
Checklist [51]	17.2	13.7	18.2	18.6	11.2	19.5	15.4	17.6	12.3	16.8
DWB [50]	67.4	12.1	68.7	19.2	58.8	09.7	66.9	14.9	55.9	10.9
HotFlip [74]	81.4	14.5	85.5	10.7	78.8	13.7	80.6	12.7	72.8	11.2
IGA [49]	72.2	14.8	77.4	13.1	69.4	14.7	70.0	13.0	67.9	12.4
InputReduction [75]	38.7	11.9	41.1	10.1	32.8	12.9	34.7	11.7	37.2	13.6
Kuleshov et al. [48]	79.8	19.0	81.9	17.6	72.1	13.7	79.6	12.4	69.7	12.9
Pruthi et al. [47]	49.7	08.9	54.2	08.2	44.3	09.5	49.9	10.9	41.5	08.9
PSO [**34**]	78.6	18.6	80.8	13.3	72.5	13.2	78.7	16.2	69.6	14.6
PWWS [**33**]	57.6	18.4	60.8	16.5	49.6	12.8	58.7	15.8	48.9	14.6
Textbugger [44]	93.5	15.1	95.8	11.8	89.6	14.1	90.6	12.7	90.2	13.7
TextFooler [34]	76.8	15.8	78.6	12.7	68.7	13.9	72.0	11.9	67.4	12.7
Inflect-Text	**93.7**	12.2	**95.4**	11.9	**91.6**	10.9	**93.3**	10.8	**91.5**	11.9

Table 3.31 Results of Adversarial Attacks on **AG News** Dataset

Attacks	Word-CNN		Bi-LSTM		BERT		DistilBERT		RoBERTa	
	ASR %	APR %	ASR %	APR %	ASR %	APR %	ASR %	APR %	ASR %	APR %
A2T [53]	46.2	13.8	54.9	15.5	30.3	16.7	34.8	13.8	30.4	14.2
BAE [73]	63.1	16.5	71.1	19.8	55.6	15.2	57.2	12.9	52.2	15.1
Checklist [51]	19.2	14.3	14.2	18.6	17.4	19.1	19.1	17.5	17.3	16.9
DWB [50]	97.3	12.9	98.2	18.5	92.9	11.2	97.7	12.8	89.6	12.7
HotFlip [74]	78.3	15.0	78.4	10.7	57.6	14.4	62.9	13.8	52.0	11.8
IGA [49]	97.7	16.3	96.7	13.1	91.0	14.1	94.3	13.9	88.9	12.7
InputReduction [75]	14.9	12.7	13.8	11.6	12.8	12.6	14.2	12.8	13.5	15.2
Kuleshov et al. [48]	97.3	18.0	97.4	16.6	91.2	13.5	95.5	12.9	84.9	16.6
Pruthi et al. [47]	35.8	09.6	54.4	08.7	53.6	09.6	48.7	08.6	53.8	10.5
PSO [**34**]	97.3	17.9	97.6	15.9	94.1	14.8	93.2	14.4	92.1	14.9
PWWS [**33**]	95.2	19.1	96.8	16.8	89.9	12.6	92.1	15.2	90.4	14.8
Textbugger [44]	86.1	12.6	82.1	13.2	62.7	14.5	80.8	14.9	62.8	13.9
TextFooler [34]	97.6	16.8	97.9	14.8	97.1	15.1	96.7	13.7	91.3	12.8
Inflect-Text	**97.7**	10.9	**98.4**	11.7	**97.9**	11.2	**98.1**	12.2	**94.4**	10.8

In addition to this, the study intends to investigate the susceptibility of different neural text classification algorithms to different kinds of adversarial intervention. When multiple classifiers are subjected to adversarial disruptions, the purpose of this investigation is to evaluate the relative vulnerability and resilience of each of the models. As can be seen in **Figure 3.19**, the ASR of each attack is evaluated on each targeted paradigm to determine the relative sensitivity of each of the simulations. **Eqn. (3.17)** is taken into consideration in order to get the average attack success rate for each classifier.

$$\mathbf{AVG_{ASR}} = \frac{\sum_{j=1}^{N} \frac{\mathbf{Success_j}}{\mathbf{Success_j + Fail_j}}}{\mathbf{N_{attacks}}} \qquad \textbf{(3.17)}$$

The Attack Success Rate will be represented as $\mathbf{AVG_{ASR}}$, where $\mathbf{Success_j}$ denotes the number of successful attacks, $\mathbf{Fail_j}$ denotes the number of unsuccessful assaults, and $\mathbf{N_{attacks}}$ is the number of attack recipes. The skipped sentences refer to the claims that the algorithm

initially made incorrect predictions throughout its training. They were omitted from the calculation.

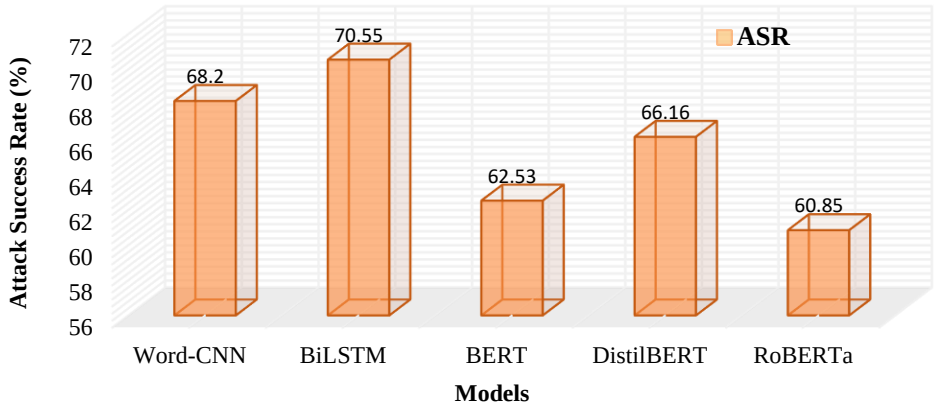


Figure 3.19 Mean ASR Scores of all the Classifiers

The **Figure 3.19** clearly demonstrates that non-attention-based models, namely word CNN and Bi-LSTM, display a notable susceptibility in comparison to other models. In addition, the Bi-LSTM model has the greatest vulnerability compared to the other classifiers, with an average ASR score of 70.55%. The $BERT_{base}$ model employs 12 layers of transformer blocks, each with a hidden size of 768. It also features 12 self-attention heads and around 110 million trainable parameters. This model has demonstrated an average ASR score of 62.53%. DistilBERT model exhibits greater vulnerability compared to BERT and RoBERTa transformer models, with an ASR score of 66.16%. On the other hand, RoBERTa model demonstrates the least fragility among each model, with an ASR score of 60.85%. The investigation's findings unequivocally indicate that light models are more susceptible to adversarial perturbations, while heavier models with a higher number of parameters are less vulnerable to hostile manipulations. BERT utilizes a process of randomly obscuring and predicting tokens. The initial BERT implementation applied masking once during the preprocessing of data, leading to the creation of a solitary and unchanging mask. In order to prevent the repetition of utilizing the same mask for each training instance in every epoch, the training data was replicated 10 times. This ensures that each sequence is masked in 10 distinct ways across the 40 training epochs. Therefore, each training sequence was observed with an identical mask on four separate occasions during the training process. The RoBERTa model, thereby, with a dataset that is ten times larger for training, also incorporates hostile samples, which enhances its resilience against adversarial manipulations. Thus, demonstrating the lowest susceptibility compared to all other classifiers. The findings could be significant for individuals who regularly employ established and advanced algorithms in their endeavours for

text classification tasks. The reader will have the capacity to identify the best suitable model that aligns with their specific challenge. Furthermore, this phenomenon serves as a stimulus for academics to create models that demonstrate antagonistic strong generalizations instead of conventional generalizations.

To determine the relative effectiveness of several attack methods in fooling a framework with an average perturbation rate. The mean rate of success and rate of modification for every kind of attack throughout the different models have been obtained and are displayed in **Figure 3.20**.

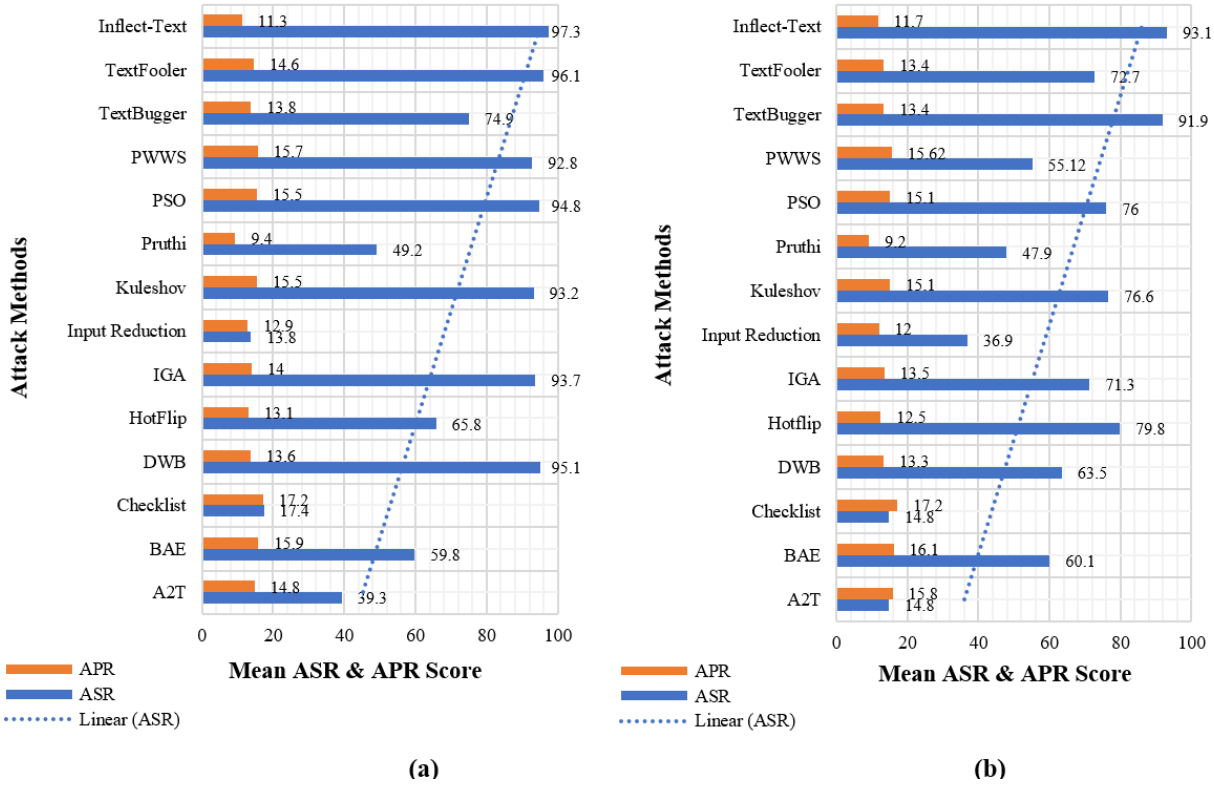


Figure 3.20 Average ASR & APR score of each attack type on **(a)** MR & **(b)** AG News dataset

Based on the findings in **Figure 3.20**, it is evident that the aforementioned Inflect-Text adversarial approach outperforms the standard baselines for comparison. This attack architecture achieved the highest ASR result while using a lower perturbation ratio. Initially, we randomly select words for alteration in the experiment. However, we obtained lower ASR scores compared to conventional ways. Consequently, we employ the beam search algorithm to identify and perturb the significant words in the input text. Upon implementing this approach, we achieved the utmost level of success. Furthermore, it has been noted that the TextBugger attack methodology, which operates by substituting characters, inserting spaces, and deleting characters within the context-aware word vector space framework, also replaces words with their nearest neighbouring characters and phrases with similar appearing letters.

The TextBugger method achieved the second-highest ASR score by operating at the level of character-level alteration granularity. In contrast, the attack technique A2T, which utilizes gradient-based synonym word substitution within the white-box adversarial environment, has been determined to be the least effective among all classifiers, as evidenced by a thorough assessment of offensive strategies. However, most attack approaches, including our suggested methodology, operate at the level of word-level granularity. Based on the data, it has been noted that word-level adversarial attacks are more prevalent than character-level attacks.

3.4.6 Additional examination and evaluation

An additional study is conducted to evaluate the effectiveness of the Inflect-Text attack method under various conditions, its impact on ASR values when randomly altering the words, and its runtime considerations in creating an adversarial example. The analysis also involves assessing the extent to which conflicting scenarios generated by the suggested method can be transferred to other adversarial settings. Furthermore, the assessment of the usefulness of the antagonistic words produced by Inflect-content indicates a notable similarity to the authentic content.

Random Word Perturbation: The information provided in **Table 3.32** shows that randomly choosing terms to alter, also known as 'Randomly Perturbing', has barely any effect on the final result. Arbitrarily changing words is unlikely to fool machine learning algorithms, hence it is crucial to carefully choose which words to change in order to carry out an effective assault.

Table 3.32 Comparing ASR values using chosen at random words against words chosen based on computed significance values for modification.

Model	Dataset	Accuracy	Randomly Perturbing		Inflect-Text (Scoring function for finding significant words)	
			ASR	APR	ASR	APR
Word-CNN	MR	79.4%	37.4%	12%	93.7%	12.2%
	AG News	91.0%	28.2%	12%	97.7%	10.9%
Bi-LSTM	MR	80.7%	39.3%	12%	95.4%	11.9%
	AG News	91.4%	34.8%	12%	98.4%	11.7%
BERT	MR	87.6%	26.9%	12%	91.6%	10.9%
	AG News	94.2%	46.3%	12%	97.9%	11.2%
DistilBERT	MR	88.7%	37.1%	12%	93.3%	10.8%
	AG News	94.4%	29.0%	12%	98.1%	12.2%
RoBERTa	MR	90.3%	31.5%	12%	91.5%	11.9%
	AG News	94.7%	24.7%	12%	94.4%	10.8%

Adversarial Transferability: The investigation aims to explore adversarial transferability in the text by evaluating how well adversarial examples created by a model may fool multiple models[76]. The present investigation collected Hundred adversarial cases created using the Inflect-Text proposed method. The cases were deliberately selected because the targeted classifier misclassified them. The success rates of these samples

were later assessed on different victim classifiers. **Table 3.33** shows that the Bi-LSTM model has an average degree of adaptability. Conversely, the transferability of transformer models is relatively reduced. The analysis revealed that the BERT model exhibits greater transferability in comparison to other transformer models.

Table 3.33 Adversarial Examples' Transferability on MR dataset. ASR for adversaries created for model a, evaluated on model b, is denoted by the intersection of row i and column j.

	Word-CNN	Bi-LSTM	BERT	DistilBERT	RoBERTa
Word-CNN	-----	37.7%	47.6%	39.2%	34.4%
Bi-LSTM	34.8%	-----	43.8%	41.9%	43.1%
BERT	48.3%	42.7%	-----	48.7%	49.0%
DistilBERT	39.6%	36.6%	53.9%	-----	36.7%
RoBERTa	37.8%	29.5%	50.7%	45.3%	-----

Runtime Analysis: An investigation was carried out to assess the computational time consequences of the proposed architecture. An adversary's primary goal is to alter a classifier by carrying out the desired assault. **Figure 3.21** illustrates the expected results of the average time needed to generate a single adversarial example using Inflect-Text attack structure for each classifier. Upon analysing the investigation's results, it is clear from the **Figure 3.21** that BERT takes the maximum time to create an adversarial instance. On the other hand, lightweight models such as DistilBERT have lower processing time in generating an adversarial instance. Additionally, the non-transformer-based Word-CNN model shows the quickest time period required to generate an adversarial example. Experimental proof indicates that creating malicious instances for the AG news dataset is quite time-consuming. The mean length of news headlines in the AG News dataset is 43 words, while in the MR dataset, the mean input review length is 20 words. There is a direct relationship between the time needed to create one hostile text and the mean input sequence length. As the input size grows, the time needed to create a single adversarial text also increases slightly because more effort is needed to find important phrases for changes.

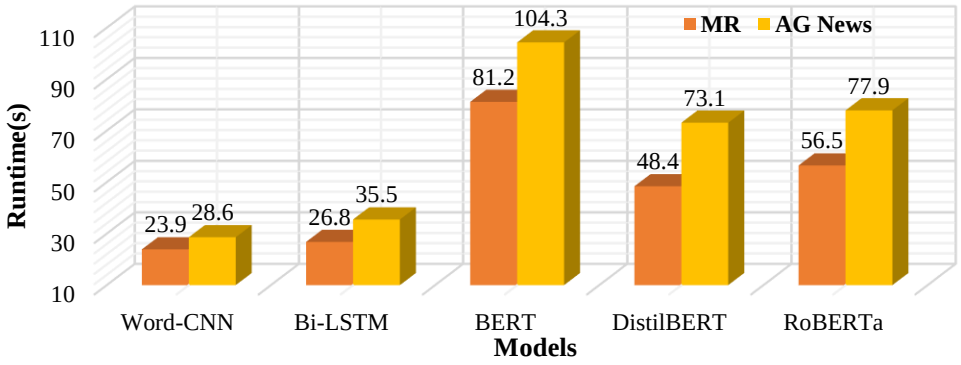
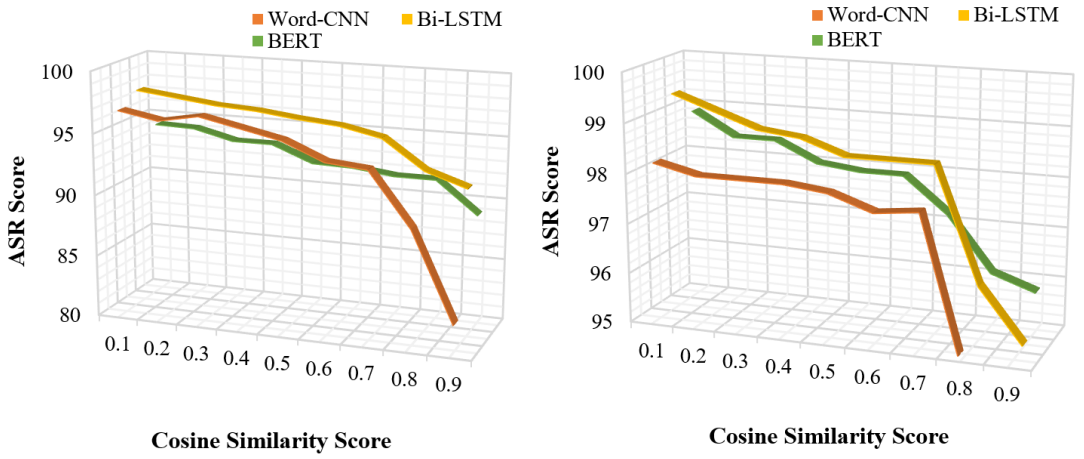


Figure 3.21 Runtime considerations of each model in developing an adversarial sequence

Utility Analysis: The assault strategy in the current investigation produces aggressive texts that closely resemble the original texts. The results in **Table 3.25** suggest that the retention of sustainability in conflicting circumstances generated by Inflect-Text is greater in terms of credibility and usability. Char-level assaults encompass various linguistic mistakes including typos, transpositions, and arbitrary character substitutions. Both humans and spell-checking systems can readily detect and distinguish attacks that target particular characters. The opponent can make undetectable alterations by using inflected word forms in the suggested solution. These changes demonstrate enhanced levels of invisibility and readability.

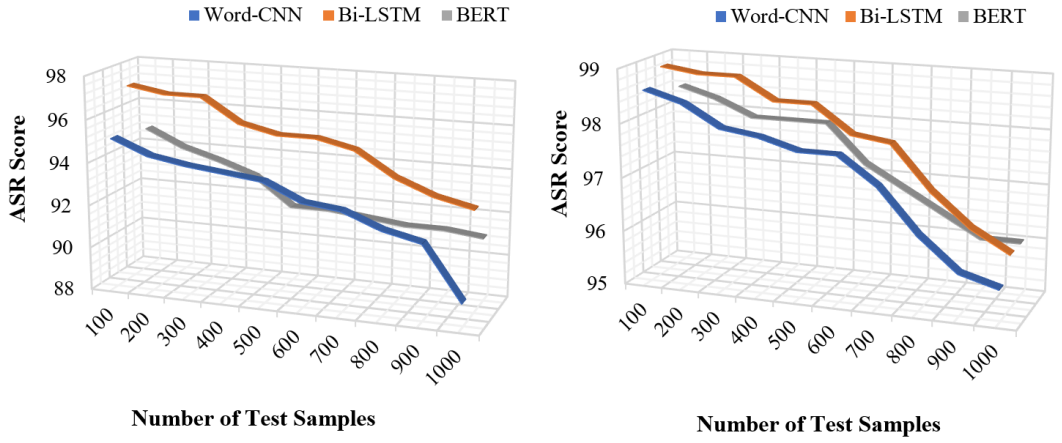
Degree of Sensitivity: The cosine similarity metric is used in the assault category to reduce the word perturbation rate. As the cosine similarity score ($\boldsymbol{\delta}$) decreases, the average word perturbation rate increases, suggesting that the framework is greater vulnerable to fluctuations. Nevertheless, it can ultimately negate the limitation of human invisibility. Sensitivities is the term typically used to describe an algorithm's reactivity. To reduce the number of interruptions. The $\boldsymbol{\delta}$ in Inflect-Text is set at $\mathbf{0.7}$ to ensure that the disruption rate remains responsive to changes while also keeping the linguistic consistency of the sequence. To evaluate the sensitiveness of a framework, one should analyse the changes in the ASR ratings in response to modifications in the parameter $\boldsymbol{\delta}$, which is specified between the interval [0.1,1) as depicted in the **Figure 3.22**. We assessed the sensitivity of Word-CNN, Bi-LSTM, and BERT models in the experiment. We have examined the variations in the ASR scores by adjusting the



perturbation threshold parameter. Upon evaluation, it was noted that the ASR score rises as the threshold values lower, leading to a higher word perturbation rate.

Figure 3.22 ASR scores fluctuate with variations in cosine similarity scores for classifiers trained on the **(a)** MR dataset and **(b)** AG News dataset.

Extensibility of test sample size: A set of testing samples were created in a particular spectrum of parameters to evaluate the effectiveness of the proposed framework after being exposed to Inflect-Text-induced changes. The variation of ASR results in response to variations in the score of the test data was assessed by examining the success rates of each classifier. As the range of the test examples expanded during the method, the average time needed to create adversarial instances also grew. As depicted in the figure. The **Figure 3.23** indicates a negative



correlation between the population sample size and the ASR scores. This association exhibits a noticeable decrease as the sample size grows. The ASR scores do not show significant variance with changes in the population size.

Figure 3.23 The ASR scores vary with changes in test sample numbers for classifiers trained on the **(a)** MR dataset and **(b)** AG News dataset.

Explainability & **Interpretability**: We use the LIME technique, developed by Ribeiro et al.[78], to offer particular characterizations concerning our techniques. The LIME methodology employs a linear approach to compute the local decision boundary for each case by adapting it to the corresponding data. The instance was altered to obtain the required information. Using the area over perturbation curve (AOPC) as a quantitative metric, the accuracy of the regional interpretations derived by LIME is evaluated [90], [91]. **Eqn. (3.18)** contains the mathematical equation for this measure.

$$\mathbf{AOPC}=\frac{1}{M+1}\sum_{m=1}^{M}\frac{1}{N}\sum_{1=1}^{N}(X_{(0)}^{(j)})-F(X_{(m)}^{(j)}) \qquad \mathbf{(3.18)}$$

The notation $\boldsymbol{X}_{(\mathbf{0})}^{(\boldsymbol{j})}$ in this research represents a particular instance where $\boldsymbol{X}^{(\boldsymbol{j})}$ is labelled as $(\mathbf{0})$ without any omissions. $\boldsymbol{X}_{(\boldsymbol{m})}^{(\boldsymbol{j})}$ denotes a scenario where the most significant words are

eliminated and represented as $(\boldsymbol{m})$. The function $\boldsymbol{F}(\boldsymbol{X})$ represents the model's confidence level on the projected target label $\boldsymbol{Y}^{(j)}$. The model's level of confidence with respect to the anticipated target label $\mathbf{Y}^{(j)}$ is represented by the function $\mathbf{F}(\mathbf{X})$. The average shift in the model's confidence towards the target label after the top $\boldsymbol{m}$ most significant words—as determined by LIME—are eliminated is known as the AOPC concept. This idea comes from an intuitive perception. From the test set, 100 randomly selected occurrences were used for assessment. When generating explanations with LIME, a set of 100 modified samples is generated for each case, employing the suggested attack method to assess the explanations. An $\boldsymbol{m}$ value of 10 was chosen for the AOPC metric. **Table 3.34** provides conclusive proof that the Bi-LSTM model achieves the greatest AOPC score. The analysis reveals that the AOPC scores of RoBERTa models are significantly inferior to those of other models, suggesting that RoBERTa may possess a diminished level of explainability in comparison to all models.

Table 3.34 The AOPC ratings were computed for the LIME interpretations of each classifier[78]. Higher AOPC score indicates more interpretability in a model.

Models	Word-CNN	Bi-LSTM	BERT	DistilBERT	**RoBERTa**
AOPC	0.48	0.59	0.37	0.42	0.25

3.5 Significant Outcomes of this Chapter

The significant outcomes of this chapter are as follows:

- Proposed three novel textual adversarial attack frameworks which are capable of bypassing textual classification algorithms.
- An effective textual adversarial approach **HOMOCHAR** is developed under black-box environment that formulates stronger adversarial examples as a combinatorial search task with the goal (untargeted attack) for deceiving neural text classifier by perturbing at character-level by replacing normal characters with homoglyph characters which adheres to specific linguistic constraints.
- **Non-Alpha-Num** adversarial assaults create adversarial examples by altering regular phrases with punctuation or non-alphanumeric characters. The outcomes demonstrate that this attack algorithm surpasses prior cutting-edge attack methods.
- **Inflect-Text** adversarial attack which uses inflectional morphology of words for perturbation i.e. replacing the normal word with its inflected form which retains its semantic meaning of the input sequence but deceive text classifier. The experimental outcomes clearly demonstrates that the attack form overcome previous cutting-edge attack algorithms.

The following research works form the basis of this chapter:

- **A. Bajaj** and D. K. Vishwakarma, “HOMOCHAR: A novel adversarial attack framework for exposing the vulnerability of text based neural sentiment classifiers,” ***Engineering Applications of Artificial Intelligence***, vol. 126, Nov. 2023, doi: 10.1016/j.engappai.2023.106815.
- **A. Bajaj** and D. K. Vishwakarma, “Non-Alpha-Num: a novel architecture for generating adversarial examples for bypassing NLP-based clickbait detection mechanisms,” ***International Journal of Information Security***, 2024, doi: 10.1007/s10207-024-00861-9.
- **A. Bajaj** and D. K. Vishwakarma, " Bypassing Neural Text Classification Mechanism by Perturbing Inflectional Morphology of Words.” Under Review in ***Neural Networks,*** June 2024.

Chapter 4: Adversarial Robustness Comparison of Neural Text Classifiers

4.1 Scope of this Chapter

Prior studies have demonstrated that Deep Neural Networks (DNNs) are susceptible to purposefully altered samples, referred to as adversarial examples. These samples are created using subtle perturbations that are not easily noticed, yet they are able to deceive the deep neural networks into producing inaccurate predictions. Diverse attack strategies are suggested to target a broad spectrum of NLP applications. This article provides a comprehensive analysis of these works. We have compiled all relevant scholarly publications starting from their initial publication in 2017. Subsequently, we proceed to choose, condense, deliberate, and scrutinise these works in a thorough manner. In order to provide a comprehensive analysis, we address the main issue of determining which neural text classifier is more susceptible and which is more resistant to adversarial manipulations. After careful analysis, we have determined which attack mechanism and perturbation granularity pose a more significant threat to machine/deep learning algorithms.

4.2 Evading Text Based Emotion Detection Mechanism via Adversarial Attacks

4.2.1 Abstract

Textual Emotion Analysis (TEA) seeks to extract and assess the emotional states of users from the text. Various Deep Learning (DL) algorithms have emerged rapidly and demonstrated success in numerous disciplines, including audio, image, and natural language processing. The trend has shifted a growing number of researchers from standard machine learning to DL for scientific study. Using DL approaches, we offer an overview of TEA in this paper. After introducing the background for emotion analysis, including the definition of emotion, emotion classification methods, and application domains of emotion analysis, we demonstrated that, despite the immense success of deep learning models in NLP-related tasks, they are susceptible to adversarial attacks, which can lead to incorrect emotion classification. An adversarial text is constructed by altering a few words or characters so as to keep the overall semantic similarity of emotion for a human reader while tricking the machine into making erroneous predictions.

This study demonstrates the vulnerability of emotion categorization by generating adversarial text using a variety of cutting-edge attack techniques. Comprehensive experiments are performed to assess the effectiveness of the attack methods against several widely-used models, such as Word-CNN, Bi-LSTM, and four powerful transformer models, namely BERT, DistilBERT, ALBERT, and RoBERTa. These models were trained on an emotion dataset utilized for the purpose of emotion classification. We evaluated and analyzed the behavior of different models under a variety of attack conditions to determine which is the most and least vulnerable. Also, we determine which perturbation technique affects transformer models the most. Using Attack Success Rates (ASR) as our evaluation metric, we have assessed the potential outcomes. The findings reveal that methodologies for classifying emotion prediction can be circumvented, which has implications for existing policy measures.

4.2.2 Textual Emotion Analysis

Emotions are a crucial aspect of human nature; hence emotion analysis has been extensively explored in psychology, neurology, & behaviour science. Emotional analysis, often known as opinion mining, is the process of recognizing and indexing content depending on the tone it expresses in the commercial world. This content may include tweets, remarks, criticisms, and even impassioned rants containing mixed or neutral views. Monitoring client feedback, identifying specific consumers to improve service, and observing how a change in a product or service affects how customers feel are examples of common uses for emotion analysis. Monitoring client emotions over time is helpful as well. This platform has fundamentally changed how firms' function, from opinion polls to inventive marketing tactics. For instance, a lot of internet recommendation algorithms analyse user reviews and comments based on their emotion. Public opinion analysis, e-commerce, personalized suggestion, healthcare (e.g., depression screening), information prediction (e.g., financial prediction, presidential election prediction) and online education all rely heavily on this type of analysis [92]. Textual Emotion Analysis (TEA), The categorization of syntactic or semantic elements within a corpus into a particular range of emotional categories, as posited by a psychological framework, is a swiftly developing subdomain of NLP. Automated TEA mechanisms employ machine learning techniques to build computational platforms that automate the emotion extraction process.

Motivated by Parrot's model, as shown in **Figure 4.1**, we have used an emotion dataset which considers six common emotions, consisting of joy, surprise, sadness, love, anger and fear [93].

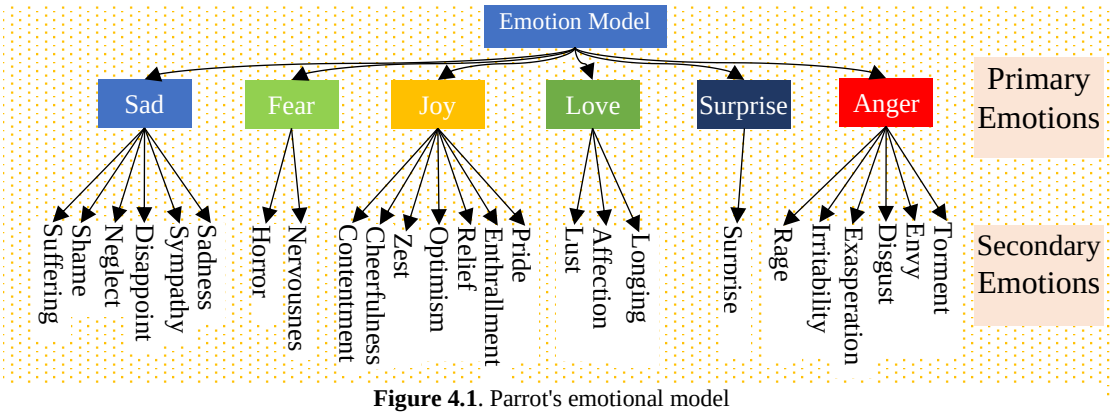


Figure 4.1. Parrot's emotional model

Classical ML algorithms have achieved huge success in emotion classification. With the advent of deep learning techniques, the sophistication and intelligence of models have increased, as exemplified by transformer models. The models have garnered significant interest due to their exceptional confidence scores in the task of text classification. In this study, we employed the most effective classification models based on deep learning. This article demonstrates the significance of validating deep learning-based emotion classifiers before using them in decision support systems by the use of practical assaults.

4.2.3 Security concerns in Textual Emotion Analysis

The study raises significant security concerns for organizations that deploy emotion detection mechanisms in multiple applications for digital marketing and for their business analytics. Malicious people can use these technologies' flaws to find vulnerabilities. A deceitful operator might change data just slightly to impact the emotion classifier's conclusion, as shown in **Figure 4.2**. Thus, this offers the decision-maker a distorted impression of reality, which might lead to wrong judgements that adversely affect the organization and raise serious security concerns.

***Impact of adversarial attacks on API platforms**:* Many businesses have developed Machine Learning-as-a-Service (MLaaS) for Deep Learning Textual Understanding applications like text categorization. MLaaS solutions install models on cloud servers and let customers access them through API [50]. An attacker is unaware of the model architecture, parameters, or training data and can only query the target model for prediction or confidence scores. An adversary can still work in the black-box settings and can readily alter the original text to

perturbed text unnoticed by human observers but can deceive the API into producing erroneous predictions, which results in major policy ramifications[94].

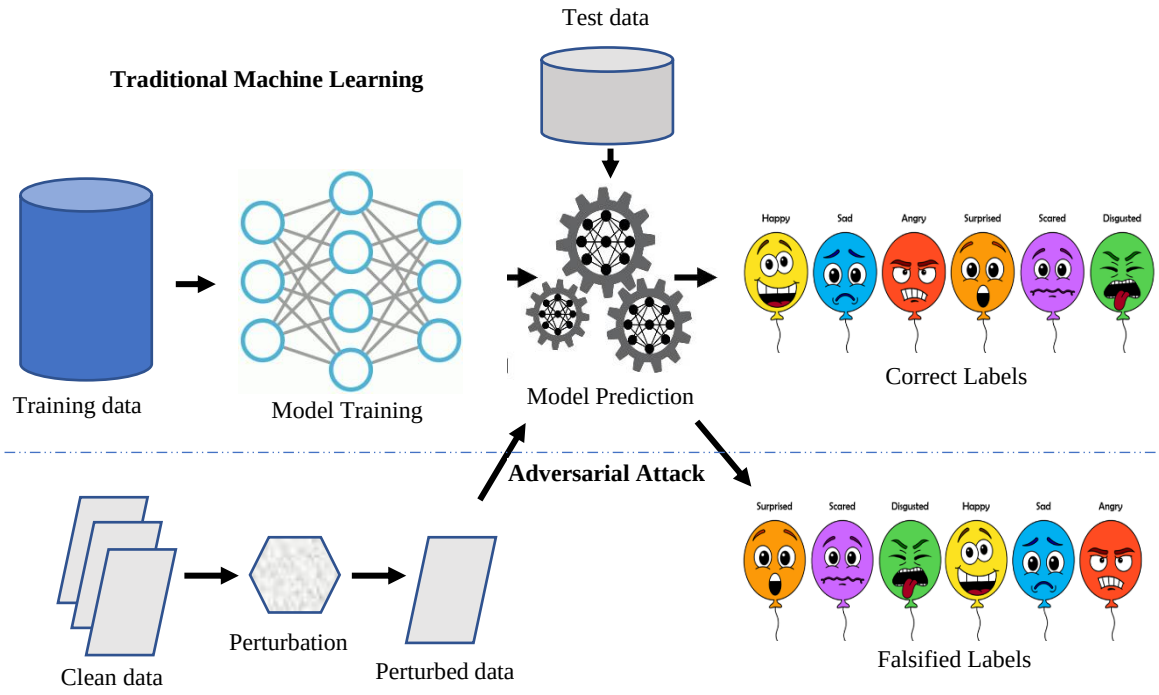


Figure 4.2 Adversarial Attack framework on emotion detection model

Resilience against attacks: Standard generalizations are produced by training the model with clean input data. But because of this, adversarial perturbations can affect the models. When a model still makes accurate predictions after being fed adversarial samples, it is said to have robust generalizations, also known as adversarially robust generalizations. Two prevalent methods for attaining robust generalizations are adversarial training[95] and knowledge distillation[96]. The practise of incorporating adversarial examples into the training process of a model is widely recognized as adversarial training. The process of knowledge distillation involves the manipulation of a neural network and subsequent training of a new model.

Although, the objective of this research is to concentrate exclusively on identifying the most effective perturbation technique and to determine which of the selected emotion classifiers is most susceptible to adversarial perturbations.

4.2.4 Procedure for evaluating emotion classifiers under adversarial settings

The initial step involved training and evaluating advanced deep-learning models using an emotion dataset. Next, a small portion of the correctly classified samples are selected at random from the test set. Afterwards, the attack is conducted on these randomly selected samples,

which are then known to be adversarial samples, these adversarial samples are then subjected to a trained model, and the model prediction changes and becomes incorrect. If the samples for the correct predictions change to incorrect, the adversarial sample is successful. The ones which do not alter the prediction are failed attacks. Hence, each attack efficacy is calculated using an Attack success rate (ASR)[97] score, i.e., $\frac{(\boldsymbol{successful\ samples})}{\boldsymbol{successful+failed\ samples}}$(*discussed in Section 4.4*). **Figure 4.3** depicts the framework for conducting an adversarial attack on emotion classification models. The mean ASR score is then evaluated using all attack techniques on different emotion classifiers to know which model is more vulnerable to adversarial perturbations. Also, which perturbation technique attack method is more potent in fooling the deep learning models. This study is considered an innovative addition to the existing literature as it provides a comprehensive evaluation of significant models that were subjected to highly efficient attack techniques. Each of these components contributes to the distinctiveness of our work. As per the study's assertion, this research represents the initial attempt to contrast emotion classifiers in order to estimate their vulnerability to adversarial circumstances. **Figure 4.10** is an illustration of an adversarial instance that has been generated by perturbing at multiple levels in such a manner that it maintains semantic similarity for humans, but it deceives the model by providing an erroneous outcome.

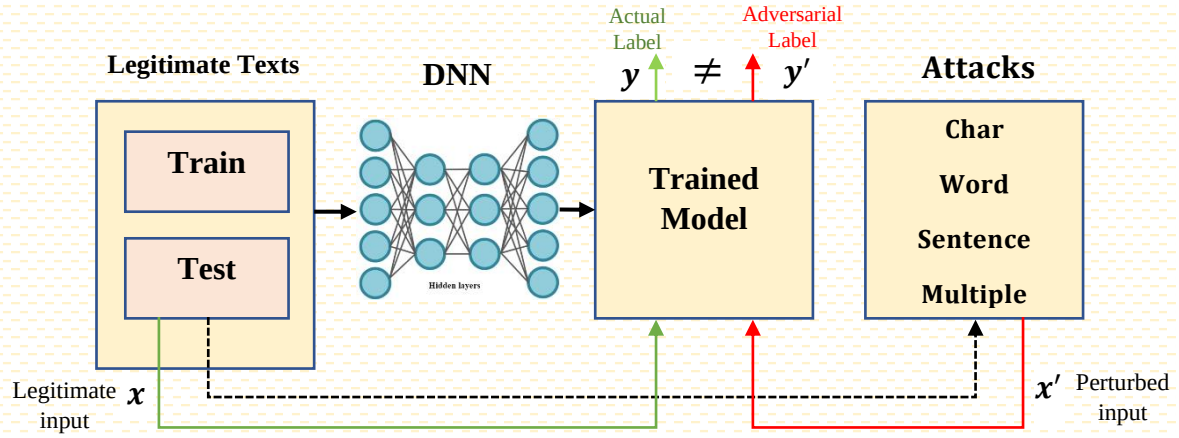


Figure 4.3. Framework for conducting adversarial attack on emotion classifier.

4.2.5 Experimental Settings

The dataset, targeted models, attack techniques, evaluation metric, and implementation details were all introduced in this part. After that, in the following section, we'll assess the findings and go over several likely causes of the observed performance.

4.2.5.1 Dataset Description

This study examines the impact of adversarial text samples on a widely utilized benchmark dataset for the purpose of emotion classification. The test set is utilized for the generation and evaluation of the adversarial examples. The presented **Table 4.1** provides a concise overview of the dataset.

emotion [98]: The present study utilizes the emotion dataset for the purpose of conducting emotion recognition tasks. The emotion dataset comes from the paper [98] by Saravia et al. The corpus comprises English tweets that are annotated with six fundamental emotions, namely: $\{\mathbf{0}: 'sadness', \mathbf{1}: 'joy', \mathbf{2}: 'love', \mathbf{3}: 'anger', \mathbf{4}: 'fear', \mathbf{5}: 'surprise'\}$. The dataset exhibits an average sample length of 15.04 words, comprises with a total of 20,000 samples. **Table 41** displays the distribution of labels within the training set. **Figure 4.4** displays the exemplification of each tweet alongside its corresponding emotional state. The data has already been pre-processed based on the approach described in their paper [7], and it is publicly available on the Hugging face library[10]. Hugging face datasets library provides API to download public datasets easily. In our experimental configuration, we partitioned the dataset into three subsets, consisting of 16,000, 2,000, and 2,000 instances for the purposes of training, testing, and validation, respectively.

Table 4.1 Overview of the dataset

Task	Application domain	Granularity	Classification	Dataset	Labels	Train	Validation	Test	Average Length
Emotion Classification	Social media	Tweets	Multi-class	emotion	6	16K	2K	2K	15.04

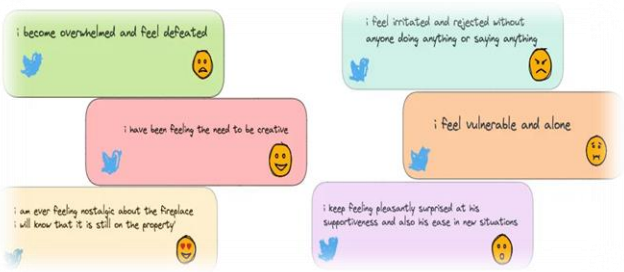


Figure 4.4 Examples of various tweets with their corresponding

Table 4.2 Distributions of labels in the training set

LABELS	DISTRIBUTION OF LABELS
Sadness	0.291625
Joy	0.335125
Fear	0.121063
Anger	0.134937
Surprise	0.035750
Love	0.081500

4.2.5.2 Victim Models

This section constitutes the description of the models that were trained on the emotion dataset, including their corresponding parameter configurations. The metric utilized for evaluating models of multi-label emotion classification is the accuracy and F1 score. The subsequent section presents experimental evidence demonstrating the susceptibility of each model to

[10] https://huggingface.co/datasets/dair-ai/emotion

adversarial conditions, as indicated by a reduction in accuracy following different adversarial attack algorithms.

Model Description

Deep learning classifiers possess the ability to autonomously learn and extract features, thereby leading to improved accuracy and performance. The investigation utilized several prominent deep learning classifiers, including Convolutional Neural Networks, Bi-LSTM & pre-trained transformer models. **Figure 4.5** depicts an overview of the classifiers used in this investigation.

Convolutional Neural Network (CNN): The CNN model is commonly utilized in various NLP applications. The process involves the utilization of a convolutional and pooling, or subsampling, layer within a deep feed-forward neural network, which subsequently transmits the data to a fully connected neural network layer[94]. Convolutional layers acquire features through the process of filtering input data, whereby multiple filters are combined to generate outputs. Pooling or subsampling is a technique employed in CNNs to reduce the feature resolution of layers, thereby enhancing the network's ability to resist distortion and noise. Pooling refers to the process of reducing the dimensionality of the output from a given layer to the subsequent layer. The classification tasks are executed by fully connected layers. CNNs exhibit a high level of proficiency in detecting local patterns and patterns that remain invariant to position. The effective use of CNNs) has been observed within the context of the categorization of text. [99].

Bi-directional Long Short-Term Memory (Bi-LSTM): Bi-LSTM models consist of a dual set of hidden layers. The forward processing of the input sequence is carried out through the utilization of the initial hidden layer, while the reverse processing is facilitated by the second hidden layer. The final layer of the neural network integrates the input from the previous layers. The utilization of hidden layers enables the system to effectively access both past and future contextual information pertaining to each individual point within the sequence LSTMs, and their bidirectional variants are quite helpful. They may learn when to ignore certain facts and when not to utilize certain gateways in their architecture. Bi-LSTM network offers the benefits of enhanced performance and a more rapid learning rate[100].

Transformer Models: Deep learning models that are based on transformers utilize a self-attention mechanism to assign varying degrees of significance to different segments of the input data. **Table 4.3** presents a descriptive analysis of various pre-trained models.

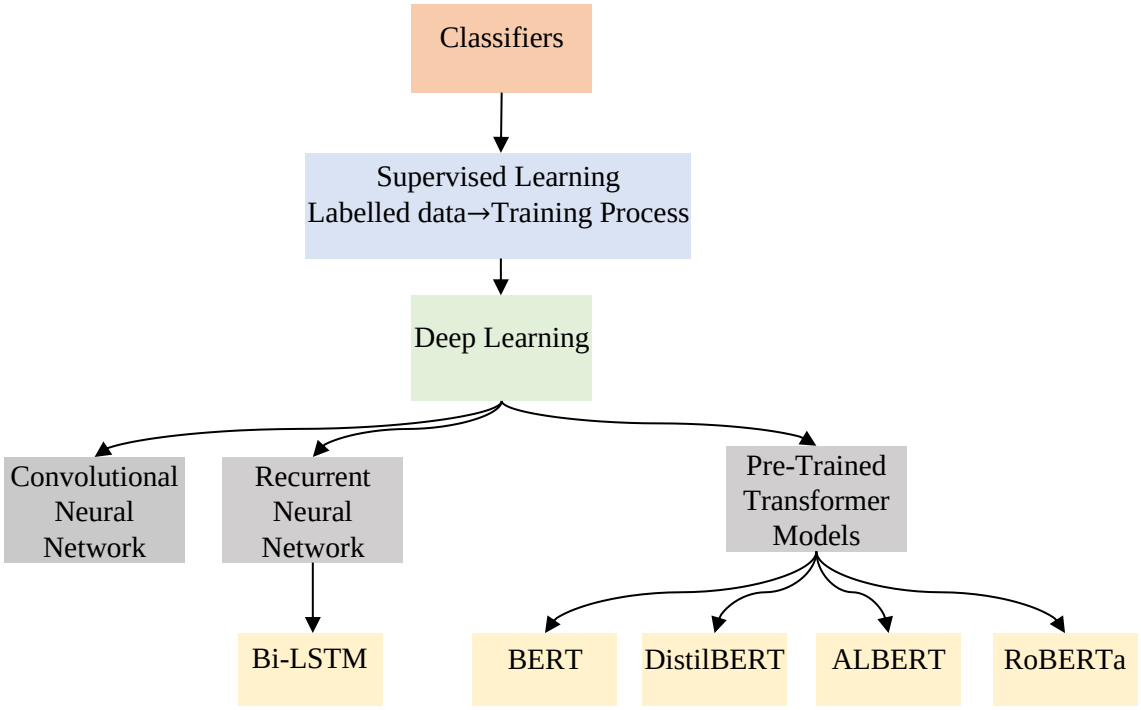


Figure 4.5 Overview of the classifiers

Table 4.3 comprehensive analysis of the transformer models

Model	Description	Tuning	Advantages	challenges	Trained on Datasets
BERT [63]	The proposed model is a bidirectional transformer architecture that integrates both Masked Language Modelling (MLM) mechanisms and Next Sentence Prediction (NSP).	Pre – Training Fine Tuning	1.) The capacity to effectively manage and process contextual information. 2.) Accelerated Training	1.) categorization is restricted to a single language 2.) The length of the input sentences is fixed. 3.) Has logical inference problems. 4.) The computational cost is high.	English Wikipedia & the BookCorpus
DistilBERT [64]	The employed methodology involves the utilization of an early version of BERT, whereby the number of layers has been reduced by a factor of two. Additionally, dynamic masking has been implemented.	Pre Training Fine Tuning	1.) The process of pre-training has been provided to improve the efficacy of language modelling. capability. 2.) In comparison to BERT, the recently developed model exhibits superior	1.) Fixed length constraint	English Wikipedia & BookCorpus

Model	Description	Tuning	Advantages	challenges	Trained on Datasets
			speed and reduced weight.		
ALBERT [65]	A variant of BERT with reduced parameters, resulting in a lighter model. The reduction of parameters can be achieved through cross-layer parameter sharing and factorized embedding layer parameterization.	Pre Training Fine Tuning	1.) lower memory consumption 2.) Enhance the training speed of BERT	1.) The model is incompatible with tasks which include text generation. 2.) Exhibits issues with logical inference.	English Wikipedia & BookCorpus
RoBERTa [66]	Replication of BERT with an expanded training set and fine-tuned hyper-parameters; makes use of dynamic masking	Pre Training Fine Tuning	1.) Utilizing a greater quantity of pre-training data improves performance. 2.) In subsequent NLP assignments, outperforms both XLNet and BERT.	1.) The resource-intensive characteristic. 2.) The task at hand requires significant computational resources and entails a lengthier processing time.	English Wikipedia, BookCorpus, CC-News, Stories, OpenWebText

Parameter Settings

The subsequent **Table 4.4** presents the parameter configurations for each model that was trained on the emotion dataset, with the purpose of assessing the efficacy of adversarial attack techniques.

Table 4.4 Parameter settings of the targeted models

Models	Parameter configurations
Word-CNN	The CNN model designed by Kim et al. [62] is selected for the purpose of the study. The Word-CNN model is employed with a configuration of 100 filters and 3 distinct window sizes, specifically 3, 4, and 5. The model's dropout rate is specified as 0.3, and it utilizes a base of 200-dimensional GLoVE embeddings. Subsequently, a fully connected layer is employed, followed by a time-dependent max-pooling layer for the purpose of classification. The model has attained an accuracy of *0.8870* and an F1 score of *0.8901* on the test set of the emotion dataset.
Bi-LSTM	LSTM model with 150 hidden states and bidirectional operation was formulated. Prior to being transmitted to the LSTM, the input is initially transformed into 200-dimensional GLoVE embeddings. Subsequently, the final execution of logistic regression is employed to predict the emotional state. This is accomplished by computing the mean of the LSTM outputs at every time step, resulting in a feature vector with a dropout rate of 0.3. The model attains a testing accuracy and F1 of *0.8934* & *0.8973* respectively on the emotion dataset.
BERT	We train the "bert-base-uncased" model for 10 iterations with a batch size of 64, a learning rate of 2e-05, and a maximum sequence length of 128 in order to optimize it for sequence classification on the emotion dataset. The training of the model was conducted through the utilization of a cross-entropy loss function. The highest performance achieved by the model in this task, as evaluated by the accuracy and F1 metrics on the test set, was *0.9405* and *0.9406* respectively, following eight epochs.
DistilBERT	We ran the "distilbert-base-uncased" model for 8 epochs with a batch size of 64, a learning rate of 2e-05, and a maximum sequence length of 128 to optimize it for sequence classification on the emotion dataset. The model was trained using a cross-entropy loss function because this task

Models	Parameter configurations
	involved classification. The evaluation set accuracy, which was discovered after 8 epochs, indicated that the model's highest accuracy and F1 score on this task were *0.9380 & 0.9379* respectively.
ALBERT	The "albert-base-v2" model was improved for sequence classification on the emotion dataset in our experiment. Running it for 8 epochs with a 64-batch size, a 2e-05 learning rate, and a 128-bit maximum sequence length. A cross-entropy loss function was used to train the model. The model's highest accuracy score on this job, as determined by the test set accuracy was *0.9360* & an F1 score of *0.9365*.
RoBERTa	The "Roberta-base" model has improved for sequence classification on the emotion dataset in our experiment. Running it with a maximum sequence length of 128 and a batch size of 64 for 8 epochs with a 2e-05 learning rate. Given that this was a classification problem, a cross-entropy loss function was used to train the model. The evaluation of the model's performance on this task yielded a maximum score of *0.9395* for accuracy and an F1 score of *0.9397*, both of which were achieved after 8 epochs.

The testing accuracy and F1 scores of each model that underwent training on the emotion dataset are presented in **Table 4.5** below.

Table 4.5 Testing Accuracy of the Targeted Models

	Word-CNN	Bi-LSTM	BERT	DistilBERT	ALBERT	RoBERTa
ACC	88.70%	89.34%	94.05%	93.80%	93.60%	93.95%
F1	89.01%	89.73%	94.06%	93.79%	93.65%	93.97%

4.2.5.3 Attacks

To generate adversarial examples, the attack techniques were applied to the test set of the emotion dataset. These adversarial samples are then utilized to misclassify the genuine emotion of the input sentence when fed to the pre-trained models. The experimentation was limited to attack methodologies that have been disseminated in highly regarded conferences and journals within the fields of AI and NLP. These publications include ACL, ICLR, EMNLP AAAI, NAACL, IJCAI, TACL, COLING, JMLR and TKDE. **Table 4.6** provides a concise summary of all adversarial attack strategies employed to perform the evaluation.

Table 4.6 Adversarial Attack Algorithms in NLP

Attacks	Granularity	Description
TextFooler[34]	Word-level	This assault approach involves exchanging words with the 50 nearest embedding neighbours of the target. optimized through BERT.
TextBugger[44]	Char-level	The effectiveness of this attack tactic has been enhanced for usage in realistic situations. They employ character substitution, space insertion, and character deletion. In context-aware word vector space, they also exchange words with their closest top neighbours and characters with letters that appear similar (for example, o with 0).
PWWS[45]	Word-level	By utilizing synonym swap, these attacks try to preserve lexical precision, grammatical correctness, and semantic proximity. Priority is determined by combining a word's saliency score and maximum word-swap efficiency.
PSO[46]	Word-level	Combining sememe-based word replacement with particle swarm optimization for word-level attacks.

Attacks	Granularity	Description
Pruthi et al.[47]	Char-level	Simulates common typographical errors using the QWERTY keyboard. This strategy employs character substitution, deletion, and insertion.
Kuleshov et al.[48]	Word-level	Replaces the important words in an input sequence with those from the counter-fitted word embedding space, according to a set of critical restrictions.
IGA[68]	Word-level	This attack technique ranks the most significant words in an input sequence using a scoring function and then replaces them with counter-fitted word embeddings. In addition to grammatical and natural checks, word embedding distance and sentence encoding cosine similarity are utilized to ensure the sample's validity.
DWB[50]	Char-level	Produces small text changes in a black-box setting. It employs a number of character-swapping strategies, including swapping, substituting, deleting, and insertion, for greedy replace-1 scoring.
BAE[52]	Char-level	This technique of attack employs a language model modification with a BERT mask. Using the language model, it replaces tokens to better fit the complete context.
A2T[53]	Word-level	This attack method employs gradient-based synonym word exchange in white-box hostile environments. It uses sentence encoding cosine similarity and grammatical checks to keep semantic similarity.
HotFlip[74]	Char-level	The methodology is predicated on an atomic flip mechanism that exchanges a token with another, contingent on the gradients of the one-hot input vectors.
InputReduction[75]	Word-level	The attack focuses on the least significant terms within a given sentence. The process involves the iterative removal of the word with the least significant importance score until a modification in the model's prediction is observed. The significance of a word can be evaluated by assessing the alteration in the level of confidence of the initial prediction upon its removal from the original sentence.
Checklist [51]	Word-level	Based on the fundamentals of behavioral testing. The use of modifications in terminology, numerical values, and locations, as well as contractions and expansions of the sentence's important terms.
CLARE [101]	Word-level	This strategy takes the use of a pre-trained linguistic model and employs greedy search with replace, merge, and insertion transformations. The USE similarity constraints are also utilized.

4.2.5.4 Evaluation metric

The efficacy of attack techniques has been exhibited through the utilization of two assessment criteria, namely, After Attack Accuracy and Attack Success Rate. The methodology for assessing and elucidating the two metrics is delineated as follows. Subsequently, the ASR metric is employed to assess the susceptibility of each deep learning-based sentiment classifier, distinguishing the most vulnerable and the more resilient.

After-attack accuracy: The purpose of adversarial attacks is to disrupt the efficacy of deep neural networks. Thus, the assessment of the attack's efficacy relies on the performance metrics of various tasks. Classification tasks are typically evaluated using performance metrics, such as accuracy. The accuracy scores prior to and after the attack have been demonstrated. Potent adversarial attacks are responsible for causing a significant decrease in accuracy scores through their attack methods.

Attack Success Rate: For determining the efficacy of the attack methods, five hundred successfully classified examples are selected at random from the test set so that the classification accuracy of the classifiers does not influence the assessment. On these source texts, the attack algorithms are then executed to generate adversarial instances. The adversarial cases are then sent to the deep learning-based emotion models to create the final prediction. The percentage of inaccurate predictions produced by these classifiers is utilized to determine the success rate of the attack algorithm. A greater success rate shows that the attack algorithm is capable of producing more powerful adversaries that may cause these emotion classifiers to behave improperly. We use attack success rate ASR (ratio of successful attack samples to the total of successful and failed samples$\frac{(\boldsymbol{successful\ samples})}{\boldsymbol{successful+failed\ samples}}$) to determine the effectiveness of an attack technique against a victim model. In formal terms, an attack is deemed successful when the classifier F is able to precisely classify the original legitimate input $\boldsymbol{F(X) = Y_{true}}$, but erroneously predicts the attacked input $\boldsymbol{F(X + \Delta X) = Y^*}$. Thus, the ASR can be expressed as shown in **Eqn. (4.1)**.

$$\frac{\boldsymbol{F(X+\Delta X)=Y^*}}{(\boldsymbol{F(X+\Delta X)=Y^*})+(\boldsymbol{F(X+\Delta X)=Y_{true}})} \quad \textbf{(4.1)}$$

In the context of untargeted attacks, $\boldsymbol{Y^*}$ denotes any label that is not $\boldsymbol{Y_{true}}$. The symbol $\boldsymbol{\Delta X}$ denotes alterations made to the original sample. A successful attack in this context means that the adversarial sample can incorrectly predict with high confidence. In the case of a failed attack, the adversarial sample is incapable of misclassifying the actual prediction. The statements that are omitted are those that the model incorrectly classified during training. We are interested in the success rates of attacks and the efficacy of Attacks in misclassifying outputs.

4.2.6 Experimental Results

We trained six cutting-edge models on the emotion dataset and attained test set accuracy scores identical to the original implementation. Different model hyperparameters and descriptions are provided in *Section 4.2*. The initial accuracy of the target models on the original test samples was recorded as the ***original accuracy***. Subsequently, the accuracy of the target models is evaluated by subjecting them to adversarial samples generated from the test samples via various attack algorithms. This metric is referred to as the ***after-attack accuracy***. Through a comparison of the two accuracy scores, the efficacy of the attack can be assessed. A larger

discrepancy between the accuracy values before and after the attack indicates a greater degree of success.

Table 4.7 Comparison of before and After-attack accuracy of each model against various adversarial attack algorithms

Attacks		Word-CNN	Bi-LSTM	BERT	DistilBERT	ALBERT	RoBERTa
(No Attack**)**	**Original Accuracy**	88.70%	89.30%	94.00%	93.80%	93.60%	93.90%
A2T [53]	**After Attack Accuracy**	42.20%	43.50%	45.30%	48.80%	66.20%	58.90%
BAE [52]		31.20%	30.40%	36.10%	35.50%	31.80%	35.40%
DWB [50]		02.20%	02.70%	04.70%	02.80%	03.70%	07.10%
IGA		08.80%	08.30%	12.20%	09.20%	08.40%	11.80%
Kuleshov et al. [48]		02.80%	02.20%	03.80%	03.60%	01.90%	07.10%
Pruthi et al. [47]		14.40%	11.20%	20.80%	17.40%	09.30%	19.60%
PSO [46]		06.60%	05.30%	09.90%	06.80%	04.80%	13.10%
PWWS [45]		04.20%	04.80%	13.40%	08.80%	04.90%	18.20%
HotFlip [74]		07.40%	07.90%	12.70%	09.30%	07.90%	17.10%
InputReduction [75]		16.20%	12.60%	19.10%	18.40%	18.60%	18.80%
Checklist [51]		47.80%	48.20%	56.50%	52.70%	49.20%	58.50%
CLARE [101]		02.60%	02.20%	03.20%	2.80%	03.10%	03.80%
Textbugger [44]		11.30%	10.60%	17.00%	20.10%	14.70%	20.30%
TextFooler [34]		01.20%	01.50%	01.80%	02.80%	01.20%	05.20%

Table 4.7 demonstrates that the TextFooler, an adversarial attack at the word-level, outperforms several state-of-the-art attack models by achieving the greatest reduction in accuracy across all emotion classifiers. The pre-attack accuracy of the Word-CNN and Bi-LSTM models is documented as 88.7 and 89.3, respectively. The Word-CNN and Bi-LSTM models exhibit a significant decrease in accuracy to 1.2% and 1.5%, respectively, when exposed to the TextFooler attack. The transformer models BERT, DistilBERT, ALBERT, and RoBERTa have their original accuracies on clean test set of 94.0%, 93.8%, 93.6%, and 93.9% respectively. After undergoing TextFooler, the accuracy metric experiences a substantial drop, with values of 1.8%, 2.8%, 1.2%, and 5.2% being observed for the BERT, DistilBERT, ALBERT, and RoBERTa models, respectively. The DWB attack methodology has been observed to cause a substantial reduction in the accuracies of WordCNN and Word LSTM models. Specifically, the accuracies of these models have been reduced from 88.7% to 2.2% and 89.3% to 2.7%, respectively. The efficacy of transformer models is considerably influenced. The accuracy of the of BERT exhibits a decline from 94% to 4.7%, while DistilBERT shows a decrease from 93.8% to 2.8%. Similarly, ALBERT's accuracy reduces from 93.6% to 3.7%, and RoBERTa's from 93.9% to 7.1%. The analysis reveals that Textfooler and DeepWordBug are the most effective word and character level attack techniques, respectively, compared to other attack methods.

Furthermore, apart from the previously mentioned drop in accuracy scores, we introduce the Attack Success Rates (ASR) as a means of assessing the effectiveness of each attack, as

described in *Section 4.4*. In addition, the Average Perturbed Rates (APR) are presented, whereby the computation involves dividing the count of perturbed words by the overall text length. The utilization of these two metrics facilitates the evaluation of various adversarial attack algorithms across numerous models. The aim of this analysis is to demonstrate the comparative degree of risk that specific attack algorithms present to particular models. Additionally, the utilization of mean ASR scores serves to demonstrate the model that exhibits the greatest vulnerability to adversarial perturbations.

Table 4.8 Attack Results on all models (***ASR***=Attack Success Rate, ***APR*** =Average Perturbed rate)

Attacks	Word-CNN		Bi-LSTM		BERT		DistilBERT		ALBERT		RoBERTa	
	ASR %	APR %	ASR %	APR %	ASR %	APR %	ASR %	APR %	ASR %	APR %	ASR %	APR %
A2T [53]	58.5	07.2	56.4	08.8	53.6	06.7	51.5	06.1	30.5	11.6	39.5	06.7
BAE [52]	67.8	11.2	68.5	10.9	62.8	10.6	64.6	10.6	67.3	10.7	63.5	10.6
Checklist [51]	41.4	09.2	40.6	08.8	34.4	08.9	35.5	09.2	38.8	10.0	32.8	08.4
CLARE [101]	97.6	12.8	97.8	13.0	94.6	13.4	95.6	12.8	94.6	13.6	94.1	12.8
DWB [50]	98.4	14.4	97.9	13.2	93.8	13.4	97.9	12.5	96.8	10.2	86.4	13.4
HotFlip [74]	94.3	12.9	94.7	10.4	88.0	11.0	90.9	11.4	92.4	10.8	87.2	10.9
IGA [68]	95.2	11.0	96.4	10.7	89.7	10.8	95.8	11.0	96.6	09.2	87.4	10.8
InputReduction [75]	83.7	13.3	88.2	13.6	76.2	14.6	81.3	14.8	80.6	12.9	75.5	15.8
Kuleshov et al. [48]	96.7	07.8	97.1	09.2	92.7	08.2	95.9	08.2	97.8	08.4	92.7	08.2
Pruthi et al. [47]	85.4	08.9	88.9	07.9	79.3	07.5	82.8	07.4	90.5	07.3	80.2	07.5
PSO [46]	93.3	10.8	94.2	10.4	83.5	14.6	93.9	12.8	94.7	10.7	92.7	14.6
PWWS [45]	94.8	10.2	95.2	11.3	86.5	10.3	91.9	10.9	95.7	09.9	81.2	10.3
Textbugger [44]	88.7	08.1	91.2	07.4	82.4	08.5	79.8	09.2	85.2	09.4	79.1	08.5
TextFooler [34]	98.9	08.8	98.2	08.3	97.9	08.3	97.9	08.6	98.9	09.9	94.7	08.3

Table 4.8, demonstrates that TextFooler [34] attains an impressive degree of attack efficacy with minimal alterations across all six emotion classifiers. Irrespective of the length of the textual sequence or the degree of accuracy of the target model, a perturbation ratio lower than 10% has the potential to deceive the model. The TextFooler method achieves the highest ASR scores compared to all other attack methods for Word-CNN and Bi-LSTM, with respective scores of 98.9% and 98.2%. Of all attack strategies applied to transformer models, the Textfooler method makes the ALBERT model exhibits the highest ASR while only perturbing an average of 9.7% of the words. The DistilBERT model's ASR score of 97.98% was compromised by a word perturbation rate of 10.64% through this attack technique. Even BERT, a heavy model with 110 million parameters that is regarded as the greatest performer in NLP for multiple tasks, is extremely susceptible to this perturbation strategy, with an ASR of 97.93% and a perturbation rate of 11.32. It outperforms past cutting-edge attack methods for RoBERTa and achieves an ASR of 94.79%. This indicates that the TextFooler[34] attack mechanism is able to successfully manipulate classifiers to assign inaccurate predictions. This attack technique is based on the premise of substituting the most essential words with their counter-fitted word synonyms, and the results indicate that this word-level attack strategy is most effective at tricking the most well-known cutting-edge transformer models. The DWB

(DeepWordBug)[50] attack approach obtains an ASR of 97.98% and 93.81 percent on DistilBERT and BERT, respectively. DWB [50] gets the maximum ASR among all assaults on the DistilBERT model and the second-highest ASR for the BERT model, following TextFooler. DWB gets 96.84% for ALBERT and 86.46% for RoBERTa. DWB also obtains ASR scores of 98.4% for Word-CNN and 97.9% for Bi-LSTM. It is concluded that DWB is the most effective character-level perturbation strategy involving character substitution, space insertion, and character deletion. The following section discusses the proper analysis of the conducted experiment, which reveals which ranks each attack type according to its potency in fooling the models and which model is more resistant to adversarial circumstances, along with plausible explanations.

4.2.7 Analysis and Discussion

To determine which attack approach is more effective at fooling the model with a lower average perturbance rate. We have determined the mean success rate and perturbation rate for each attack type across all models, which can be seen in **Figure 4.6**. Then, we rank various attack strategies as given in **Table 4.9**. It clearly shows that TextFooler [34] is the most effective word-level perturbation also CLARE [101], which also works on counter-fitted synonym substitution, ranks second, whereas DWB [50] is the most effective character-level perturbation, according to a comprehensive evaluation of assault tactics, the attack method, A2T [53], which uses gradient-based synonym word swap in the white-box adversarial setting is least effective on all models.

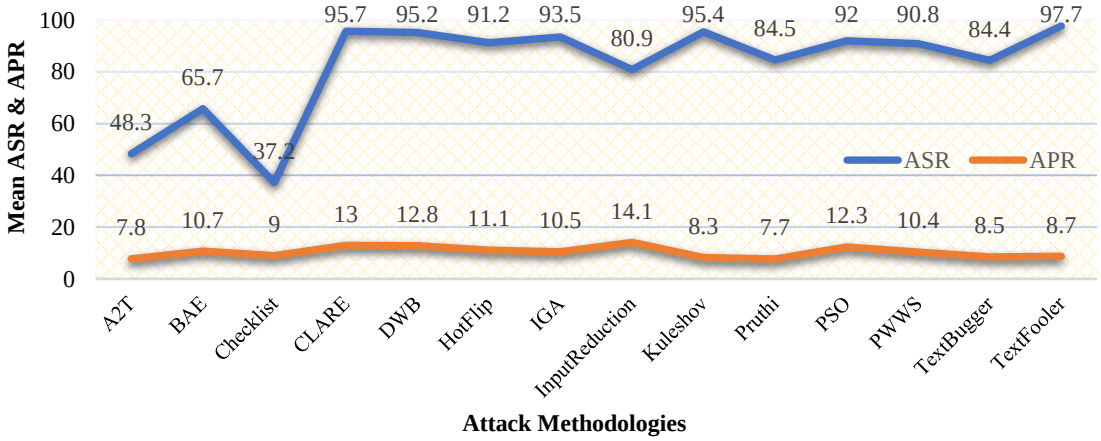


Figure 4.6. Mean attack success rate (ASR) of each attack algorithm along with mean average perturbed rate (APR) on all models.

Word-level perturbation attacks are at the top of **Table 4.9**'s list of the most effective attacks. The top three attack methods are word perturbation-based. This demonstrates conclusively that models tend to be more susceptible to word-level attacks as opposed to character-level attacks.

Table 4.9 Mean attack success rate of each attack type on all models

Attack Methodology	Mean ASR%	Mean APR%	Perturbation level
TextFooler [34]	97.7	08.7	Word-level
CLARE [101]	95.7	13.0	Word-level
Kuleshov et al. [48]	95.4	08.3	Word-level
DWB [50]	95.2	12.8	Character-level
IGA [68]	93.5	10.5	Word-level
PSO [46]	92.0	12.3	Word-level
HotFlip [74]	91.2	11.1	Word-level
PWWS [45]	90.8	10.4	Word-level
Pruthi et al. [47]	84.5	07.7	Character-level
Textbugger [44]	84.4	08.5	Character-level
InputReduction [75]	80.9	14.1	Word-level
BAE [52]	65.7	10.7	Character-level
A2T [53]	48.3	07.8	Word-level
Checklist [51]	37.2	09.0	Word-level

Following the identification of the perturbation level that exerts the greatest impact on transformer models, we proceed to assess the models' resilience against all adversarial configurations, with the aim of finding the model that exhibits the highest and lowest susceptibility. The average ASR for each model is calculated using the subsequent **Eqn. (4.2)**:

$$\boldsymbol{S_r} = \frac{\sum_{i=1}^{a} \frac{S_i}{S_i + F_i}}{a} \qquad \textbf{(4.2)}$$

$\boldsymbol{S_r} = Attack\ Success\ rate;\ \ \boldsymbol{a} = attack;\ \ \boldsymbol{S_i} = successful\ attack;\ \ \boldsymbol{F}_i = Failed\ attack$

(The Attack Success Rate is $\boldsymbol{S_r}$, no. of successful attacks is $\boldsymbol{S_i}$, the no. of unsuccessful attacks

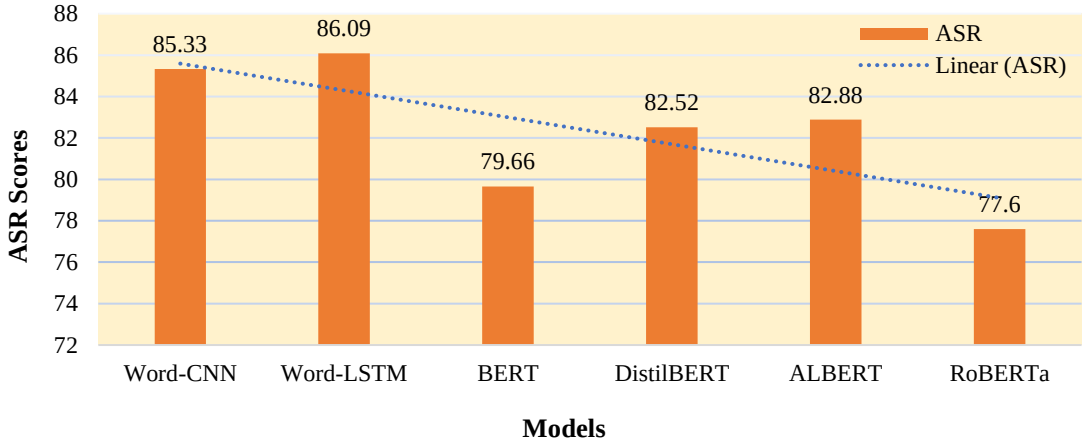


Figure 4.7. Average ASR of different emotion classifiers

is $\boldsymbol{F}_i$, and the no. of attack recipes is $\boldsymbol{a}$. The statements the model initially incorrectly anticipated during its training are skipped statements. They were not included in the calculation.)

From an information theoretic perspective, we investigated the robustness of language models against adversarial manipulations and calculated the average ASR of these models as shown in **Figure 4.7**. On evaluation, it has been determined that the Bi-LSTM model exhibits the highest

susceptibility to adversarial attacks, as evidenced by an ASR score of 86.09%. The Word-CNN model, on the other hand, is the second most vulnerable, with an average ASR score of 85.33%. RoBERTa is the least & ALBERT is the most vulnerable among the transformer models; our observations conclude that the lighter model ALBERT model comprises 128 embedding layers, 768 hidden layers, and 12 million parameters got an ASR of 82.88% which clearly states that the ALBERT model will predict 82.88% erroneous prediction with high confidence. The DistilBERT, which is also a lighter version of BERT model, achieves an ASR of 82.52%, showing that it is also highly vulnerable to adversarial examples; on the other hand, BERT base has 110 million parameters, which is a heavy model with a very high computational complexity which makes it less vulnerable as compared to DistilBERT & ALBERT model. RoBERTa exhibits the lowest results, with an ASR of 77.60%. The likely explanations for the framework's reduced vulnerability are as follows: A lightweight variant of BERT, robustly optimized BERT, is optimized for both local features (word-level representation) and global characteristics (sentence-level representation). It includes a regularizer that selects local stable features that are immune to adversarial modifications and that maximizes the mutual information between local stable features and global features, hence contributing to a more robust global representation [66] . RoBERTa is a retraining of BERT with improved training techniques, one thousand times more data, and one thousand times more computational power. RoBERTa omits the Next Sentence Prediction (NSP) task from BERT's pre-training and adds dynamic masking so that the masked token varies between training epochs. It was also discovered that training was more successful with bigger batch sizes. Notably, RoBERTa uses 160 GB of text for pre-training, which consists of 16 GB of Books Corpus and English Wikipedia employed by BERT. CommonCrawl News dataset (63 million articles, 76 GB), Web text corpus (38 GB), and Tales from Common Crawl comprised the supplementary data (31 GB). This, along with 1024 V100 Tesla GPUs operating for 24 hours, led to the pre-training of RoBERTa. All of these factors conclude that RoBERTa is best for sequence classification in terms of better accuracy scores and less vulnerability to adversarial perturbed input samples. With this, it proves that RoBERTa outperforms BERT, ALBERT and DistilBERT in terms of robusticity against adversarial circumstances. The declining trendline illustrated in **Figure 57** suggests that the non-transformer-based Word-CNN and Bi-LSTM models are relatively more susceptible as compared to the transformer-based models. Further Analysis

This section presents an analysis of various factors pertaining to the generation of adversarial samples on each model, including execution time, scalability, sensitivity, utility, transferability property and interpretability.

Runtime Considerations: An investigation was carried out to assess the computational time consequences of the different attacks. The potential results of the average runtime to generate a single adversarial sequence using various attacks for each particular model is presented in **Figure 4.8**. The illustrations from the Figure suggest that producing adversarial samples for transformer models is time-consuming. The average runtime required to generate an adversarial example is observed to be at its lowest and highest for DeepWordBug (DWB) on WordCNN and TextFooler on BERT, respectively, with values of 18.76 and 104.55 seconds.

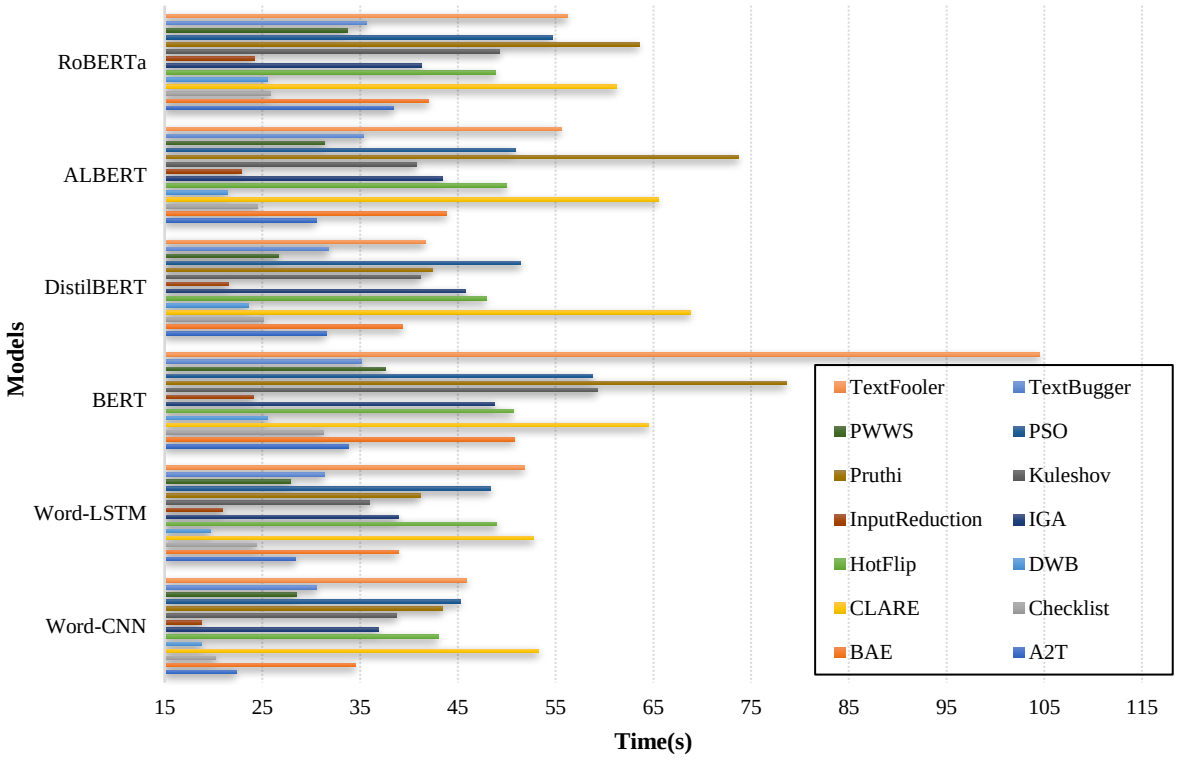


Figure 4.8 Average run time of generating an adversarial example from different attacks on various models

Scalability: To assess the efficacy of multiple attacks, a collection of five hundred test samples has been considered and subjected to diverse adversarial perturbations in the former section. The objective of scalability analysis is to evaluate the fluctuations in ASR values in relation to variations in the number of test samples. In order to accomplish this objective, an examination was conducted on the ASR scores for a range of test samples taken from 100 to 1000, targeting all models. The observation from **Figure 4.9** shows an inverse relationship between the magnitude of the sample population and the ASR scores. The correlation observed displays a significant negative slope as the number of samples increases, indicating that the models have

to predict the broader spectrum of perturbation with a larger number of samples. It is noteworthy that modifications to the sample count do not yield considerable variations in the ASR scores across all attack methods. The ASR scores obtained from Word-CNN and Bi-LSTM exhibit a high degree of similarity, despite the differences in the test samples. As such, the scalability analysis depicted in the Figure pertains to four transformer models.

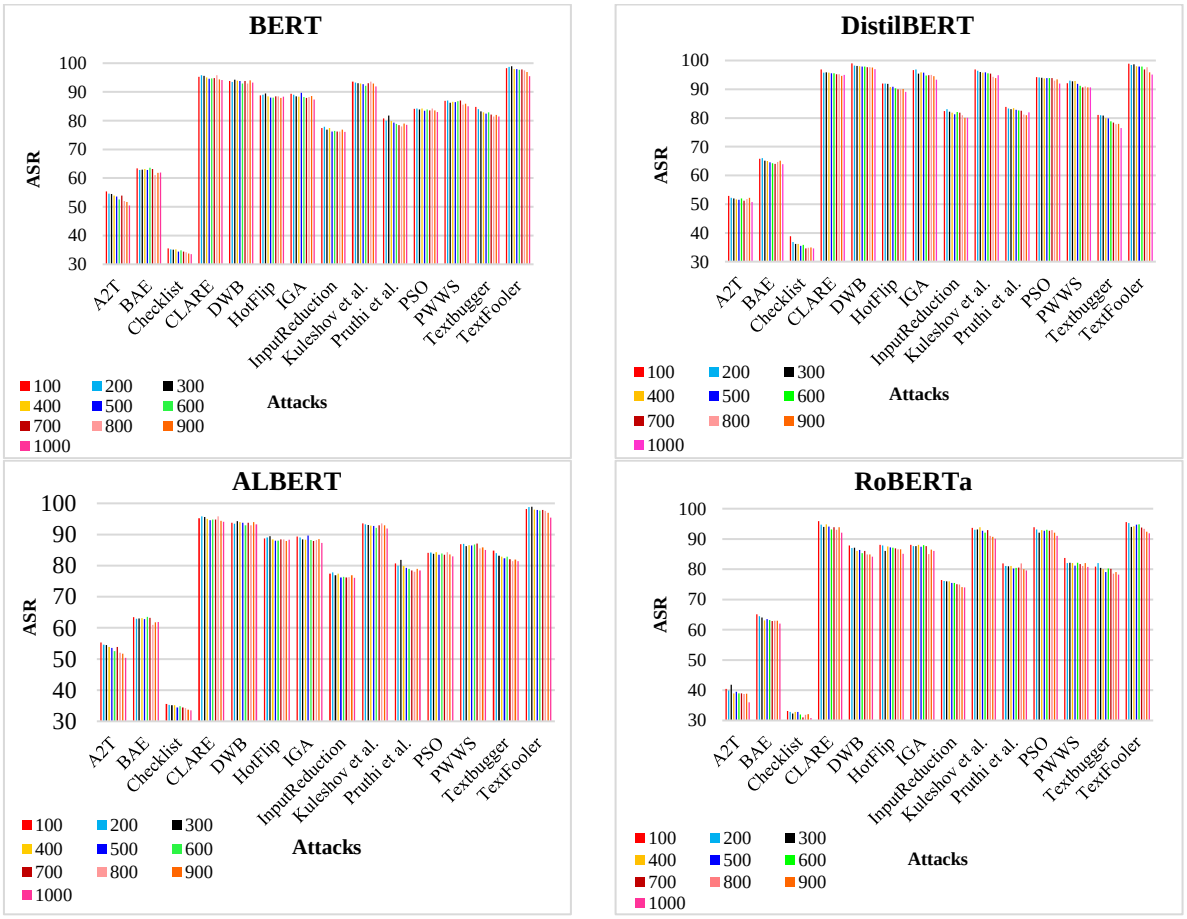


Figure 4.9 Disparities in ASR scores due to variations in the test sample scale

Sensitivity: The utilization of cosine similarity function and word embedding distance is prevalent in multiple attack techniques as a constraint to uphold the semantic significance of the modified input following alterations. The negative correlation between cosine similarity score (**δ**) and word error rate (**WER**) suggests that the model's vulnerability to perturbations increases as **δ** decreases. The constraint of word embedding distance is primarily employed in attacks involving the replacement of synonyms. The semantic proximity between the original and its corresponding synonym must be carefully considered to ensure that the fundamental meaning of the sentence is preserved. The optimal replacement word from the embedding space should be selected for substitution. The attribute of a model that relates to its ability to react to these limitations is frequently denoted as sensitivity in academic discourse. Henceforth, to curtail the extent of disturbance caused by all attacks, the parameter **δ** is allocated a numerical

value of **0.7**, thereby limiting the **WER**. Additionally, the word embedding distance is established at **0.6**, ensuring that a significant proportion of perturbations produced by the adversary culminate in an "unknown" token at the input of the model while simultaneously preserving the semantic consistency of the sentence.

Utility Analysis: Evidently, the adversarial texts generated by various attacks bear a higher degree of similarity to the originals. It can be concluded from **Figure 4.10** that adversarial examples generated by word perturbations are more effective at preserving utility. This is because char-level assaults encompass a variety of linguistic errors, including misspellings,

Figure 4.10. Adversarial examples generated through various word-level and char-level perturbation techniques

insertions, transpositions, arbitrary character substitutions, etc. Both humans and spell-checkers can readily detect character-level attacks. The adversary must increase transformed characters to generate adversarial instances, which can reduce imperceptibility and legibility.

Interpretability and Fairness: The LIME methodology, as proposed by Ribeiro et al.[78] , is employed in order to produce explanations at the local level for our models. LIME utilizes a linear model to approximate the local decision boundary for each example by fitting it over the corresponding samples. Acquired through the process of perturbation of the given example. In

order to assess the fidelity of the regional explications derived from LIME, the area over perturbation curve (AOPC) is employed as a metric[90], [91]. The formulae of this metric are given in **Eqn. (4.3)**.

$$\textbf{AOPC}=\frac{1}{K+1}\sum_{k=1}^{K}\frac{1}{N}\sum_{1=1}^{N} f(x_{(0)}^{(i)}) - f(x_{(k)}^{(i)}) \quad \textbf{(4.3)}$$

In the context of this study, $\boldsymbol{x}_{(\mathbf{0})}^{(\boldsymbol{i})}$ denotes an instance wherein $\boldsymbol{x}^{(\boldsymbol{i})}$ with no words eliminated is represented by $(\mathbf{0})$, while $\boldsymbol{x}_{(\boldsymbol{k})}^{(\boldsymbol{i})}$ with the exclusion of the $\boldsymbol{k}$ most significant words represented by $(\boldsymbol{k})$**.** The function $\boldsymbol{f}(\boldsymbol{x})$ is utilized to denote the level of assurance of the model regarding the target label $\boldsymbol{y}^{(\boldsymbol{i})}$. The concept of AOPC pertains to the average alteration in the model's confidence towards the target label upon removal of the top-k most significant words as identified by LIME. This is a notion that is derived from intuition. The function $\boldsymbol{f}(\boldsymbol{x})$ is utilized to denote the level of assurance of the model regarding the target label $\boldsymbol{y}^{(\boldsymbol{i})}$. The concept of AOPC pertains to the average alteration in the model's confidence towards the target label upon removal of the top-k most significant words as identified by LIME [78]. This is a notion that is derived from intuition. A sample size of 1000 instances was randomly selected from the test set for the purpose of evaluation. In the process of obtaining explanations using LIME [78], a total of 1000 perturbed samples are generated for each instance from TextFooler[34] attack for the evaluation. The value of $K = 10$ was chosen for the AOPC metric. According to **Table 4.10**, it can be observed that the Bi-LSTM model attains the highest AOPC. It is observed that the AOPC scores of RoBERTa models are significantly beneath other models, indicating that RoBERTa may possess a lower level of interpretability compared to all models.

Table 4.10 The AOPC scores for each model's LIME explanations [78]. A model with a higher AOPC score is more interpretable.

Models	Word-CNN	Bi-LSTM	BERT	DistilBERT	ALBERT	RoBERTa
AOPC Scores	0.67	0.74	0.61	0.49	0.57	0.22

Property of Transferability: The current investigation examined the transferability of adversarial text, specifically, the extent to which adversarial samples generated using one model can successfully deceive a different model[102]. We subsequently evaluated the **ASR** scores of these examples against alternative target models. The findings presented in **Table 4.11** indicate that there exists a moderate level of transferability among the non-transformer-based models, whereas the transferability is comparatively lesser in the transformer models. The BERT model exhibits greater transferability among transformer models.

Table 4.11 Transferability of Adversarial examples on emotion dataset. Row *i* and column *j* is the **ASR** of adversaries generated for model *i* evaluated on model *j*.

	Word-CNN	**Bi-LSTM**	**BERT**	**DistilBERT**	**ALBERT**	**RoBERTa**
Word-CNN	----	79.26	79.77	59.04	57.72	55.02
Bi-LSTM	84.34	----	72.34	60.78	52.22	68.16
BERT	72.82	67.75	----	58.88	64.67	59.56
DistilBERT	56.02	44.76	52.56	----	45.00	49.96
ALBERT	58.09	52.82	47.98	48.47	----	47.23
RoBERTa	45.45	58.89	59.56	42.85	48.38	----

Hardware specification & memory details: The experimental procedure was conducted using NVIDIA RTX A5000 Graphics Processing Units (GPUs), which were equipped with a system memory of 128 gigabytes. The system is outfitted with a cumulative graphics memory of 48GB, driver version 460.32.03, CUDA version 11.2, and a hard disc capacity of 10TB. The study was conducted using a repeated measurements approach with each experimental condition being replicated five times. The calculated quantity represents the arithmetic average of the acquired outcomes. The importance of this replication lies in the probabilistic nature of the training process, which leads to fluctuations in the level of performance. Stop-words tend to be eliminated during the process of feature extraction in diverse natural language processing tasks, as is the case in this particular experiment. This phenomenon can be attributed to the observation that the inclusion or exclusion of stop-words has negligible impact on the predictive outcomes. In the course of generating 500 adversarial samples on the emotion dataset, it was observed that the memory usage varied across different adversarial attack algorithms. On average, the development of adversarial samples through each attack algorithm required roughly 8.3 GB of RAM, 3.9 GB of graphics memory, and 26.2 GB of disc space. The percentage error associated with this investigation was approximately $\pm 4\%$.

4.2.8 Conclusion

The present study aims to investigate the vulnerability of text-based emotion detection mechanisms to adversarial attacks. The results unambiguously indicate that the analysis of emotional content in text can be disrupted by altering the words and characters utilized in machine learning models while still preserving semantic similarity for human observers. In this research, we discuss a flaw in deep learning models for emotional analysis. By taking advantage of this weakness, we have demonstrated a comparative analysis of various deep-learning-based emotion classifiers to determine which model is more susceptible to adversarial perturbations and which is more robust against them. In general, empirical evidence has demonstrated the feasibility of perturbing automatic emotion detection models through

adversarial alterations, resulting in obfuscation. Therefore, it is imperative to prioritize the development of adversarial robust generalizations over standard generalizations in order to promote societal progress. Furthermore, this study advocates for the exploration of models that exhibit higher resilience against adversarial attacks, as opposed to solely pursuing heightened confidence scores.

4.3 Significant Outcomes of this Chapter

The significant outcomes of this chapter are as follows:

- To determine which perturbation technique (attack method) most influences deep learning models trained on text classification dataset. The models include two commonly used Word-CNN and Bi-LSTM, as well as four powerful transformer models, BERT, DistilBERT, ALBERT and RoBERTa.
- Compare and explain the performance of DL-based text classifiers to determine which model is most sensitive and which is most resilient in all adversarial conditions.

The following research works form the basis of this chapter:

- **A. Bajaj** and D. Kumar Vishwakarma, "Evading text-based emotion detection mechanism via adversarial attacks," ***Neurocomputing***, vol. 558, p. 126787, Nov. 2023, doi: 10.1016/j.neucom.2023.126787.

Chapter 5: Adversarial Defense Against Word-level Textual Adversarial Attacks

5.1 Scope of this Chapter

Text classification is a developing subject in the realm of text data mining. However, the existing approaches for determining phrase polarity have significant drawbacks. In particular, deep learning algorithms are highly susceptible to attacks from adversarial samples. Existing word-level textual adversarial attacks primarily involve substituting synonyms, which leads to modified text that typically retains correct syntax and meaning. Protecting against hostile attacks at the individual word level presents additional difficulties. This article presents a new system called Adversarial Robust Generalised Network (ARG-Net) designed to defend against word-level adversarial attacks. ARG-Net enhances the model's performance by incorporating adversarial training and data perturbation techniques throughout the training phase. Our studies on two datasets confirm that the model, developed using our approach, effectively counteracts word-level adversarial attacks.

5.2 ARG-Net: Adversarial Robust Generalized Network to Defend Against Word-Level Textual Adversarial Attacks

5.2.1 Abstract

Natural Language Processing models have strong performance across various applications, although they are susceptible to manipulation by adversarial instances. A minor disturbance has the potential to alter the outcome of the deep learning algorithm. Humans find it difficult to detect this type of disturbance, particularly adversarial instances created using word-level adversarial attacks. Char-level adversarial assault can be countered using grammar detection and word recognition. The current word-level textual adversarial attacks rely on the substitution of synonyms, resulting in perturbed text that often maintains proper syntax and semantics. Defending against adversarial attacks at the word level poses more challenges. This study introduces a novel system called Adversarial Robust Generalized Network (ARG-Net) that aims to protect against word-level adversarial assaults. ARG-Net improves the model's performance by using both adversarial training and data perturbation techniques during the training process. The results of our tests on two datasets demonstrate that the model, which is

built upon our framework, successfully mitigates word-level adversarial assaults. The defence success rate of the model trained using ARG-Net is greater than that of the previous defence approaches when tested on 1000 adversarial samples. Furthermore, our model exhibits superior accuracy on the standard testing set compared to current defence techniques. The accuracy is comparable to, or even surpasses, that of the conventional model.

5.2.2 Fundamentals of textual adversarial attack on granularity of word-level

5.2.2.1 Textual Adversarial Attack

Textual adversarial assaults aim to provide adversarial instances that can deceive a victim model $\boldsymbol{F}$. In this case, the victim model is assumed to be a text classifier that relies on a Pretrained Language Model. Provided a dataset including $\boldsymbol{N}$ input sequences $\boldsymbol{X} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ and their corresponding labels $\boldsymbol{Y} = \{\boldsymbol{y_1}, \ldots, \boldsymbol{y_n}\}$, the victim classifier $\boldsymbol{F}$: $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ is a mapping that transforms the input space $\boldsymbol{X}$ into the label space $\boldsymbol{Y}$. An input sequence $\boldsymbol{x}$ in $\boldsymbol{X}$ should have a matching adversary example $\boldsymbol{x_{adv}}$ that satisfies the following condition as given in **Eqn. (5.1):**

$$\boldsymbol{F}(\boldsymbol{x_{adv}}) \neq \boldsymbol{F}(\boldsymbol{x}) \;\&\; \boldsymbol{dis}(\boldsymbol{x_{adv}} - \boldsymbol{x}) \leq \boldsymbol{\delta} \qquad \textbf{(5.1)}$$

The function $\boldsymbol{dis}()$ quantifies the perceived discrepancy between $\boldsymbol{x_{adv}}$ and $\boldsymbol{x}$, whereas $\boldsymbol{\delta}$ acts as a threshold to restrict the magnitude of disturbances[103].

5.2.2.2 Word-level Textual Adversarial Attack

Perturbations in language can be divided as sentence-level, word and char-level attacks based on their granularity[43]. Word-level assaults include replacing many terms with their equivalents in order to deceive the model, either through a heuristic or a contextualized approach. Word-level attacks include the computation of the word vector $V(\boldsymbol{x})$ for each individual word $\boldsymbol{w_i}$ in the text $\boldsymbol{x}$. When selecting priority replacement words, a complete analysis is conducted to determine the extent of change in the classification likelihood following the substitution, as well as the relevance of every term. Subsequently, the notation $\boldsymbol{x'} = (\boldsymbol{w_1}, \boldsymbol{w_2} \ldots \boldsymbol{w_n})$ is employed to represent the text with $\boldsymbol{w_i'}$ replacing $\boldsymbol{w_i}$, while $\boldsymbol{\Delta P_i'} = F_y(\boldsymbol{x}) - F_y(\boldsymbol{x'})$ is used to denote the importance of replacing $\boldsymbol{w_i}$. Ultimately, the score of $\boldsymbol{w_i}$ is determined by the subsequent function as shown in **Eqn. (5.2)** below:

$$\boldsymbol{H}(\boldsymbol{x}, \boldsymbol{x_i'}, \boldsymbol{w_i}) = \boldsymbol{\phi}\,(V(\boldsymbol{x}))_{\boldsymbol{i}} * \boldsymbol{P_i'} \qquad \textbf{(5.2)}$$

The function $\phi()_i$ refers to the softmax function. The process involves sorting all terms in decreasing order depending on the $\boldsymbol{H}$ value and selecting contender terms. It then utilizes a searching strategy to travel through all the contender terms until the classifier label varies[104].

5.2.2.3 Defence Against Word-level Textual Adversarial Attack

The objective of adversarial defences is to train a classifier that can attain a high level of accuracy when tested with both legitimate and non-legitimate i.e., adversarial instances. Adversarial defences should not just protect against static adversarial instances but also guard against repeated attacks. There are two kinds of word-level textual adversarial defences: Synonym Encoding Method (**SEM**) and Random Substitution Encoding (**RSE**) explained in detail below.

Limitation: Both techniques are effective in defending against hostile instances. Research has indicated that models with low performance on the testing set typically demonstrate a high level of adversarial resilience. Thus, it is necessary to train an improved network model that exhibits comparable performance during testing. This standard guarantees that the defense mechanism enhances resilience. Test accuracy should not be compromised in favor of robustness.

- ***Synonym Encoding Method (SEM):*** SEM is suggested for finding input texts' neighbors. SEM assumes synonymous texts are input texts' neighbors. SEM replaces words with synonyms to create synonymous texts. A reliable model labels synonymous texts the same. Synonyms must be combined and allocated unique encodings to create a map. SEM generates and stores the synonym encoding dictionary word vector matrix, which is subsequently used to train models.

- ***Random Substitution Encoding (RSE)***[105]***:*** RSE randomly picks a substitution rate within the specified range for the input text. It then produces a candidate word set $\boldsymbol{C}$ from the input sequence & finds altered words for every term in $\boldsymbol{C}$ to obtain the perturbed text $\boldsymbol{s'}$. RSE substitutes the word $\boldsymbol{s}$ with $\boldsymbol{s'}$ during training. During the testing step, the hostile cases from the testing set are fed into the upgraded classifier to evaluate impact of RSE.

Considering the limitations of current defense approaches, we provide a new defense framework called Adversarial Robust Generalized Network (ARG-Net). This framework

incorporates a step of modifying words during the training phase, building upon the concept of adversarial training. The studies demonstrate that our approach successfully protects against word level adversarial instances and surpasses the most recent defense techniques.

5.2.3 Proposed Adversarial defense Mechanism

This section introduces our method ARG-Net, which aims to enhance and make adversarial training for NLP more effective and feasible. We use both clean and adversarial cases in the training of our model. Our objective is to reduce both the loss incurred on the original training dataset and the loss incurred on the adversarial cases. We define $\boldsymbol{L}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ as the loss function for input sequence $\boldsymbol{x}$ & label $\boldsymbol{y}$, $\boldsymbol{A}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ be the adversarial assault that generates the hostile sample $\boldsymbol{x_{adv}}$. Our training aim is as shown in **Eqn. (5.3)**:

$$\boldsymbol{arg_{\theta} min\ E_{(x,y)\sim D}[L(\theta, x, y) + \alpha L(\theta, A(\theta, x, y), y)]} \quad \textbf{(5.3)}$$

The variable $\boldsymbol{\alpha}$ is employed to assign a weight to the adversarial loss. In this study, we assigned a value of $\boldsymbol{\alpha}$ = 1, ensuring that the two losses were given equal weight. The architecture of proposed defence mechanism is shown in **Table 5.1** Algorithm of defence against word level adversarial attack.

Algorithm 1: **ARG Defence Methodology**
Aim: Enhance **Robustness** of the model against $\boldsymbol{x_{adv}}$ **inputs**.
Requisite: Number of clean epochs ($\boldsymbol{n_{clean}}$),Number of Adversarial epochs ($\boldsymbol{n_{adv}}$), percentage of dataset to attack $\boldsymbol{\gamma}$, Attack $\boldsymbol{A}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$, training data $\mathbf{D} = \{\boldsymbol{x^i}, \boldsymbol{y^i}\}$, Smoothing proportion $\boldsymbol{\alpha}$.
Output: Generate legit output $\boldsymbol{y}$ **on** giving $\boldsymbol{x_{adv}}$ & $\boldsymbol{x}$ as input.

1. **Initialize model θ**
2. **for clean epoch** $= \mathbf{1}, \ldots\ldots\ldots, \boldsymbol{n_{clean}}$ **do**
3. **Train θ on D**
4. **end for**
   a. **for adversarial epoch** $= \mathbf{1}, \ldots\ldots\ldots, \boldsymbol{n_{adv}}$ **do**
5. **Randomly Shuffle D**
6. $\boldsymbol{D_{adv}} \Leftarrow \{\}$
7. $\boldsymbol{i} \Leftarrow \mathbf{1}$
8. **While** $|\boldsymbol{D_{adv}}| < \boldsymbol{\gamma} * |\mathbf{D}|$ $\boldsymbol{and}$ $\boldsymbol{i} \leq |\mathbf{D}|$ **do**
9. $\boldsymbol{x^i_{adv}} \Leftarrow \boldsymbol{A}(\boldsymbol{\theta}, \boldsymbol{x^i}, \boldsymbol{y^i})$
10. $\boldsymbol{D_{adv}} \Leftarrow \boldsymbol{D_{adv}} \cup \{\boldsymbol{x^i_{adv}}, \boldsymbol{y^i}\}$
11. $\boldsymbol{i} \Leftarrow \boldsymbol{i} + \mathbf{1}$
12. **end while**
13. **end for**
14. $\boldsymbol{D}, \Leftarrow \boldsymbol{D} \cup \boldsymbol{D_{adv}}$
15. **Train θ** $\boldsymbol{D}$, **with** $\boldsymbol{\alpha}$ used to weigh the loss

Table 5.1 provides a comprehensive depiction of the suggested adversarial defence algorithm. We conduct pristine training for $\boldsymbol{n_{clean}}$ epochs followed by $\boldsymbol{n_{adv}}$ epochs of adversarial training.

We produce the adversarial cases iteratively until we achieve a proportion of γ of the training dataset. When there are numerous GPUs available, we employ data parallelism to accelerate the generating process. In addition, we employ dataset shuffling prior to assaulting in order to prevent targeting the same sample in each epoch as shown in **Figure 5.1**. The subsequent section provides an elaborate explanation of the **ARG defence technique:**

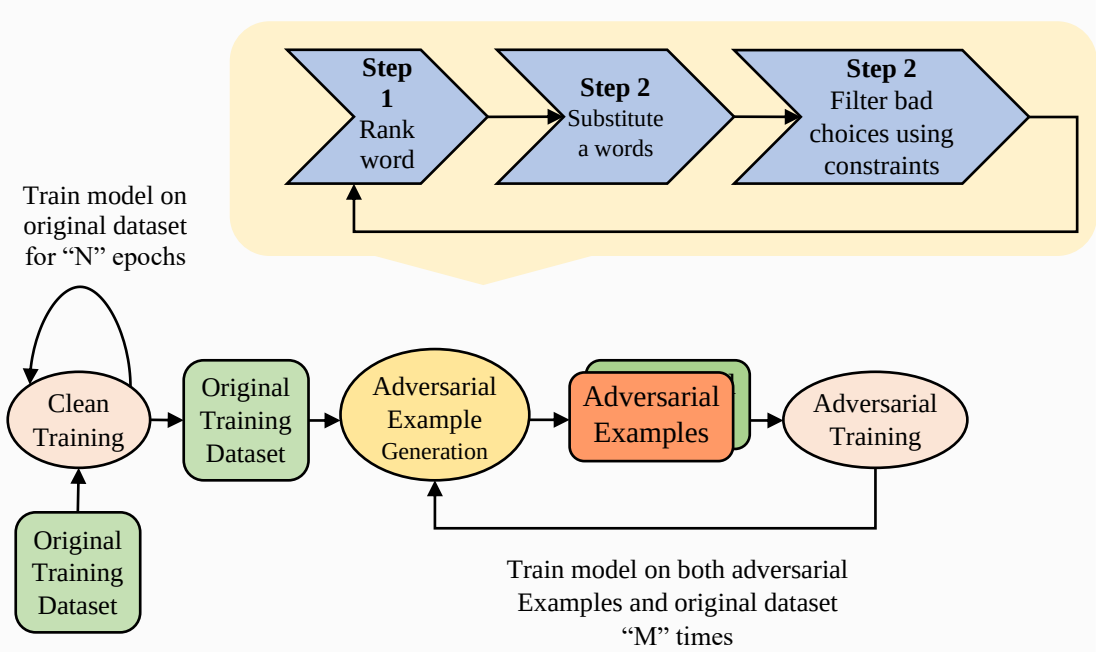


Figure 5.1 Proposed Architecture of Adversarial Defence Mechanism.

- We employ an assault to produce $\boldsymbol{k}$ hostile cases inside the conventional learning set. Subsequently, we include these adversarial examples into the training set, resulting in an augmented dataset. Given that the new training set includes adversarial instances from the conventional training set, the retrained classifier is capable of recognizing adversarial instances created by the clean test set.

- The loss function, denoted as $\boldsymbol{L}(\boldsymbol{s}, \boldsymbol{y})$, is utilised to enhance the classifier's ability to accurately anticipate the right label y based on the provided text $\boldsymbol{s}$. The loss function $\boldsymbol{L}$ is a type of loss that can be either binary or cross-entropy. It is accompanied with softmax activation as shown in **Eqn. (5.4)** below:

$$\boldsymbol{L}(\boldsymbol{s}, \boldsymbol{y}) = \sum_{i=1}^{m} -\log(y^i|s^i) \quad \textbf{(5.4)}$$

5.2.4 Experimental Approach

The subsequent part offers a thorough analysis of the dataset and models used in the inquiry, utilizing appropriate configurations to ensure a robust empirical evaluation. Following that, two advanced adversarial strategies were utilized with the explicit purpose of attacking and undermining the current trained models. Additionally, two fundamental defensive strategies are examined for the sake of comparison.

5.2.4.1 Dataset Description

The present investigation examines adversarial textual specimens on two widely known standard datasets that are commonly employed for text categorization problems. The final adversarial instances are created and assessed on the test set. **Table 5.2** provides a concise overview of the datasets.

- ***Internet Movie Database (IMDB)[11]:*** The IMDB dataset consists of 50,000 movie reviews that demonstrate a significant level of polarity. Out of these, 25,000 reviews are designated for training purposes, while the remaining 25,000 reviews are used for testing. The dataset has mean word length ranging from 215 to 216 words. The classifiers were learned to conduct a binary categorization on evaluations of movies, aiming to categorize them as either positive or negative emotion.

- ***AG news classification[12]:*** The dataset for AG news categorization is obtained from the hugging face. The AG corpus contains more than one million news pieces. A portion of AG's collection of news items consists of the headings and summaries of articles from the 4 primaries categories (Sports, Sci/Tech World and Business). The AG News dataset consists of 1,900 test instances and 30,000 learning examples for each category.

Based on the aforementioned datasets, the set of ARG has learning incorporated an additional 10% of adversarial cases derived from the conventional training set. The adversarial instances employed for adversarial learning are exclusively created with a maximum substitution rate of 25% to guarantee the efficacy of the adversarial instances. Simultaneously, these instances are all accurately identified by the model to mitigate mistakes stemming from the model's

[11] https://huggingface.co/datasets/imdb

[12] https://huggingface.co/datasets/ag_news

correctness. The defense model is tested using 1000 randomly generated adversarial cases derived from the adversarial assault on the conventional test set. This ensures that the training and testing processes are completely unrelated. The statistical data related to the dataset is presented in **Table 5.2**.

Table 5.2 Dataset Splits

Dataset	Class	Training	Testing	ARG Training	ARG Testing	Task
IMDB	2	25,000	25,000	27,500	1000	Sentiment Classification
AG News	4	120,000	76,00	132,000	1000	News Classification

5.2.4.2 Victim Models

Our ARG framework for text categorization challenge employs two deep-learning algorithms. Word-CNN and Bi-LSTM. The description of the model along with their parameter settings is shown in **Table 5.3**.

Table 5.3 Models

Model	Parameter Settings
Word-CNN	The Word-Convolutional Neural Network model utilises a collection of **100** filters and incorporates three separate window sizes, namely **3, 4**, and **5**. The system utilises a dropout rate of **0.3** and incorporates **200**-dimensional GLoVE embeddings as its foundation. Next, a fully linked layer is utilised, followed by a time-dependent max-pooling layer to improve the classification process.
Word-LSTM	A Long Short-Term Memory model was created with **150** hidden states and bidirectional functionality. Prior to being inputted into the framework, the data encounters an initial transformation step where information is transformed into **200**-dimensional GLoVE embeddings. Subsequently, the logistic regression model is employed for categorization. A feature vector is generated by computing the mean of the LSTM outputs at each time step while incorporating a rate of dropout of **0.3**.

5.2.4.3 Baseline Defence Techniques

We consider 3 baselines defence techniques: Normal Training (NT), Adversarial Training (AT), & RSE.

- **Normal Training (NT):** NT is a conventional training paradigm that does not incorporate any of the defensive strategies.
- **Adversarial Training (AT)**[53]***:*** AT is a training framework that uses adversarial

instances to enhance the robustness of an algorithm. We produce 10% of instances that are adversarial from every set of data. The number of hostile cases is confirmed by additional trials. Subsequently, the adversarial instances and original training examples are combined throughout the training procedure.

- **Random Substitution Encoding (RSE)**[105]***:*** During the training procedure, the model randomly replaces words in order to enhance its resilience. *Section II* of this article provides an elaborate description.

We examine the influence of the quantity of conventional learning set data augmentations on the precision of the classifier in regular adversarial training. In contrast to the traditional approach, we simply augment the size of the training set and momentarily disregard the disruption procedure in the ARG throughout the training stage. The technique of generating adversarial instances is quite time-consuming. We are only able to produce 15% of hostile examples inside the typical training set. It is evident that when more hostile cases are incorporated into the training set, the correctness of the model on the conventional testing set decreases. The explanation is that the training set introduced more data with distinct properties compared to the original dataset. For future studies, we set the ARG training to 27500, adding 10% of hostile cases. The model accuracy seldom falls in the conventional testing set, but approaches 90% on the adversarial example set. After using the whole ARG training approach, the model exceeds the accuracy of the conventional classifier on the clean test set as shown in **Table 5.4**.

Table 5.4 Accuracy Score of Adversarial Training

Additional Samples	1,250	2,500	3,750
Normal Accuracy (%)	86.69	86.28	86.66
Adversarial Accuracy (%)	88.12	90.91	93.80

5.2.5 Results & Analysis

We identify two parameters that impact the performance of defence: the rate at which text is replaced i.e., **TRR** (Text Replacement Rate) and the rate at which words are replaced i.e., Word Replacement Rate (**WRR**). This part examines the impact of classifier trained on the IMDB dataset using the ARG approach, while considering various **TRR** and **WRR** parameters. The experimental findings depicted in the **Figure 5.2, Figure 5.3** and **Figure 5.4** allow us to derive the various presumptions.

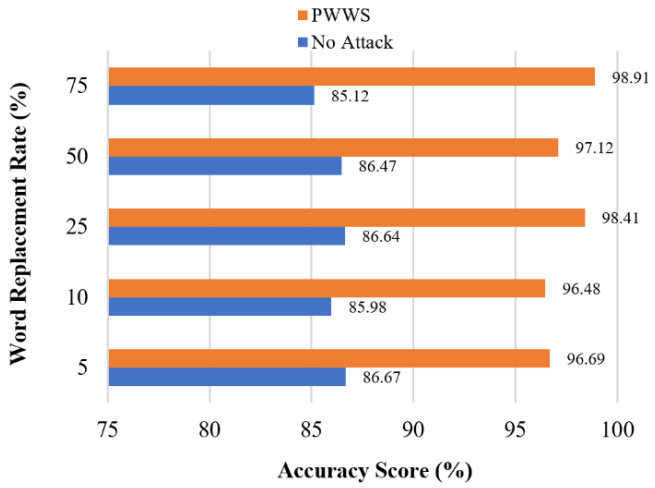


Figure 5.2 Bi-LSTM Accuracy Score when **TRR=12.5%**

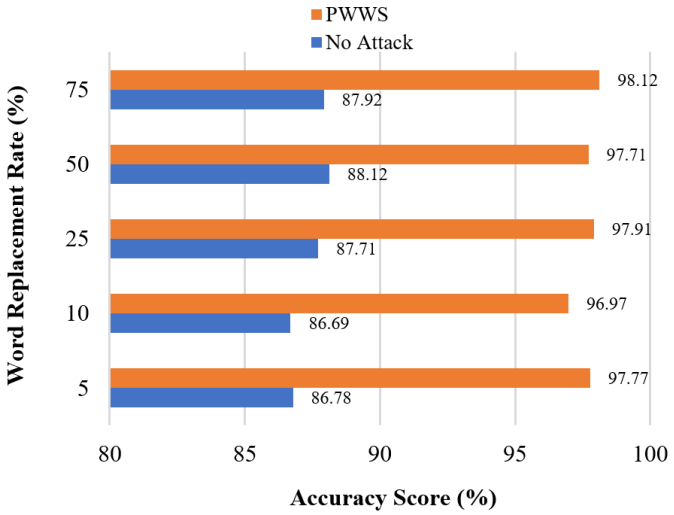


Figure 5.3 Bi-LSTM Accuracy Score when **TRR=25%**

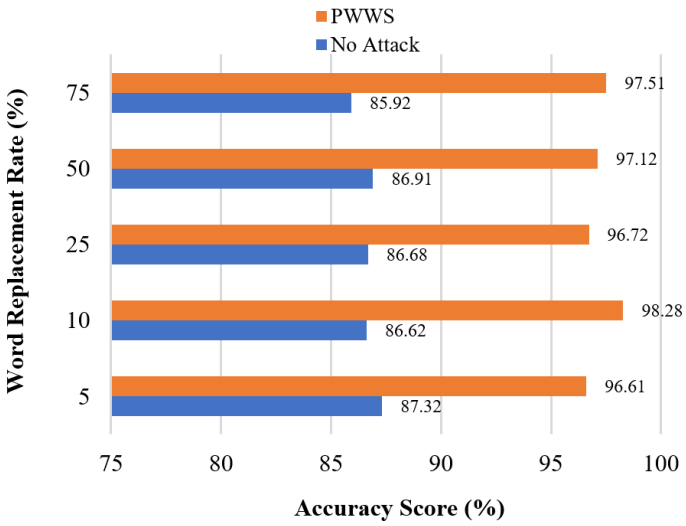


Figure 5.4 Bi-LSTM Accuracy Score when TRR=50%

Whenever the pace at which texts are replaced is small, the rate at which words are replaced is high, and the accuracy of the classifiers on both the clean test set and the hostile instances testing set is inversely related. For instance, using a **TRR** of 12.5% and a **WRR** of 75%, the classifier achieves an accuracy of 85.12% on the standard testing set. However, the model performs much better on the adversarial instance test set, reaching an accuracy of 98.91%. This demonstrates that as the quantity of disturbed words increases, the model's ability to learn the characteristics of adversarial cases improves. However, this comes at the cost of losing significant amounts of information from the regular training set, ultimately resulting in subpar performance on the test set. **Table 5.5** displays the results of classifiers that were trained using different defence techniques on the conventional test set. The efficacy of our approach on the network has been exceptional. It nearly matches or surpasses the original classifier in terms of categorization accuracy. In all scenarios, our approach surpasses adversarial learning and RSE defence.

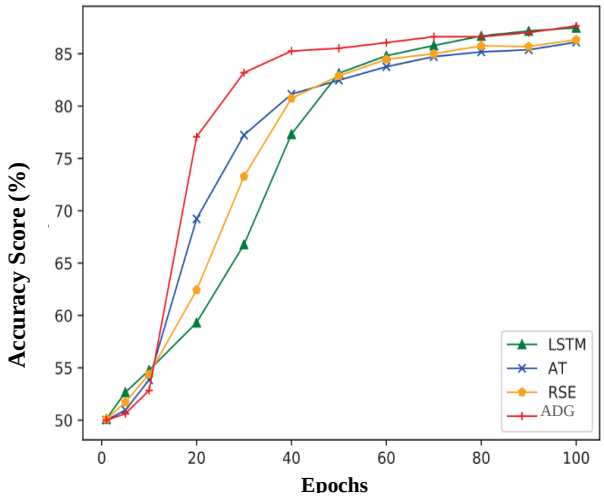


Figure 5.5 variation in the accuracy scores for each epoch

Table 5.5 Test set evaluation results

Framework	Defence Technique	IMDB	AG News
Word-CNN	NT	86.63	90.33
	AT[53]	87.17	89.57
	RSE[105]	87.56	89.98
	ADG (Ours)	**88.31**	**90.52**
Bi-LSTM	NT	89.78	90.22
	AT[53]	89.01	90.56
	RSE[105]	90.01	89.97
	ADG (Ours)	**90.87**	**91.05**

Figure 5.5 depicts the fluctuation in the precision of the classifier on the conventional test set as the duration for training rises. We document 100 epochs of the complete training procedure to monitor the progress throughout the training procedure. The green line depicts the alterations in the conventional LSTM classifier, while the red line shows the modifications in the ARG defense classifier. The remaining two curves correspond to the adversarial learning and the RSE defense architecture. The graphic illustrates that the accuracy of all classifiers reaches a stable state after the $50th$ epoch. The ARG model exhibits quicker convergence. The area reaches a state of stability by the 40th epoch. Furthermore, beyond the 10th epoch, the accuracy of the ARG approach surpasses that of other models throughout the whole training period due to the data growth in ARG.

Table 5.6 displays the accuracy outcomes of several configurations, encompassing two models: The 2 datasets utilized are IMDB & AG News. The 2 textual assault techniques employed are random & PWWS. 2 defence mechanism, namely RSE, AT & ADG, are utilised. The 2 rows in each dataset provide the test outcomes of the defence framework on 1000 adversarial samples from the adversarial testing set, using 2 different attack strategies. All 1000 hostile testing sets are derived from the standard testing set. All of these are adversarial cases that have been correctly classified by the model, and their replacement rates are below 25%. Based on the data presented in the table, it can be inferred that our defence approach surpasses the performance of existing defence methods. ADG demonstrates superior performance in handling adversarial cases compared to the defence techniques AT and RSE. This superiority is shown across two datasets and two attack mechanisms.

Table 5.6 The assessment outcomes of 1000 adversarial instances across various configurations.

Dataset	Attack	Word-CNN			Bi-LSTM		
		AT	RSE	ADG	AT	RSE	ADG
IMDB	Random[105]	87.40	78.40	**94.50**	71.30	58.60	**76.90**
	PWWS[45]	90.20	81.20	**97.30**	90.10	63.70	**95.70**
AG News	Random[105]	61.50	58.80	**66.10**	68.70	59.70	**70.10**
	PWWS[45]	87.80	72.20	**89.40**	87.40	64.90	**92.50**

5.2.6 Conclusion

This article examines the defence mechanisms against adversarial instances and explores the present findings in word-level text adversarial instances. Our research indicates that generating adversarial instances at the word level is more difficult compared to assaults at the character level. This paper's defense strategy involves the integration of adversarial training and the

introduction of text perturbations throughout the training process. The defense mechanism it employs against hostile instances surpasses those of the current word-level defence approaches. Simultaneously, the defence strategy presented in this research produces an example with just a marginal efficiency, which can even surpass the conventional model under some scenarios on the testing set. Our technique demonstrates a level of accuracy on the testing set that is most similar to that of the original model when compared to the current strategy.

5.3 Significant Outcomes of this Chapter

The significant outcomes of this chapter are as follows:

- The study introduces a novel system called Adversarial Robust Generalized Network (ARG-Net) that aims to protect against word-level adversarial assaults. ARG-Net improves the model's performance by using both adversarial training and data perturbation techniques during the training process. The results of our tests on two datasets demonstrate that the model, which is built upon our framework, successfully mitigates word-level adversarial assaults.

The following research works form the basis of this chapter:

❖ **A. Bajaj** and D. K. Vishwakarma, "ARG-Net: Adversarial Robust Generalized Network to Defend Against Word-Level Textual Adversarial Attacks," in *2024 IEEE **9th International Conference for Convergence in Technology** (I2CT)*, Institute of Electrical and Electronics Engineers (IEEE), Jun. 2024, pp. 1–7. doi: 10.1109/i2ct61223.2024.10543623.

Chapter 6: Conclusion & Future Scope

6.1 Conclusion

This chapter serves as the finalization of the research conducted in this thesis. In summary, this work introduces three new methods for attacking text-based systems and one defence mechanism to counteract different types of malicious manipulations. Furthermore, we conducted a comparison analysis on the susceptibility of various neural text classifiers to adversarial attacks in order to determine the level of vulnerability and robustness exhibited by each model. Additionally, we have determined which perturbation approach poses a more significant risk to neural text classifiers. The details are as follows:

- **HOMOCHAR** is a novel textual adversarial attack operating within a black box setting. The proposed method generates more resilient adversarial examples by considering the task of perturbing a text input with transformations at the character level by replacing normal characters with imperceptible homoglyph characters. The objective is to deceive a target NLP model while adhering to specific linguistic constraints in a way such that the perturbations are unnoticeable under human observation.

- **Non-Alpha-Num**: A novel architecture for generating adversarial examples using Punctuations and Non-alphanumeric character insertion as perturbations for bypassing NLP-based clickbait detection mechanisms.

- **Inflect-Text:** Training on only perfect Standard English corpora predisposes pre-trained neural networks to discriminate against minorities from nonstandard linguistic backgrounds (e.g., African American Vernacular English, Colloquial Singapore English, etc.). We propose an **Inflect-Text** word-level attack that perturbs the inflectional morphology of words to craft plausible and semantically similar adversarial examples that expose these biases in popular NLP models.

- **ARG-Net:** Due to the recent increase in textual adversarial attack methods, neural text classifiers are facing a more significant risk. In response to this, we have suggested a strategy to enhance the generalization capability of these classifiers by implementing adversarial training (defensive strategy). In the proposed **ARG-Net** model, it utilizes Augmented text to generate adversarial examples. The model

undergoes training using both clean and adversarial cases to develop robust classification capabilities against word-level synonym replacement assaults.

- **Comparative Analysis:** Trained the most popular models on the emotion dataset and applied conventional adversarial attacks on the pre-trained models to have a comparative analysis among models to find out which model is more vulnerable and which attack method is a greater threat to state-of-the-art Classifiers.

6.2 Discussion & Future Scope

NLP algorithms relying upon text information can be vulnerable to subtle disturbances which may affect their computational results & prolong inferences time yet preserve the sensory representation unchanged. These attacks include arbitrary letter substitutions, as well as adding, removing, & replacing significant phrases using synonyms. The study presents three novel method that utilize character & word perturbations to trick neural text classifiers. The developers of many algorithms for NLP presently in use have failed to address those obstacles. This study examines how adversarial attacks impact text-based neural classifiers by using these adversarial manipulations. The results from experiments show that the novel attack mechanisms can surpass conventional methods with respect of the two; its effectiveness and average disturbances. In recent years, extensive research has been conducted to detect manipulation in multimedia content. While the performance has consistently improved in detecting or localizing these manipulations, several promising research directions need to be addressed. Also, we will discuss some real-world use cases for adversarial attacks and defenses in the text modality.

- **Apply novel attacks on API platforms:** The advent of machine learning has spurred a proliferation of companies offering their own Machine-Learning-as-a-Service (MLaaS) platforms, which are tailored to Deep Learning Text Understanding tasks, such as text classification. The models are deployed on cloud-based servers and user access is limited to utilising an application programming interface exclusively. In situations where such a setting is present, a perpetrator is devoid of information regarding the structure of the model, its parameters, or the data used for training. Their sole capability lies in querying the target model, with the output being in the form of prediction or probability scores. The attack models, as developed in the present study, has demonstrated efficacy in operating within

black-box scenarios. In future research, it may be feasible to carry out these attack mechanisms on digital platforms.

- **Expansion to broader NLP usage:** We plan to expand the scope of our novel assaults to include a wide variety of NLP applications in our upcoming endeavors[44]. The tasks encompass sentiment classification, poisonous content detection, phishing and smishing, detection, alongside the current emphasis on generalized neural text classifiers.
- **Adversarial Robustness:** Researchers desire to create a system that shows robustness to adversarial disruptions, particularly in the setting of our attack mechanisms, by using an adversarial training approach[53]. This work offers a thorough investigation of adversarial examples in text classification. Our next project is to enhance these models and enhance the accuracy of neural networks by integrating adversarial training techniques[106]. In subsequent studies, we plan to overcome these constraints by explicitly analyzing the L2 and dialectal populations and exploring ways to enhance the robustness of these frameworks at an earlier stage. It is advised that all firms involved in developing and incorporating these steps strengthen the products towards harmful alterations.
- **Extension to Targeted Attack:** The present research focuses solely on untargeted attacks that involve manipulating the algorithm's outcomes[43]. It is vital to recognize that our attack architectures can adjust to particular attacks, allowing the system to be manipulated into producing an identified categorization. Modifying an arrangement deliberately to achieve a certain outcome, also referred to as a targeted assault, involves changing the primary functioning element within the proposed strategy.
- **Extension of Adversarial Techniques to Other Languages:** In future work, we aim to extend our research to include the generation of adversarial examples in languages beyond English, such as Spanish, to evaluate the robustness of machine learning models in a more diverse linguistic context. By generating adversarial samples in multiple languages, we can better assess the vulnerabilities of models trained on different language datasets and improve their generalizability. For example, in Spanish, slight alterations like misspellings, synonym substitutions, or punctuation changes could lead to misclassifications. Below are a few adversarial examples in Spanish that could potentially deceive a text classification model:

  Original: "Me encanta este lugar, es muy bonito."

Adversarial: "Me encantha este lugár, es muy bonito." (Alterations: "encanta" -> "encantha", "lugar" -> "lugár")

Original: "Me encanta este lugar, es muy bonito."

Adversarial: "Me fascina este lugar, es muy hermoso." (Synonym replacement: "encanta" -> "fascina", "bonito" -> "hermoso")

Original: "Me encanta este lugar, es muy bonito."

Adversarial: "Es muy bonito este lugar, me encanta." (Reordering of sentence structure)

By incorporating adversarial example generation in Spanish and other languages, we can significantly broaden the scope of our study and create more robust NLP models capable of handling a wide variety of linguistic challenges.

- ❖ **Exploring Adversarial Examples on LLMs:** In future work, we plan to explore the use of Large Language Models (LLMs) for generating and analysing adversarial examples. By leveraging the power of LLMs, we could automate the generation of adversarial inputs across different languages and contexts, enabling a more efficient and scalable approach to testing model robustness. Furthermore, LLMs can help simulate a wider variety of linguistic nuances, such as syntax variations, colloquialisms, and even subtle semantic shifts, which could potentially bypass traditional defense mechanisms. This exploration could significantly advance our understanding of how adversarial attacks function in multilingual settings and improve the resilience of NLP systems.
- ❖ **Extension to Other Modalities:** In future work, we plan to extend our exploration of adversarial examples to include different modalities such as images, audio, and video. This will help assess the robustness of multimodal models and explore cross-modal adversarial attacks, where perturbations in one modality (e.g., text) affect model performance across others (e.g., image or audio). By broadening our focus, we aim to improve the resilience of models in real-world, multimodal applications.

The following are some real-world use cases that highlight adversarial attacks and defences in the text modality:

1. Sentiment Analysis

- **Use Case**: Sentiment analysis models are widely used in social media monitoring, customer feedback analysis, and brand management. Companies use these models to

automatically assess customer opinions about products, services, or brand reputation.

- **Adversarial Attack**: An attacker might subtly modify a review or social media post (e.g., changing words, introducing typos, or altering sentence structure) to mislead the sentiment analysis model into misclassifying a negative review as positive.
- **Defense**: Robustifying sentiment analysis models by training them on adversarial examples or using techniques like adversarial training could help improve model accuracy and reduce the likelihood of misclassification.

2. Spam Detection

- **Use Case**: Email services and messaging platforms use spam detection models to automatically filter out unwanted or malicious messages. This is crucial for preventing phishing attacks, malware distribution, or other malicious activities.
- **Adversarial Attack**: Attackers could craft spam messages that are subtly altered to evade detection (e.g., using obfuscated URLs, slight word modifications, or using uncommon synonyms).
- **Defense**: Developing advanced spam filters that can detect adversarial patterns and integrate multiple layers of checks (such as URL analysis, email header inspection, and linguistic pattern recognition) would strengthen defenses against such attacks.

3. Machine Translation

- **Use Case**: Machine translation models, like Google Translate or DeepL, are used in a wide range of applications including business, education, and government for real-time translation between languages.
- **Adversarial Attack**: Attackers may introduce subtle changes in the input text, such as misspelled words, or use syntactic ambiguities to alter the translation output. For example, changing a single character or word can cause a translation to be misleading or completely incorrect.
- **Defense**: Robust machine translation models could be trained with adversarial examples in multiple languages to reduce vulnerabilities to attacks and maintain accurate translations even in the presence of subtle adversarial input.

4. Autonomous Chatbots and Virtual Assistants

- **Use Case**: Virtual assistants like Siri, Alexa, and chatbots used in customer support rely on natural language understanding to interpret and respond to user queries.
- **Adversarial Attack**: Attackers could manipulate user queries by rephrasing or embedding irrelevant information to confuse the assistant or make it respond incorrectly (e.g., providing malicious commands or requests disguised as regular questions).
- **Defense**: Using adversarial training and enhanced contextual analysis could make these models more resilient to such manipulations, ensuring they respond correctly even to misleading or adversarial input.

5. Content Moderation

- **Use Case**: Social media platforms, forums, and online communities rely on automated content moderation systems to detect hate speech, harassment, or explicit content in user posts and comments.
- **Adversarial Attack**: Malicious users might try to evade content moderation by using homophones, slang, or creative misspellings to bypass detection of harmful content.
- **Defense**: Defenses could involve improving the model's ability to detect hidden meanings, slang, and other forms of obfuscation, as well as implementing a combination of keyword-based and machine learning-based approaches.

6. Financial Fraud Detection

- **Use Case**: Text analysis is used in fraud detection systems to flag suspicious financial transactions, especially those involving customer support interactions or abnormal patterns in transactional communication.
- **Adversarial Attack**: Fraudsters might alter the wording of messages or create fake communication that mimics legitimate customer service inquiries, tricking the model into categorizing fraudulent activities as valid.
- **Defense**: Developing fraud detection systems that consider not only the content but also the context, historical patterns, and linguistic features of communication could improve their robustness against adversarial manipulations.

Integrating these strategies into these real-world use cases can enhance the security, reliability, and fairness of NLP systems in these domains.

References

[1] G. W. Lindsay and D. Bau, “Testing methods of neural systems understanding,” *Cogn Syst Res*, vol. 82, 2023, doi: 10.1016/j.cogsys.2023.101156.

[2] X. Liu, Z. Zhuo, X. Du, X. Zhang, Q. Zhu, and M. Guizani, “Adversarial attacks against profile HMM website fingerprinting detection model,” *Cogn Syst Res*, vol. 54, 2019, doi: 10.1016/j.cogsys.2018.12.005.

[3] S. Jusoh, “A study on nlp applications and ambiguity problems,” *J Theor Appl Inf Technol*, vol. 96, no. 6, 2018.

[4] M. M. Uddin Rony, N. Hassan, and M. Yousuf, “Diving deep into clickbaits: Who use them to what extents in which topics with what effects?,” in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2017*, 2017. doi: 10.1145/3110025.3110054.

[5] N. H. Vo, K. D. Phan, A. D. Tran, and D. T. Dang-Nguyen, “Adversarial Attacks on Deepfake Detectors: A Practical Analysis,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022. doi: 10.1007/978-3-030-98355-0_27.

[6] M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C.-J. Hsieh, and M. Srivastava, “GenAttack: Practical Black-box Attacks with Gradient-Free Optimization,” in *GECCO 2019 - Proceedings of the 2019 Genetic and Evolutionary Computation Conference (2019)*, May 2019, pp. 1111–1119. [Online]. Available: http://arxiv.org/abs/1805.11090

[7] N. Boucher, I. Shumailov, R. Anderson, and N. Papernot, “Bad Characters: Imperceptible NLP Attacks,” Jun. 2021, [Online]. Available: http://arxiv.org/abs/2106.09898

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 6562–6572.

[9] R. Geirhos *et al.*, “Shortcut learning in deep neural networks,” *Nat Mach Intell*, vol. 2, no. 11, 2020, doi: 10.1038/s42256-020-00257-z.

[10] T. Gu, B. Dolan-Gavitt, and S. Garg, “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain,” Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.06733

[11] S. Tan, S. Joty, M. Y. Kan, and R. Socher, “It’s morphin’ time! combating linguistic discrimination with inflectional perturbations,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2020.acl-main.263.

[12] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011. doi: 10.1109/CVPR.2011.5995347.

[13] J. Wang and H. Zhang, “Bilateral Adversarial Training: Towards Fast Training of More Robust Models Against Adversarial Attacks,” Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.10716

[14] L. Schönherr, T. Eisenhofer, S. Zeiler, T. Holz, and D. Kolossa, “Imperio: Robust Over-the-Air Adversarial Examples for Automatic Speech Recognition Systems,” in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, 2020. doi: 10.1145/3427228.3427276.

[15] B. Zheng *et al.*, “Black-box Adversarial Attacks on Commercial Speech Platforms with Minimal Information,” in *Proceedings of the ACM Conference on Computer and Communications Security*, 2021, pp. 86–107. doi: 10.1145/3460120.3485383.

[16] A. Sharma, Y. Bian, P. Munz, and A. Narayan, “Adversarial Patch Attacks and Defences in Vision-Based Tasks: A Survey,” Jun. 2022, [Online]. Available: http://arxiv.org/abs/2206.08304

[17] E. R. Sykes, “A deep learning computer vision iPad application for Sales Rep optimization in the field,” *Visual Computer*, vol. 38, no. 2, 2022, doi: 10.1007/s00371-020-02047-5.

[18] B. Nelson *et al.*, “Misleading learners: Co-opting your spam filter,” in *Machine Learning in Cyber Trust: Security, Privacy, and Reliability*, 2009, pp. 17–51. doi: 10.1007/978-0-387-88735-7_2.

[19] P. W. Koh, J. Steinhardt, and P. Liang, “Stronger data poisoning attacks break data sanitization defenses,” *Mach Learn*, vol. 111, no. 1, 2022, doi: 10.1007/s10994-021-06119-y.

[20] B. I. P. Rubinstein *et al.*, “Antidote: Understanding and defending against poisoning of anomaly detectors,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 2009, pp. 1–14. doi: 10.1145/1644893.1644895.

[21] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, “Bagging classifiers for fighting poisoning attacks in adversarial classification tasks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011. doi: 10.1007/978-3-642-21557-5_37.

[22] B. Biggio, G. Fumera, and F. Roli, “Multiple classifier systems for robust classifier design in adversarial environments,” *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1–4, 2010, doi: 10.1007/s13042-010-0007-7.

[23] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, “Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning,” in *Proceedings - IEEE Symposium on Security and Privacy*, 2018. doi: 10.1109/SP.2018.00057.

[24] J. Ho, B. G. Lee, and D. K. Kang, “Attack-less adversarial training for a robust adversarial defense,” *Applied Intelligence*, vol. 52, no. 4, 2022, doi: 10.1007/s10489-021-02523-y.

[25] S. Sankaranarayanan, R. Chellappa, A. Jain, and S. N. Lim, “Regularizing deep networks using efficient layerwise adversarial training,” in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018, pp. 4008–4015.

[26] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples,” in *35th International Conference on Machine Learning, ICML 2018 (2018)*, Feb. 2018, pp. 274–283. [Online]. Available: http://arxiv.org/abs/1802.00420

[27] U. Shaham, Y. Yamada, and S. Negahban, “Understanding Adversarial Training: Increasing Local Stability of Neural Nets through Robust Optimization,” *Neurocomputing*, vol. 307, pp. 195–204, 2018.

[28] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks,” in *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, 2016, pp. 582–597. doi: 10.1109/SP.2016.41.

[29] N. Papernot, “A Marauder’s Map of Security and Privacy in Machine Learning,” in *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, Nov. 2018, pp. 1–1. [Online]. Available: http://arxiv.org/abs/1811.01134

[30] N. Akhtar, J. Liu, and A. Mian, “Defense against Universal Adversarial Perturbations,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2018)*, 2017, pp. 3389–3398.

[31] W. Wang, R. Wang, L. Wang, Z. Wang, and A. Ye, “Towards a Robust Deep Neural Network Against Adversarial Texts: A Survey,” *IEEE Trans Knowl Data Eng*, vol. 35, no. 3, 2023, doi: 10.1109/TKDE.2021.3117608.

[32] X. Han, Y. Zhang, W. Wang, and B. Wang, “Text Adversarial Attacks and Defenses: Issues, Taxonomy, and Perspectives,” *Security and Communication Networks*, vol. 2022. Hindawi Limited, 2022. doi: 10.1155/2022/6458488.

[33] S. Qiu, Q. Liu, S. Zhou, and W. Huang, “Adversarial attack and defense technologies in natural language processing: A survey,” *Neurocomputing*, 2022.

[34] 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,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Jul. 2019, pp. 8018–8025. [Online]. Available: http://arxiv.org/abs/1907.11932

[35] A. Bajaj and D. K. Vishwakarma, “A state-of-the-art review on adversarial machine learning in image classification,” *Multimed Tools Appl*, 2023, doi: 10.1007/s11042-023-15883-z.

[36] S. Goyal, S. Doddapaneni, M. M. Khapra, and B. Ravindran, “A Survey of Adversarial Defences and Robustness in NLP,” *ACM Comput Surv*, 2023, doi: 10.1145/3593042.

[37] Z. Liu *et al.*, “HyGloadAttack: Hard-label black-box textual adversarial attacks via hybrid optimization,” *Neural Networks*, p. 106461, Jun. 2024, doi: 10.1016/j.neunet.2024.106461.

[38] S. Lu *et al.*, “Black-box attacks against log anomaly detection with adversarial examples,” *Inf Sci (N Y)*, vol. 619, 2023, doi: 10.1016/j.ins.2022.11.007.

[39] X. Liu *et al.*, “Privacy and Security Issues in Deep Learning: A Survey,” *IEEE Access*, vol. 9. 2021. doi: 10.1109/ACCESS.2020.3045078.

[40] W. Wang, B. Tang, R. Wang, L. Wang, and A. Ye, “A survey on Adversarial Attacks and Defenses in Text,” *ArXiv*, vol. 2, 2019.

[41] I. Alsmadi *et al.*, “Adversarial Machine Learning in Text Processing: A Literature Survey,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3146405.

[42] J. Li, S. Zhang, J. Cao, and M. Tan, “Learning defense transformations for counterattacking adversarial examples,” *Neural Networks*, vol. 164, 2023, doi: 10.1016/j.neunet.2023.03.008.

[43] J. X. Morris, E. Lifland, J. Y. Yoo, and Y. Qi, “TextAttack: A framework for adversarial attacks in natural language processing,” *ArXiv*, pp. 119–126, 2020.

[44] J. Li, S. Ji, T. Du, B. Li, and T. Wang, “TextBugger: Generating Adversarial Text Against Real-world Applications,” in *26th Annual Network and Distributed System Security Symposium*, 2019, pp. 1–15. doi: 10.14722/ndss.2019.23138.

[45] S. Ren, Y. Deng, K. He, and W. Che, “Generating natural language adversarial examples through probability weighted word saliency,” in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020. doi: 10.18653/v1/p19-1103.

[46] Y. Zang *et al.*, “Word-level Textual Adversarial Attacking as Combinatorial Optimization,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6067–6080. doi: 10.18653/v1/2020.acl-main.540.

[47] D. Pruthi, B. Dhingra, and Z. C. Lipton, “Combating adversarial misspellings with robust word recognition,” in *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020. doi: 10.18653/v1/p19-1561.

[48] V. Kuleshov, S. Thakoor, T. Lau, and S. Ermon, “Adversarial Examples for Natural Language Classification Problems,” in *ICLR 2018 : International Conference on Learning Representations*, 2018.

[49] X. Wang, H. Jin, and K. He, “Natural language adversarial attacks and defenses in word level,” *ArXiv*, 2019.

[50] J. Gao, J. Lanchantin, M. Lou Soffa, and Y. Qi, “Black-box generation of adversarial text sequences to evade deep learning classifiers,” in *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, 2018, pp. 1–21. doi: 10.1109/SPW.2018.00016.

[51] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, “Beyond Accuracy: Behavioral Testing of NLP models with CheckList,” *ACL 2020 - 58th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, pp. 4902–4912, 2020.

[52] S. Garg and G. Ramakrishnan, “BAE: BERT-based Adversarial Examples for Text Classi□cation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6174–6181.

[53] J. Y. Yoo and Y. Qi, “Towards Improving Adversarial Training of NLP Models,” in *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, 2021. doi: 10.18653/v1/2021.findings-emnlp.81.

[54] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, “Adversarial example generation with syntactically controlled paraphrase networks,” in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018. doi: 10.18653/v1/n18-1170.

[55] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, “Deep text classification can be fooled,” in *IJCAI International Joint Conference on Artificial Intelligence*, 2018, pp. 4208–4215. doi: 10.24963/ijcai.2018/585.

[56] M. Wolff, “Attacking neural text detectors,” pp. 1–8, 2020.

[57] D. Cer *et al.*, “Universal Sentence Encoder,” Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.11175

[58] D. Naber, P. F. Kummert, T. Fakultät, and A. Witt, “A Rule-Based Style and Grammar Checker,” Technische Fakultät, Universität Bielefeld, 2003. Accessed: May 10, 2024. [Online]. Available: https://www.danielnaber.de/languagetool/download/style_and_grammar_checker.pdf

[59] B. Pang and L. Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” in *ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2005.

[60] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis.”

[61] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, 1997, doi: 10.1162/neco.1997.9.8.1735.

[62] Y. Kim, “Convolutional neural networks for sentence classification,” in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014. doi: 10.3115/v1/d14-1181.

[63] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019.

[64] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT , a distilled version of BERT : smaller , faster , cheaper and lighter,” pp. 2–6, 2019.

[65] Z. Lan *et al.*, “ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS,” in *International Conference on Learning Representations (ICLR*, 2020, pp. 1–17. [Online]. Available: https://github.com/google-research/ALBERT.

[66] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” in *International Conference on Learning Representations (ICLR*, Jul. 2019, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1907.11692

[67] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized autoregressive pretraining for language understanding,” in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019, pp. 1–11.

[68] X. Wang, H. Jin, Y. Yang, and K. He, “Natural Language Adversarial Defense through Synonym Encoding,” in *37th Conference on Uncertainty in Artificial Intelligence, UAI 2021*, 2021.

[69] R. Jia, A. Raghunathan, K. Göksel, and P. Liang, “Certified robustness to adversarial word substitutions,” in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019. doi: 10.18653/v1/d19-1423.

[70] M. Terzi, G. A. Susto, and P. Chaudhari, “Directional adversarial training for cost sensitive deep learning classification applications,” *Eng Appl Artif Intell*, vol. 91, 2020, doi: 10.1016/j.engappai.2020.103550.

[71] J. Y. Yoo, J. X. Morris, E. Lifland, and Y. Qi, “Searching for a Search Method: Benchmarking Search Algorithms for Generating NLP Adversarial Examples,” in *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2020, pp. 323–332. [Online]. Available: https://github.com/QData/TextAttack

[72] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, “Stop Clickbait: Detecting and preventing clickbaits in online news media,” in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*, 2016. doi: 10.1109/ASONAM.2016.7752207.

[73] S. Garg and G. Ramakrishnan, “BAE: BERT-based adversarial examples for text classification,” in *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2020. doi: 10.18653/v1/2020.emnlp-main.498.

[74] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, “HotFlip: White-Box Adversarial Examples for NLP,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 2018, pp. 31–36.

[75] S. Feng, E. Wallace, A. Grissom, M. Iyyer, P. Rodriguez, and J. Boyd-Graber, "Pathologies of neural models make interpretations difficult," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018. doi: 10.18653/v1/d18-1407.

[76] C. Xie *et al.*, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. doi: 10.1109/CVPR.2019.00284.

[77] J. Zhang, W. Peng, R. Wang, Y. Lin, W. Zhou, and G. Lan, "Enhance Domain-Invariant Transferability of Adversarial Examples via Distance Metric Attack," *Mathematics*, vol. 10, no. 8, 2022, doi: 10.3390/math10081249.

[78] M. T. Ribeiro, S. Singh, and C. Guestrin, "'why should i trust you?' explaining the predictions of any classifier," in *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, 2016. doi: 10.18653/v1/n16-3020.

[79] J. R. Rickford and S. King, "Language and linguistics on trial: Hearing rachel jeantel (and other vernacular speakers) in the courtroom and beyond," *Language (Baltim)*, vol. 92, no. 4, 2016, doi: 10.1353/lan.2016.0078.

[80] W. Bright and B. F. Grimes, "Ethnologue: Languages of the World," *Language (Baltim)*, vol. 62, no. 3, 1986, doi: 10.2307/415492.

[81] E. G. Winkler, "English as a Global Language (review)," *Language (Baltim)*, vol. 81, no. 4, 2005, doi: 10.1353/lan.2005.0220.

[82] H. N. Seymour, "The challenge of language assessment for African American English-speaking children: A historical perspective," *Semin Speech Lang*, vol. 25, no. 1, 2004, doi: 10.1055/s-2004-824821.

[83] L. WHITE, "Fossilization in steady state L2 grammars: Persistent problems with inflectional morphology," *Bilingualism: Language and Cognition*, vol. 6, no. 2, 2003, doi: 10.1017/s1366728903001081.

[84] C. G. Haswell, "A Global Model of English: How New Modeling Can Improve the Appreciation of English Usage in the Asia Pacific Region," *Asia Pacific World*, vol. 4, no. 2, 2013, doi: 10.3167/apw.2013.040208.

[85] B. Haznedar, "Missing surface inflection in adult and child L2 acquisition," *... Approaches to Second Language Acquisition ...*, no. Gasla 2002, 2003.

[86] C. L. Nelson, Z. G. Proshina, and D. R. Davis, *The Handbook of World Englishes*. 2020. doi: 10.1002/9781119147282.

[87] X. Yang, Y. Qi, H. Chen, B. Liu, and W. Liu, "Generation-based parallel particle swarm optimization for adversarial text attacks," *Inf Sci (N Y)*, vol. 644, 2023, doi: 10.1016/j.ins.2023.119237.

[88] X. Zhang, J. Zhao, and Y. Lecun, “Character-level convolutional networks for text classification,” in *Advances in Neural Information Processing Systems*, 2015.

[89] A. Raghunathan, J. Steinhardt, and P. Liang, “Certified defenses against adversarial examples,” in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[90] D. Nguyen, “Comparing automatic and human evaluation of local explanations for text classification,” in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018. doi: 10.18653/v1/n18-1097.

[91] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE Trans Neural Netw Learn Syst*, vol. 28, no. 11, 2017, doi: 10.1109/TNNLS.2016.2599820.

[92] S. Kusal, S. Patil, K. Kotecha, R. Aluvalu, and V. Varadarajan, “Ai based emotion detection for textual big data: Techniques and contribution,” *Big Data and Cognitive Computing*, vol. 5, no. 3, 2021, doi: 10.3390/bdcc5030043.

[93] G. W. Parrot, *Emotions in social psychology: Key readings in social psychology*. 2001.

[94] S. Kusal, S. Patil, · Jyoti Choudrie, · Ketan Kotecha, · Deepali Vora, and I. Pappas, “A Review on Text-Based Emotion Detection-Techniques, Applications, Datasets, and Future Directions”.

[95] C. Szegedy *et al.*, “Intriguing properties of neural networks,” in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014, pp. 1–10.

[96] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks,” in *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, Institute of Electrical and Electronics Engineers Inc., Aug. 2016, pp. 582–597. doi: 10.1109/SP.2016.41.

[97] Y.-T. Tsai, M.-C. Yang, and H.-Y. Chen, “Adversarial Attack on Sentiment Classification,” in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019, pp. 233–240. doi: 10.18653/v1/w19-4824.

[98] E. Saravia, H. C. Toby Liu, Y. H. Huang, J. Wu, and Y. S. Chen, “Carer: Contextualized affect representations for emotion recognition,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018. doi: 10.18653/v1/d18-1404.

[99] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning-Based Text Classification,” *ACM Computing Surveys*, vol. 54, no. 3. 2021. doi: 10.1145/3439726.

[100] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, 1997, doi: 10.1109/78.650093.

[101] D. Li *et al.*, “Contextualized Perturbation for Textual Adversarial Attack,” 2021. doi: 10.18653/v1/2021.naacl-main.400.

[102] N. Papernot, P. McDaniel, and I. Goodfellow, “Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples,” May 2016, [Online]. Available: http://arxiv.org/abs/1605.07277

[103] A. Bajaj and D. K. Vishwakarma, “Exposing the Vulnerabilities of Deep Learning Models in News Classification,” in *2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT)*, IEEE, Feb. 2023, pp. 1–5. doi: 10.1109/ICITIIT57246.2023.10068577.

[104] 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,” Jul. 2019. [Online]. Available: http://arxiv.org/abs/1907.11932

[105] Z. Wang and H. Wang, “Defense of Word-Level Adversarial Attacks via Random Substitution Encoding,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020. doi: 10.1007/978-3-030-55393-7_28.

[106] H. Zhang *et al.*, “Masking and purifying inputs for blocking textual adversarial attacks,” *Inf Sci (N Y)*, vol. 648, 2023, doi: 10.1016/j.ins.2023.119501.

[107] A. Turner, D. Tsipras, and A. Madry, “Clean-Label Backdoor Attacks,” *The International Conference on Learning Representations*, 2019.

[108] J. Steinhardt, P. W. Koh, and P. Liang, “Certified defenses for data poisoning attacks,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1–13.

[109] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *ASIA CCS 2017 - Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security*, 2017, pp. 506–519. doi: 10.1145/3052973.3053009.

[110] Y. Zhou, M. Han, L. Liu, J. He, and X. Gao, “The Adversarial Attacks Threats on Computer Vision: A Survey,” in *Proceedings - 2019 IEEE 16th International Conference on Mobile Ad Hoc and Smart Systems Workshops, MASSW 2019*, Institute of Electrical and Electronics Engineers Inc., Nov. 2019, pp. 25–30. doi: 10.1109/MASSW.2019.00012.

[111] W. Brendel, J. Rauber, and M. Bethge, “Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models,” in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings (2018)*, Dec. 2017, pp. 1–12. [Online]. Available: http://arxiv.org/abs/1712.04248

[112] S. Shen, G. Jin, K. Gao, and Y. Zhang, “APE-GAN: Adversarial Perturbation Elimination with GAN,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings (2019)*, Jul. 2019, pp. 3842–3846. [Online]. Available: http://arxiv.org/abs/1707.05474

[113] W. Xu, D. Evans, and Y. Qi, “Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks,” in *Network and distributed systems security symposium*, 2018, pp. 1–16. doi: 10.14722/ndss.2018.23198.

[114] P. Samangouei, M. Kabkab, and R. Chellappa, “Defense-Gan: Protecting classifiers against adversarial attacks using generative models,” in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018, pp. 1–17.

PROOF OF PUBLICATIONS

SCIE Journal Paper 1:

❖ **A. Bajaj** and D. K. Vishwakarma, "HOMOCHAR: A novel adversarial attack framework for exposing the vulnerability of text based neural sentiment classifiers," ***Engineering Applications of Artificial Intelligence***, vol. 126, Nov. 2023, doi: 10.1016/j.engappai.2023.106815.

Engineering Applications of Artificial Intelligence 126 (2023) 106815

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

HOMOCHAR: A novel adversarial attack framework for exposing the vulnerability of text based neural sentiment classifiers

Ashish Bajaj, Dinesh Kumar Vishwakarma *

Biometric Research Laboratory, Department of Information Technology, Delhi Technological University, Bawana Road, Delhi 110042, India

ARTICLE INFO

Keywords:
Adversarial attack
Vulnerability
Transformers
Natural language processing) NLP
Sentiment classification

ABSTRACT

State-of-the-art deep learning algorithms have demonstrated remarkable proficiency in the task of text classification. Despite the widespread use of deep learning-based language models, there remains much work to be done in order to improve the security of these models. This is particularly concerning for their growing use in sensitive applications, such as sentiment analysis. This study demonstrates that language models possess inherent susceptibility to textual adversarial attacks, wherein a small number of words or characters are modified to produce an adversarial text that deceives the machine into producing erroneous predictions while maintaining its true meaning for human readers. The current study offers HOMOCHAR, a novel textual adversarial attack that operates within a black box setting. The proposed method generates more robust adversarial examples by considering the task of perturbing a text input with transformations at the character level. The objective is to deceive a target NLP model while adhering to specific linguistic constraints in a way such that the perturbations are imperceptible to humans. Comprehensive experiments are performed to assess the effectiveness of the proposed attack method against several popular models, including Word-CNN, Word-LSTM along with five powerful transformer models on two benchmark datasets, i.e., MR & IMDB utilized for sentiment analysis task. Empirical findings indicate that the proposed attack model consistently attains significantly greater attack success rates (ASR) and generates high-quality adversarial examples when compared to conventional methods. The results indicate that text-based sentiment prediction techniques can be circumvented, leading to potential consequences for existing policy measures.

SCIE Journal Paper 2:

❖ **A. Bajaj** and D. K. Vishwakarma, "A state-of-the-art review on adversarial machine learning in image classification," ***Multimedia Tools & Applications***, 2023, doi: 10.1007/s11042-023-15883-z.

Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-15883-z

Check for updates

A state-of-the-art review on adversarial machine learning in image classification

Ashish Bajaj[1] · Dinesh Kumar Vishwakarma[1]

Received: 26 November 2022 / Revised: 3 April 2023 / Accepted: 22 May 2023

Abstract
Computer vision applications like traffic monitoring, security checks, self-driving cars, medical imaging, etc., rely heavily on machine learning models. It raises an essential growing concern regarding the dependability of machine learning algorithms, which cannot be entirely trusted due to their fragile nature. This leads us to a dire need for systematic analysis of adversarial settings in neural networks. Hence, this article presents a comprehensive study of vulnerabilities, possible attacks such as data poisoning and data access during training, evasion, and oracle attacks at the test time, and their defensive and preventive measures using novel taxonomies. The survey has covered the complete scenario where an adversary can make malicious manipulations and elaborated more on the most potent threat, i.e., test time evasion attack using an adversarial image (maliciously perturbed image). It expounds an intuition behind generating an adversarial image, covering all relevant adversarial attack algorithms and strategies for increasing robustness against adversarial images. The existence and effect of adversarial images, as well as their transferability, are also examined. The article guides the reader with an approach on building new models to enhance their reliability. Additionally, the survey presents the procedures that still demand further exploration with limitations in existing methods, enhancing future research directions.

Keywords Attacks · Adversarial machine learning (AML) · Deep neural networks (DNNs) · Convolutional neural networks (CNN) · Defences · Data poisoning · Backdoor attacks · Robustness · Evasion · Brittle

SCIE Journal Paper 3:

❖ **A. Bajaj** and D. Kumar Vishwakarma, "Evading text-based emotion detection mechanism via adversarial attacks," ***Neurocomputing***, vol. 558, p. 126787, Nov. 2023, doi: 10.1016/j.neucom.2023.126787.

Neurocomputing 558 (2023) 126787

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Evading text based emotion detection mechanism via adversarial attacks

Ashish Bajaj, Dinesh Kumar Vishwakarma *

Biometric Research Laboratory, Department of Information Technology, Delhi Technological University, Bawana Road, Delhi 110042, India

ARTICLE INFO

Keywords:
Adversarial Attack
Textual Emotion Analysis (TEA)
Natural Language Processing (NLP)
Deep Learning (DL)
Vulnerability
Transformers Semantic
Similarity

ABSTRACT

Textual Emotion Analysis (TEA) seeks to extract and assess the emotional states of users from the text. Various Deep Learning (DL) algorithms have emerged rapidly and demonstrated success in numerous disciplines, including audio, image, and natural language processing. The trend has shifted a growing number of researchers from standard machine learning to DL for scientific study. Using DL approaches, we offer an overview of TEA in this paper. After introducing the background for emotion analysis, including the definition of emotion, emotion classification methods, and application domains of emotion analysis, we demonstrated that, despite the immense success of deep learning models in NLP-related tasks, they are susceptible to adversarial attacks, which can lead to incorrect emotion classification. An adversarial text is constructed by altering a few words or characters so as to keep the overall semantic similarity of emotion for a human reader while tricking the machine into making erroneous predictions. This study demonstrates the vulnerability of emotion categorization by generating adversarial text using a variety of cutting-edge attack techniques. Comprehensive experiments are performed to assess the effectiveness of the attack methods against several widely-used models, such as Word-CNN, Bi-LSTM, and four powerful transformer models, namely BERT, DistilBERT, ALBERT, and RoBERTa. These models were trained on an emotion dataset utilized for the purpose of emotion classification. We evaluated and analyzed the behavior of different models under a variety of attack conditions to determine which is the most and least vulnerable. Also, we determine which perturbation technique affects transformer models the most. Using Attack Success Rates (ASR) as our evaluation metric, we have assessed the potential outcomes. The findings reveal that methodologies for classifying emotion prediction can be circumvented, which has implications for existing policy measures.

SCIE Journal Paper 4:

❖ **A. Bajaj** and D. K. Vishwakarma, "Non-Alpha-Num: a novel architecture for generating adversarial examples for bypassing NLP-based clickbait detection mechanisms," ***International Journal of Information Security***, 2024, doi: 10.1007/s10207-024-00861-9.

International Journal of Information Security
https://doi.org/10.1007/s10207-024-00861-9

REGULAR CONTRIBUTION

Non-Alpha-Num: a novel architecture for generating adversarial examples for bypassing NLP-based clickbait detection mechanisms

Ashish Bajaj[1] · Dinesh Kumar Vishwakarma[1]

Accepted: 23 April 2024

Abstract
The vast majority of online media rely heavily on the revenues generated by their readers' views, and due to the abundance of such outlets, they must compete for reader attention. It is a common practise for publishers to employ attention-grabbing headlines as a means to entice users to visit their websites. These headlines, commonly referred to as clickbaits, strategically leverage the curiosity gap experienced by users, enticing them to click on hyperlinks that frequently fail to meet their expectations. Therefore, the identification of clickbaits is a significant NLP application. Previous studies have demonstrated that language models can effectively detect clickbaits. Deep learning models have attained great success in text-based assignments, but these are vulnerable to adversarial modifications. These attacks involve making undetectable alterations to a small number of words or characters in order to create a deceptive text that misleads the machine into making incorrect predictions. The present work introduces "*Non-Alpha-Num*", a newly proposed textual adversarial assault that functions in a black box setting, operating at the character level. The primary goal is to manipulate a certain NLP model in a manner that the alterations made to the input data are undetectable by human observers. A series of comprehensive tests were conducted to evaluate the efficacy of the suggested attack approach on several widely-used models, including Word-CNN, BERT, DistilBERT, ALBERTA, RoBERTa, and XLNet. These models were fine-tuned using the clickbait dataset, which is commonly employed for clickbait detection purposes. The empirical evidence suggests that the attack model being offered routinely achieves much higher attack success rates (ASR) and produces high-quality adversarial instances in comparison to traditional adversarial manipulations. The findings suggest that the clickbait detection system has the potential to be circumvented, which might have significant implications for current policy efforts.

Keywords Clickbait · Convolutional Neural Network · Transformer models · Adversarial Attacks · Deep Learning · Vulnerability

Conference Paper 1:

❖ **A. Bajaj** and D. K. Vishwakarma, "ARG-Net: Adversarial Robust Generalized Network to Defend Against Word-Level Textual Adversarial Attacks," in *2024 IEEE **9th International Conference for Convergence in Technology** (I2CT)*, Institute of Electrical and Electronics Engineers (IEEE), Jun. 2024, pp. 1–7. doi: 10.1109/i2ct61223.2024.10543623.

ARG-Net: Adversarial Robust Generalized Network to Defend Against Word-Level Textual Adversarial Attacks

2024 IEEE 9th International Conference for Convergence in Technology (I2CT) | 979-8-3503-9447-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/I2CT61223.2024.10543623

Ashish Bajaj
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
bajaj.ashish25@gmail.com

Dinesh Kumar Vishwakarma
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
dvishwakarma@gmail.com

***Abstract*—Natural Language Processing models have strong performance across various applications, although they are susceptible to manipulation by adversarial instances. A minor disturbance has the potential to alter the outcome of the deep learning algorithm. Humans find it difficult to detect this type of disturbance, particularly adversarial instances created using word-level adversarial attacks. Char-level adversarial assault can be countered using grammar detection and word recognition. The current word-level textual adversarial attacks rely on the substitution of synonyms, resulting in perturbed text that often maintains proper syntax and semantics. Defending against adversarial attacks at the word level poses more challenges. This study introduces a novel system called Adversarial Robust Generalized Network (ARG-Net) that aims to protect against word-level adversarial assaults. ARG-Net improves the model's performance by using both adversarial training and data perturbation techniques during the training process. The results of our tests on two datasets demonstrate that the model, which is built upon our framework, successfully mitigates word-level adversarial assaults. The defense success rate of the model trained using ARG-Net is greater than that of the previous defense approaches when tested on 1000 adversarial samples. Furthermore, our model exhibits superior accuracy on the standard testing set compared to current defense techniques. The accuracy is comparable to, or even surpasses, that of the conventional model.**

Keywords*—Adversarial Machine Learning, Adversarial Training, Robustness, Deep Learning, Transformers, Natural Language Processing (NLP).

I. Introduction

The field of text categorization has shown notable progress by employing neural network architectures[1]. Neural Networkshas exhibited remarkable results in several text-processing assignments, such as sentiment analysis, machine translation & relation extraction. However, recent studies have revealed that the introduction of slight modifications to test inputs might potentially mislead sophisticated deep classifiers, resulting in erroneous categorizations. The phenomena were first introduced by incorporating subtle and often imperceptible disturbances into images, leading to the capability of misleading deep classifiers within the domain of image classification tasks. The issue of the robustness of deep learning systems has been a matter of significant interest particularly in light of their extensive utilization in security-sensitive domains. The classification of adversarialexamples in textual domain is shown in **Figure 1**.

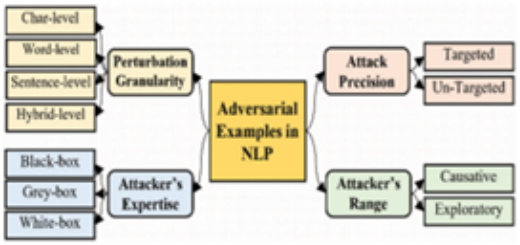


Fig. 1. Taxonomy of *Adversarial Examples in Text Domain*

Several scientific inquiries have been conducted to examine the security ramifications of existing neural network structures[2]. These research studies have suggested a wide range of attack methods, including both causative assaults and exploratory attacks. The concept of "$\boldsymbol{causative\ attacks}$" pertains to intentional alterations applied to training samples to mislead the machine learning algorithm. Conversely, "$\boldsymbol{exploratoryattacks}$" assaults involve generating adverse test cases, also known as adversarial instances, with the explicit goal of evading a given classifier[3]. **Figure 1**illustrates the classification of the categorization of adversarial cases in the field of text processing based on several characteristics. Adversarial instances are generated within the context of a dualistic situation. The term "$\boldsymbol{black\text{-}box\ situation}$" refers to a particular scenario in which an opponent has no knowledge of the classifier or the training dataset when an instance is generated. Conversely, a "$\boldsymbol{white\text{-}box}$" arrangement denotes a situation when the attacker possesses a comprehensive understanding of both the machine learning algorithm and the initial training dataset[4]. The above examples are carefully crafted to maintain semantic accuracy for humans while deliberately deceiving the algorithm to provide inaccurate results. Furthermore, it is possible to classify the attack according to its intended target or its indiscriminate. Within the framework of a "$\boldsymbol{targeted}$" assault, the input data $\boldsymbol{x}$ is subjected to a modification process, yielding $\boldsymbol{x_{adv}}$. This modified data is subsequently employed to predict a particular class $\boldsymbol{C_T}$, deviating from its initial classification of $\boldsymbol{C_i}$. The objective of this work is to attain a pre-established goal designation. In an assault that is characterized as "$\boldsymbol{untargeted}$," the primary objective is to change the input variable $\boldsymbol{x}$ in a way that causes it to depart from its original class[5], regardless of which alternative class label it may be assigned to.

Conference Paper 2:

❖ **A. Bajaj** and D. K. Vishwakarma, "Deceiving Deep Learning-based Fraud SMS Detection Models through Adversarial Attacks," in *Proceedings* - ***17th International Conference on Signal-Image Technology and Internet-Based Systems***, *SITIS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 327–332. doi: 10.1109/SITIS61268.2023.00059.

Deceiving Deep Learning-based Fraud SMS Detection Models through Adversarial Attacks

2023 17th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) | 979-8-3503-7091-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/SITIS61268.2023.00059

Ashish Bajaj
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
bajaj.ashish25@gmail.com

Dinesh Kumar Vishwakarma
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
dvishwakarma@gmail.com

Abstract— **Short Messaging Service (SMS) is one of the most extensively utilized mobile applications for communication around the world. Smishing is the technique of delivering harmful SMS to users in which intruders send malicious SMS to the victim. This content may contain links that direct the user to websites that contain harmful software and user interfaces. Researchers have acquired outstanding accuracy scores in proposing SMS spam detectors utilizing transformer-based deep learning algorithms. Despite their superior performance in Natural Language Processing-related tasks, deep learning models are vulnerable to adversarial attacks that result in misclassification. A few words or characters are altered to create adversarial text, fooling the machine into making inaccurate predictions. This research aims to analyze the security weaknesses of the smishing detection method by employing advanced attack techniques to generate adversarial text. In this study, we conducted a comparative analysis of various transformer models, including BERT, DistilBERT and RoBERTa. These models were trained on the SMS spam dataset, which is often used for detecting fraudulent SMS messages. Through our analysis, we examined and evaluated the behavior of these models to know which model is more vulnerable or which is more robust. The prospective outcomes have been assessed by the computation of Attack Success Rates (ASR) for each model. The findings indicate the feasibility of circumventing automated spam SMS detection systems, hence highlighting potential implications for existing regulatory interventions.**

Keywords— Textual Adversarial Attacks, Smishing Detection, Transformers, Natural Language Processing (NLP), Vulnerability.

I. Introduction

The discipline of Natural Language Processing has witnessed significant advancements through the utilization of deep learning methodologies. Notably, deep learning has demonstrated exceptional outcomes in several Natural Language Processing (NLP) encompasses several tasks, including relation extraction, sentiment analysis, and machine translation. Nevertheless, recent research has indicated that the incorporation of minor alterations to test inputs has the potential to deceive advanced deep classifiers, leading to inaccurate classifications. The initial formulation of this phenomenon was the introduction of minute and frequently undetectable perturbations onto pictures[1], resulting in the ability to deceive deep classifiers in the context of image classification tasks. The resilience of deep learning systems is a subject of concern, particularly due to their widespread use in security-sensitive applications like text-based spam detection.

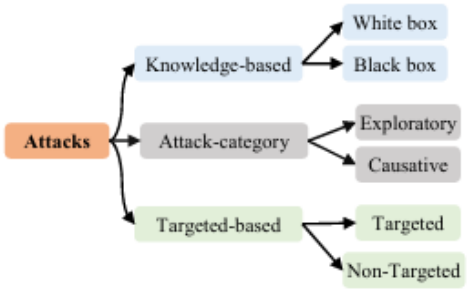


Figure 1 *Taxonomy of Adversarial Attack*

Numerous scholarly inquiries have been conducted to examine the security aspects of existing neural network models, wherein various assault techniques have been suggested, including causative attacks as well as exploratory attacks. "***Causative attacks"*** are designed to modify the training data in order to deceive the classifier, whereas "***exploratory"*** assaults provide harmful testing cases, known as adversarial examples, with the intention of evading a certain classifier. This research exclusively focuses on the analysis and examination of assaults carried out through the utilization of adversarial instances[2]. Adversarial instances are developed within the context of two adversarial scenarios. The term "***black-box situation***" refers to the scenario in which an example is generated without the adversary having any knowledge about the classifier or the training collection. Conversely, a "***white-box***" configuration exists whereby the adversary possesses a comprehensive understanding of both the classifier and the training data. The examples are constructed in a manner that maintains semantic significance for human readers while deceiving the computer into producing erroneous predictions[3]. In addition, the attack may be categorized based on whether it is directed against a specific target or if it is indiscriminate in nature as shown in **Figure 1**. In the context of a "***targeted***" assault, the input data $\boldsymbol{x}$ undergoes a modification resulting in $\boldsymbol{x_{adv}}$, which is then used to forecast a specific class $\boldsymbol{C_T}$ instead of its original class $\boldsymbol{C_t}$. The aim of this task is to achieve a predetermined goal label. In an "***untargeted***" assault, the primary goal is to manipulate the input variable $\boldsymbol{x}$ in such a

Conference Paper 3:

❖ **A. Bajaj** and D. K. Vishwakarma, "Bypassing Deep Learning based Sentiment Analysis from Business Reviews," in *2023* ***2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies*** *(ViTECoN)*, IEEE, May 2023, pp. 1–6. doi: 10.1109/ViTECoN58111.2023.10157098.

Bypassing Deep Learning based Sentiment Analysis from Business Reviews

Ashish Bajaj
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
bajaj.ashish25@gmail.com

Dinesh Kumar Vishwakarma
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
dvishwakarma@gmail.com

2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN) | 979-8-3503-4798-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ViTECoN58111.2023.10157098

Abstract— **In recent years, online reviews of businesses have grown increasingly significant, as customers and even competitors use them to evaluate a company's quality. Yelp is one of the most popular review websites, and it would be advantageous for them to be capable of predicting the sentiment or even the star rating of a review. Current deep-learning algorithms excel at sentiment classification. With the tremendous performance of models based on deep learning in text-related problems, they are susceptible to adversarial manipulations that result in inaccurate sentiment classification. An adversarial text is created by manipulating just few letters or words in such a manner so that general meaning of the text remains unchanged for humans but fooling a system into making false predictions. This study highlights the shortcomings of sentiment categorization by employing a range of cutting-edge attack techniques to generate perturbed text. We examined the performance of several models, including BERT, an advanced transformer model, and the extensively used LSTM and Word-CNN classifiers trained on the Yelp polarity dataset. For each model, Attack Success Rates (ASR) are calculated as the evaluation metric. Based on the experimental results, we determined which sentiment classifier is more vulnerable to adversarial perturbations and which is more resistant. The results demonstrate that automatic sentiment classification techniques can be circumvented, which has implications for present policy approaches.**

Keywords— (Natural Language Processing) NLP, Sentiment Classification, Adversarial Attack, Transformers, Semantic Similarity, Vulnerability.

I. Introduction

Across the past decade, Machine Learning (ML) approaches have flourished at a range of tasks, including regression, classification, and decision processing. Yet, these models are fragile to adversarial situations, which are genuine inputs that have been intentionally modified by minute, frequently imperceptible variations. Emerging research has generated adversarial perturbed images which render algorithms for computer vision ineffective[1]. A few research on adversarial cases in text-categorization problems have been undertaken, such as emotional analysis, topic classification[2], machine translation, fake news classification, hate content detection, etc. Yet, due to the adversarial machine learning's achievement in visuals, it is a relatively recent topic that has received more attention and is interesting to examine [3]. To produce adversarial cases, two hostile situations are used. A *"black-box setup"* is the design of adversarial perturbed sample in which an attacker is clueless of classification algorithm or the training information. In contrast, in a *"white-box"* situation, the latter one has detailed understanding of the algorithm and the training set [4]. In addition, the attack is also classified according to whether it is targeted or untargeted. Consider the input class to be C_i and the output class to be C_T. The input x belongs to the class C_i, and we desire that, after perturbation, x' set belongs to a class other than C_i. In a *"targeted"* assault, the input data x is altered to x' so that it predicts a targeted class C_T rather than its genuine class C_i. Here, the objective is to reach a specific target label. In an *"untargeted"* attack, the objective is to shift input x away from its true class C_i, regardless of which other classes are struck [5]. Technically, in virtue to fooling the target models, the outcomes of a natural language parsing framework must fulfill 3 key utility-preserving functionalities :***1.)*** same resemblance in meaning—the produced example should have the same significance as the actual based on human judgement; ***2.)*** created instances of opposition shall seem grammatical and genuine. ***3.)*** Human predictions should be consistent and remain constant[6]. **Figure 1** demonstrates how assaults are classified depending on attack specificity and attacker knowledge.

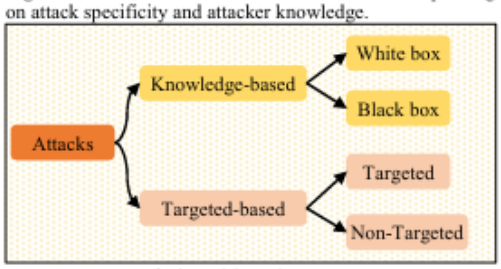


Figure 1. Taxonomy of Adversarial Attack.

This research focuses on the widely used word convolution neural-networks and bi-directional long short-term memory, along with the potent transformer model, that is, BERT, for various natural language processing tasks, to illustrate the weakness in sentiment classification. Firstly, the classifiers are trained on the (Yelp polarity dataset), a well-known set of business review emotions. The deterioration of these pre-trained models' performance is then examined by conducting attacks utilizing various cutting-edge adversarial attack methodologies. The findings may be of interest to users who habitually use well-known cutting-edge classification methods. The reader will be able to choose which model is

Conference Paper 4:

❖ **A. Bajaj** and D. K. Vishwakarma, "Exposing the Vulnerabilities of Deep Learning Models in News Classification," in *2023 4th* ***International Conference on Innovative Trends in Information Technology*** *(ICITIIT)*, IEEE, Feb. 2023, pp. 1–5. doi: 10.1109/ICITIIT57246.2023.10068577

Exposing the Vulnerabilities of Deep Learning Models in News Classification

Ashish Bajaj
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
bajaj.ashish25@gmail.com

Dinesh Kumar Vishwakarma
Biometric Research Laboratory, Department of Information Technology, Delhi Technological University
Bawana Road, Delhi-110042, India
dvishwakarma@gmail.com

2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT) | 978-1-6654-9414-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICITIIT57246.2023.10068577

Abstract—**News websites need to divide their articles into categories that make it easier for readers to find news of their interest. Recent deep-learning models have excelled in this news classification task. Despite the tremendous success of deep learning models in NLP-related tasks, it is vulnerable to adversarial attacks, which lead to misclassification of the news category. An adversarial text is generated by changing a few words or characters in a way that retains the overall semantic similarity of news for a human reader but deceives the machine into giving inaccurate predictions. This paper presents the vulnerability in news classification by generating adversarial text using various state-of-the-art attack algorithms. We have compared and analyzed the behavior of different models, including the powerful transformer model, BERT, and the widely used Word-CNN and LSTM models trained on AG news classification dataset. We have evaluated the potential results by calculating Attack Success Rates (ASR) for each model. The results show that it is possible to automatically bypass News topic classification mechanisms, resulting in repercussions for current policy measures**.

Keywords— Adversarial Attack, News Classification, (Natural Language Processing) NLP, Semantic Similarity, Vulnerability, Transformers.

I. Introduction

Machine Learning (ML) models have excelled in various tasks during the past ten years, including classification, regression, and decision-making. Though, it has been discovered that these models are susceptible to adversarial examples, which are actual inputs modified by tiny, frequently undetectable perturbations, as shown in Figure 1. Recent studies have successfully generated adversarial images[1] that render computer vision algorithms useless. There are few studies of adversarial instances in natural language processing applications like topic classification, sentiment analysis, fake news detection, hate content detection, machine translation, etc. Nevertheless, it is a newer topic that is interesting to investigate and has recently received more attention due to the success of adversarial learning in images. Adversarial examples are generated under two adversarial settings. One is a "black-box setting," i.e., creating an adversarial example when the adversary is unaware of the classifier or the training set. On the other hand, there is a white-box setup in which the adversary has complete knowledge of the classifier and the training data. Formally, outputs of a natural language assaulting system should also satisfy three important utility-preserving features in addition to their capacity to deceive the target models:1.) semantic similarity—as determined by humans, the constructed example should have the same meaning as the source, 2.) Adversarial examples generated should appear natural and grammatical, 3.) Consistency of human prediction—human predictions should not change.

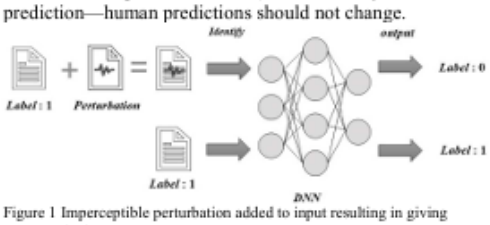


Figure 1 Imperceptible perturbation added to input resulting in giving wrong output

In this work, we present the vulnerability in news topic classification by targeting the widely used long short-term memory and convolution neural networks, including the most potent transformer model, i.e., BERT, for text classification. The models are first trained on the (AG news dataset) a popular dataset for news topic classification. These pre-trained models are then evaluated on their performance degradation by conducting attacks using various *state-of-the-art* adversarial attack algorithms. The results are of potential interest to users who frequently use famous *state-of-the-art* models for their classification tasks. The reader will be able to find out the best fit model for their problem. Also, this motivates the researchers to build models with adversarial robust generalizations instead of standard generalizations. The authors claim that this is the first work raised in the literature that has shown a comparative analysis of the vulnerability of different models on various adversarial attack algorithms for the news classification task.

II. Related Work

A. News Topic Classification

With the rise in the usage of social media applications, the user usually gathers important news from social media sources. Users often seek interest in reading news articles related to them. Based on their interest, the google recommendation system suggests news articles to their users for their benefit. Before the news articles are recommended, the article is first classified into various categories, i.e., sports, entertainment, technical, business, etc. Since news is of multiple types, this is considered a multi-class classification. The classical ML algorithms, such as Naïve

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis: Design of Framework for Adversarial Attacks & Defences in Classification Models

Total Pages: 175

Name of the Scholar: Ashish Bajaj

Supervisor: Prof. Dinesh Kumar Vishwakarma

Department: Information Technology

This is to report that the above thesis was scanned for similarity detection. Process and outcome are given below:

Software used: Turnitin Similarity Index: 06% Word Count: 50900 Words

Date: 30/07/2024

Candidate's Signature **Signature of Supervisor**

PLAGIARISM REPORT

Similarity Report

PAPER NAME

Final Thesis Ashish Bajaj.docx

WORD COUNT	CHARACTER COUNT
50900 Words	**295246 Characters**
PAGE COUNT	FILE SIZE
175 Pages	**20.5MB**
SUBMISSION DATE	REPORT DATE
Jul 30, 2024 12:40 PM GMT+5:30	**Jul 30, 2024 12:43 PM GMT+5:30**

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 3% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 10 words)
- Manually excluded sources

Author Biography

Ashish Bajaj received Bachelor of Technology in Computer Science & Engineering degree in 2019 and Master of Technology in Information Technology degree in 2021 from Guru Gobind Singh Indraprastha University, Delhi, India. He is a senior research scholar at the Department of Information Technology, Delhi Technological University, Delhi. The topic of his doctoral dissertation is Design of Framework for Adversarial Attacks & Defences in Classification Models. He is primarily interested in deep-learning-based Natural Language Processing tasks, such as Textual Sentiment Analysis, Clickbait Detection, Smishing, and Phishing Detection. The primary objective is to identify weaknesses in the models and develop novel adversarial attack and defence frameworks to obtain robust generalizations in adversarial scenarios.