

**NATURAL LANGUAGE PROCESSING FOR ADVERSARIAL ATTACK DETECTION IN AI TRAINING DATASET**

BY

NKANSAH KWADWO EDWARD

THOMPSON JUNIOR WILLIAMS

AKWASI NYAMEKYE KUSI

ASAFO ADJEI EMMANUEL

OWUSU RICHARD

A PROJECT WORK SUBMITTED TO THE DEPARTMENT OF INFORMATION TECHNOLOGY AND DECISION SCIENCES,

UNIVERSITY OF ENERGY AND NATURAL RESOURCES,

IN PARTIAL FULFILMENT OF THE REQUIREMENT OF THE DEGREE OF

BACHELOR OF SCIENCE

IN

INFORMATION TECHNOLOGY,

SCHOOL OF SCIENCE

SEPTEMBER, 2024

**DECLARATION**

I, Nkansah Kwadwo Edward, Thompson Junior Williams, Akwasi Nyamekye Kusi, Asafo Adjei Emmanuel, Owusu Richard declare that the submission is our own work towards the award of a BSc. Information Technology. And that to the best of our knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgment has been made to the text.

NKANSAH KWADWO EDWARD ----------------------- -----------------------

(UEB3210420) DATE SIGNATURE

THOMPSON JUNIOR WILLIAMS ----------------------- -----------------------

(UEB3267822) DATE SIGNATURE

AKWASI NYAMEKYE KUSI ----------------------- -----------------------

(UEB3209620) DATE SIGNATURE

ASAFO ADJEI EMMANUEL ----------------------- -----------------------

(UEB3212420) DATE SIGNATURE

OWUSU RICHARD ----------------------- -----------------------

(UEB3208320) DATE SIGNATURE

PROF. PETER APPIAHENE ----------------------- -----------------------

(SUPERVISOR) DATE SIGNATURE

PROF. PETER APPIAHENE ----------------------- -----------------------

(HEAD OF DEPARTMENT) DATE SIGNATURE

**ACKNOWLEDGEMENT**

We would like to express profound gratitude to God for the strength, resilience, and clarity of mind that have seen us through this task. Above and beyond all, we are grateful to our supervisor, Prof. Peter Appiahene, for his enormous guidance, sustained support, and valuable expertise during the course of this research. His insightful feedback and mentorship were crucial to this project's success. In addition, we would like to extend our greatest appreciation to the Department of Information Technology and Decision Sciences, UENR, for the general support and help received throughout the tenure of this project. In fact, it was the guidance by the faculty, encouragement, and provision of necessary resources that enhanced a smooth completion of this research study.

We are also grateful to Mr. Arthur Enoch Justice, Senior Research Assistant, Department of Information Technology and Decision Sciences, UENR, for his enormous contribution and technical advice on the project. We would finally like to express our deep gratitude to our families, who had been so constant with encouragement, patience, and support; without the latter, this research could never have been satisfactorily concluded.

**ABSTRACT**

NLP has nowadays become one of the main technologies in the digital world. It helps enhance communication and is mainly applied for system automatization. But together with big advantages, the NLP models suffer increasingly from adversarial attacks-such are smart manipulations that influence model predictions and raise serious questions about the reliability and effectiveness of these systems. The presented work aims at certain of these risks by building a framework for improved security of NLP models in detail. Advanced defense algorithms are proposed to reduce the impact of such adversarial attacks by up to 50%. This will also make the applications of NLP more strong in different domains. We will combine some approaches, such as ensemble learning, with the anomaly detection techniques, including Local Outlier Factor and Isolation Forest, all set up within a Python-based environment. Early testing showed that around 12% of the data was flagged as adversarial, strengthening the security of NLP models. In the end, the work presented here provides a necessary foundation for future research to be done on protecting AI from malicious attacks so that these systems remain reliable and trustworthy, especially when used in critical applications.

Keywords: NLP, Adversarial Attack, Anomaly Detection, Ensemble Methods, Security in AI, Defense Machine Learning, Python, Isolation Forest, Local Outlier Factor.

**Table of Contents**

[CHAPTER ONE 10](#_Toc178763759)

[1.0 Introduction 10](#_Toc178763760)

[1.1 Background of the Study 10](#_Toc178763761)

[1.2 Problem Statement 11](#_Toc178763762)

[1.3 Main Objective 12](#_Toc178763763)

[1.3.1 Specific Objectives 12](#_Toc178763764)

[1.4 Research Questions 12](#_Toc178763765)

[1.5. Significance of the study 13](#_Toc178763766)

[1.5.1 Theoretical Significance 13](#_Toc178763767)

[1.5.2 Practical Significance 14](#_Toc178763768)

[1.6 Structure of Subsequent Chapters. 14](#_Toc178763769)

[CHAPTER TWO (Literature Review) 16](#_Toc178763770)

[2.0 Introduction 16](#_Toc178763771)

[2.1 Definition of Concepts 16](#_Toc178763772)

[2.1.1 Natural Language Processing 16](#_Toc178763773)

[2.1.2 Adversarial Attacks 16](#_Toc178763774)

[2.1.3 Adversarial Attacks in Natural Language Processing (NLP) 17](#_Toc178763775)

[2.2.0 Existing Detection Methods for Adversarial Attacks 21](#_Toc178763776)

[2.2.1 Traditional Detection Techniques 21](#_Toc178763777)

[2.2.2 Machine Learning-Based Detection 22](#_Toc178763778)

[2.3. Local Outlier Factor (LOF) Algorithm 23](#_Toc178763779)

[2.3.1 Principles of Local Outlier Factor 23](#_Toc178763780)

[2.3.2 How LOF Identifies Outliers: 24](#_Toc178763781)

[2.3.3 Applications of LOF in Different Domains 24](#_Toc178763782)

[2.4 Isolation Forest (IF) Algorithm 25](#_Toc178763783)

[2.4.1 Principles of IF: 25](#_Toc178763784)

[2.4.2 Applicability to Adversarial Attack Detection 26](#_Toc178763785)

[2.5 Review of Related works 26](#_Toc178763786)

[2.6 Summary of Related Works/Systems 30](#_Toc178763787)

[2.7 Summary of Literature Review 34](#_Toc178763788)

[CHAPTER THREE (Proposed System Design and Methodology) 35](#_Toc178763789)

[3.0 Introduction 35](#_Toc178763790)

[3.1 Research Design 35](#_Toc178763791)

[3.1.1 Research Philosophy 35](#_Toc178763792)

[3.1.2 Research Type 36](#_Toc178763793)

[3.1.3 Research Strategy 37](#_Toc178763794)

[3.1.4. Data Collection Methods 38](#_Toc178763795)

[3.1.5 TF-IDF (Term Frequency-Inverse Document Frequency): 38](#_Toc178763796)

[3.3 System Components 40](#_Toc178763797)

[3.3.1 Front End 40](#_Toc178763798)

[3.3.2 Back End 41](#_Toc178763799)

[3.3.3 Algorithms 41](#_Toc178763800)

[3.4 Operational Methods (Basic Logic) 43](#_Toc178763801)

[3.5 Methodological Limitations 43](#_Toc178763802)

[3.6 Summary of Methodology 44](#_Toc178763803)

[CHAPTER 4 (Analysis and Results) 45](#_Toc178763804)

[4.0 Introduction 45](#_Toc178763805)

[4.1 Dataset Used 45](#_Toc178763806)

[4.2 Adversarial Attack Detection 46](#_Toc178763807)

[4.2.1 Combined Detection Approach 47](#_Toc178763808)

[4.2.2 Detection Results 48](#_Toc178763809)

[4.3 Performance Metrics 49](#_Toc178763810)

[4.3.1 Accuracy 50](#_Toc178763811)

[4.3.2 Precision 51](#_Toc178763812)

[4.3.3 F1 Score 51](#_Toc178763813)

[4.3.4 Interpretation of Results 52](#_Toc178763814)

[4.4 Using the web app 52](#_Toc178763815)

[CHAPTER 5 (Discussion and Conclusion) 54](#_Toc178763816)

[5.0 Introduction 54](#_Toc178763817)

[5.1 Discussion of Results 54](#_Toc178763818)

[5.2 Future Works/Recommendation 55](#_Toc178763819)

[5.3 Conclusion 56](#_Toc178763820)

**Table of Figures**

[Figure 1 System Architecture 40](#_Toc178366446)

[Figure 2 Operational Flow of the System 43](#_Toc178366447)

[Figure 3 Adversarial attack detection results 48](#_Toc178366448)

[Figure 4 Heat map of Evaluation metrics of Local Outlier Factor and Isolation Forest 49](#_Toc178366449)

[Figure 5 Accuracy, Precision and F1 Score of IF and LOF 50](#_Toc178366450)

[Figure 6 Using The Web App 53](#_Toc178366451)

# **CHAPTER ONE**

## **Introduction**

This section contains, the background, problem statement, objectives, and organisation of this study.

## **1.1 Background of the Study**

During the last couple of years, it had been viewed that NLP was one of those cornerstone technologies that enabled computers and machines to understand, interpret, and generate human language. According to Shah (2020), the revolutionized NLP models have swept virtual assistants, language translation, sentiment analysis, and content creation among several applications. The wide adoptions brought them into the territory of vulnerability against adversarial attacks mounting serious threats to their reliability, security, and trustworthiness (Ibitoye et al., 2019 & Khurana et al., 2023).

AI and ML can change everything, from health care to finance. These systems learn from data with intelligent decisions. The quality and integrity of the training datasets are perhaps the most critical link in the chain of developing these AI models. Natural language processing, as a subbranch of AI, deals with interaction between computers and natural human language ranging from chatbots to sentiment analysis to machine translation.

Goyal et al., (2023) defined Adversarial attacks in NLP as intelligently crafted perturbations added to input text data with the intention of misleading or manipulating NLP models' predictions.

Prevalence of adversarial attacks in NLP underlines an urgent need in robust mechanisms of defense or strategy for detection to protect NLP models from malignant manipulation (Wang et al., 2019; Randaliev & Santos, 2023).

Generally speaking, the adversarial defense strategy in NLP can be divided into three main categories: adversarial training-based methods, perturbation control-based methods, and certification-based methods. These attacks can compromise the integrity of training datasets, leading to degraded model performance and mistrust in AI systems. Most research in this area makes use of adversarial training, while techniques are then further divided based on how adversarial instances or noise will be generated in the course of the defense process. Methods in the advantages offered include data augmentation-based adversarial training, adversarial training used as regularization technique, GAN-based adversarial training, VAT, and HITL approaches. The identification and mitigation of adversarial threats would add value by strengthening the security, dependability, and trustworthiness of NLP models for wider adoptions and deployments in real-world applications.

## **1.2 Problem Statement**

The vulnerability of deep learning-based models in natural language processing to adversarial attacks involves critical challenges related to their reliability and security (Alshemali & Kalita, 2021). This, therefore, calls for a dire need for effective defense mechanisms and detection strategies to be designed and deployed so as to safeguard the NLP systems against malicious manipulations and deceptions, hence the robustness and trustworthiness of the systems in real-world applications.

Current methods of adversarial attack detection suffer from either low detection accuracy or low counterfactual efficiency, thus mostly failing to protect NLP systems against manipulation and exploitation (Qiu et al., 2019). Besides, because of the dynamic nature of adversarial attacks, how to develop robust and resilient defense mechanisms has been a constant challenge to researchers and practitioners in the domain of NLP (Zhang et al., 2020; Omar et al., 2022).

## **1.3 Main Objective**

The general objective of this project is to enhance the security and robustness of natural language processing (NLP) models against adversarial attacks through the development of effective detection algorithms and defense mechanisms.

### **1.3.1 Specific Objectives**

The specific objectives are to:

1. Investigate the underlying principles and mechanisms of adversarial attacks in natural language processing (NLP) models.
2. Design and develop a user-friendly web interface for interacting with the NLP model and accessing detection results.
3. Comparatively detect adversarial attacks in AI dataset and evaluate the effectiveness and performance of the proposed detection algorithms using benchmark datasets and standardized evaluation metrics.
4. Deploy the developed detection algorithms integrated into web interface.

## **1.4 Research Questions**

This study aims to address the following research questions:

1. What are the underlying principles and mechanisms of adversarial attacks in NLP models, and how can these be effectively detected using Local Outlier Factor (LOF) and Isolation Forest (IF) algorithms?
2. How effective are the Local Outlier Factor (LOF) and Isolation Forest (IF) algorithms in detecting adversarial attacks in AI datasets, and how do they perform relative to each other and existing methods when evaluated using benchmark datasets and standardized metrics?
3. What are the essential features and challenges in designing and deploying a user-friendly web interface that integrates the developed detection algorithms for practical use, and how can the system be optimized for accessibility, scalability, and continuous improvement?

## **1.5. Significance of the study**

### **1.5.1 Theoretical Significance**

This research contributes to the ever-growing knowledge base relating to adversarial attack detection, particularly for NLP datasets. With a narrow focus on LOF and IF algorithms, this effort will tend to seek new vistas for amping up the mechanism of detection.

The combination of the two outlier detection algorithms is considered relatively new, and this investigation could present an insight into their performance regarding real-world data exploration. It may serve as a precinct for further academic exploration and refinement.

The research integrates LOF and IF machine learning techniques into NLP to bridge two very important areas of study in AI. This integration is important in that it opens fresh opportunities for cross-disciplinary applications, thus enriching the two fields.

The results of this study will help in the formulation of more advanced models using the benefits of NLP and algorithms to detect outliers to move the frontiers of AI one step forward.

This serves as a baseline for future research of AI model hardening and security. Other researchers could extend this work by applying other outlier detection methods, other kinds of adversarial attacks, or simply apply the methodology described herein for other areas besides NLP.

The methodological approach of this study, based on LOF and IF, and the metrics being used provide a perhaps valuable reference for future studies in the domain of adversarial attack detection by allowing consistency and comparability.

### **1.5.2 Practical Significance**

The most practical benefit that can be expected from this research is related to the enhancement in the robustness of the AI models applied in various applications. In essence, by the detection and elimination of adversarial attacks, the proposed system will ensure that the AI models also act reliably in cases when maliciously tampered data is available for representatives.

This is of utter importance in sensitive NLP applications, ranging from healthcare to finance and security, where the integrity of AI predictions can have huge real-world consequences.

## **1.6 Structure of Subsequent Chapters.**

This project is comprised of the following sections:

**Chapter 2:** The paper reviews the existing works on adversarial attack detection of NLP models. In this chapter, a critical review of related literature about the methodologies for detecting adversarial examples through traditional and machine learning-based approaches is discussed.

**Chapter 3:** This chapter describes the methods employed during the design and development stages of the project. It covers the implementation of LOF and Isolation Forest and the development of an intuitive web-based interface for adversarial attack detection.

**Chapter 4:** The chapter reflects on results from the design process. It has also given the results of applying the developed algorithms for detection on benchmark datasets, including a detailed analysis of performance and effectiveness regarding adversarial attack detection.

**Chapter 5:** This chapter summarizes the entire project with knowledge contribution by highlighting the importance of the findings, implications for further research, applications, and possible ways to improve in adversarial attack detection in NLP models.

# **CHAPTER TWO (Literature Review)**

## **Introduction**

This Chapter provide s a comprehensive literature review focusing on adversarial attacks and defense mechanisms specifically within the realm of Natural Language Processing (NLP). It explores the vulnerabilities of NLP models to adversarial manipulations, highlighting the unique challenges they face compared to other domains, such as image processing. Additionally, the chapter reviews existing detection and mitigation strategies, identifies gaps in current research, and underscores the necessity for novel approaches tailored to enhance the security and reliability of NLP systems against these attacks.

## **2.1 Definition of Concepts**

### **2.1.1 Natural Language Processing**

Natural Language Processing (NLP) is a sub field of artificial intelligence (AI) and linguistics that focuses on the interaction between computers and human (natural) languages. It involves the development of algorithms and models that enable computers to understand, interpret, generate, and respond to human language in a way that is both meaningful and useful. The goal of NLP is to bridge the gap between human communication and computer understanding, allowing for more natural and efficient interactions between people and machines.

### **2.1.2 Adversarial Attacks**

Adversarial attacks, a concept borne from the intersection of cybersecurity and artificial intelligence, constitute a formidable challenge in the realm of machine learning, particularly in the nuanced’ landscape of natural language processing (NLP). At its core, an adversarial attack entails the deliberate manipulation of input data with the intent to deceive or mislead a machine learning model, resulting in erroneous predictions or classifications (Pitropakis et al., 2019). While the phenomenon of adversarial attacks is not new, its emergence in NLP has introduced a new dimension of complexity and intrigue.

#### **2.1.2.1 Types of Adversarial Attacks:**

Depending on the motivation, adversarial attacks can be further categorized into two classes: targeted attacks and non-targeted attacks. In targeted attacks, the aim is misclassifying inputs to a specific class. In the case of non-targeted attacks, the attempts are to shift the decision boundary of the classifier to misclassify the inputs. Another important classification, depending on the access to the model's parameters, includes white-box and black-box attacks (Goya et al., 2023& Dey et al., 2023).

In fact, adversarial attacks for NLP would arguably be the most popular kind of evasion attack: crafting the inputs to be subtly altered in such a way as to cause misclassification while still retaining semantic coherence. On the other hand, poisoning attacks try to introduce integrity through malicious injection of data in the model training phase and thus affect its decision-making process. These attacks pose serious threats to the reliability and robustness of NLP models, shake trust in their predictions, and hamper real-world applications.

### **2.1.3 Adversarial Attacks in NLP**

Adversarial attacks in NLP come in many forms-perhaps most harmfully from hardly noticeable perturbations to blatant distortions in textual data. As observed by Goyal et al.,, "Adversarial attacks in NLP operate at character, word, or sentence levels with various proposals of perturbations via deletion, insertion, swapping, flipping, synonyms, concatenation with characters or words, and insertion of numeric or alphanumeric characters.". Adversarial attacks in NLP direct the vulnerabilities of machine learning models to small perturbations in the input data in such a way that the model makes a wrong prediction or classification. The various attacks can be differentiated regarding the level of granularity involved in the modifications: character level, word level, sentence level, and multi-level. Each category unique strategies is utilized to mislead NLP models while maintaining semantic coherence to human observers.

#### **2.1.3.1 Character Level Adversarial Attacks**

Character-level attacks are those where individual characters within words are changed to create perturbations that mislead NLP models. These simple kinds of attacks may easily impact model performance much because, by their nature, NLP models often rely heavily on the specific sequence of characters in words. Accordingly, some techniques and impacts include:

1. Character Swapping: Swapping in similar characters, like adjacent ones, disrupts word recognition, such as changing "model" to "moedl".
2. Homoglyph Substitution: One of the ways to get the models to misinterpret the text is to substitute similar looking homoglyphs for the characters, for example, "O" with "0", or "l" with "1".
3. Insertions and Deletions: Addition or removal of characters; for instance, "hello" being turned into "hhello" or "helo". Such input transformations may lead to differences in model predictions.

Character-level attacks decrease the model accuracy that is based on word embeddings or character-level representations. Most of them have been quite effective against spell-checking systems, sentiment analysis, and named entity recognition tasks in which character sequences are crucial.

#### **2.1.3.2 Word Level Adversarial Attacks**

Word-level adversarial attacks are variants inside the sentence that change words. They create adversarial examples which are semantically similar to the original text; however, they have different model output. Such an attack takes advantage of the given model's overdependence on specific words and their embeddings. Techniques and impacts include the following:

1. Replace synonyms: word change with synonyms, e.g., "happy" becomes "joyful," hence changing model prediction without touching sentence meaning.
2. Paraphrasing: The test involves the use of different words and phrases for the same meaning, such as "buy" becoming "purchase," in order to test the strength of the models against lexical variations.
3. Homophones: Substituting words with their phonetic equivalents, such as "two" with "too", to perplex the models that depend on text-to-speech or speech-to-text applications.
4. Word Insertion/Deletion: Inserting or deleting certain words-for example, "I am happy" can be rephrased as either "I am not happy" or "I happy"-modifies the semantic context.

Word-level attacks degrade the performance of models, including those used for sentiment analysis, machine translation, and text classification.

In particular, such attacks would highlight weaknesses in the performance of models on inputs with paraphrased or lexically varied content.

#### **2.1.3.3 Sentence Level Adversarial Attacks**

Sentence-level adversarial attacks are the kind of attacks that create adversarial examples by manipulating whole sentences or phrases in order to deceive the NLP model. They are thus able to modify the contextual setting or meaning of the input text in ways much more significant than the character or word-level attacks could do. Some of the techniques and their impacts include:

1. Sentence Paraphrasing: This is the rewriting of complete sentences through changing the structure while maintaining the meaning. For instance, a sentence "The cat sat on the mat" would be paraphrased to "The mat was sat on by the cat.".
2. Contextual alteration: changing the context within a sentence to make it come out differently; for example, "He said he didn't steal the money" compared to "He said, 'He didn't steal the money'".
3. Sentence insertion or deletion: changing the overall meaning of something by insertion or deletion of certain sentences, like adding "However, he lied" after a positive statement.
4. Negation: This would introduce negation to the sentence in order to give it the opposite meaning, such as "I liked the movie" versus "I did not like the movie.".

Models could be vulnerable to sentence-level attacks during complex NLP tasks such as sentiment analysis, document classification, and summarization.

Such attacks force the model to make a challenge on its contextual understanding, coherence, and semantic relationships within a text.

#### **2.1.3.4 Multi-Level Adversarial Attacks**

Multi-level attacks can then be described as character, word, and sentence-level perturbations combined into one to make more sophisticated adversarial examples not as easily detected. They leverage each level in such a way as to maximize the chances of deceiving NLP models. A few such techniques along with impacts include:

1. Combined Perturbations: Changing character, word and sentence in their own way within the same text-in other words, "Th3 cat sat oon the mat. But the dog did not bark" combines typos, word substitution, and sentence insertions.
2. Layered Attacks: Running multiple levels of attacks sequentially in building up the strengths, such as introducing typos first, followed by word replacement, and finally sentence editing.
3. Contextual and Lexical Variations: Combinations of contextual changes with lexical changes generate sophisticated adversarial examples semantically well-formed but hard for models to interpret correctly.

What makes such an attack even more effective is the multi-level approach it takes. These attacks are especially effective in uncovering NLP models' weaknesses dependent on either contextual embedding or deep learning architectures.

These include attacks testing the overall robustness of the model: noise, paraphrasing, and contextual shifts.

## **2.2.0 Existing Detection Methods for Adversarial Attacks**

Detection methods against adversarial attacks play an important role in the realm of NLP to ensure the integrity and reliability of AI systems. The overall detection methods can be segregated into two broad parts: traditional methods of detection, and those that are machine learning-based. Each category has its particular strategy to identify and mitigate these adversarial attacks.

### **2.2.1 Traditional Detection Techniques**

Conventional adversarial attack detection methods are statistical and heuristic methods that are not based on machine learning models. Probably, most of these approaches will have a theoretical basis within the analysis of text characteristics and patterns that could denote the presence of adversarial modifications.

**Statistical Methods:** Anomaly Detection Such a method would point to the data that significantly differs from the rest. In NLP applications, statistical tests might be conducted to find anomalies of texts by using frequency distributions of words, n-grams or other linguistic features.

For example, large deviations in lexical or character usage patterns may be quantified by chi-square test or z-scores to raise an adversarial attack.

**Language Models:** For instance, pre-trained language models such as n-gram models can be used to estimate the probability of sequences in texts. Text with low scores of probability may be marked as a potential adversarial example.

Example: To find out the probability of sequences of texts with the help of an n-gram language model and flag those which have improbably low scores.

**Heuristic Methods**: Rule-Based Detection This usually detects suspicious patterns or anomalies within the text based on predefined rules. The rules might be according to domain knowledge or common attack vectors, such as unexpected character substitutions or unusual word choices.

Example: a rule that flags text containing a high number of homograph substitutions, such as "0" for "O".

Editing Distance: The Levenshtein distance or edit distance between the original and modified text provides the presence of minor perturbations introduced by the adversarial attack.

For example, high edit distances between words in similar contexts could indicate an adversarial manipulation.

### **2.2.2 Machine Learning-Based Detection**

Machine learning-based detection methods refer to those approaches that use the power of advanced algorithms and models in order to detect adversarial attacks. These are typically composed of training models on labeled datasets that contain examples both normal and adversarial, so the training learns distinguishing features.

Advantages and Limitations

Advantages: The machine learning-based methods learn autonomously, displaying their capability to find complex patterns of data. Hence, they give better performance in higher accuracy and robustness against still-sophisticated attacks. If appropriately trained, they could generalize to unseen types of adversarial attacks also.

Limitations: This technique requires significant computational resources and big labeled datasets for training. They may also easily be subjected to overfitting and hence may need often to be retrained so that they adapt to new strategies of attack.

## **2.3. Local Outlier Factor (LOF) Algorithm**

The Local Outlier Factor (LOF) algorithm is a density-based method used to identify anomalies in data by comparing the local density of a data point to that of its neighbors. The central premise of LOF is that anomalies are data points that have a significantly lower density than their surrounding points (Foorthuis, 2021; Alimohammadi & Chen, 2022).

### **2.3.1 Principles of Local Outlier Factor**

Local Density Deviation: LOF measures the local density deviation of a given data point with respect to its neighbors. It computes the density around a point and compares it to the density around its neighbors.

Reachability Distance: The reachability distance is used to determine the density around a point. It is defined as the distance to the k-th nearest neighbor.

LOF Score: The LOF score quantifies how isolated a data point is compared to its neighbors. A higher LOF score indicates a higher likelihood of the point being an outlier.

### **2.3.2 How LOF Identifies Outliers:**

Neighborhood Comparison: LOF, for any given data point, calculates the average local density of its k-nearest neighbors and compares it with the density of the point itself.

Density Ratios: Points whose density is much smaller compared to its neighbors are assigned with higher LOF scores and considered as the potential outlier.

The threshold can be set to determine thresholds on LOF scores to label an anomalous point as an outlier, labeling data points between normal and anomaly.

### **2.3.3 Applications of LOF in Different Domains**

1. As such, LOF has been in wide applications over a variety of domains, including anomaly detection like fraud detection, network security, and industrial monitoring.
2. 1. Fraud Detection: LOF finds its application in detecting fraudulent financial transactional activity by spotting those transactions that are significantly far from normal patterns.
3. 2. Network Security: LOF finds applications in Network Intrusion Detection to identify abnormal network traffic patterns, possibly pointing out malicious activity.
4. 3. Industrial Monitoring: LOF is also applied in monitoring equipment and machinery for any abnormal deviation that may show failure has occurred or maintenance is required.

#### **2.3.3.1 Applicability to Natural Language Processing:**

1. Text Anomaly Detection: LOF can be used for text anomaly detection in NLP datasets, mainly to find text sequences that are not normal and are considerably different from patterns used in the normal language, hence notifying adversarial text modifications.
2. Spam Detection: LOF can be used in filtering emails to detect spam messages by considering anomalies in the content with respect to normal emails.

## **2.4 Isolation Forest (IF) Algorithm**

Overview of the Isolation Forest Algorithm

Isolation Forest is an efficient anomaly detection method featuring a straightforward notion: it spatially isolates observations by way of recursive partitioning of the data. Unlike traditional distance-based methods, IF puts its emphasis on how easily points can be isolated. That is to say, it intrinsically recognizes anomalies (Ripan et al., 2021).

### **2.4.1 Principles of IF:**

Isolation by Partitioning: IF isSelected randomly a feature and then randomly chose a split value between the maximum and minimum values of the selected feature.

Isolation Trees: The algorithm ensembles an isolation tree-iTrees-by recursively partitioning the data. Every step of partition is modeled as a node in a tree.

Path Length: The length of the path from the root of the tree down to the terminating node indicates how many splits it takes to isolate a data point. In other words, anomalies are easily isolatable, and PATH length for those is usually much shorter (Chater et al., 2022).

Efficiency of Anomaly Detection:

Random Partitioning: IF random partitioning process isolates those anomalies that are few and different much more quickly compared to the normal points (Marteau, 2021)..

Use Cases in Different Fields

Cybersecurity: IF finds wide applications in network intrusion detection and further in identifying compromised devices through isolation of unusual patterns of flow in network traffic.

Finance: IF detects outliers in transaction data, which can indicate fraud.

Healthcare: Anomaly detection would be applied to patient data by IF to analyze whether there is a deviation in medical records or some unusual medical condition.

### **2.4.2 Applicability to Adversarial Attack Detection**

Adversarial Text Detection: IF can be utilized to detect adversarial attacks in NLP datasets by screening out those text sequences that appear with abnormal patterns as compared to the normal text.

Model Robustness Testing: IF makes it easier to test the robustness of NLP models by finding adversarial examples that are most likely to fool a model.

## **2.5 Review of Related works**

Katzir and Elovici, in 2019, presented a novel approach to adversarial perturbation detection, utilizing the spatial behavior of the activation values in a neural network layer. The authors created from those values Euclidean spaces, joined them for every layer by k-nearest neighbor classifiers, where adversarial examples were detected by class label change between layers. Despite promising results from the theoretical discussion of their method, it reached an accuracy of only 75%.

Kim et al. (2024) came up with a two-phase adversarial attack detection technique in CAVs called TAAD-CAV. This paper proposed using an ensemble voting classifier that consisted of three machine learning classifiers and one deep learning classifier, followed by training a meta-classifier, namely Decision Tree, on the combined predictions made. The approach reported an accuracy of 70%, while both precision, recall, and F1-score were 0.70. TAAD-CAV was more robust in adversarial attack detection compared to other methods; however, this again introduces high computational complexity since several classifier outcomes are taken into view. This might be costly for runtime applications.

In the same vein, recently in the year 2023, Watson and Moubayed proposed the adversarial detection technique over medical data using explainability via SHAP values. The reported detection accuracies have been 77% in MIMIC-III, 81% in Henan-Renmin, and 88% in the MIMIC-CXR dataset. Although their method performed better without the need for retraining than the current techniques, it tended to vary a lot between the different datasets considered-a fact that may indicate some problems it could face in generalizing across different types of medical data.

So, in 2019, Colombo and his team came up with the attack-detection technique called HAMPER that could detect adversarial attacks in high-dimensional data using this halfspace-mass depth. HAMPER outperformed all prior methods, sometimes increasing AUROC scores as high as 22.6% and reducing false positive rates significantly by up to 49%. Fairly good overshoots-but because HAMPER is so computationally intensive, it may not work fast enough for real-time applications.

X et al. (2021) studied feature space adversarial attacks by introducing style perturbation that resulted in more realistic adversarial examples. These samples had an extremely low detection rate of only 0.04% on CIFAR-10 and a prediction accuracy of only 1.25% on ImageNet, indicating that the current pixel-space defense methods are rather ineffective against them.

This work proposed the DADA framework, which improves the generalization of replay spoofing detection systems through dual-adversarial domain adaptation with two domain discriminators. Consequently, it outperforms baseline models by a large margin with very low EERs, improving generalization across domains. However, there is still a big challenge in its complexity and performance consistency across different domains.

However, a significant disadvantage is the intensive computation needed to establish and maintain such Euclidean spaces, which may raise barriers in real-time usage and scalability on different datasets. Regarding datasets and model features used, relying on these could raise questions about the general applicability of this approach to practical, real-world scenarios.

Kebande et al. (2021) conducted a study on adversarial attack detection in the User Feedback Process of active machine learning systems. They successfully simulated targeted attacks by intentionally mislabeling smart environment data. Their experimental results showed that real-time identification of adversarial examples during UFP greatly enhances the accuracy of the learning process, though the detection results using their method came out in the range between 0.9474 and 0.5158 F1-Scores. It had two major limitations: being dependent on the particular dataset coming from CASA and further adjustments for other contexts. Besides, an adversarial detection in real time may face substantial technical and practical problems in dynamically changing environments.

Picot et al. (2022) developed REFEREE, an adversarial attack detector intended to solve three major requirements: adaptability in black-box settings, fast inference, and no training phase. ReferEE employed I-projection and achieved 78% accuracy due to performance enhancement and reduction of the inference times. However, the softmax output dependency of the approach binds its utilization for models with varied output configurations.

Finally, Metzen et al. (2017) suggested the extension of deep neural networks by a detector subnetwork that recognizes adversarial perturbations. Their method revealed such a possibility to detect even nearly invisible adversarial attacks with accuracy as high as 70%. However, again, performance showed strong dependence on the type of adversarial attack, thus limiting general robustness and the possibility of an application.

Chakraborty et al. (2021) presented an extensive survey on adversarial attacks and defenses, including this work, by digging deep into the various attack types that may emerge in different kinds of threat models. They have really explored the efficacy of state-of-the-art defense strategies and related challenges. Their conclusion gives an overview of the effectiveness of those defenses while pointing out some hurdles that are yet to be overcome.

Gales and Raina, in their work "Residue-Based Natural Language Adversarial Attack Detection," go further to show how adversarial attack methods can be applied both to images and text. They have tried to detail certain methods, such as BERT-on-BERT and PWWS attacks. The results had some promise, with an F1 score of 0.8 and a Pearson Correlation Coefficient of 0.749, but they also brought some concerns into the limelight. This raises questions as to how well the approach would generalize across diverse NLP tasks and its strong dependence on the BERT architecture, which throws doubt on its generalization ability across different contexts.

Meanwhile, Qiu et al. (2019) reviewed the latter developments of adversarial attack and defense technologies in deep learning. They wanted to summarize what has happened in this area. Their study provides an in-depth analysis of various attacks and defenses by showing how they work and how effective they are.

Goyal et al. (2023) also reviewed adversarial methods of defense in NLP and suggested a new taxonomy, pointing to unresolved issues for future exploration. Their comprehensive review underlines the crucial importance of solid methods of defense and the specific challenges that face NLP. The survey identifies an urgent need that better adversarial defenses are high in demand and suggests some directions of future research for improving robustness in NLP.

The paper "Deep Isolation Forest for Anomaly Detection" by Xu et al. (2023) enhanced the classical technique of the Isolation Forest to better catch anomalies in big and complex datasets. Despite the fact that this method had some prospects, at the same time it showed very poor accuracy and F1 scores, below 80% and 0.8 correspondingly. Another big drawback of this technique is that it is quite resource-heavy, which makes real-time applications quite tricky. On top of that, how well it works really depends on the specific dataset, raising questions about whether it can handle different types of data effectively.

Similarly, Ripan et al. (2021) came up with the "Isolation Forest Learning-Based Outlier Detection Approach for Effectively Classifying Cyber Anomalies," which used the Isolation Forests for identifying cyber threats. Despite some promise, this also had its issues of accuracy and F1 scores not hitting the target. One notable issue they observed was that it was less effective on higher complexity and constantly evolving cyber threats that normally require more adaptive and advanced detection techniques.

Pitropakis et al., in their 2019 paper, provided an in-depth study of how machine learning systems could be attacked with a rich taxonomy of vulnerabilities. While the associated risks were very well identified, they did not really offer anything in terms of solutions. For an informative study, this was way beyond expectation; I was leaving with the thought that more needed to be done in the way of practical defenses against such attacks.

In the year 2024, Dey et al. took on the challenge of looking into adversarial attacks on NLP models. Original work: "Semantic Stealth: Adversarial Text Attacks on NLP Using Several Methods" The authors of this research found that tricking these models using various attack methods was not all that hard. If there was anything worse, it was how weak the current detection systems were. An F1 score less than 0.8 was proof that there was still a lot more to be done in constructing defenses that can keep pace.

In 2019, Qiu and his colleagues reviewed the landscape of "Artificial Intelligence Adversarial Attack and Defense Technologies." They gave a good overview of the progress that has been seen in developing the defense strategies. They realized that although some milestones have occurred, many of the proposed techniques to date remain highly theoretical. What this essentially means is that they have not as yet been put through the process of real-world testing, leaving some doubt about how well they would function effectively in practice.

Goyal et al., while working on "A Survey of Adversarial Defenses and Robustness in NLP," surveyed a few of the defenses in the NLP domain in 2023. Many current mechanisms among them were found incapable of keeping pace, especially when adversarial attacks become sophisticated. The study that the group had done showed that there was still a fair amount of work yet to be done to make these systems truly reliable amidst a myriad challenges they are likely to face in future.

## **2.6 Summary of Related Works/Systems**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | **Main Objectives** | **Methodology** | **Result** | **Evaluation Metrics** | **conclusion** |
| Adversarial Attack Detection in Connected and Automated Vehicles (CAVs) (Kim et al., 2024) | To propose a new dataset and a two-phased adversarial attack detection technique named TAAD-CAV. | An ensemble voting classifier with three ML classifiers and one DL classifier, followed by a meta classifier (Decision Tree) trained on combined predictions. | TAAD-CAV achieves the highest accuracy of 70% compared with individual ML and DL classifiers. | Accuracy: 70%,  Precision: 71%,  Recall: 70%, F1-Score: 70% | TAAD-CAV effectively detects adversarial attacks in CAVs, outperforming individual. |
| Residue-Based Natural Language Adversarial Attack Detection (Raina & Gales 2022) | To examine what differences result when porting these image and text designed strategies to Natural Language Processing (NLP) tasks. | BERT-on-BERT attack, PWWS attack. | An adversarial attack detection approach is proposed that specifically exploits the discrete nature of perturbations for attacks on discrete sequential inputs. | F1 score: 80%  Pearson Correlation Coefficient: 0.749 | The nature of the data (e.g. discrete or continuous) strongly influences the success of detection systems and hence it is important to consider the domain when designing defence strategies. |
| Active Machine Learning Adversarial Attack Detection in the User Feedback Process (Kebande et al., 2021) | To explore adversarial attacks on ML models during the User Feedback Process (UFP) and propose detection methods. | Utilized CASA dataset and nine ML algorithms to detect adversarial attacks during active learning. | The study found that active learning improves classifier performance even under adversarial attacks. | Precision: 94% to 78%,   Recall: 60% to 94%,   F1-Score: 51% to 94%,  Kappa: 92.802% to 25.082%. | Active learning improves classifier performance after an attack. |
| Adversarial Attack Detection Under Realistic Constraints (Picot et al., 2022) | Introduce REFEREE, a detector meeting real-world requirements for adversarial attack detection. | Uses information projections (I-projection) to extract relevant information from softmax outputs. | REFEREE improves state-of-the-art performances and reduces inference time to less than 0.05 seconds per test input. | AUROC: 91.1% (CIFAR10),  90.0% (CIFAR100),  83.2% (Tiny ImageNet);  FPR: 25.9% (CIFAR10),  24.0% (CIFAR100),  38.5% (Tiny ImageNet). | REFEREE is an efficient, real-life adapted adversarial detector that outperforms existing methods and is ready for deployment in real-world applications. |
| GADoT: GAN-based Adversarial Training for Robust DDoS Attack Detection (Abdelati et al., 2021) | To increase the robustness of ML models used for DDoS attack detection by leveraging GAN-based adversarial training. | Uses a GAN to generate fake-benign samples to perturb DDoS samples, creating an augmented dataset for adversarial training. | The state-of-the-art NIDS accuracy drops to 1.8% or less after adversarial training using GADoT. | F1 Score: 99% (before) to 98% (after);  FNR: 0.0000 (before) to 0.0021 (after). | GADoT effectively increases the robustness of ML-based DDoS classifiers against adversarial attacks. |
| A Halfspace-Mass Depth-Based Method for Adversarial Attack Detection (Colombo et al., 2019) | Introduce HAMPER for detecting adversarial examples using data depths. | Leverage halfspace-mass depth for adversarial detection in high-dimensional spaces. | HAMPER outperforms SOTA methods in adversarial detection. | AUROC↑: 13.1%-22.6%,  FPR↓95%: 25.7%-49.0% | HAMPER is effective and robust for adversarial attack detection. |
| Dual-adversarial domain adaptation for generalized replay attack detection (Wang et al., 2020). | Improve generalization of replay spoofing detection systems | Use dual-adversarial domain adaptation with two domain discriminators | DADA framework outperforms baseline and DAT models | EERs: LCNN-DADA (10.06%),  ResNet10-DADA (13.77%),  CGCNN-DADA (12.45%) | DADA framework significantly improves cross-domain generalization performance |
| Detecting Adversarial Perturbations Through Spatial Behavior in Activation Spaces (Katzir et al., 2019) | Propose a novel adversarial example detection method based on activation spaces. | Construct Euclidean spaces from activation values, train k-NN classifiers, and estimate class label change probabilities. | The detector achieves an AUC of 0.95 for the CIFAR-10 dataset with minimal computational complexity increase. | AUC: 95%   Accuracy: 75% | The proposed method effectively detects adversarial examples by leveraging spatial behavior in activation spaces, showing resilience to future adversarial manipulations. |
| Towards Feature Space Adversarial Attack by Style Perturbation (X et al., 2021) | Propose a new adversarial attack focusing on perturbing abstract features. | Perturb abstract features using optimization to induce model misclassification. | Generated adversarial samples are more natural-looking than state-of-the-art attacks. | Detection rate of 0.04% on CIFAR-10 and prediction accuracy of 1.25% on ImageNet. | Existing pixel-space defense techniques are ineffective against feature space attacks. |
| Attack-Agnostic Adversarial Detection on Medical Data Using Explainable Machine Learning (Watson & Moubayed et al., 2023) | Propose a model-agnostic explainability-based method for detecting adversarial samples in medical data. | Utilized SHAP values to detect adversarial samples on EHR and CXR datasets using both supervised and semi-supervised methods. | Achieved 77% detection accuracy on MIMIC-III and Henan-Renmin EHR datasets, and 88% on MIMIC-CXR dataset. | Accuracy: 77% (MIMIC-III),  81% (Henan-Renmin),  88% (MIMIC-CXR). | The proposed method outperforms state-of-the-art techniques and generalizes to different attack types without retraining. |

## **2.7 Summary of Literature Review**

Although there have been quite a few targeted NLP attacks created with the goal of generating adversarial attacks for NLP inputs, several works have presented these—Xu et al., Ripan et al., Pitropakis et al., Dey et al., Boucher et al., Schneider et al., Vaccari et al., Wenger et al., and Zhang et al.—not much emphasis has been put on developing strong detection mechanisms to counter them. A few works introduced different approaches for the detection of these adversarial attacks: TAAD-CAV for CAVs reached an accuracy of 70% by Kim et al. (2024), and the explainability-based approach of Watson and Moubayed et al. for medical data with accuracies as high as 88%.

Whereas most of these methods do show improvement over others, they still face generalization issues and usually incur heavy computational costs; hence, these remain less practical for everyday usage. The clear need is evident for more efficient, scalable, and adaptive methods for detection to keep pace with increasing sophistication in adversarial attacks.

# **CHAPTER THREE (Proposed System Design and Methodology)**

## **3.0 Introduction**

Following this, we undertook the holistic approach for this research project. This approach focuses on the various stages of the design and development of the adversarial attack detection system for NLP datasets. In relation to design considerations, after discussing the research study design, the philosophy, type, and the strategy adopted are discussed in detail. The next section outlines the sampling strategy applied, techniques for data collection, and data analysis methods. This chapter also contemplates the operational procedures that govern or lead to the implementation of the system. Critically re-evaluating the limitations and challenges of our chosen methodology provides a balanced view of the study. Finally, this chapter concludes with a summary that summarizes the main points discussed to pave the way forward for subsequent sections of the documentation.

## **3.1 Research Design**

### **3.1.1 Research Philosophy**

This is supported by the fact that, for example, the positivist philosophy will be particularly apt in the study of adversarial attack detection in the framework of NLP datasets. It is of critical importance to note at this point that, by definition, positivism is characterized by an emphasis on observable, objective phenomena, and quantitative methods for empirical data collection. In Ayeni et al. (2019); Maksimovic & Evtimov (2023). This is the philosophy that supports the application of statistical analysis and machine-learning algorithms in delimitating and analyzing data patterns that implicate an adversarial attack. A positivist approach will help us establish results that are reliable and consistent and also unbiased and reproducible. As pointed out by Khraisat & Alazab (2021), the positivist paradigm plays a very important role in ensuring the standard of research outputs for which validity and reliability could be well-established in later studies, thereby helping to develop the area of adversarial attack detection. This philosophy underpins our intention of modeling a strong detection system that could be empirically experimented on and proved to add to the wider knowledge base in AI and cybersecurity. Besides, the positivistic approach gives us the real opportunity to focus our endeavors on the question of precise measurements and analyses, which justifies the derivation of correct conclusions regarding the efficiency of LOF and IF algorithms against adversarial attacks.

### **3.1.2 Research Type**

An adversarial attack detection system using machine learning algorithms such as LOF and IF is proposed to be designed and developed on NLP datasets. This is based on an assumption that both algorithms have the inbuilt potential to recognize such anomalies caused by adversarial attacks, which stands as a hypothesis that our research will follow on a deductive basis. While the original data this research would employ, namely NLP datasets, are qualitative, the objective nevertheless remains to conduct a quantitative analysis on such datasets in search of adversarial patterns and the effectiveness of our algorithms. To that extent, this research is largely quantitative. We expect in our case that LOF and IF algorithms may contribute towards detecting adversarial attacks, and our analysis follows upon completion of the research by using inductive generalizations that allow us to draw broader conclusions.

### **3.1.3 Research Strategy**

The experimental approach of the project's proposed research strategy is based on developing and validating an adversarial attack detection system with experiments on NLP datasets. The dataset will be subdivided into three distinct subsets, namely, train.csv, dev.csv (development), and test.csv, each serving a particular purpose in the research process. Subsequently, train.csv will be utilized for the training of LOF and IF algorithms. This involves the training of the algorithms by feeding them labeled data such that they can learn patterns of data and detect anomalies that may signify an adversarial attack. The dev.csv dataset, also called the validation set, will be used for tuning the models. Model parameters adjustment for optimized performance is done in this step. The performance of the models is tested using a development set in order to avoid overfitting and generalization on new and unseen data. Finally, the test.csv dataset will be used as an unbiased evaluation set. It consists of the data which the models have never seen during either training or validation phases. Such separation leads to careful assessment of how formidable these models are in finding adversarial attacks in NLP datasets. Basically, we would like to provide empirical evidence to prove our hypothesis that LOF and IF algorithms give good detection performance of adversarial attacks, and then contribute to developing adversarial attack detection methods in AI training datasets by following this systematic experimentation strategy.

### **3.1.4. Data Collection Methods**

The ANLI-Adversarial Natural Language Inference dataset, ANLI, by Nie et al. (n.d.), is the dataset considered for this research, downloaded from Kaggle. ANLI is a large-scale NLI benchmark that has been collected through an iterative, adversarial human-and-model-in-the-loop procedure and hence is more challenging compared to its predecessors like SNLI and MNLI. ANLI consists of three rounds, each with train, dev, and test splits. Schemas of the fields are uniform across splits. The train.csv is used to train the algorithm LOF and IF. In this, the dev.csv file acts as the validation set to tune the models, and test.csv is the file that the model has never seen; it is used in final evaluation to get an unbiased estimate of the performance of the model. The license for this dataset is CC0 1.0 Universal (Public Domain Dedication). Originally published on Hugging face Hub. This dataset makes for an excellent benchmark in testing and validating the robustness of our adversarial attack detection methods in NLP. Then, premise and hypothesis are pre-processed using TF-IDF.

## **3.1.5 TF-IDF (Term Frequency-Inverse Document Frequency):**

It's a numerical statistic used to evaluate how important a word is to a document in a collection or corpus.

Term Frequency (TF): Measures how frequently a word appears in a document. If a word appears often in a document, it has a high term frequency.

Example: In a document with 100 words, if the word "cat" appears 5 times, the term frequency (TF) for "cat" is

Inverse Document Frequency (IDF): Measures how important a word is across all documents in the corpus. Words that appear in many documents (common words like "the", "is", etc.) have a low IDF, while words that appear in fewer documents have a high IDF.

Example: If the word "cat" appears in 10 out of 1000 documents, the IDF is calculated as ) = log (100) = 2  
TF-IDF Score: The TF-IDF score is calculated by multiplying the TF and IDF scores. It helps to weigh the word's importance in the document relative to the corpus.

Example: If the TF for "cat" is 0.05 and the IDF is 2, the TF-IDF score for "cat" is

0.05 × 2 = 0.1  
**3.2 System Architecture**

The system architecture of our project is designed to integrate advanced detection algorithms and provide a user-friendly interface for interacting with NLP models.

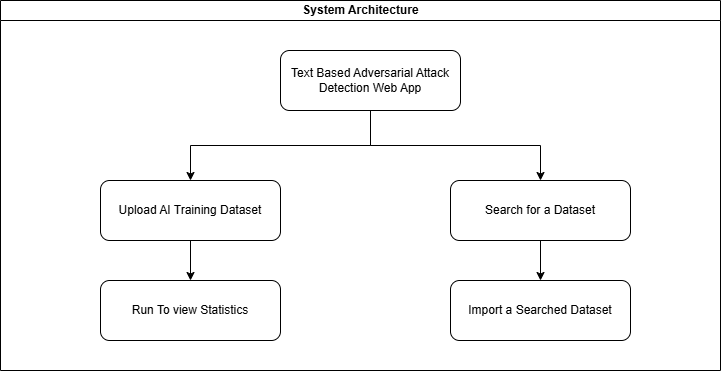


Figure 3.1 System Architecture

The system consist of three main functionalities:

Uploading training dataset, interacting with AI, and searching for a dataset based on keyword

## **3.3 System Components**

The system consist of two main components, Front end and back end.

### **3.3.1 Front End**

It is a very friendly user interface that allows users to effectively interact with the system through the web application frontend. We developed this interface using HTML for structuring, CSS for styling, and JavaScript for functionality. We aslo frameworks like Bootstrap for responsive design, and React or Vue.js for rendering content dynamically. With this clean and intuitive interface, one will comfortably upload a dataset, trigger detection, and dig into the results. This would include file uploads complete with drag-and-drop, progress indicators, and interactive charts and graphs to show the results of detection. Results are to be relayed async to and from the back end using AJAX or the Fetch API to maintain smooth and seamless user experiences.

It provides a small frontend interaction with the backend through the RESTful API. In that respect, it enables smooth data transfer without any need for page reloads. Dataset upload, detection triggering, fetching of results, along with the visualizations—the APIs take care of everything. Therefore, the design ensures that the user can navigate easily around the application, understand the status of their uploads and processing, and interpret the detection results themselves through good-looking, easy-to-understand diagrams.

### **3.3.2 Back End**

The core processing tasks, data handling, running detection algorithms, serving the front end—are handled by the back end with a Django build. The web server will handle HTTP requests and perform user authentication and authorization, routing the request to appropriate services. User information, uploaded datasets, and detection results would be stored in a robust database system such as PostgreSQL or MySQL for secure and efficient handling.

It provides for data storage using AWS S3 or local storage solutions to securely manage uploaded datasets and generated results. The RESTful API is supported by the backend, which supplies endpoints like /upload for uploading datasets, /process for beginning the detection process, and /results for fetching results about detections along with visualizations. The integrated components listed here form the backend, which ensures smooth runs and efficiency in the application for reliable service delivery to the frontend.

### **3.3.3 Algorithms**

A module called algorithms is implanted into the core detection system, which includes LOF and Isolation Forest algorithms. These will find the adversarial attack in the datasets uploaded by detecting outliers or anomalies from normal data points. The LOF algorithm identifies outliers based on the sheer deviation of their local density, while the IF algorithm spots the anomalies efficiently by the isolation of observations.

It thus brings these algorithms in the sense that the uploaded datasets are processed towards the detection of any adversarial attack. The results derived from the algorithms are then analyzed with Pandas and NumPy data analysis libraries. It creates visualizations with Matplotlib or Plotly, enabling statistical diagrams in respect to the nature and extent of adversarial attacks that may be detected in the datasets. This, therefore, ensures comprehensive and robust adversarial attack analysis, raising well the reliability and effectiveness bar for the detection system.

# **3.4 Operational Methods (Basic Logic)**

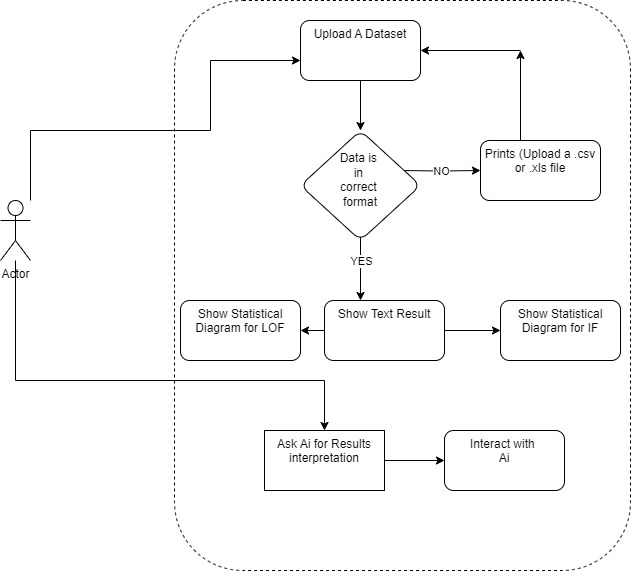


Figure 3.2 Operational Flow of the System

## **3.5 Methodological Limitations**

In the benchmark datasets used to train or test our detection algorithms, there could be possible biasness. These datasets, although comprehensive, may not include the full diversity or variability of real-world adversarial attacks. In this case, the results in these experiments may overestimate how well our detection algorithms generalize to new, unseen data. Also, though remedies like isolation forest and Local Outlier Factor are immensely powerful, there are certain theoretical limitations in their detection for all types of adversarial attacks, especially those that are sophisticated or novel.

Another limitation is the generalization challenge between different NLP models and applications. We focus on specific NLP models, which do not take into account certain nuances and variations that are rather specific to other models or domains. That is, their actual performance might be different if applied to other types of NLP tasks or integrated into other model architectures. Furthermore, the computational cost for executing these algorithms could pose practical challenges, in particular with regard to real-time processing applications, that may inhibit scalability and wider adoption of the proposed algorithm.

## **3.6 Summary of Methodology**

Our approach is systematic in beefing up the security of NLP models against adversarial attacks. We focus on developing strong detection algorithms while keeping a user-friendly web interface. It starts with benchmark dataset collection, including benign and adversarial examples, and preparing them. These are instrumental datasets for training and testing our detection algorithms so as to have them identify the adversarial threats as and when they occur and handle them suitably.

Methodologies' core detection algorithms isolate the forest technique and LOF. Since these techniques are well recognized for their effectiveness in anomaly detection, therefore we think they are quite all right.

Subsequently, we integrate these detection algorithms into NLP models for real-time protection against any adversarial shenanigans. Finally, we are providing a user-friendly web interface using state-of-the-art technologies like Django, HTML, CSS, and JavaScript. This would allow users to interact with NLP models and see through the results of the detection. Intuitive design shall be targeted for users to easily submit their inputs and observe how the detection process would go on. Along all this development process, some thorough testing and validation will be done, to guarantee that everything works seamlessly and with reliability for both detection algorithms and web interface.

# **CHAPTER 4 (Analysis and Results)**

## **4.0 Introduction**

In this chapter, we’ll take a closer look at the analysis and outcomes of the adversarial attack detection system developed for this research. The main objective is to figure out how well the Local Outlier Factor (LOF) and Isolation Forest (IF) algorithms perform when it comes to spotting adversarial attacks in Natural Language Processing (NLP) datasets. We’ll be evaluating these algorithms by looking at various performance metrics, like precision, recall, F1-score, and accuracy. This section also covers the results of applying these models to the dataset, provides insights into the visual data generated by the system, and explains the findings in detail.

## **4.1 Dataset Used**

To do this, I will be working with the ANLI dataset, downloaded from Kaggle. Due to its ability to test models using adversarial examples, the ANLI dataset is one of the most popular benchmarks in the field of NLP. Probably one of the most striking features of ANLI can be seen in how this dataset was created, using human input along with machine-generated data, yielding particularly complex examples that are challenging for models to handle. This makes it a good dataset to test how well the LOF and IF algorithms are able to detect the adversarial cases.

The above dataset is divided into three key parts: the train.csv, which contains data for training the model; dev.csv for development and validation; and finally test.csv for testing. Each of these sections contains the same data field types.

We started with the train.csv file, training our models on this, so both LOF and IF could find patterns in possibly adversarial attacks. Then, using dev.csv, the models were fine-tuned and validated in order to see that they would generalize well on new, unseen data. We finally apply these models on test.csv in order to check the accuracy and, in general, the capability of the models in detecting adversarial attacks.

This dataset is a great benchmark, because of the many different types of adversarial examples that are in it, intended to test how well our models can identify and handle adversarial threats.

## **4.2 Adversarial Attack Detection**

In this research, we plunge into adversarial attack detection by using the hybrid of two most effective anomaly detection algorithms: LOF and IF. Both these models cooperatively find the anomalies in our dataset, which would hint at probable adversarial attacks.

Let me break it down a bit. The LOF algorithm is interesting because, for the identification of outliers, the LOF looks into data points' local density in comparison with their neighbors; in other words, whether it calculates a point that has denser neighbors or not.

The LOF score for a data point is given by:

----------------------------- 1

Where;

is the k-nearest neighbor of the point p

Local Reachability Density (LRD) is computed as the inverse of the average distance from the point to its neighbors.

If the LOF score is greater than a threshold, the point is considered an anomaly. In this case, LOF helps to detect points with locally sparse densities in the dataset, identifying potential adversarial examples.

The **Isolation Forest (IF)** algorithm is designed to isolate observations by recursively partitioning the data. Anomalies are isolated more quickly because they are rare and different from the majority of the data.

The **Isolation Path Length** for a point is the number of partitions required to isolate it. The **Anomaly Score** is derived from the average path length of the point across all isolation trees, calculated as:

------------------------------------- 2

Where:

(ℎ(𝑥)) is the average path length of point

c(n) is the average path length of a binary search tree built with n observations.

A higher anomaly score indicates that the point is an outlier, likely representing adversarial behavior in the dataset.

### **4.2.1 Combined Detection Approach**

The combination of LOF and IF in this study does indeed provide a solid approach toward the detection of adversarial attacks. LOF detects local anomalies by comparing density deviations, while IF isolates global anomalies through recursive data splitting. With their combination, we make sure that the proposed discriminator will detect both local and global anomalies effectively to provide comprehensive adversarial attack detection in an NLP dataset.

### **4.2.2 Detection Results**

A total of 16,947 records were used for testing the model. The LOF and IF combined model identified 12% of the dataset as adversarial examples and labeled them as -1 (outliers), while 88% of the data was labeled as normal data-1.

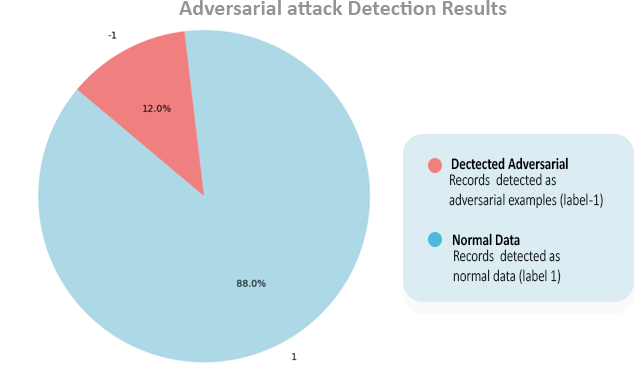


Figure 4.1 Adversarial attack detection results

Number of Adversarial Records:

Number of Normal Records:

These results show that 2,033 records were detected as adversarial examples (label -1), while 14,914 records were classified as normal (label 1). The model's detection provides an effective breakdown of adversarial versus normal data, helping to protect NLP models from attacks designed to deceive the system.

## **4.3 Performance Metrics**

Such metrics as accuracy, precision, and the F1 score can be used in order to test the efficiency of the adversarial attack detection system. The use of such metrics provides the insight of how well the combined model LOF-IF detects adversarial examples and differentiates them from data that is normal.

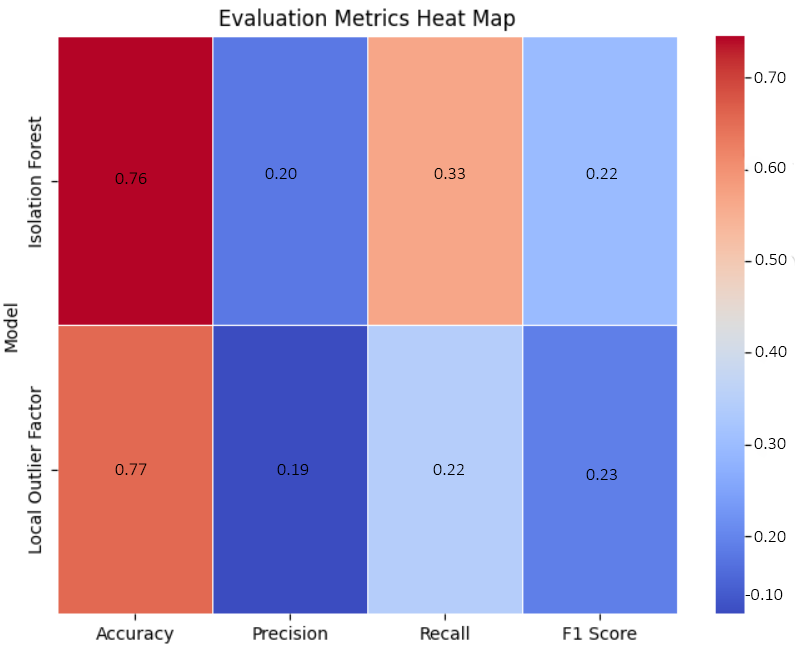


Figure 4.2 Heat map of Evaluation metrics of Local Outlier Factor and Isolation Forest

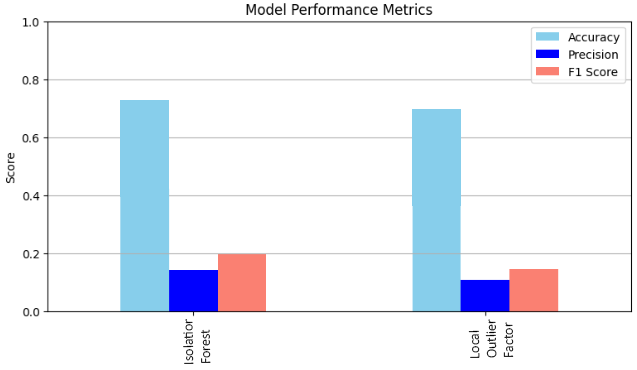


Figure 4.3 Accuracy, Precision and F1 Score of IF and LOF

### **4.3.1 Accuracy**

Accuracy measures the overall correctness of the model by calculating the proportion of correct predictions (both adversarial and normal) over the total number of predictions.

Our accuracy was calculated using the formula:

--------------------------------- 3

Where:

TP: True Positives (correctly detected adversarial examples)

TN: True Negatives (correctly identified normal data)

FP: False Positives (normal data incorrectly identified as adversarial)

FN: False Negatives (adversarial examples missed by the model)

With an accuracy of 76%, the model correctly identifies both adversarial and normal examples in the majority of cases, meaning that approximately three-quarters of the records in the dataset were correctly classified.

### **4.3.2 Precision**

Precision is the proportion of true positive adversarial detections out of all the records predicted as adversarial. It reflects the model’s ability to correctly identify adversarial examples while avoiding false alarms (false positives).

The formula for precision is:

---------------------------------- 4

In this study, the precision is 19%, meaning that only 19% of the records flagged as adversarial are truly adversarial. This lower precision indicates that the model produces a significant number of false positives, identifying some normal records as adversarial, which can lead to over-detection.

### **4.3.3 F1 Score**

The F1 score is the harmonic mean of precision and recall, providing a balanced metric that accounts for both false positives and false negatives.

The formula below is used in the calculation of F1 score:

----------------------- 5

A higher F1 score indicates a balance between precision and recall, showing the model's overall effectiveness in detecting adversarial attacks. With an F1 score of 22%, the model’s performance in detecting adversarial examples is modest but highlights the need for further refinement to balance the trade-offs between precision and recall.

### **4.3.4 Interpretation of Results**

While this model has a relatively good overall accuracy of 76.5%, it has a rather low precision of 19%, and its F1 score is only 22%. This reflects that although the system is somewhat good at determining adversarial examples, it fails in cases of false alarms-in other words, it misclassifies most of the normal data points as adversarial. The imbalance between precision and accuracy shows that while the model properly captures most of the normal examples, it is bad at precisely pinpointing adversarial attacks, making it less targeted towards detection.

## **4.4 Using the web app**

1. Perform a web search for adversarial attack detection tool or enter the url of the app
2. Login if you already have an account
3. Sign up if you have not already registered
4. Click and upload a csv or excel file (Dataset) to predict
5. Click on ‘Predict’ button
6. Re-upload the same file for confirmation of first results or load a new dataset

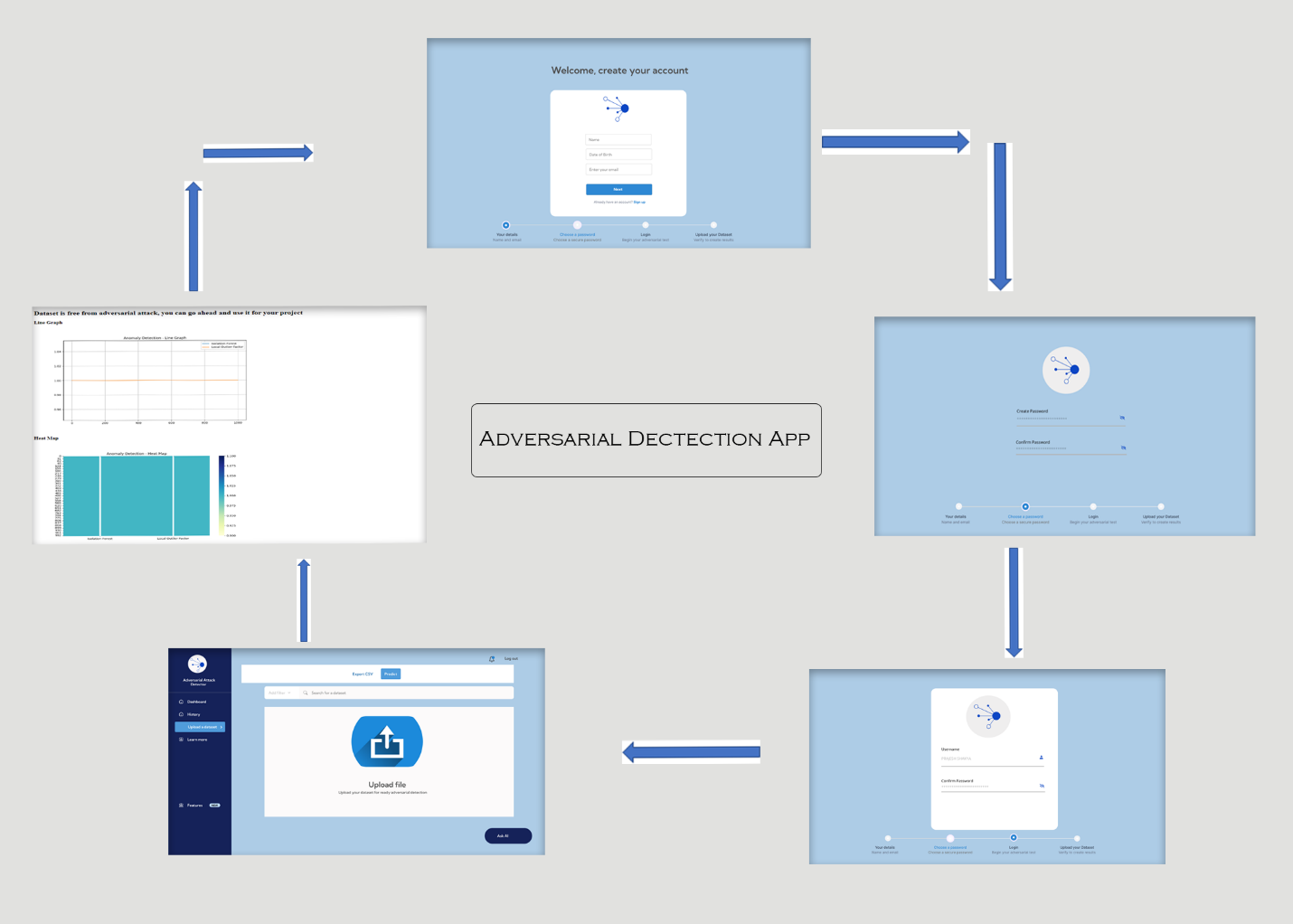


Figure 4.4 Using The Web App

# **CHAPTER 5 (Discussion and Conclusion)**

## **5.0 Introduction**

This chapter reflects on the results obtained during the research and provides a broad discussion with respect to the objectives of the project. The study aimed to develop a robust NLP-based adversarial attack detection system by combining LOF and IF models. Additionally, the section discusses the limitations encountered, recommendations for future improvements, and summarizes some key conclusions derived from the project.

## **5.1 Discussion of Results**

Results from this research show the strengths and limitations of the adversarial attack detection model implemented in this paper, which coupled the LOF and IF algorithms. This model was experimented with and tested against the ANLI dataset, which had a record count of 16,947, from which 12% of data instances were found to be adversarial-labeled as -1-whereas 88% were found to be normal data points, labeled as such with label 1. These results are important because they show that the system can indeed pick out a significant portion of adversarial examples, something that confirms their utility for hybrid approaches to attack detection in NLP tasks.

The combination of LOF and IF proved useful in capturing local and global outliers from the dataset. LOF works in comparing local densities created by the neighborhood of data points to find instances where data is sparse or anomalous with respect to its neighbors. On the other hand, Isolation Forest works efficiently by the isolation of data points on random partitioning and is well-suited for high-dimensional data such as text. What this does, in combination, is balance precision and recall for the system in a way that can be more holistic in its targeting of adversarial data.

However, the performance metrics still hint at some scope for improvement. The accuracy was at 76%, and the precision was 19%, with an F1 score of 22%. On one hand, the 76% accuracy hints at the overall reliability of this model in terms of the correct classification of data points. Then again, the 19% low precision indicates that the model tends to struggle with false positives. That is, the model very often classifies normal data as adversarial, which seriously diminishes its usefulness in real-world practice, given that a misclassification of normal instances might result in some inefficient or costly outcome. This further leads to an F1 score of 22%, which balances precision and recall, hence underlining an imbalance in this model and sustaining the idea that more tuning is necessary for better performance.

These metrics demonstrate that, even though the model has shown potential, more especially in detecting adversarial attacks, it is not optimal yet. The low precision and F1 score hint at a more refined or probably complex model that can capture the subtlety of adversarial attacks better in natural language. These results could partly be due to the inherently hard nature of the ANLI dataset, which is so by design, and comparably simpler feature extraction methods adopted.

## **5.2 Future Works/Recommendation**

For future work on this project, we would develop this project further by deploying our adversarial attack detection web application to popular cloud platforms such as AWS or Microsoft Azure so that more users can access it and also scale it. In this way, adoptability of the tool in a wide variety of AI and cybersecurity use cases by organizations and individuals will be possible. Secondly, we will optimize the system towards real-time adversarial detection, so it can work as a live-monitoring tool within AI pipelines that will enhance its practical utilities in sensitive applications such as finance, healthcare, and autonomous systems.

In future, the model should be further expanded to include more adversarial detection techniques. Expert machine learning methods will be applied to ensure a high degree of precision. The web-based interface will include feedback mechanisms that enable users to report cases of false positives or misclassifications. These will, in turn, assist in constantly improving the model to evolve with new emerging adversarial attack patterns. We also call on AI practitioners and organizations to field the tool in real-world environments and provide feedback that will help us further develop the robustness and accuracy of the system.

## **5.3 Conclusion**

It is a web-based project based on the usage of NLP and machine learning algorithms to identify adversarial attacks within an AI training dataset. In this developed system, LOF and IF models have been integrated in order to find the anomalies. As mentioned, it labeled correctly 12% of the data as adversarial (label -1) and 88% as normal ones on the ANLI dataset, which contained 16,947 records. And the performance metrics-accuracy being at 76%, precision at 19%, and an F1 score also at 22%-prove the fact that the model can accurately detect these adversarial instances; however, much improvement in precision is necessary in order to reduce the number of false positives.

It provides a clear statistical overview and visualizations of the results obtained through adversarial detection by allowing the user to supply their dataset and retrieve quick feedback on potential threats. On the other hand, the relatively low precision suggests that the model still misclassifies a good chunk of the non-adversarial data as adversarial, which may make it somewhat problematic for use in practical settings. Hence, our future update considers responsively enhancing methodologies of feature extraction and fine-tuning the detection algorithms to perform better. This tool, even with these limitations, paves a major step toward securing AI models with adversarial attacks, especially in the realm of in-vocabulary Natural Language datasets.

**References**

Abdelaty, M., Scott-Hayward, S., Doriguzzi-Corin, R., & Siracusa, D. (2021, October). Gadot:

Gan-based adversarial training for robust ddos attack detection. In *2021 IEEE Conference on Communications and Network Security (CNS)* (pp. 119-127). IEEE.

Ahmadi, M. A., Dianat, R., & Amirkhani, H. (2021). An adversarial attack detection method in

deep neural networks based on re-attacking approach. *Multimedia Tools and Applications*, *80*, 10985-11014.

Akhtar, N., Mian, A., Kardan, N., & Shah, M. (2021). Advances in adversarial attacks and

defenses in computer vision: A survey. IEEE Access, 9, 155161-155196.

Alimohammadi, H., & Chen, S. N. (2022). Performance evaluation of outlier detection techniques

in production timeseries: A systematic review and meta-analysis. Expert Systems with Applications, 191, 116371.

Alshemali, B., & Kalita, J. (2020). Improving the reliability of deep neural networks in NLP: A

review. Knowledge-Based Systems, 191, 105210.

Alotaibi, A., & Rassam, M. A. (2023). Adversarial machine learning attacks against intrusion

detection systems: A survey on strategies and defense. Future Internet, 15(2), 62.

Boucher, N., Shumailov, I., Anderson, R., & Papernot, N. (2022, May). Bad characters:

Imperceptible NLP attacks. In 2022 IEEE Symposium on Security and Privacy (SP) (pp. 1987-2004). IEEE.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., & Mukhopadhyay, D. (2021). A survey

on adversarial attacks and defences. CAAI Transactions on Intelligence Technology, 6(1), 25-45.

Chang, G., Gao, H., Yao, Z., & Xiong, H. (2023). TextGuise: Adaptive adversarial example attacks

on text classification model. Neurocomputing, 529, 190-203.

Chater, M., Borgi, A., Slama, M. T., Sfar-Gandoura, K., & Landoulsi, M. I. (2022). Fuzzy isolation

forest for anomaly detection. Procedia Computer Science, 207, 916-925.

Colombo, P., Picot, M., Granese, F., Romanelli, M., Messina, F., & Piant, P. (2023). A

halfspace-mass depth-based method for adversarial attack detection. *Transactions on Machine Learning Research Journal*.

Dey, R., Debnath, A., Dutta, S. K., Ghosh, K., Mitra, A., Chowdhury, A. R., & Sen, J. (2024).

Semantic Stealth: Adversarial text attacks on NLP using several methods. arXiv preprint

arXiv:2404.05159.

Foorthuis, R. (2021). On the nature and types of anomalies: A review of deviations in data.

International Journal of Data Science and Analytics, 12(4), 297-331.

George, A., & Marcel, S. (2020). Learning one class representations for face presentation attack

detection using multi-channel convolutional neural networks. IEEE Transactions on

Information Forensics and Security, 16, 361-375.

Goyal, S., Doddapaneni, S., Khapra, M. M., & Ravindran, B. (2023). A survey of adversarial

defenses and robustness in NLP. ACM Computing Surveys, 55(14s), 1-39.

Huang, R., & Li, Y. (2022). Adversarial attack mitigation strategy for machine learning-based

network attack detection model in power system. *IEEE Transactions on Smart Grid*, *14*(3), 2367-2376.

Ibitoye, O., Abou-Khamis, R., Shehaby, M. E., Matrawy, A., & Shafiq, M. O. (2019). The threat

of adversarial attacks on machine learning in network security--A survey. arXiv preprint arXiv:1911.02621.

Katzir, Z., & Elovici, Y. (2019, July). Detecting adversarial perturbations through spatial

behavior in activation spaces. In *2019 international joint conference on neural networks (IJCNN)* (pp. 1-9). IEEE.

Kebande, V. R., Alawadi, S., Awaysheh, F. M., & Persson, J. A. (2021). Active machine

learning adversarial attack detection in the user feedback process. *IEEE Access*, *9*, 36908-36923.

Khan, A., Qureshi, A. S., Wahab, N., Hussain, M., & Hamza, M. Y. (2021). A recent survey on

the applications of genetic programming in image processing. Computational Intelligence, 37(4), 1745-1778.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the

art, current trends and challenges. Multimedia Tools and Applications, 82(3), 3713-3744.

Kim, T. H., Krichen, M., Alamro, M. A., & Sampedro, G. A. (2024). A Novel Dataset and

Approach for Adversarial Attack Detection in Connected and Automated Vehicles. *Electronics*, *13*(12), 2420.

Lauriola, I., Lavelli, A., & Aiolli, F. (2022). An introduction to deep learning in natural language

processing: Models, techniques, and tools. Neurocomputing, 470, 443-456.

Larijani, H., Mtetwa, N., Yousefi, M., & Javed, A. (2020, July). An adversarial attack detection

paradigm with swarm optimization. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE.

Marteau, P. F. (2021). Random partitioning forest for point-wise and collective anomaly

detection—Application to network intrusion detection. IEEE Transactions on Information Forensics and Security, 16, 2157-2172.

Metzen, J. H., Genewein, T., Fischer, V., & Bischoff, B. (2017). On detecting adversarial

perturbations. *arXiv preprint arXiv:1702.04267*.

Nguyen, H. H., Yamagishi, J., & Echizen, I. (2019). Use of a capsule network to detect fake images

and videos. arXiv preprint arXiv:1910.12467.

Omar, M., Choi, S., Nyang, D., & Mohaisen, D. (2022). Robust natural language processing:

Recent advances, challenges, and future directions. IEEE Access, 10, 86038-86056.

Picot, M., Noiry, N., Piantanida, P., & Colombo, P. (2022). Adversarial attack detection under

realistic constraints.

Pitropakis, N., Panaousis, E., Giannetsos, T., Anastasiadis, E., & Loukas, G. (2019). A taxonomy

and survey of attacks against machine learning. Computer Science Review, 34, 100199.

Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and

defense technologies. Applied Sciences, 9(5), 909.

Radanliev, P., & Santos, O. (2023). Adversarial attacks can deceive AI systems, leading to

misclassification or incorrect decisions.

Raina, V., & Gales, M. (2022). Residue-based natural language adversarial attack detection. arXiv

preprint arXiv:2204.10192.

Ripan, R. C., Sarker, I. H., Anwar, M. M., Furhad, M. H., Rahat, F., Hoque, M. M., & Sarfraz, M.

(2021). An isolation forest learning-based outlier detection approach for effectively classifying cyber anomalies. In Hybrid Intelligent Systems: 20th International Conference on Hybrid Intelligent Systems (HIS 2020), December 14-16, 2020 (pp. 270-279). Springer International Publishing.

Rose, R. L., Puranik, T. G., & Mavris, D. N. (2020). Natural language processing based method

for clustering and analysis of aviation safety narratives. Aerospace, 7(10), 143.

Ryciak, P., Wasielewska, K., & Janicki, A. (2022). Anomaly detection in log files using selected

natural language processing methods. Applied Sciences, 12(10), 5089.

Schneider, J., Meske, C., & Vlachos, M. (2023). Deceptive XAI: Typology, creation and detection.

SN Computer Science, 5(1), 81.

Shah, V. (2020). Advancements in deep learning for natural language processing in software

applications. International Journal of Computer Science and Technology, 4(3), 45-56.

Vaccari, A., Carlevaro, S., Narteni, E., Cambiaso, M., & Mongelli, M. (2022). eXplainable and

reliable against adversarial machine learning in data analytics. IEEE Access, 10, 83949-83970.https://doi.org/10.1109/ACCESS.2022.3197299. https://ieeexplore.ieee.org/document/9852204

Wang, C., Chen, J., Yang, Y., Ma, X., & Liu, J. (2022). Poisoning attacks and countermeasures in

intelligent networks: Status quo and prospects. Digital Communications and Networks, 8(2), 225-234.

Wang, H., Dinkel, H., Wang, S., Qian, Y., & Yu, K. (2020). Dual-Adversarial Domain

Adaptation for Generalized Replay Attack Detection. In *Interspeech* (pp. 1086-1090).

Wang, X., Li, J., Kuang, X., Tan, Y. A., & Li, J. (2019). The security of machine learning in an

adversarial setting: A survey. Journal of Parallel and Distributed Computing, 130, 12-23.

Wenger, E., Passananti, J., Bhagoji, A. N., Yao, Y., Zheng, H., & Zhao, B. Y. (2021). Backdoor

attacks against deep learning systems in the physical world. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6206-6215).

Watson, M., & Al Moubayed, N. (2021, January). Attack-agnostic adversarial detection on

medical data using explainable machine learning. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 8180-8187). IEEE.

Xu, H., Pang, G., Wang, Y., & Wang, Y. (2023). Deep isolation forest for anomaly detection. IEEE

Transactions on Knowledge and Data Engineering, 35(12), 12591-12604.

Xu, Q., Tao, G., Cheng, S., & Zhang, X. (2021, May). Towards feature space adversarial attack

by style perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 12, pp. 10523-10531).

Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning

models in natural language processing: A survey. ACM Transactions on Intelligent

Systems and Technology (TIST), 11(3), 1-41.