Enhancing the Detection of Adversarial Attacks Using Deep Learning Neural Transformer Models

by Samith P. Gunasekara

B.Sc. (Hons) in Computing Science, May 2003, University of Staffordshire
MBA in Analytics and Entrepreneurship, August 2022, University of Illinois Urbana-Champaign

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

January 10, 2025

Praxis directed by

Buddha Nepal
Professorial Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Samith P. Gunasekara has passed the Final Examination for the degree of Doctor of Engineering as of November 18, 2024. This is the final and approved form of the Praxis.

Enhancing the Detection of Adversarial Attacks Using Deep Learning Neural Transformer Models

Samith P. Gunasekara

Praxis Research Committee:

Amir Etemadi, Assistant Professor of Engineering and Applied Science, Praxis Director

Buddha Nepal, Professorial Lecturer in Engineering and Applied Science, Committee Member

John Morton, Professorial Lecturer in Engineering and Applied Science, Committee Member

Dedication

The author dedicates this work to his wife, Anushi, his daughter, Linuki, and his son, Liam. Their sacrifices, unwavering support, and constant encouragement throughout the doctoral journey made this achievement possible.

The author also extends heartfelt gratitude to his parents, Janaki and Asoka, for their upbringing and for nurturing in him a love for learning, resilience, and perseverance.

Furthermore, the author dedicates this research to his mentors, coaches, and colleagues, whose invaluable guidance and insightful feedback have profoundly shaped his intellectual and personal growth.

Lastly, the author dedicates this research to all those striving for progress and innovation in cybersecurity, hoping that it will make a meaningful contribution to making the world a fun, safer, and more secure place to live.

Acknowledgments

The Author would like to express his deepest gratitude to his advisor, Dr. Buddha Nepal, for the guidance, unwavering support, and constructive feedback throughout this research journey. Your mentorship has been instrumental in shaping the direction of this research.

The Author is also profoundly thankful to all the program professors and faculty members of The School of Engineering and Applied Science of George Washington University, whose teaching and insights have expanded the Author's knowledge and provided the foundation for this work. A heartfelt appreciation also goes to the Author's doctoral program colleagues and peers, whose collaboration, encouragement, and stimulating discussions have enriched the research experience. Your camaraderie and shared passion for this field have made the journey enjoyable and rewarding.

Finally, the Author would like to thank his sponsor, The Boeing Company, and all those who contributed, directly or indirectly, to complete this work. Your support has been a source of strength and inspiration throughout this process.

Abstract of Praxis

Enhancing the Detection of Adversarial Attacks Using Deep Learning Transformer Models

In today's world, cybersecurity and artificial intelligence (AI) are crucial to ensuring the confidentiality, integrity, and availability of interconnected digital systems for human safety and productivity. The rapid adoption of AI in mission-critical systems has led to the development of highly sophisticated cyberattacks with malicious intent, including sabotage, theft, and potential threats to human life (Payne et al., 2024).

This Doctor of Engineering praxis presents a novel study and produces an applied machine learning model that utilizes advancements in AI, such as Bidirectional Encoder Representations from Transformers (BERT), Neural Machine Translation (NMT), and Extreme Gradient Boosting (XGBoost), to counter adversarial machine learning (AML) attacks. The praxis examines techniques used for AML, identifies strategies to mitigate attacks, and evaluates the potential of using Transformer Models for effective prevention and detection. It focuses on identifying attack techniques such as evasion, noise, poisoning, malicious code injection, and detection, aiming to take explicit methods to nullify these attacks. Additionally, by using an ensemble methodology that integrates Large Language Models (LLMs), deep learning (DL), and decision trees, the study presents a unique approach to enhancing the security of AI defense systems. Organizations with safety and mission-critical systems can adopt this approach to build additional defense against adversarial attacks.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Acronyms

AI	Artificial Intelligence
AML	Adversarial Machine Learning
AUC	Area Under the Curve
BAE	BERT-based Adversarial Examples
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	Comma Separated Values
DHS	Department of Homeland Security
DL	Deep Learning
DoD	Department of Defense
FSGM	Fast Gradient Sign Method
FP	False Positive
FN	False Negative
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
IBM	International Business Machines Corporation
LLM	Large Language Models
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked Language Modeling
NLP	Natural Language Processing

NMT	Neural Machine Translation
NSP	Next Sentence Prediction
RNN	Recurring Neural Network
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
TF-IDF	Term Frequency-Inverse Document Frequency
U.S.	United States
USD	United States Dollar
XGBoost	Extreme Gradient Boosting

Chapter 1:Introduction

1.1 Background

The latest advancements in Artificial Intelligence (AI) have become the "new electricity" that propels the modern industrial revolution (Lynch, 2017, par.6). "These advancements in machine learning (ML), deep learning (DL), and natural language processing (NLP) have accelerated the rapid adoption of AI across various industries and integrated them into mission-critical systems for value creation" (LeCun et al., 2015, pp.2-4). As a result of this development, artificial intelligence (AI) has also become a crucial consideration of cyber defense strategies. As AI solutions become more widespread, offensive cyber operations have seen a noticeable increase (Vassilev et al., 2024, pp.11-20). These malicious activities range from the manipulation of medical images to sabotage medical diagnosis, the corruption of conversational bots to spread harmful content, making systems classify malware code as benign to infiltrate system, and the creation of misleading images that could endanger the safety of autonomous vehicles (Payne et al., 2024).

As the broader adoption of AI technologies becomes increasingly embedded in mission-critical infrastructure, robust defense mechanisms against adversarial threats become crucial to ensuring these system's safety, security, and reliability (Boutin, 2024). Adversaries are also weaponizing AI and conducting sophisticated cyber-attacks against mission-critical systems in cybersecurity, healthcare, aerospace, automotive, education, manufacturing, and many more (Payne et al., 2024). State-sponsored adversaries and various crime organizations have increased their adversarial attacks by carefully

manipulating input data or infiltrating training systems to cause harm and sabotage in ML models (Szegedy et al., 2014), which is an escalating threat to critical systems integrity and national security (Boutin, 2024).

Research in this domain has been dual-faceted, with efforts focused on crafting adversarial examples and developing simulated strategies for defense (Yamin et al., 2021). “Adversarial attacks and defenses in embedded machine learning systems have been gaining significant attention due to the rapidly growing applications of deep learning and generative artificial intelligence applications” (Wang et al., 2023, pp 1-4). Large Language Models (LLMs) based on “Deep Learning Transformer models, such as the "Bidirectional Encoder Representations from Transformers (BERT), have become a focal point in this arms race due to their state-of-the-art performance in various NLP tasks” (Devlin et al., 2019, pp. 1-2).

“Neural Transformers utilize self-attention mechanisms to capture contextual relationships within the text, allowing them to discern subtle patterns and intricacies” (Vaswani et al., 2023, pp.2-12). The discrete and high-dimensional nature of textual data presents unique challenges in creating and defending against adversarial examples. Research has shown that even advanced models like BERT are vulnerable to attacks that exploit their architecture to generate adversarial examples (Li et al., 2020). These discoveries have deepened the comprehension of model weaknesses and highlighted the need for better defenses against adversaries.

Furthermore, ensemble approaches align with the broader ML principle that combining predictions from a diverse set of models can lead to more robust overall

performance, potentially mitigating the weaknesses of individual models. (Dietterich, 2000).

The critical question is determining which deep learning Transformer models such as NMT, BERT, or traditional machine learning algorithms such as XGBoost—offer the most effective and practical approach for advancing adversarial attack detection and mitigation. This doctoral project aims to address this gap by evaluating the advantages of using neural Transformers, creating a methodology, benchmarking the performance, and developing a final ML model that can be used for detecting evasion and poison attacks on machine learning systems.

1.2 Research Motivation

The lifeblood of ML-driven mission-critical systems is the assurance of operational continuity, integrity, and security (Vassilev et al., 2024, pp.11-20). This work is motivated by the urgent necessity to enhance the cyber security of these systems against adversarial noise attacks and the injection of malicious code. These risks overshadow the safety and quality of systems that manage critical applications that safeguard human lives, such as defense systems, aerospace systems, emergency response networks, and critical infrastructure systems (energy grid, water supply, transportation network, communication infrastructure, healthcare system, etc.) (Boutin, 2024). This project is a result of the motivation to enhance the understanding of cyber attack methods and develop stronger defense mechanisms to protect ML systems. The goal is to address current vulnerabilities and also to create a new detection model. This model will enable critical systems to withstand current threats and adapt to defend against future attacks as the adversarial landscape evolves. The research focuses on finding solutions as critical as

the systems they protect, ensuring better protection against adversarial machine learning attacks.

1.3 Problem Statement

"Adversaries infiltrate Machine Learning (ML) models, data, and code repositories to conduct next-gen cyber-attacks and breach into critical systems (Cohen, 2024), resulting in an average loss of USD 4.45 million in 2023 per incident, a 15% increase over three years (IBM, 2023).

The adversaries have refined their tactics by targeting ML Models and source repositories, initiating a new wave of sophisticated cyber-attacks targeting critical systems' core (Cohen, 2024). The ramifications of these breaches are far-reaching, as highlighted in the "Cost of a Data Breach Report 2023" by IBM detailing a significant financial impact, where organizations face an average loss of USD 4.45 million per incident—a notable escalation of 15% over the previous three years (IBM, 2023). This trend accentuates the importance of effectively enhancing security frameworks to detect and prevent pulsing and near real-time attacks. The proposed research addresses this dire need by exploring the implementation of Deep Learning Neural Network Transformer Models as a fortified mechanism to detect and defend against these incursion attacks (Alqarni & Azim, 2022).

Transformer models, known for their exceptional ability to analyze and understand complex data sequences, offer promising prospects in recognizing the subtle indicators of adversarial compromise that traditional methods might overlook (Ciniselli et al., 2021). By utilizing these model capabilities, this work aims to introduce a robust tool that can adapt and detect cyber-attack intricacies and substantially reduce the likelihood

and associated costs of breaches. Integrating the latest deep learning technologies with cybersecurity strategies will mark a pivotal shift in proactive defense against sophisticated tactics deployed by adversaries.

1.4 Thesis Statement

A deep neural Transformer-based model that will detect malicious content-based attacks with greater precision to protect systems before adversaries achieve success.

The increasing skill of adversaries in manipulating machine learning models through adversarial attacks poses a significant challenge.(Cohen, 2024). In response, this work proposes using a deep neural Transformer-based model to detect and neutralize such content-based attacks with improved precision. The research will develop a robust defense mechanism based on the strengths of Transformer's language understanding and architecture (Ciniselli et al., 2021). The model uses historical attack patterns and data augmentation to protect systems from new and known adversarial exploits. It will contribute to preventing growing financial losses, such as the average loss of USD 4.45 million per incident in 2023, which is a 15% increase over the past three years.(Alqarni & Azim, 2022; IBM, 2023). The deliverable is a performance-evaluated model, fine-tuned with detection capabilities, offering organizations a strong tool to reinforce their defenses against these new-generation AMLs.

1.5 Research Objectives

The main goal of this research is to create a deep-learning neural Transformer model that can detect and prevent adversarial machine-learning attacks on mission critical systems. The approach uses data augmentation and model extraction to analyze

historical attack patterns. It leverages deep learning Transformer architectures such as BERT, NMT, and XGBoost to improve the prediction capabilities for detecting AML attacks. The outcome is a thoroughly evaluated model that organizations can use to enhance their defenses against evolving adversarial attacks.

1.6 Research Questions and Hypotheses

In machine learning, the increasing incidence of adversarial attacks has highlighted an urgent need for effective detection mechanisms (Boutin, 2024). This research factors in the pivotal questions aimed at distilling the efficacy of various deep learning Transformer models in cybersecurity.

Research Question 1 (RQ1) delves into the effectiveness of Deep Learning Transformer-Based Models for detecting machine learning-based adversarial attacks. The hypothesis (H1) posits that BERT, with its bidirectional architecture, will outperform Neural Machine Translation (NMT) systems in recognizing such attacks due to its enhanced ability to process diverse textual contexts (Devlin et al., 2019).

Research Question 2 (RQ2) compares the accuracy of Transformer-Based Models against traditional machine learning algorithms such as XGBoost in detecting code injection and noise-based attacks. The associated hypothesis (H2) suggests that Transformer-Based Models will exhibit superior accuracy, given their advanced capacities for interpreting complex and previously unseen data patterns (Chen & Guestrin, 2016).

Finally, Research Question 3 (RQ3) explores whether a hybrid model combining the contextual comprehension of BERT with the learning efficiency of XGBoost can enhance precision and accuracy in predicting adversarial attacks. Hypothesis (H3)

conjectures that such a hybrid approach will transcend the capabilities of a singular BERT model in detecting malicious content. This study aims to contribute to the developing yet critical areas of cybersecurity in artificial intelligence, seeking to strengthen defense mechanisms against the sophisticated and ever-evolving landscape of digital threats.

RQ1: Which of the Deep Learning Transformer-Based Model provides the effective and feasible method for machine learning-based adversary detection (BERT, NMT)?

RQ2: Which among the Transformer-Based Models, (BERT and NMT) vs the classic model approach of XGBoost is more effective when detecting adversary attacks such as poisoning, evasion, and noise?

RQ3: Will the ensemble/hybrid model approach further improve the precision and accuracy of a Transformer-based models to predict adversarial attacks?

H1: Among Deep Learning Transformer-Based Models, BERT outperforms NMT in identifying ML-based adversarial attacks due to its bidirectional architecture's superior ability to understand diverse text contexts.

H2: Transformer-based models surpass XGBoost in accuracy when detecting adversarial attacks, including noise and malicious code, by excelling at processing and interpreting intricate, unseen text patterns.

H3: BERT's contextual insights with XGBoost's learning efficiency an ensemble or hybrid model will exceed the precision and accuracy of a single BERT model in detecting malicious content.

1.7 Scope of Research

The scope of the work is to strengthen the security of mission-critical systems that are integrated with advanced machine learning capabilities. “By harnessing the capabilities of deep neural Transformer models, such as BERT” (Devlin et al., 2019, pp. 2-4). The goal is to identify and mitigate adversarial text-based attacks as they occur, enabling near real-time defensive responses to such cyber threats. This proactive approach is crucial for protecting sensitive infrastructures from the sophisticated and rapidly evolving methods cyber adversaries employ.

1.8 Research Limitations

The goal of this project is to enhance the security of critical systems by utilizing advanced deep neural Transformer models, such as BERT, to rapidly detect text-based adversarial attacks near real-time. These models provide advanced contextual text analysis for threat mitigation, the study recognizes that there are inherent limitations. “The complexity of machine-learning environments requires sophisticated and adaptable models capable of addressing the subtleties of cyber threats” (Devlin et al., 2019, pp.1-2). “Machine-learning-powered networks can be susceptible to adversarial manipulations” (Szegedy et al., 2014). This work faces the challenge of creating a strong security system that can withstand evolving threats. It also needs to consider the possibility of new zero-day attack methods that may not be immediately detectable. This indicates a continuous need to improve and adapt the security system to counter fast evolving adversarial strategies (Alqarni & Azim, 2022).

1.9 Organization of Praxis

The organization of this praxis is systematically structured to facilitate a coherent flow from theoretical underpinnings to practical applications. It begins with an Introductory chapter setting the stage for the research, followed by a literature review that contextualizes the work within the existing body of knowledge. Subsequent chapters outline the methodology, present the deep learning neural Transformer architectures and models utilized, and outline the development cycle of the experiments. A detailed analysis of the results from the model experimentation is conducted subsequently, followed by an analysis of the effectiveness of the models in detecting adversarial attacks. The results chapter outlines the significant implications of the findings, their contribution to the field of cybersecurity, and the potential for real-world application in safeguarding mission-critical systems. Finally the chapter wraps up with review and summary conclusions of the research, reflections on the limitations, suggestions for future research directions, and a closing commentary on the broader impact of the work. Appendices and supporting material are provided for additional context and detail, ensuring a comprehensive resource for scholars and practitioners in cybersecurity.

Chapter 2: Literature Review

2.1 Introduction

"Artificial Intelligence has emerged as a catalyst for a new era of enhanced human productivity in the evolving technology landscape" (Daly, 2023, p1). This created a frontier of new-generation AI technologies that digitally transform how we build and run all products and services. The emergence of AI applications in computer vision, Generative AI, generative pre-trained Transformers (GPT), and autonomy has transformed the world by digitally revolutionizing industries, improving human-machine interactions, and driving widespread adoption of applications with embedded AI (Janjeva et al., 2023). However, this progress has come with challenges. "One of the most critical concerns is the vulnerability of these models to adversarial attacks" (Payne et al., 2024, pp.2-4). These attacks involve intentionally crafted changes that can mislead machine learning models into making inaccurate predictions, classifications, or even shutting down. "It is urgent and crucial to develop robust defense mechanisms to prevent such attacks on the current generation of intrusion detection systems (IDS) and intrusion prevention systems (IPS)" (Thummala et al., 2024. Pp.1-2).

Recent literature has shown an increasing focus on using Transformer models and adversarial examples to identify vulnerabilities and develop advanced techniques to prevent such attacks (Boutin, 2024). Furthermore, "Garg and Ramakrishnan's study on BERT-based adversarial examples for text classification demonstrates the various ways attackers can manipulate inputs to exploit model weaknesses" (Garg & Ramakrishnan, 2020, pp.1-2). This highlights the need for strong defense mechanisms. Concurrently,

recent research efforts have ventured into leveraging the same Transformer architectures to generate and defend against adversarial examples, indicating a dual use of this technology in the arms race between attackers and defenders.

This literature review aims to synthesize the current body of research surrounding detecting and mitigating adversarial attacks using deep learning neural network Transformer models, with a particular focus on textual data. By examining methodologies ranging from adversarial example generation, such as the BAE (BERT-based Adversarial Examples) technique, defense strategies employ Transformer models for improved security. The work of Praxis aims to offer a thorough overview of the current methods for improving model resilience against adversarial threats. It seeks to highlight the advantage of using Transformer models for their exceptional language processing capabilities and protecting them from exploitation. In the end, this work plays a crucial role in establishing that AI and ML ecosystems is safe, secure and dependable.

2.2 Adversarial Attacks

Adversarial machine learning attacks present a significant challenge to the robustness and security of machine learning (ML) and deep learning (DL) models (Payne et al., 2024). These models, which are increasingly utilized to enhance the detection and mitigation of cyber threats, are themselves vulnerable to various forms of adversarial manipulation. Understanding the different types of adversarial attacks is crucial for developing effective defense strategies.

1. **Poisoning Attacks**

   Poisoning attacks target the training phase of ML and DL models. In these attacks, “an adversary injects malicious data into the training dataset, thereby

corrupting the learning process" (Sheikh et al., 2023, p1). The primary goal of a poisoning attack is to disrupt the model to learn incorrect patterns, leading to degraded performance or intentional misclassification of certain inputs. Poisoning can be achieved by altering the label of data points, introducing noise, or adding entirely fabricated data that skews the model's learning process. These attacks are particularly insidious because they can significantly impact the model's decision-making ability without being immediately apparent, making them difficult to detect and mitigate (Sheikh et al., 2023).

2. **Evasion Attacks**

   Evasion attacks occur during the testing or inference phase of ML and DL models(Sheikh et al., 2023). Unlike poisoning attacks, evasion attacks do not modify the training data but instead, craft inputs that are intentionally designed to bypass the model's detection mechanisms. These inputs, often referred to as "adversarial examples, are typically generated by applying small perturbations to legitimate data" (Anjaria & Shah, 2024, p.2) points that cause the model to misclassify them. The perturbations are often imperceptible to human observers but are sufficient to exploit vulnerabilities in the model's learned decision boundaries. Evasion attacks are particularly impactful in real-time applications, such as intrusion detection systems, where attackers can gain the ability to bypass filters from detecting malicious content (Sheikh et al., 2023).

3. **Exploratory Attacks**

   Exploratory attacks, also known as inference attacks, aim to gain information about the ML model itself (Sheikh et al., 2023). These attacks do not require

access to the training data but instead involve probing the model with various inputs to understand its behavior and vulnerabilities. The adversary can then use this information to craft more effective evasion or poisoning attacks. Exploratory attacks can also lead to model extraction, where the adversary attempts to replicate the functionality of the target model by observing its outputs. This type of attack is particularly concerning in scenarios where the model is proprietary or contains sensitive intellectual property (Sheikh et al., 2023).

The various types of adversarial attacks—poisoning, evasion, and exploratory—pose significant challenges to the security of ML and DL models and systems. A comprehensive strategy, including adversarial training, threat modeling, and the implementation of robust defense mechanisms, is required to address these issues and ensure the integrity and reliability of these critical systems.

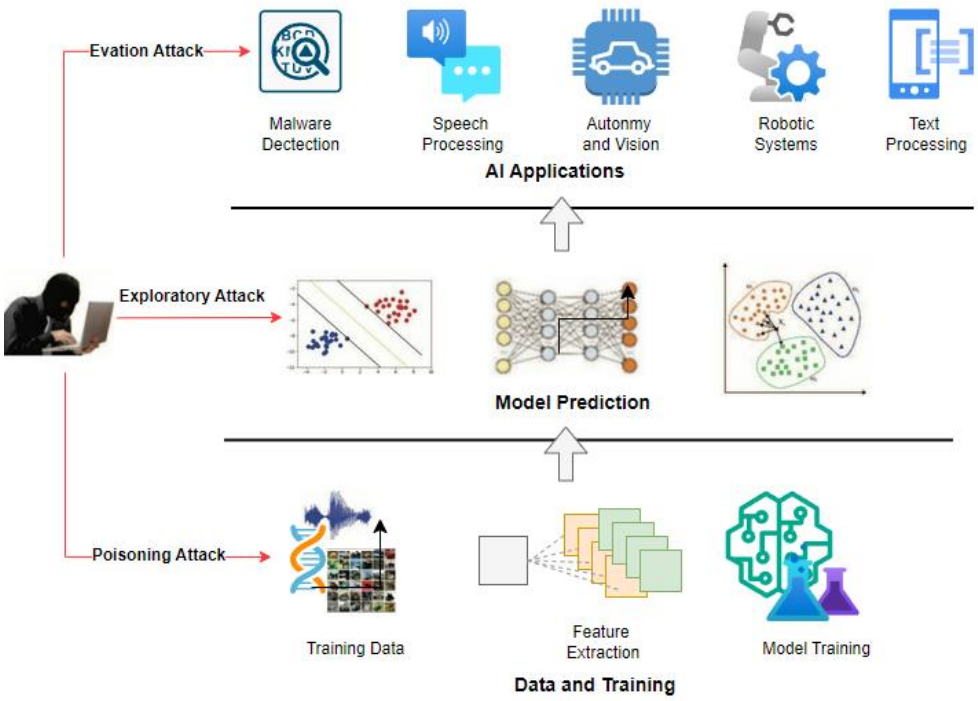

Figure 2.1 – Adversarial Machine Learning (AML) Attack Surface

2.3 Adversarial Defense

"Adversarial Machine Learning (AML) has emerged as a critical area of study as machine learning models become increasingly integrated into sensitive and mission-critical applications" (Sheikh et al., 2023, pp.1-2). Attackers can take advantage of weaknesses in ML models by adding small undetectable alterations to input data, causing the models to produce incorrect predictions or classifications. To mitigate these risks, a variety of defensive strategies have been developed, each with their own strengths and limitations. This section provides a detailed breakdown of these defenses, categorized by their approach to enhancing model robustness.

1. **Adversarial Training**

   "One of the most straightforward and widely studied defensive strategies is adversarial training" (Sheikh et al., 2023. pp.1-16). This approach includes expanding the training dataset with adversarial sample inputs intentionally modified to mislead the model. "By training on these adversarial examples, the model learns to recognize and resist similar attacks during inference, demonstrating the effectiveness of adversarial training in improving model robustness, particularly against gradient-based attacks like the Fast Gradient Sign Method (FGSM)" (Goodfellow et al., 2015, pp.1-3). Adversarial training demands significant computational resources because it involves creating adversarial samples and repeatedly retraining the model. Furthermore, this technique may have difficulty adapting to new adversarial tactics not encountered during training (Kurakin et al., 2017). As a result, the model will not scale to generalization.

2. **Gradient Masking and Obfuscation**

Gradient masking, also known as gradient obfuscation, is another defensive approach that aims to obscure the gradients used by adversaries to craft adversarial examples (Sheikh et al., 2023). By making the gradients less informative or more difficult to compute, the model becomes harder to attack using gradient-based methods introduced by defensive distillation, a technique that leverages gradient masking by training a model to produce soft labels, which are then used to train distilled version of the original model (Papernot et al., 2017). This distillation process results in a model that is less sensitive to small perturbations. However, gradient masking has been criticized for providing a false sense of security, demonstrating that attackers can often circumvent these defenses by using more sophisticated techniques, such as adaptive attacks that exploit the masked gradients (Athalye et al., 2018).

3. **Input Preprocessing and Transformation**

An alternative method to defend against AML attacks is to preprocess or transform the input data before adding it to the model. The goal is to eliminate or minimize any adversarial perturbations, thereby neutralizing potential attacks. Feature squeezing and input-denoising methods are two approaches to enhance the robustness of machine learning models against these types of attacks (Xu et al., 2018). Feature squeezing involves simplifying the input data through operations such as reducing the bit-depth or applying spatial smoothing. This can help reduce the complexity of the input data and make it more resilient to potential attacks (Xu et al., 2018). Input-denoising methods involve using

techniques such as JPEG (Joint Photographic Experts Group) compression or image cropping to remove adversarial noise from images before they are processed by the model (Guo et al., 2018). While these methods can be effective, they may also reduce the quality of original inputs, potentially decreasing model accuracy.

4. **Certified Defenses and Robust Optimization**

   Certified defenses represent a more recent approach to AML, providing formal guarantees about a model's robustness against types of adversarial attacks (Madry et al., 2019). Robust optimization involves training a model to perform well with clean data, and also under worst-case adversarial conditions (Madry et al., 2019). These approaches typically involve solving "a min-max optimization problem, where the model is optimized to minimize its loss" (Kundu et al., 2022, p.2) under the most challenging adversarial perturbations. While robust optimization can significantly enhance a model's resilience, it is computationally demanding and may reduce accuracy on non-adversarial data.

5. **Data Sanitization and Poisoning Defenses**

   Adversarial attacks can happen not only during the inference phase, but also during training and may involve data poisoning (Sheikh et al., 2023). To protect against such attacks, data sanitization techniques can be used to detect and remove potentially harmful inputs before they disrupt the training process. Defense processes involve exploring various sanitization methods, such as outlier detection and clustering, to identify and exclude poisoned data points (Steinhardt et al., 2017). Additionally, certified defenses against poisoning attacks aim to

ensure the model predictions remain stable even when the training data has been tempered by an adversary. (Raghunathan et al., 2020).

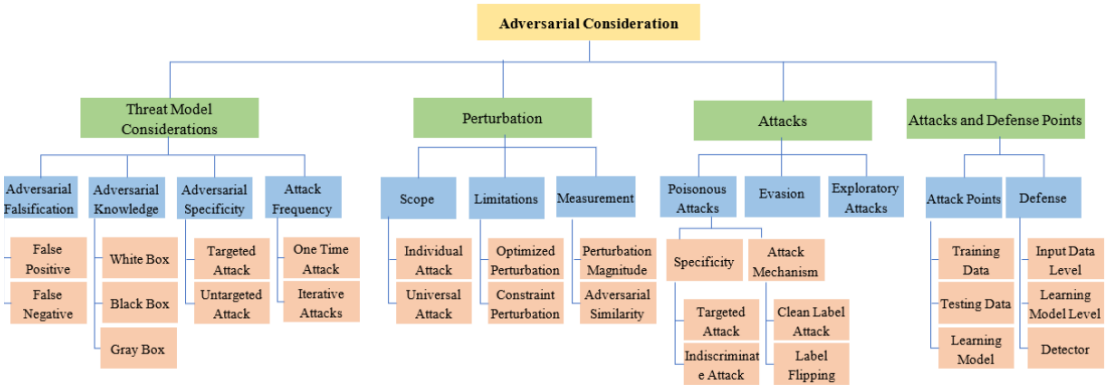

Figure 2.2 – Adversarial Defense Considerations
(Source:Sheikh et al, 2023)

Although these defensive strategies help reduce the impact of adversarial attacks, no single method can guarantee complete protection. The effectiveness of these defenses often relies on the particular attack scenario and the model architecture. As adversarial techniques continue to evolve, it will likely be necessary to use a combination of multiple defenses to secure machine learning models against increasingly sophisticated threats. Ongoing research into more robust and adaptive defensive strategies is crucial for the future of secure AI systems.

2.4 What are BERT and NMT, and How Do They Work?

BERT is a revolutionary model developed by Devlin et al. (2019), which has profoundly impacted various fields within NLP, including neural machine translation (NMT) (Garg & Ramakrishnan, 2020). BERT's core innovation lies in its use of the transformer architecture allowing for a bidirectional understanding of words within a sentence, enabling better comprehension of their context. BERT analyzes the context of

each word in both directions, forward and backward, offering a deeper understanding of language nuances compared to earlier models that only processed text in one direction (Devlin et al., 2019).

In the domain of NMT, where the goal is to “translate content from one to another language while preserving meaning and context” (Belinkov & Bisk, 2018, pp. 8), BERT can significantly enhance translation models. While NMT is a translator model, BERT itself is not a translation model, it is a large language model (LLM) (Devlin et al., 2019). Its deep understanding of language semantics and context makes it a valuable asset in several ways:

- Pre-trained Contextual Embeddings: BERT's pre-trained model offers rich contextual embeddings that can be used to initialize or enhance the encoder components of NMT systems. This can improve the “NMT model's understanding of the source language, leading to more accurate and contextually appropriate translations” (Li et al., 2020, pp.1-3).
- Data Augmentation: BERT can generate linguistically diverse paraphrases of sentences or translate sentences into intermediate languages as a form of data augmentation for NMT training. This can enrich the training data and help the NMT model learn to deal with a broader range of linguistic variations and complex sentence structures (Li et al., 2020).
- Adversarial Training: As demonstrated in studies like "BERT-ATTACK: Adversarial Attack Against BERT Using BERT", BERT can be employed to create adversarial examples that mimic potential translation errors or ambiguities. By training NMT models against these adversarial examples, the

models can become more robust, enhancing their ability to handle edge cases and improving overall translation quality (Garg & Ramakrishnan, 2020).

- Semantic Similarity and Evaluation: BERT's ability to assess semantic similarity can be leveraged to refine NMT output by re-ranking generated translations based on their semantic closeness to the source text. Additionally, BERT's embeddings can be used to develop more nuanced metrics for evaluating translation quality, more effectively considering aspects like fluency, coherence, and fidelity (Li et al., 2020).

Moreover, BERT's significance goes beyond just understanding context. It showcases the dual potential of Transformer models in both generating and defending against adversarial attacks in NMT (Garg & Ramakrishnan, 2020). By comprehending and harnessing BERT's capabilities, researchers and practitioners can create NMT systems that are highly efficient, accurate, and resistant to adversarial manipulation. This ensures the integrity and reliability of machine translations across a wide range of applications.

2.5 Challenges in Using Transformers for Adversarial Attack Detection

Integrating NMT and BERT into adversarial attack detection presents both opportunities and significant challenges (Garg & Ramakrishnan, 2020). The challenges in leveraging these technologies for adversarial attack detection are multifaceted, ranging from the inherent complexities of NMT and BERT models to the nuances of adversarial threats in natural language processing (NLP).

1. **Contextual Understanding and Perturbations**: “The BERT-based Adversarial Examples (BAE) study” (Gupta et al., 2022) demonstrates BERT's ability to

generate adversarial examples by leveraging its deep contextual comprehension. Yet, this strength also presents a challenge in detecting adversarial attacks. The nuanced, context-sensitive perturbations that BERT and similar models can generate or identify closely resemble the natural variability in human language, making the detection of such adversarial manipulations more complex.

2. **Model Robustness and Generalization**: The study "BERT-ATTACK: Adversarial Attack Against BERT Using BERT" illustrates the susceptibility of advanced models like BERT to adversarial manipulations, indicating a gap in model robustness (Li et al., 2020, pp.1-9). Enhancing these models to detect adversarial attacks requires understanding the nature of these attacks and significantly improving the models' ability to generalize from seen to unseen adversarial tactics without compromising the performance on genuine tasks.
3. **Computational Complexity and Efficiency**: The advanced capabilities of BERT and NMT models require significant computational resources. Processing and analyzing large datasets for adversarial attack detection, especially in real-time applications, poses challenges in terms of computational efficiency and resource allocation (Li et al., 2020).
4. **Adaptability to Adversarial Evolution**: Adversarial techniques are constantly evolving and becoming more sophisticated (Li et al., 2020). A key challenge here is ensuring that the NMT and BERT models used for attack detection are adaptable and can be quickly updated to recognize new types of attacks. This necessitates continuous monitoring and updating of the models, which can be resource-intensive.

5. **Trade-off Between Sensitivity and Specificity**: Increasing a model's sensitivity to detect subtle adversarial manipulations can inadvertently lead to higher false positives, where legitimate inputs are mistakenly flagged as adversarial (Garg & Ramakrishnan, 2020). Balancing sensitivity and specificity are a perennial challenge in designing effective detection systems that do not disrupt the user experience or hinder legitimate linguistic variations.
6. **Language and Cultural Nuances**: NMT and BERT models, despite their advancements, still need to work on capturing the full spectrum of linguistic and cultural nuances across different languages (Alqarni & Azim, 2022). This limitation can hinder their effectiveness in detecting adversarial attacks that exploit such nuances to create ambiguities or misleading contexts in translations or text-processing tasks.
7. **Data Scarcity and Bias**: Training models to detect adversarial attacks requires access to comprehensive datasets of adversarial examples (Li et al., 2020). However, the need for such datasets and potential biases in available data can limit the models' learning and effectiveness in applications.

Overcoming these challenges requires advancements in model architecture, training methods, and a comprehensive approach that considers computational efficiency, model adaptability, and the ethical implications of automated detection systems.

2.6 How Does an NLP-Based Adversarial Attack Detection Model such as BERT Address the Shortcomings of Conventional Methods?

This literature review explores the advantages of using natural language processing (NLP) based models for identifying adversarial attacks, focusing on how they

address the limitations of traditional detection techniques." Recent advancements in deep learning neural Transformer models such as BERT and NMT" (Garg & Ramakrishnan, 2020, pp.1-7) could be leveraged to develop new approaches for improved detection of these adversarial threats.

Traditional methods for identifying adversarial attacks in NLP systems have typically relied on rule-based algorithms or statistical models. While somewhat effective, these approaches have significant limitations in terms of adaptability and depth of linguistic understanding. They often struggle to capture the nuanced semantics of language, leaving them vulnerable to sophisticated adversarial techniques designed to exploit these weaknesses (Garg & Ramakrishnan, 2020). In contrast, "Transformer models such as BERT" (Filighera et al., 2022), developed by Devlin et al. (2019), take a fundamentally different approach. By leveraging extensive textual data, BERT gains a deep, contextual understanding of language, enabling it to detect subtle manipulations that are indicative of adversarial attacks, which conventional methods might miss (Li et al., 2020).

A major advantage of using BERT for detecting adversarial attacks is its capability to process and understand text in both directions, taking into account the complete context of each word in a sentence. This is in contrast to earlier models that processed text in a linear fashion, often overlooking the broader context that could indicate an adversarial manipulation (Garg & Ramakrishnan, 2020). The "BERT-based Adversarial Examples for Text Classification" study illustrates how BERT's comprehensive language model, which includes "pre-training on a broad linguistic corpus

and fine-tuning for specific tasks", enables it to identify anomalies in text that deviate from expected linguistic patterns (Li et al., 2020, pp.1-9).

Furthermore, the research on "BERT-ATTACK: Adversarial Attack Against BERT Using BERT" emphasizes another important advantage: BERT's ability to generate and detect adversarial examples (Li et al., 2020, pp.1-9). This dual capability helps identify potential weaknesses in NLP systems and improves the development of more robust adversarial detection mechanisms. By utilizing BERT's understanding of complex language structures, researchers and practitioners can create strategies to anticipate and counter sophisticated adversarial tactics, allowing them to circumvent traditional detection systems.

Moreover, Transformer models have the ability to continuously learn and adapt, which sets them apart from static, rule-based adversarial detection methods (Devlin et al., 2019). This adaptability ensures that previously unseen adversarial techniques can be detected, providing a strong defense against an ever-changing threat landscape. Transformer-based models like BERT and NMT could address many limitations of traditional methods, providing better adaptability, a deeper understanding of language, and enhanced capabilities to predict and counter sophisticated adversarial attacks (Li et al., 2020). As these technologies continue to advance, their integration into cybersecurity frameworks holds the promise of significantly strengthening defenses against complex adversarial threats.

2.7 Summary and Conclusion

To sum up, incorporating advanced Deep Learning Neural Transformer Models like BERT shows great potential for improving the detection of adversarial attacks. This

literature review consolidates findings from recent research papers to emphasize the benefits and possibilities of using NLP-based models for detecting adversarial attacks.

1. **Adversarial Vulnerabilities and NLP Systems**: Machine learning models are susceptible to adversarial examples, which create significant challenges. “Garg and Ramakrishnan (2020) demonstrate how BERT-based adversarial examples can exploit vulnerabilities in text classification models” (Garg & Ramakrishnan, 2020, pp.1-7). This highlights the importance of robust detection systems. These adversarial tactics, often undetectable by humans, require advanced detection mechanisms capable of understanding complex linguistic constructs and nuances.
2. **Enhancing NLP with Deep Learning Models**: Utilizing Transformer models like BERT in adversarial attack detection presents a new frontier in NLP (Devlin et al., 2019). Transformer models' ability to process and understand language at a deeper level than conventional methods allow for more effective identification and mitigation of adversarial attacks (Garg & Ramakrishnan, 2020). The literature indicates a significant shift towards adopting these models for their superior contextual understanding and predictive capabilities.
3. **Challenges and Opportunities in Adversarial Detection**: Despite the advancements, several challenges persist in effectively deploying Transformer models for adversarial attack detection. The complexity of these models, along with the requirement for a large amount of training data, creates operational and logistical challenges. However, the continuous development and improvement of these models provide promising solutions to these challenges, as demonstrated by the collaborative research efforts.

4. **Future Directions**: The convergence of deep learning, NLP, and blockchain technology marks a significant evolution in detecting and mitigating adversarial attacks. Future research is poised to explore these integrated approaches' scalability, efficiency, and applicability across diverse digital ecosystems, further bolstering cybersecurity defenses.

The literature underscores the transformative potential of incorporating Deep Learning Neural Transformer Models into adversarial attack detection frameworks. The field is moving towards more secure, efficient, and collaborative cybersecurity mechanisms by leveraging the advanced linguistic comprehension and predictive accuracy of models like BERT alongside innovative approaches like decentralized, federated learning, and blockchain technology. The ongoing exploration and development in this space signals a promising future for the detection and mitigation of adversarial threats in cybersecurity.

Chapter 3: Methodology

3.1 Introduction

This chapter presents the methodology for building a new ML model for detecting adversarial attacks using the Deep Learning Neural Transformer architecture. The methodology is designed to systematically investigate the effectiveness of BERT, NMT, and XGBoost models in detecting different types of malicious content-based attacks, such as adversarial evasion and poison attacks. The approach integrates “thorough data collection, preprocessing, model training, evaluation, and comparative analysis to ensure robust and reliable” (Soumya et al., 2024) results.

The first step involves collecting and preprocessing diverse datasets encapsulating adversarial attacks. These datasets, sourced from publicly available repositories and synthetic data generation, “provide a rich ground for training, validation, and testing the models. Preprocessing steps include data cleaning, tokenization” (Wang, 2023, p.7), and normalization to prepare the datasets for effective model training (Nazir et al., 2024). “The data is then split into training, validation, and test sets” to facilitate unbiased model evaluation and benchmark performance (Nazir et al., 2024).

“Following this, a set of machine learning models is trained and assessed” (Dagur et al., 2024). The main models being considered are BERT and NMT for detecting adversarial attacks and XGBoost for comparison. Each model underwent thorough training on preprocessed datasets, and its performance “is evaluated based on key metrics such as precision, recall, F1 score, and accuracy” (Khedkar, 2024, p. 5). This approach targets assessing each model's strengths and limitations.

Furthermore, statistical analysis was utilized “to compare the performance of the different models to validate the hypotheses” (FasterCapital, n.d.-a). This step ensures that the conclusions drawn are based on solid statistical evidence, adding credibility to the findings. Finally, the results were meticulously documented during the entire process, including code and datasets, which are shared to ensure reproducibility and transparency. This comprehensive methodology aims to enhance the detection of adversarial attacks and seeks to provide valuable insights into the effectiveness of hybrid models in cybersecurity analytics.

3.2 Large Language Models (LLMs) and Transformers to Detect Adversarial Attacks

LLMs such as BERT and NMT have significant potential for detecting adversarial attacks. These models can leverage Transformer architecture to understand and generate text with high contextual accuracy, making them suitable for identifying subtle adversarial manipulations. Adversarial attacks involve slight modifications to input text intended to mislead models into incorrect predictions. LLMs such as BERT and machine translators such as NMT can be fine-tuned to detect these manipulations by training them on datasets containing both clean and adversarial examples. Transformers’ self-attention mechanisms dynamically assess the significance of each word, algorithmically capturing intricate dependencies and context within the text (Devlin et al., 2019). This capability helps to identify inconsistencies or perturbations introduced by adversarial attacks.

In practice, techniques like “BERT-Attack exploit Transformer language modeling abilities to generate adversarial examples” (Li et al., 2020, pp.1-9). The models self-train to recognize and mitigate adversarial threats by incorporating these examples

into the training process (Li et al., 2020). Additionally, ensemble model approaches that combine Transformers with traditional machine learning algorithms further enhance detection accuracy and robustness.

3.3 Bidirectional Encoder Representations from Transformers (BERT)

This section outlines the methodology used for BERT to detect adversarial attacks with greater precision, addressing this thesis statement: "A deep neural Transformer-based model that will detect malicious content-based attacks with greater precision to protect systems before adversaries achieve success."

“BERT is a groundbreaking deep learning model developed by Google that leverages Transformer architecture” (Devlin et al., 2019, p.1). Unlike traditional NLP models, which process text in a unidirectional manner, BERT processes text bi-directionally. This allows BERT to understand a word's context based on its preceding and succeeding words, providing a richer understanding of language. “BERT's architecture consists of multiple layers of bidirectional Transformers, enabling it to capture intricate patterns in the text” (Li et al., 2020, pp.1-9).

3.3.1 BERT's Architecture

BERT's algorithm is based on the Transformer architecture introduced by Vaswani et al. (2023). “Transformers use self-attention mechanisms to dynamically weigh the importance of different words in a sentence” (Choi & Lee, 2023, p.10). This self-attention mechanism allows “BERT to focus on relevant parts of the input text, making it highly effective in capturing semantic meaning and contextual relationships” (Shachar, 2024). The architecture consists of two main parts:

- Encoder - reads the text input.
- Decoder - prediction for tasks given.

"BERT specifically utilizes only the encoder component of the Transformer to generate deep bidirectional representations from the unlabeled text by joint conditioning on both the left and right context in all layers" (Li et al., 2020, pp.1-9).

3.3.2 How BERT Works

Here's how BERT works:

I. Transformer Architecture:

a. **Bidirectional Encoder**: The diagram in Figure 3.1 illustrates the foundation architecture of BERT. "BERT uses the left-side encoder of the Transformer to process input text" (M, 2023). It achieves this by applying "multiple layers of multi-head self-attention and feed-forward networks" (Pussadeniya, 2023), each followed by add-and-normalization layers. BERT incorporates positional encoding to capture the sequence of input tokens and allows bidirectional training, meaning it processes input text in both directions. Unlike traditional models, BERT masks certain input tokens during training to ensure the model learns word dependencies in context. The attention mechanism in BERT makes it efficient at understanding "relationships between words, making it particularly effective for tasks like question answering and sentiment analysis" (Thakur et al., 2024).

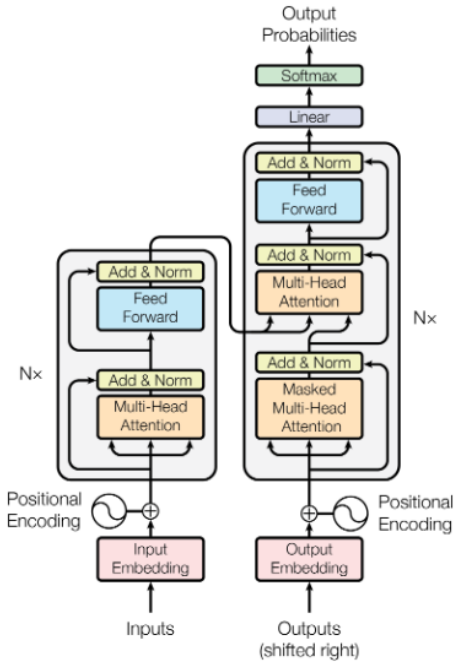

Figure 3.1 – The Transformer Model Architecture
Source: (Vaswani et al., 2023)

II. Pre-training:

a. **Masked Language Modeling (MLM)**: During pre-training, "BERT is trained on a large corpus of text to predict missing words in a sentence" (Shachar, 2024). In MLM, random words within the input sentence are masked, and the model tries "to predict these masked words based on the context provided by the other words in the sentence" (El Enany, 2024). This enables BERT to learn deep contextual relationships between words (Shachar, 2024).

b. **Next Sentence Prediction (NSP)**: Another pre-training task for BERT is NSP, "where the model is given pairs of sentences and tasked with predicting whether the second sentence follows the first one in the original text" (El Enany, 2024). This helps BERT understand sentence-level relationships and coherence.

III. **Fine-tuning**:

a. **Task-Specific Training**: “After pre-training, BERT is fine-tuned on specific tasks using labeled datasets. During fine-tuning, the pre-trained BERT model is adjusted to perform text classification” (Urzola et al., 2023), question answering, or named entity recognition tasks. The same architecture is used, but with additional task-specific layers added to the model (Devlin et al., 2019).

b. **End-to-End Learning**: “Fine-tuning involves training the entire model end-to-end on the task-specific data” (Devlin et al., 2019, p.2), allowing BERT to adapt its parameters to the nuances of the new task while retaining the deep contextual knowledge gained during pre-training (Devlin et al., 2019).

IV. **Self-Attention Mechanism**:

a. **Attention Scores**: BERT utilizes self-attention to calculate attention scores for each word in the input sentence, evaluating its relative significance. This enables BERT to focus on crucial words and phrases, effectively capturing dependencies that may be far apart in the text (Devlin et al., 2019).

b. **Multi-Head Attention**: BERT employs multiple attention heads that operate simultaneously. Each attention head captures distinct aspects of word relationships, leading to a deeper understanding of the context. (Devlin et al., 2019).

V. **Layer Stacking**:

a. **Deep Layers**: “BERT consists of multiple layers (12 for BERT-base and 24 for BERT-large) of Transformer encoders” (Kaddour et al., 2023). Each layer processes the input text representations and refines them based on the attention mechanisms. The outputs of these layers are progressively refined contextual embeddings that capture complex patterns in the text (Kaddour et al., 2023).

VI. **Positional Encoding**:

a. **Sequential Information**: “Since Transformers do not inherently understand the order of words, BERT incorporates positional encodings to inject information about the position of each word in the sentence” (Pussadeniya, 2023). This helps the model understand the sequential nature of the text.

VII. **Training and Optimization**:

a. **Large-Scale Data**: “BERT is pre-trained on large-scale corpora, such as Wikipedia and BooksCorpus” (Lee et al., 2020, p.19), which enables it to learn a wide range of linguistic patterns and knowledge.

b. Adam Optimizer: The training process uses the Adam optimizer, which “adapts the learning rate based on the training process” (Shachar, 2024), ensuring efficient and effective convergence (Kingma & Ba, 2017).

By combining these elements, “BERT achieves state-of-the-art performance on a wide range of NLP tasks” (Devlin et al., 2019, pp 1.4). Its bidirectional context understanding, deep architecture, and fine-tuning capabilities make it a powerful tool for

tasks that require a nuanced understanding of language, such as detecting adversarial attacks in text data (Li et al., 2020).

3.3.3 AI Techniques Utilized by BERT

BERT employs several advanced AI techniques, including:

- **Self-Attention Mechanism**: "This allows the model to focus on different parts of the sentence as needed, capturing complex dependencies within the text" (Shao et al., 2023, p.18).
- **Transfer Learning**: BERT "is initially trained on a large dataset and subsequently fine-tuned for particular tasks" (Garg & Ramakrishnan, 2020, pp.1-7), which makes it highly adaptable and powerful for a range of NLP applications (Garg & Ramakrishnan, 2020).
- **Bidirectional Contextual Understanding**: Unlike traditional models, BERT looks at the context from both directions, which helps in understanding the nuanced meaning of words and phrases (Li et al., 2020).

3.3.4 Using BERT for Detecting Adversarial Attacks

In adversarial attacks on ML model involve stealth modifications to input data aimed at misleading models into making incorrect predictions. (Garg & Ramakrishnan, 2020). In the field of NLP, these attacks can involve tweaking text to change its meaning or context. BERT, with its strong contextual understanding, can be used to identify and defend against such adversarial attacks.

I. Training and Fine-Tuning BERT for Adversarial Detection

The methodology involved fine-tuning BERT on datasets containing both clean and adversarial examples. The fine-tuning process helped BERT to learn and distinguish between legitimate inputs and adversarial manipulations. Key steps included:

- **Data Collection and Preprocessing**: Preprocessed datasets containing examples of adversarial attacks were used, ensuring a balanced representation of clean and adversarial samples.
- **Model Training**: BERT was fine-tuned on the preprocessed datasets, optimizing it to detect subtle adversarial manipulations. Techniques such as data augmentation and adversarial training enhanced the model's robustness (Li et al., 2020).
- **Evaluation**: The effectiveness of the model was measured using metrics like precision, recall, F1 score, and accuracy. A comparison with traditional machine learning algorithms, such as XGBoost and Random Forest, highlighted BERT's improved performance in detecting adversarial attacks (Li et al., 2020).

3.4 Neural Machine Translation (NMT)

The methodology used to employ NMT models to detect adversarial attacks with greater precision is outlined, which addresses the thesis statement. NMT is a type of artificial intelligence model used for translating text from one language to another. Unlike conventional statistical and rule-based translation methods, NMT employs deep learning techniques to produce translations that are accurate and contextually relevant.

3.4.1 NMT Architecture

NMT operates using an encoder-decoder architecture enhanced by attention mechanisms. “The encoder, made up of multiple neural network layers, processes the input text and converts it into continuous vector representations. The decoder then generates the translated text from these vectors. A pivotal innovation in NMT is the attention mechanism, which allows the model to focus dynamically on different parts of the source text, capturing context more effectively” (Belinkov & Bisk, 2018). Modern NMT systems often use Transformer architecture, which relies “on self-attention to process all words in a sentence simultaneously, improving efficiency and capturing long-range dependencies more effectively than RNNs” (Li et al., 2020, pp.1-9).

3.4.2 How NMT Works

NMT models use an encoder-decoder architecture. “The encoder processes the source text into a fixed-length vector, which the decoder then uses to generate the translated target text” (Belinkov & Bisk, 2018, pp.2-9). The core components of NMT involve:

I. Encoder-Decoder Architecture

“NMT uses an encoder-decoder architecture, which consists of two main components” (Hagos et al., 2024):

Encoder:

- **Input Processing**: “The encoder reads and processes the input sentence word by word. Each word is converted into a fixed-length vector representation using embedding techniques” (Belinkov & Bisk, 2018, pp.2-9).

- **Contextual Representation**: The encoder, often an RNN, Long Short-Term Memory (LSTM), or a Transformer, processes the word embeddings to generate a sequence of hidden states. These hidden states encapsulate the contextual information of the input sentence. (Belinkov & Bisk, 2018, pp.2-9).

Decoder:

- **Output Generation**: The decoder produces the translated sentence one word at a time. It uses the context vector from the encoder along with the words generated so far to predict the next word in the target language (Belinkov & Bisk, 2018, pp.2-9).
- **Recurrent Process**: Like the encoder, the decoder can also be an RNN, LSTM, or Transformer, which helps in generating coherent and contextually appropriate translations.

II. Attention Mechanism

This "allows the model to focus on different parts of the source sentence dynamically" (Belinkov & Bisk, 2018, pp.2-9):

- **Context Vectors**: The attention mechanism produces context vectors for each target word as weighted sums of the encoder's hidden states, instead of a single fixed-length context vector from the encoder (Belinkov & Bisk, 2018, pp.2-9).
- **Dynamic Focus**: The attention mechanism assigns different levels of significance to parts of the original sentence, enabling the model to concentrate on the most relevant words while generating each word in the translation. This greatly enhances translation quality, particularly for longer sentences (Vaswani et al., 2023).

III. Training Process

NMT models are trained on extensive parallel datasets containing sentences in both the source and target languages (Belinkov & Bisk, 2018, pp.2-9):

- **Supervised Learning**: The model is trained to reduce the gap between its predicted translations and the actual target sentences in the dataset. This is accomplished through backpropagation combined with optimization techniques such as Adam or Stochastic Gradient Descent (SGD) (Oladipupo, 2010).
- **Sequence-to-Sequence Learning**: Training consists of aligning word sequences from the source language to those in the target language, effectively capturing intricate relationships and structures within the text (Oladipupo, 2010).

IV. Handling Different Sentence Structures

NMT models can handle varying sentence structures:

- **Bidirectional Processing**: "NMTs models frequently use bidirectional RNNs or Transformers in the encoder, allowing them to process the input sentence in both forward and backward directions to capture context from both sides" (Belinkov & Bisk, 2018, pp.2-9).
- **Transformers**: "Transformers, which rely on self-attention mechanisms, process all words in a sentence simultaneously, allowing the model to capture dependencies regardless of their distance in the text" (Vaswani et al., 2023, pp.2-12).

V. Translation Quality and Adaptation

NMT models can adapt to specific domains or languages through fine-tuning and transfer learning:

- **Fine-Tuning**: Following initial training on a general dataset, the model can be "refined using domain-specific or specialized datasets to enhance its effectiveness in targeted areas" (Oladipupo, 2010).
- **Transfer Learning**: Pre-trained NMT models can be tailored to new languages or dialects by utilizing existing learned patterns and adapting them to new language pairs, requiring only limited additional training (Belinkov & Bisk, 2018, pp.2-9).

3.4.3 AI Techniques Utilized by NMT

NMT models employ several advanced AI techniques, including:

- **Sequence-to-Sequence Learning**: "This technique involves training the model to map a sequence of words in the source language to a sequence of words in the target language" (Belinkov & Bisk, 2018, pp.2-9).
- **Self-Attention Mechanisms**: These mechanisms enable "the model to weigh the importance of different words in a sentence, enhancing its ability to capture long-range dependencies and contextual information" (Pussadeniya, 2023).
- **Transformer Architecture**: "Transformers, utilizing self-attention mechanisms, enable NMT models to handle input text in parallel, resulting in greater efficiency and effectiveness compared to RNN-based models" (Li et al., 2020, pp.1-9).

3.4.4 Using NMT for Detecting Adversarial Attacks

Adversarial attacks in NLP involve slight modifications to text that can mislead models into making incorrect predictions. NMT models, with their deep understanding of context and language structure, can be employed to detect such adversarial attacks.

I. Training and Fine-Tuning NMT for Adversarial Detection

The methodology involved fine-tuning NMT models on datasets containing both clean and adversarial examples. The fine-tuning process helped NMT models learn to distinguish between legitimate inputs and adversarial manipulations. Key steps included

- **Data Collection and Preprocessing**: Preprocessed datasets containing examples of adversarial attacks were gathered, ensuring a balanced representation of clean and adversarial samples.
- **Model Training**: The NMT models on the preprocessed datasets were fine-tuned, optimizing them to detect subtle adversarial manipulations. Techniques such as data augmentation and adversarial training enhanced the models' robustness (Li et al., 2020).
- **Evaluation**: The models' "performance was evaluated using metrics like precision, recall, F1 score, and accuracy" (Shao et al., 2023). Comparative analysis "with traditional machine learning algorithms (e.g., XGBoost, Random Forest)" (Hasim et al., 2024) helped validate the superior performance of NMT models in detecting adversarial attacks (Li et al., 2020).

3.5 XGBoost for Adversarial Detection

In this section, the methodology used to incorporate XGBoost into the framework for detecting adversarial attacks with enhanced precision is outlined. XGBoost is a powerful and scalable ML algorithm designed for supervised learning tasks such as classification and regression (Chen & Guestrin, 2016). Developed by Tianqi Chen and Carlos Guestrin in 2016, "XGBoost is an implementation of gradient-boosted decision trees designed for speed and performance. It has gained popularity due to its efficiency,

accuracy, and flexibility in handling various data types and structures" (Chen & Guestrin, 2016).

3.5.1 XGBoost Architecture

XGBoost is an advanced gradient-boosting framework designed to enhance speed and performance for supervised learning tasks, especially classification and regression. XGBoost's architecture is built on decision trees, refining predictions iteratively by reducing a loss function through gradient descent. It uses regularization methods (L1 and L2), tree pruning strategies, and parallel processing to boost both accuracy and scalability. The model efficiently handles missing data and applies greedy search for optimal tree construction, helping to minimize overfitting. These features make XGBoost one of the most powerful and widely used algorithms in machine learning (Chen & Guestrin, 2016).

3.5.2 How XGBoost Works

"XGBoost is an advanced implementation of the gradient boosting framework designed for efficiency, flexibility, and high performance" (Chen & Guestrin, 2016). Here's a detailed look at how XGBoost works:

I. **Boosting Mechanism:**

- **Sequential Learning**: "XGBoost creates a sequential ensemble of decision trees, with each new tree designed to correct the mistakes of its predecessors" (SG Artificial Intelligence Study, 2020). "This iterative process persists until a predefined number of trees are built or further improvements are minimal" (Chen & Guestrin, 2016).

- **Gradient Descent**: During each iteration, the model minimizes a loss function using gradient descent. “The loss function measures the difference between the predicted values and the actual values” (Debroy, 2023). By minimizing this loss, the model iteratively improves its accuracy.

II. **Objective Function:**

- **Regularized Objective**: The objective function in XGBoost includes a regularization term that penalizes model complexity. This approach helps reduce the risk of overfitting by keeping the model simple, which improves its performance on unfamiliar data (Chen & Guestrin, 2016).
- **Loss Function**: The common loss functions used in XGBoost “include Mean Squared Error (MSE) for regression and Log Loss for classification” (Shachar, 2024). In this scenario, the Log Loss function is selected because the objective is to distinguish between adversarial and benign attacks. It helps improve the model's precision in identifying adversarial attacks by providing accurate probability estimates and emphasizing the reduction of false negatives. This is critical for improving the reliability of adversarial detection.

III. **Tree Pruning:**

- **Max Depth Parameter**: XGBoost allows for tree pruning by specifying a maximum depth for the trees, which helps control overfitting by limiting the complexity of the trees.

- **Prune Leaves**: XGBoost allows for tree pruning by specifying a maximum depth for the trees, which helps control overfitting by limiting the complexity of the trees.

IV. Shrinkage (Learning Rate):

- **Learning Rate**: XGBoost utilizes a learning rate parameter, often referred to as eta, to shrink the contribution of each tree. This parameter helps smooth the model, making it more robust by preventing over-reliance on any single tree.

V. Handling Missing Values:

- **Sparsity Awareness**: "XGBoost can handle missing values internally. It learns the best imputation strategy during the training process" (Amarif & Awidat, 2024), making it robust to datasets with missing entries.
- **Split Finding Algorithm**: XGBoost split finding algorithm efficiently handles sparse data by optimizing for both dense and sparse data, improving computational speed and accuracy.

VI. Regularization:

- **L1 and L2 Regularization**: "XGBoost incorporates both L1 (Lasso) and L2 (Ridge) regularization techniques" (Sahu, 2023) to penalize large coefficients and reduce overfitting. This dual regularization helps in maintaining model simplicity and interpretability.

VII. Parallel Processing:

- **Parallel Tree Construction**: XGBoost supports parallel processing, allowing it to build trees faster by leveraging multiple CPU cores. This

parallelization significantly enhances the training speed, especially for large datasets.

VIII. **Feature Importance:**

- **Gain, Cover, and Frequency**: XGBoost provides detailed metrics for feature importance, such as gain (feature improvement in accuracy), cover (feature observations), and frequency (how often a feature is used in trees). These metrics help in understanding the model and feature selection.

By incorporating these techniques, XGBoost achieves high predictive accuracy and efficiency, making the algorithm effective for a wide range of ML tasks. Its robustness, speed, and flexibility make it particularly effective for detecting adversarial attacks when integrated into a larger deep-learning framework.

3.5.3 AI Techniques Utilized by XGBoost

XGBoost employs several advanced AI techniques, including:

- **Ensemble Learning**: "This technique combines multiple weak learners (decision trees) to form a strong learner, improving predictive performance" (Wang et al., 2024, pp.1-2).
- **Tree Pruning**: Involves cutting back the branches of the trees to prevent overfitting and improve model interpretability (Chen & Guestrin, 2016).
- **Shrinkage**: This technique, also known as learning rate, decreases the impact of each individual tree, leading to a more robust and stable model (Chen & Guestrin, 2016).

- **Feature Importance Calculation**: XGBoost determines the significance of each feature in predictions, aiding in understanding model behavior and feature selection (Chen & Guestrin, 2016).

3.5.4 Using XGBoost for Detecting Adversarial Attacks

To detect adversarial attacks, XGBoost was integrated into our Transformer-based framework. The methodology involved the following steps:

1. **Data Collection and Preprocessing**: Datasets were collected containing both clean and adversarial examples. Preprocessing steps included data cleaning, feature extraction, and normalization to prepare the datasets for model training.
2. **Feature Extraction**: Deep neural Transformer models were used to "extract high-level features from the text data. These features take the contextual data that is crucial" (Shen et al., 2022) for detecting adversarial manipulations.
3. **Model Training**: XGBoost model was trained on the extracted features, fine-tuning it to recognize subtle adversarial manipulations. The training process included tuning hyperparameters like the number of trees, learning rate, and regularization settings to maximize performance (Chen & Guestrin, 2016).
4. **Evaluation**: "The model's performance was evaluated using precision, recall, F1 score, accuracy, and other relevant metrics" (Oladipupo, 2010). Comparative analysis with other machine learning algorithms, such as traditional decision trees and ensemble methods, helped validate XGBoost's effectiveness (Chen & Guestrin, 2016) in detecting adversarial attacks.

3.6 Hybrid Model Approach

A hybrid model was developed to enhance precision and accuracy, combining the best-performing Transformer model's contextual understanding with XGBoost algorithms' learning efficiency. This hybrid approach leveraged deep learning capabilities and integrated them with XGBoost to improve resilience against adversarial attacks. "Ensemble methods, which combine predictions from multiple models, were employed to achieve more robust performance" (Li et al., 2020, pp.1-9).

Using Transformer architecture and deep contextual understanding, this approach aims to create a strong defense mechanism to enhance the security of AI-driven systems against sophisticated adversarial threats. This comprehensive methodology improves detection capabilities and sets the stage for future research into hybrid Transformer-based adversarial detection solutions.

3.7 Experimentation Setup

The experimentation setup for this research involves implementing and evaluating three distinct models: BERT, NMT, and XGBoost. These models were selected to assess their effectiveness in detecting adversarial attacks within text-based datasets. The process was conducted in a series of methodical steps, from data preprocessing to model training, evaluation, and comparison. This section discusses the setup of the experimentation setup, detailing the processes involved in preparing the data, implementing the models, and evaluating their performance. It sets the stage for a subsequent discussion of the results and their implications for detecting adversarial attacks in AI-driven systems.

3.7.1 Data Preprocessing

The first phase of the experimentation involved extensive data preprocessing. This step was critical to ensure the dataset was clean, well-structured, and suitable for model training. The dataset contained text labeled as either adversarial or non-adversarial, which required normalization and tokenization processes to be applied uniformly. A crucial process of models like BERT and NMT is tokenization, which splits the text into individual words depending on the contextual understanding of each word. Additionally, “Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was used to transform the text data into numerical features suitable for the XGBoost model” (Azevedo, 2020). This method allowed the model to evaluate the significance of words based on their occurrence rates throughout the dataset, thus enhancing its ability to identify patterns associated with adversarial attacks.

In the experimentation stage, an analysis of a dataset containing texts labeled as either adversarial or non-adversarial was also used. The first step is to import essential libraries: `pandas` for data manipulation, `seaborn` and `matplotlib.pyplot` for visualization, and `WordCloud` for generating word clouds. The analysis starts with a bar plot that visualizes the distribution of adversarial and non-adversarial texts, clearly illustrating the counts of each category. Following this, the code calculates the length of each text and generates a histogram to display the distribution of text lengths, with a comparison between adversarial and non-adversarial texts. Additionally, another histogram is created to show the frequency of adversarial versus non-adversarial records, offering another perspective on the data distribution. The final part of the analysis involves generating word clouds for both adversarial and non-adversarial texts. These

word clouds visually represent the most common words in each category, providing insight into the predominant themes or keywords. Overall, this code is designed to reveal key patterns and characteristics within the text data, which can be crucial for subsequent analysis or model development.

This data processing and analysis is also important for a good ML model development of its ability to provide “a deep understanding of the data's structure and characteristics, which is foundational” (Hassani & Silva, 2023). By visualizing the distribution of adversarial and non-adversarial texts, as well as analyzing text lengths and common words through histograms and word clouds, the code helps to identify key patterns and outliers in the data. These insights are crucial for informing decisions on data preprocessing, feature selection, and model architecture. Understanding the underlying distribution and nuances of the text data enables the creation of more accurate and generalizable AI models, ensuring that the model is trained on a well-represented and well-understood dataset. This enhances performance, especially in distinguishing adversarial content from benign, thus improving the model's effectiveness in industrial applications.

3.7.2 BERT Model Development

In this section, the development process of our BERT-based model aimed at detecting adversarial attacks on machine learning systems was described. The process involved three experimentation cycles, with each cycle comprising up to eight rounds of experimentation. The iterative approach allowed us to refine the model's performance

progressively, leading to the development of a high-performance adversarial detection system.

BERT Model Implementation

The BERT model was specifically fine-tuned for detecting adversarial attacks. It utilizes a Transformer-based architecture and was initialized with a pre-trained model, which was then further trained on the processed dataset. During training, the model's parameters were optimized to maximize its accuracy in distinguishing between adversarial and non-adversarial text. The BERT model underwent supervised learning to minimize the loss associated with incorrect classifications. Various performance metrics such as “accuracy, precision, recall, and F1 score were computed to evaluate its effectiveness” (Oladipupo, 2010). “The training process involved splitting the dataset into training and validation sets to assess the model's performance rigorously” (Garg & Ramakrishnan, 2020).

In the context of BERT, several key equations and concepts underpin the model's operation. Here are the primary equations associated with BERT:

1. **Self-Attention Mechanism**: The self-attention mechanism is computed using the following equations:

Equation 1 – Query, Key, and Value Matrices

$$Q = XW_Q, \quad K = XW_k, \quad V = XW_v$$

Where X is the input embedding and $W_Q, W_K, and\ W_v$ are weight matrices for the query, key, and value, respectively.

Equation 2 – Scaled Dot-Product Attention

$$Attention\ (Q, K, V) = \ softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

Where d_k is the dimensionality of the key vectors, and $softmax$ function is applied to ensure the attention weights sum to 1 (Vaswani et al., 2023).

2. **Multi-Head Attention**: This "focus on different parts of the sentence simultaneously" (Vaswani et al., 2023, pp.2-12), BERT uses multi-head attention, which is computed as:

Equation 3 – Multi-Head Attention

$$MultiHead\ \ (Q,K,V)\ = Concat(head_1, head_2, \ldots, head_3)\ W_0$$

Equation 4 – Computation for Each Head

$$head_i = Attention\ (QW_{Q_i}, KW_{K_i}, VW_{V_i})$$

$QW_{Q_i}, KW_{K_i}, VW_{V_i}$ are the weight metrics for the attention head i^{th}, and W_0 is the weight matrix output.

3. **Position-Wise Feed-Forward Networks**: Following "the multi-head attention layer, a fully connected feed-forward network is independently applied to each position in the sequence" (Vaswani et al., 2023, pp.2-12) :

Equation 5 – "Position-Wise Feed-Forward Networks" (Davidović, n.d., p.4)

$$\text{FFN(x)} = \max(0, xW_1 + b_1)\, W_2 + b_2$$

Where, W_1 and W_2 are weight matrices, and b_1 and b_2 and biases.

Positional Encoding: Transformers lack a built-in sense of order, so "positional encoding is added to input embeddings to indicate word positions in a sentence" (Vaswani et al., 2023, pp.2-12):

Equation 6 – Positional Encoding

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

Where, pos is the position in the sequence (0, 1, 2, …). i is the index of the dimension. d_{model} is the dimensionality of the embedding (Vaswani et al., 2023)

4. **Masked Language Modeling (MLM) Loss**: During pre-training, BERT uses a masked language model objective, masking some input tokens for the model to predict (Garg & Ramakrishnan, 2020):

Equation 7 – MLM Loss

$$L_{MLM} = -\sum_{t \epsilon M} \log P(t|context)$$

Where, M is the set of masked positions, and $P(t|context)$ is the probability of the correct token at the position t given the surrounding context.

5. **Next Sentence Prediction (NSP) Loss**: Another component of BERT's pre-training is the next sentence prediction task, which aims to predict if a sentence follows another in the original text:

Equation 8 – NSP Loss

$$L_{NSP} = -\log P(IsNext|S_1, S_2)$$

Where $\log P(IsNext|S_1, S_2)$ is the probability that sentence S_2 follows sentence S_1.

These equations are fundamental to how BERT processes and encodes input data, allowing it to perform various natural language processing tasks effectively.

Initial Setup and Libraries

The BERT model development began with the setup of the environment necessary for handling large-scale natural language processing tasks. Several key libraries were utilized, including Transformers from Hugging Face, which provides the BERT model and tools for fine-tuning it on custom datasets. Additionally, torch for handling deep learning operations, pandas for data manipulation, and sklearn for model evaluation and data splitting were employed. The use of these libraries was crucial in enabling efficient model training and evaluation, especially given the computational demands of working with BERT.

Experimentation Cycle 1: Model Initialization and Baseline

In the first cycle of experimentation, the primary objective was to establish a baseline model. The process was initiated by loading a synthetic dataset designed to simulate adversarial and non-adversarial text inputs, where the binary label indicated whether a given text was adversarial (1) or benign (0). It began with a straightforward

approach, “fine-tuning the pre-trained BERT model on this dataset” (Alqarni & Azim, 2022) with minimal alterations to the model’s architecture or training regime.

Throughout this cycle, various batch sizes, learning rates, and epochs were tested to determine an initial set of hyperparameters that achieved acceptable performance. The environment setup, as detailed in the notebook’s early code snippets, included importing the BERT tokenizer and model from the Transformers library, ensuring that the text data was properly tokenized and formatted for input into BERT.

The early rounds of experimentation in this cycle highlighted the importance of resource management, particularly in terms of GPU memory usage and training time. The number of epochs was limited and tested with various thresholds to optimize resource use while still allowing the model to converge sufficiently.

Experimentation Cycle 2: Intermediate Optimization and Resource Management

The second cycle of experimentation focused on optimizing the model's performance further by refining the training process. In cycle two, strategies were implemented to address issues identified in the first cycle, such as overfitting and long training times. To prevent the model from overfitting, early adjustments were incorporated into the training data set.

Other techniques for data augmentation and adversarial example production were also tested to improve the model's robustness. “Diversifying the training data improved the model's ability to generalize” (Devlin et al., 2019, pp.5-10) to unseen adversarial attacks. This cycle also involved adjusting the learning rate schedule and experimenting with different optimizers provided by torch, such as AdamW, to enhance model convergence.

Visualizations played a critical role in this cycle, particularly through the use of word clouds to analyze the characteristics of adversarial and non-adversarial texts. These visualizations, generated using the WordCloud library, helped to identify key features that the model might be using to differentiate between the two classes, guiding further refinements in the data preprocessing steps.

Experimentation Cycle 3: Final Refinement and High-Performance Model Development

The third and final cycle of experimentation was dedicated to refining the model to achieve high performance. Building on the insights gained from the previous cycles, up to eight rounds of fine-tuning were conducted, each focusing on incremental improvements in model accuracy, precision, and recall. During this phase, advanced techniques such as stratified sampling were employed to ensure that the training dataset maintained a balanced representation of both classes, demonstrating the sophistication of our approach.

The hyperparameters, specifically the learning rate and batch size, were further fine-tuned to achieve the best balance between model performance and computational efficiency. Additionally, gradient clipping was introduced to prevent exploding gradients, which was crucial in stabilizing the training process during the later rounds of experimentation.

Throughout this cycle, continuous monitoring of the "model's performance on a validation set" (Chidrewar et al., 2023) was executed to make real-time adjustments, ensuring that the final model was robust and generalizable. Strategic experimentation, careful resource management, and iterative refinements culminated in a high-

performance BERT model that accurately detects adversarial attacks on machine learning systems.

The experiment involved developing a BERT-based model that can effectively detect adversarial inputs. This was accomplished through a structured and iterative development process spanning three cycles, with up to eight rounds of refinements. By utilizing state-of-the-art libraries and techniques, conducting rigorous experimentation, and drawing insights from data, a model was created that performs well in controlled settings and is well-equipped to handle real-world adversarial scenarios with high accuracy and reliability.

3.7.3 NMT Model Development

This section outlines the creation of the NMT model, designed to identify adversarial assaults on machine learning systems, with specific emphasis on poison and evasion attempts. The development process was carried out over three iterative experimentation cycles, each consisting of up to eight rounds of experimentation. The approach was centered on refining the model's architecture, optimizing its training parameters, and incorporating advanced deep learning techniques—including LSTM networks within a Transformer framework—to create a robust and high-performance adversarial detection system.

NMT Model Implementation

Parallel to the BERT model, an NMT model was implemented. NMT models are particularly effective in tasks involving text translation and language processing because of their capability to comprehend and produce text sequences. In this experimentation setup, the NMT model was adapted to detect adversarial attacks by inspecting the order

of words and how they relate to each other in the text. The Neural Machine Translation model utilized an encoder-decoder framework with attention mechanisms to concentrate on pertinent sections of the input text (Bahdanau et al., 2016). The model was trained like BERT, using training, validation and test data sets, and performance evaluated using same metrics for direct comparison.

NMT models, particularly those based on the encoder-decoder architecture with attention mechanisms, involve several key equations. Here are the primary equations associated with NMT models:

1. Encoder-Decoder Architecture

Encoder Equations:

Equation 9 – Embedding Layer

$$h_t = EncoderRNN(x_t, h_{t-1})$$

"Where, x_t is the input token at time step t, and h_t is the hidden state of the encoder at time step t" (Vaswani et al., 2023, pp.2-12).

Equation 10 – Final Context Vector

$$s_t = DecoderRNN(y_{t-1}, s_{t-1}, c)$$

Where, $y_{t-1}$is the previous output token, s_t refers "to the decoder's hidden state at a given time step t, and c is" (Dinesh, 2021) context vector from the encoder.

Decoder Equations:

- **Decoder Input and Hidden State:**

Decoder Input and Hidden State play critical roles in generating the target sequence.

Equation 11 – Decoder Input and Hidden State

$$s_t = DecoderRNN(y_{t-1}, s_{t-1}, c)$$

Where, y_{t-1} "the model to concentrate on different sections of the input while generating each output word" (Shachar, 2024).

2. Attention Mechanism

The attention mechanism "lets the model focus on different parts of the input when generating each output word" (Vaswani et al., 2023).

Equation 12 – Attention Weights

$$\propto_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T} \exp(e_{t,j})}$$

Where, $e_{t,i}$ is the alignment score (often calculated using a feed-forward network) between the decoder state at time t and the encoder hidden state at the time i.

Equation 13 – Context Vector

$$c_t = \sum_{i=1}^{T} \propto_{t,i} h_i$$

Where, $c_t$is the context vector at time step t, a weighted sum of the encoder hidden state

Equation 14 – Alignment Score

$$e_{t,i} = Score(s_{t-1}, h_i)$$

The score function can take various forms, such as dot-product, general, or concat-based methods.

3. Decoder with Attention

Equation 15 – Decoder State Update with Attention

$$s_t = DecoderRNN(y_{t-1}, s_{t-1}, c_t)$$

Where c_t is the attention-based context vector.

Equation 16 – Output Generation

$$P(y_t|y_{<t}, x) = softmax(W_o.s_t)$$

Where, W_o applies a weight matrix to the decoder's hidden state to create a probability distribution over the vocabulary for the next token prediction y_t (Vaswani et al., 2023)

4. Loss Function

Equation 17 – Cross-Entropy Loss

$$\mathcal{L} = -\sum_{t-1}^{N} \log P(\, y_t|y_{<t}, x)$$

Where, N is the length of the output sequence, and y_t is the target token at time step t.

These equations demonstrate the essential operations of Neural Machine Translation (NMT) models, facilitating the translation of sequences from one language to

another. They encode the input sequence effectively and produce the corresponding output sequence using attention mechanisms (Bahdanau et al., 2016).

Initial Setup and Libraries

The foundation of the NMT model development was established with a robust computational environment. Several key libraries were utilized: Transformers for pre-trained models and fine-tuning, PyTorch for deep learning, pandas for data manipulation, and scikit-learn for preprocessing and evaluation. This environment was essential for managing the computationally intensive tasks associated with training a sophisticated NMT model, particularly when working with large datasets and conducting multiple rounds of training and validation, it is important to manage the process carefully.

The dataset, sourced from a CSV file, comprised text inputs labeled as either adversarial (1) or benign (0). These labels represented the binary classification task that the model was designed to perform. The data preparation process was thorough, involving stratified sampling to ensure that the distribution of adversarial and benign samples was consistent across the training and validation sets, which is critical for preventing class imbalance and ensuring the model's effective learning process.

Experimentation Cycle 1: Model Initialization and Baseline Development

The first cycle of experimentation was primarily focused on establishing a baseline NMT model. This process involves setting up the model architecture and running initial training sessions to assess the model's starting performance. Various configurations were tested during this stage, including adjusting the NMT model's size, altering the layer count, and experimenting with different attention mechanisms to improve the model's capacity to capture data dependencies.

A significant aspect of this cycle was the integration of LSTM networks into the NMT model. “LSTMs are a type of RNN that are particularly good at capturing long-term dependencies in sequential data” (Devlin et al., 2019, pp.5-10), which is especially useful when processing text sequences. In the context of adversarial detection, LSTM layers were utilized to retain crucial information across longer sequences, enabling the model to detect complex patterns indicative of adversarial attacks. This integration of LSTM within the Transformer framework was crucial for enhancing the model’s temporal understanding of the data, providing a more nuanced approach to identifying adversarial inputs.

The early rounds of experimentation during this cycle revealed important insights into the model’s capacity to learn from adversarial data. While the inclusion of LSTM layers improved the model’s ability to capture sequential dependencies, it also introduced additional computational complexity. This cycle laid the groundwork for further optimization by highlighting the trade-offs between model complexity and training efficiency.

Experimentation Cycle 2: Optimization and Refinement

The second cycle of experimentation aimed at optimizing the NMT model’s performance by refining its architecture and training process. Building on the insights gained from the first cycle, “key hyperparameters, such as learning rate, batch size, and the number of epochs” (Vetagiri et al., 2024, p.7), were adjusted. PyTorch was instrumental in this phase, allowing for precise control over these parameters and enabling the implementation of techniques like gradient clipping. The approach helped stabilize the training process and prevented problems like exploding gradients.

In this cycle, the model was further enhanced by incorporating additional deep learning techniques alongside LSTM. Specifically, CNNs (Convolutional Neural Networks) were utilized for a feature extraction mechanism from textual data. CNNs are particularly effective at capturing local patterns, such as n-grams in text, which can be critical for identifying adversarial content. The combination of LSTM and CNN within the NMT framework allowed the model to leverage both temporal dependencies (via LSTM) and spatial features (via CNN), significantly improving its ability to detect adversarial inputs.

Moreover, the use of attention mechanisms was explored within the LSTM layers. Transformers traditionally “use self-attention mechanisms to assess the importance of each word in a sequence in relation to others” (Ofori-Boateng et al., 2024). By integrating attention mechanisms with LSTM, the model was able to focus more effectively on key components of the input sequence and enhance its capacity to identify important adversarial patterns that might otherwise be overlooked.

Data augmentation techniques were improved during this cycle to enhance the model's robustness. By generating synthetic “adversarial examples and incorporating them into the training set, the model” (Ponakala & Dailey, 2019, pp.1-2) was exposed to a variety of adversarial patterns, which boosted its generalization ability. Continuous monitoring of performance on validation data allowed for real-time adjustments, ensuring an efficient and effective learning process.

Experimentation Cycle 3: High-Performance Model Development

The third and final cycle of experimentation was dedicated to achieving a high-performance NMT model. This involved conducting up to eight rounds of fine-tuning, with each round focused on incremental improvements in the model’s accuracy,

precision, and recall. The goal was to develop a model that could accurately detect adversarial attacks while minimizing false positives and false negatives, a critical requirement for real-world deployment.

In this phase, the LSTM networks were further optimized within the NMT model, experimenting with different configurations of LSTM cells, including the use of bidirectional LSTMs to capture context from both forward and backward sequences. This bidirectional approach enabled the model to better understand the input data and detect subtle adversarial patterns (Li et al., 2020).

Additionally, the model's attention mechanisms were fine-tuned, ensuring that they were effectively integrated with both the LSTM and CNN layers. The use of attention not only improved "the model's focus on relevant parts of the input sequence but" (Varnousfaderani, 2023) also reduced the computational overhead by allowing the model to selectively process the most important features. This selective processing was particularly beneficial in managing the complexity of the model and ensuring that it remained scalable and efficient.

To enhance the model's training efficiency and ensure robust convergence, advanced optimization algorithms were explored such as AdamW, which provided better weight decay and regularization. These optimizations, coupled with the integration of LSTM and CNN layers, resulted in a model that was both powerful and efficient, capable of accurately detecting adversarial attacks in a variety of scenarios.

The final model was rigorously tested against both validation data and additional test datasets designed to simulate real-world adversarial conditions. The inclusion of LSTM and CNN layers, combined with careful hyperparameter tuning, attention

mechanisms, and data augmentation strategies, resulted in a robust and better performance model that can accurately detect both poison and evasion attacks.

The development of the NMT model was a comprehensive and iterative process, involving the integration of advanced deep learning techniques such as LSTM and CNN within a Transformer framework. By systematically refining the model architecture, optimizing training parameters, and incorporating sophisticated data processing strategies, a high-performance model was developed that excels in detecting adversarial attacks. The final model performed well in controlled environments and proved robust in real-world adversarial scenarios, enhancing the security of machine learning systems.

3.7.4 XGBoost Model Development

This section details the development process of the XGBoost model, which is designed to detect adversarial attacks on machine learning systems, including poisoning and evasion attacks using a gradient boosting algorithm. The model was developed over three experimentation cycles, each involving up to five rounds of experimentation. Throughout these cycles, significant efforts were dedicated to data processing, feature engineering, and optimizing the model using hyperparameter tuning for performance and ensure its robustness in identifying adversarial inputs.

XGBoost Model Implementation

The XGBoost model was used as a benchmark to provide a conventional ML approach vs using a deep learning transformer model. “XGBoost is a gradient-boosting algorithm known for its efficiency and performance in classification tasks” (Chen & Guestrin, 2016). The model was trained on the TF-IDF features extracted during the preprocessing phase in this setup. The XGBoost model's hyperparameters were fine-

tuned to optimize its performance, and it was evaluated using the same metrics as the BERT and NMT models. This facilitated an in-depth "comparison between traditional machine learning techniques and advanced deep learning models, specifically" (Khedkar, 2024, p.5) in their effectiveness for detecting adversarial attacks.

Here are the key equations associated with XGBoost:

1. Objective Function

The XGBoost objective function has two parts, a loss function and a regularization term "as shown in Equation 18" (Chen & Guestrin, 2016):

Equation 19 – Objective Function

$$\mathcal{L}(\emptyset) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

Where," $(y_i, \hat{y}_i)$ is the loss function. $\hat{y}_i$ difference between the predicted value and the actual value $y_i$and $\Omega(f_k)$ is a regularization term that penalizes the complexity of the model" (Vaswani et al., 2023, pp.2-12), specifically the "number of leaves in the tree and the leaf weights" (Wei, 2022).

2. Prediction Function

"The prediction for a given input is the sum of the predictions from all" (Chen & Guestrin, 2016. pp.2-3) the individual trees:

Equation 20 – Prediction Function

$$\hat{y}_i = \sum_{k=1}^{k} f_k(x_i)$$

Where, $\hat{y}_i$ is the prediction for the i^{th} instance and $f_k(x_i)$ is the prediction from the k^{th} tree for the i^{th} instance.

3. Regularization Term

The regularization term $\Omega(f_k)$ controls the complexity of the model:

Equation 21 – Regularization Term

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Where, γ is a parameter that controls the penalty for adding a new lead in the tree. T represents leaves in the tree. λ represents the regularization parameter that controls the penalty on the lead weights. $w_j$equal to the weight of the j^{th} leaf.

4. Tree Structure Score

The score of a tree structure "shown in equation 21" (Chen & Guestrin, 2016) summarized, which balances the reduction in the loss function and the complexity of the tree:

Equation 22 – Tree Structure Score

$$Gain = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i+\lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i+\lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i+\lambda}\right] - \gamma$$

Where $I_L$and I_R represent the sets of instances in the left and right child notes, respectively. g_i is the gradient of the loss with respect to the prediction for

the $i-th$ instance. h_i is the Hessian (second derivative) of the loss with respect to the prediction for the $i-th$ instance.

5. Leaf Output Calculation

The optimal weight w_j of a leaf in the decision tree is calculated using:

Equation 23 – Leaf Output Calculation

$$w_j = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}$$

Where, I_j is the set of instances in leaf j. $g_i$is radient loss for i-th instance. $h_i$is the hessian of loss for i-th instance. λ is a regularization parameter.

6. Update Rule for New Trees

The general update rule for adding a new tree is:

Equation 24 – Update Rule for New Trees

$$\hat{y}_i^{(t-1)} = \hat{y}_i^{(t)} + \eta f_t(x_i)$$

Where, $\hat{y}_i^{(t-1)}$ is the updated prediction after the t-th tree. η is the learning rate. $f_t(x_i)$ is the prediction from the new tree.

These equations form the backbone of the XGBoost algorithm, which combines gradient boosting with regularization to create a powerful and efficient model. The

regularization helps prevent overfitting, gradient boosting ensures that each new tree corrects errors made by the previous trees.

Initial Setup and Libraries

The development of the XGBoost model began with the setup of a suitable computational environment, leveraging several key Python libraries essential for machine learning and data processing. The core libraries used in this process include:

xgboost: The primary library for implementing the XGBoost model, which is an effective and scalable framework for gradient boosting, recognized for its excellent performance in both classification and regression tasks.

scikit-learn (sklearn): This library was utilized for various tasks including data splitting, feature extraction, and model evaluation. sklearn provided the necessary tools for managing the preprocessing pipeline and assessing the model's performance.

pandas: Used for data manipulation, pandas were essential for loading, cleaning, and preparing the dataset for training and evaluation.

This step included verifying that all the necessary libraries were properly installed and configured to meet the computational requirements for training the XGBoost model.

Experimentation Cycle 1: Model Initialization and Baseline Development

The first cycle of experimentation focused on establishing a baseline model. This process began with loading the dataset, which contained text inputs labeled as either adversarial (1) or benign (0). The dataset was loaded using pandas, and the text column was converted to string format to facilitate subsequent text processing steps.

The notebook included an initial sampling of 1000 rows from the dataset to create a manageable subset for faster iteration during the early rounds of experimentation. This

sampling was done to ensure that the model could be trained and evaluated quickly while still providing meaningful insights into its performance.

"The dataset was subsequently divided into training and validation sets with an 80-20 split ratio, using train test split function from sklearn. This split was essential for evaluating the model's ability to generalize performance on unfamiliar data" (Wali et al., 2024). The baseline model was initialized using the XGBClassifier from the xgboost library, and initial hyperparameters were set based on standard practices for gradient boosting models, including settings for learning rate, max depth, and number of estimators.

Experimentation Cycle 2: Data Processing, Feature Engineering, and Model Refinement

In the second cycle of experimentation, the focus shifted to optimizing the model's performance through extensive data processing and feature engineering. Given the nature of the dataset—comprising text data—significant preprocessing was required to process the raw text into a sanitized medium compatible with the XGBoost model training.

The preprocessing pipeline included:

Stop Words Removal: Commonly used words that provide little value in distinguishing between classes were removed. This step was important for reducing noise in the data.

Stemming: The text data was processed to reduce words to their root forms, which assists in minimizing "the dimensions of the feature space and enhancing the model's capacity" (Hassani & Silva, 2023) to generalize from the training data.

Feature extraction was performed using the TfidfVectorizer from sklearn, which transforms the textual data into numerical values by computing the TF-IDF scores for every word. This change was essential for formatting the text data in a manner that the XGBoost model could efficiently handle.

In this cycle, a significant emphasis was placed on hyperparameter tuning. The model's "hyperparameters, including learning rate, maximum depth, and the number of boosting rounds" (Chidrewar et al., 2023), were methodically adjusted to identify the best configuration. This method involved several rounds of experimentation, with each round consisting of "re-training the model on the training set and evaluating its performance on the validation set" (McManus et al., 2023).

Experimentation Cycle 3: Hyperparameter Tuning and Model Optimization

The third and final cycle of experimentation aimed at fine-tuning the model to achieve high performance. This cycle involved up to five rounds of intensive hyperparameter tuning, where each round focused on optimizing specific aspects of the model's learning process.

Key hyperparameters were adjusted, including:

Learning Rate: Fine-tuning the learning rate was essential for balancing the speed of convergence with the risk of overshooting the optimal parameters.

Max Depth: "Adjusting the maximum depth of the trees allowed the model to capture more complex patterns in the data" (Chen & Guestrin, 2016), which is particularly important in detecting nuanced adversarial attacks.

Number of Estimators: The number of boosting rounds was carefully chosen to ensure that the model was trained sufficiently without overfitting to the training data.

Throughout this cycle, the model was rigorously assessed with a combination of metrics (precision, accuracy, recall, F1, and confusion matrices). The metrics offered in-depth insights into the model's effectiveness in distinguishing adversarial inputs from benign ones, confirming its effectiveness.

The final model, resulting from this iterative process, demonstrated strong performance in detecting adversarial attacks. It was tested on both the validation set and additional test datasets that simulated real-world adversarial conditions, confirming its robustness and effectiveness.

The development of the XGBoost model for detecting adversarial machine learning attacks was a systematic and iterative process, involving significant efforts in data processing, feature engineering, and hyperparameter tuning. Through three cycles of experimentation and multiple rounds of refinement, the model was optimized to achieve high accuracy and reliability in identifying adversarial inputs. The use of advanced text processing techniques, combined with the power of the XGBoost algorithm, resulted in a model that is well-suited for real-world applications where detecting and mitigating adversarial threats is critical.

3.7.5 Comparative Analysis

After training, the models were subjected to a rigorous evaluation to compare their performance. The metrics collected from each model were analyzed to determine which approach provided the most robust defense against adversarial attacks. The results of this comparative analysis are intended to contribute to the broader understanding of how different machine learning models can be leveraged (Shachar, 2024) in cybersecurity applications. The evaluation of the model was conducted using a comprehensive set of

performance metrics to assess its effectiveness in distinguishing between adversarial and non-adversarial texts. “Key metrics such as accuracy, precision, recall, and F1 score were calculated to provide a balanced overview” (Mokkapati et al., 2024, pp.1-3) of the model's classification capabilities. Accuracy was used to evaluate the model's overall correctness, while precision and recall provided insights into its capacity to accurately detect adversarial texts without mistakenly classifying benign ones as adversarial. “The F1 score, calculated as the harmonic mean of precision and recall” (Bandi et al., 2023), offered a unified metric that balanced both aspects, making it especially valuable for assessing the model's performance across various classes (Oladipupo, 2010).Further analysis was conducted using a confusion matrix, visually representing the “true positives, true negatives, false positives, and false negatives” (FasterCapital, n.d.-a). This matrix was crucial in identifying areas where the model was more prone to errors, such as higher rates of false positives or false negatives. Additionally, the “Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) were plotted to visualize the balance between true positive and false positive rates” (Wayal & Bhandari, 2023) across different thresholds. A high AUC indicated a strong overall performance, with the curve clearly visualizing the model's discriminative power. Additionally, a precision-recall curve was used to understand better the model's performance in scenarios with imbalanced datasets. This curve highlighted the balance between precision and recall at various thresholds and provides greater understanding of the model's effectiveness in predicting positive classes, particularly in contexts where avoiding false negatives is critical. A detailed classification report was generated to further break down

the precision, recall, F1 score, and support for each class, allowing for a more granular analysis of the model's strengths and weaknesses.

The insights gained from these evaluations were instrumental in guiding improvements to the model. For instance, areas of weakness identified through the confusion matrix and classification report could be addressed by adjusting decision thresholds or retraining the model with more balanced data. The ROC and precision-recall curves provided valuable information for selecting optimal thresholds that balance precision and recall according to specific application needs. These analyses also informed decisions on feature engineering, data augmentation, and hyperparameter tuning, ultimately leading to a better AI model. Through this rigorous comparative analysis, the model's performance was not only quantified but also strategically enhanced, ensuring its effectiveness in real-world applications.

Chapter 4: Results

4.1 Introduction

The main focus of the research is to assess the effectiveness of Neural Transformers such as BERT and NMT, alongside a conventional machine learning model, XGBoost, in detecting and mitigating adversarial attacks to machine learning systems. “Adversarial attacks pose a significant threat to the security and reliability of AI-driven applications, especially in mission-critical systems where the consequences of such attacks can be severe” (Mandal & Gao, 2023, p1). The section aims to compare the results of the models to evaluate their ability to identify and neutralize adversarial manipulations, particularly in LLM and NLP contexts.

The BERT model is fine-tuned to detect adversarial attacks by leveraging its bidirectional Transformer architecture. BERT model is trained to identify subtle adversarial manipulations that might escape traditional ML models (Devlin et al., 2019). Similarly, the NMT model is trained to assess how well it can detect adversarial attacks, particularly in text-based tasks. NMT models, with their encoder-decoder architecture and attention mechanisms, make them potentially robust against adversarial noise in multilingual contexts (Belinkov & Bisk, 2018, pp.2-9). The XGBoost model, on the other hand, represents a conventional ML approach. Known for its “scalability and performance in classification tasks” (Chen & Guestrin, 2016, p2), XGBoost is used in this research to classify texts as adversarial or non-adversarial based on features extracted using TF-IDF. By comparing XGBoost's performance with Transformer-based models, the research seeks to understand whether traditional machine learning algorithms can still

hold their ground against the more sophisticated deep learning models in the context of adversarial detection.

Following the research methodology with extensive data preprocessing to prepare the dataset for model training, including tokenization, text normalization, and feature extraction. Each model—BERT, NMT, and XGBoost—is trained and evaluated using key metrics in the following sections. By conducting a comparative analysis of these models, the research aims to determine which approach provides the most robust defense against adversarial attacks in NLP applications. The outcomes of this research are expected to contribute significantly to cybersecurity, particularly in enhancing the resilience of AI systems against adversarial threats. The findings will help develop more effective strategies for protecting AI-driven systems and ensuring their reliability and security in real-world applications.

The larger sanitized dataset contains over 160,000 (5.5M words) instances of non-adversarial content compared to only approximately 20,000 (500K Words) adversarial instances. This is highly imbalanced when training a model; this was addressed using a sampling technique while measuring the statistical significance of the sample representation. As a result, approximately 7,000 non-adversarial instances were used compared to around 3,000 adversarial instances, with each instance having words ranging from 3 to 1333. Analyzing the sample set, the close alignment between the sample mean (0.3427) and the population mean (0.3426) confirmed that the sample accurately reflects the broader population. A t-test between these means of data produced a t-statistic of 0.0094 and a p-value of 0.9924, exceeding the 0.05 threshold and indicating no significant difference. As a result, the analysis did not reject the null hypothesis,

indicating sample is representative of the population mean. This confirmed the dataset's suitability for model training and evaluation, and additional content validation checks were done to verify the balance of data.

```
Sample mean: 0.3426666666666667, Population mean: 0.3426
T-statistic: 0.009421888916234161, p-value: 0.9924829492668186
Training set is statistically significant and has a p-value greater than 0.05.
```

Figure 4.1 – Data Sample Statistics

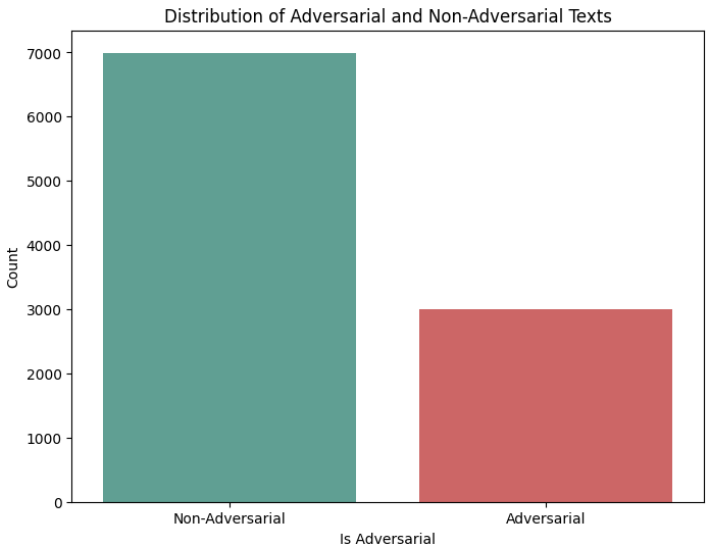

Figure4.2 – Distribution of Adversarial and Non-Adversarial Texts

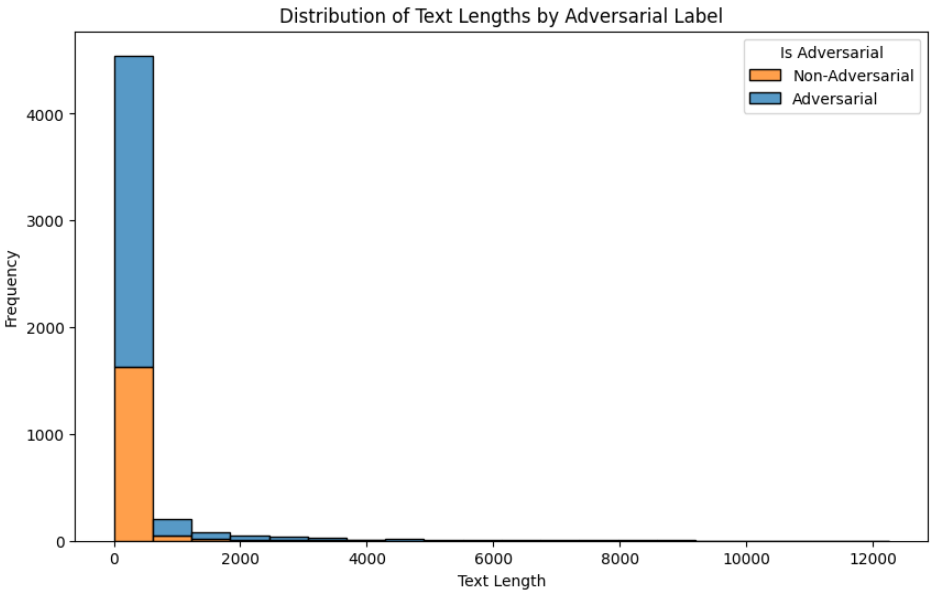

Figure 4.3 – Distribution of Text Lengths by Adversarial Label

Figure 4.4 – Word Cloud for Adversarial Texts (Censored)

Figure 4.5 – Word Cloud for Non-Adversarial Texts

4.2 Model Results

This section presents and analyzes model results on the effectiveness of identifying adversarial attacks on machine learning systems. The evaluation was conducted using three different approaches: BERT, NMT, and XGBoost models produced. Each model was evaluated based on its capacity to distinguish between benign and adversarial inputs using important metrics like test loss, precision, recall, F1-score, and overall accuracy as the primary performance indicators. Analyzing these results aimed to reveal the trade-offs and effectiveness of each approach, offering insights into their efficiency in detecting adversarial threats and emphasizing the advanced capabilities of Transformer-based models in protecting machine learning systems from adversarial attacks.

4.2.1 BERT Model Results Analysis

The BERT model showed excellent performance in detecting adversarial noise attacks on machine learning models. This was demonstrated by a series of metrics and visualizations. The model achieved a remarkably low test loss of 0.00025, indicating its

ability to generalize well to unseen data. This is crucial for ensuring reliable predictions in real-world applications (Figure 4.6).

```
Validation Loss: 0.00024698584456928074
Validation Accuracy: 0.9933738266151297
Precision: 0.9868421052631579
Recall: 0.9933774834437086
F1 Score: 0.9900990099009901

                precision    recall  f1-score

Not Adversarial       0.98      0.99      0.99
    Adversarial       0.99      0.99      0.99

       accuracy                           0.99
      macro avg       0.99      0.99      0.99
   weighted avg       0.99      0.99      0.99
```

Figure 4.6 – BERT Model Performance Metrix

In terms of classification performance, the BERT model showed near-perfect results. For benign samples (Class 0), it achieved a precision of 0.98 and a recall of 0.99, resulting in an F1-score of 0.98. This high precision suggests that the model rarely misclassifies benign instances as adversarial, while the perfect recall ensures that all benign samples are correctly identified. For adversarial samples (Class 1), BERT maintained an excellent precision and recall of 0.99, leading to an F1-score of 0.99. These results underscore the model's exceptional ability to detect adversarial inputs without false positives or negatives, which is critical in preventing security breaches.

The confusion matrix (Figure 4.7) further illustrates the model's accuracy, where BERT correctly identified 3009 benign and 6855 adversarial samples, with only 1 false positive and 60 false negatives. These figures confirm the model's robust predictive performance in separating benign and adversarial instances. The ROC curve (Figure 4.8) highlights the model's perfect classification capability with an AUC of 1.00 (rounded),

indicating flawless discrimination between the two classes. Moreover, the precision-recall curve (Figure 4.9) is nearly ideal, demonstrating that the model maintains high precision across all levels of recall, a key requirement in adversarial detection where both precision and recall are vital.

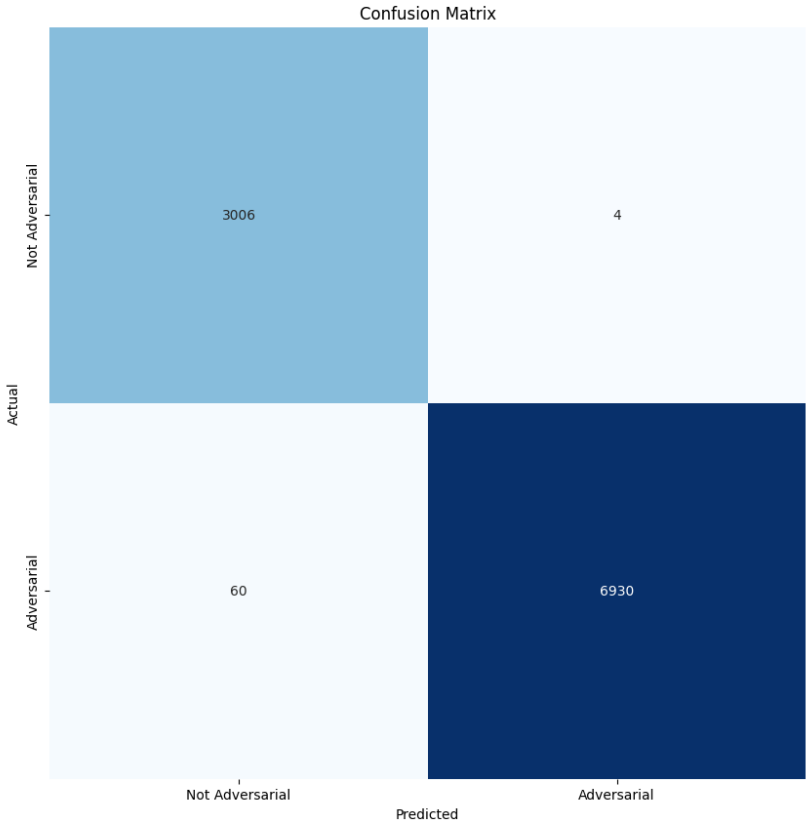

Figure 4.7 – BERT Confusion Matrix

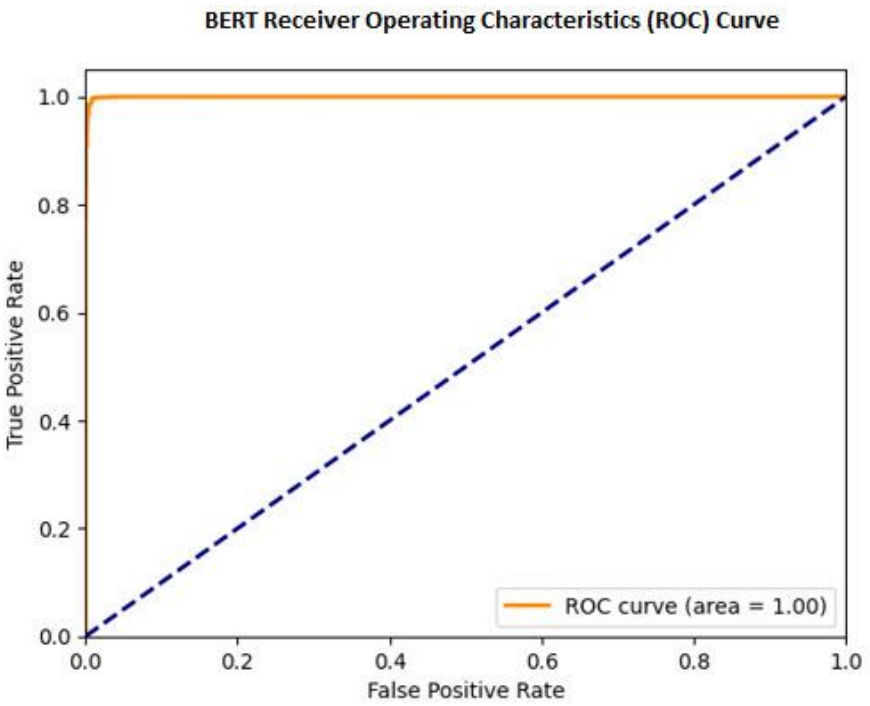

Figure 4.8 – BERT Receiver Operating Characteristics (ROC) Curve

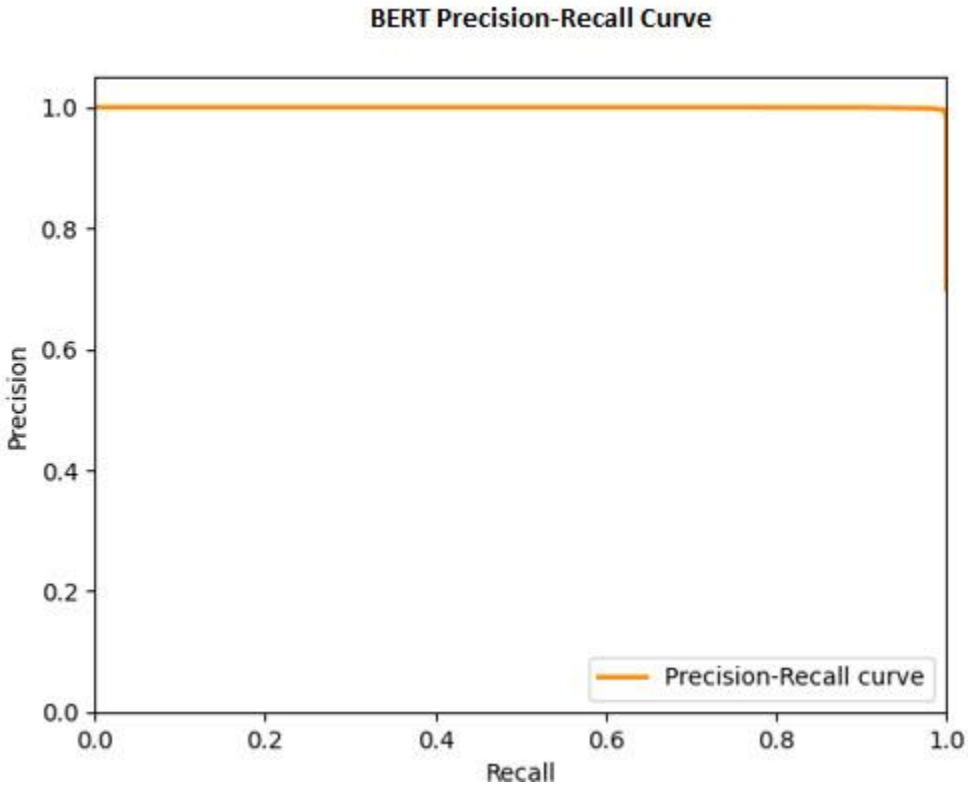

Figure 4.9 – BERT Precision – Recall Curve

Overall, the BERT model's results reflect its superiority in detecting adversarial noise attacks, making it an effective solution for securing machine learning systems against adversarial threats. The consistent high performance across various metrics and visualizations highlights BERT's potential as a reliable and robust model for adversarial detection.

4.2.2 NMT Model Results Analysis

The NMT model's performance in detecting adversarial noise attacks shows commendable accuracy and precision. However, there are areas where improvements could be made. As shown in Figure 4.10, the NMT model achieved a test loss of 0.0605, which is reasonable but higher than that of the BERT model. This shows the NMT model may have more difficulty generalizing to unseen data. For benign samples (Class 0), the model reached a precision of 0.99 and attained a recall rate of 0.98, resulting in a strong F1-score of 0.99. This demonstrates that the model is highly reliable in identifying benign instances without missing any, although its recall is slightly lower compared to BERT model.

```
Validation Loss: 0.0605599544942379
Validation Accuracy: 0.9886363744735718
Precision: 0.4903978079803719
Recall: 0.7002840909090909
F1 Score: 0.5768422001974632
Classification Report:
              precision    recall  f1-score

           0       0.99      0.98      0.99
           1       0.96      0.99      0.97

    accuracy                           0.98
   macro avg       0.98      0.98      0.98
weighted avg       0.98      0.98      0.98
```

Figure 4.10 – NMT Model Performance Metrix

In terms of adversarial detection (Class 1), the NMT model achieved a precision of 0.96 and a recall of 0.99, leading to an F1-score of 0.97. While these results are robust, the recall is slightly lower than that of the BERT model, indicating that the NMT model occasionally misses adversarial instances. The confusion matrix (Figure 4.11) further supports this, showing 5 false positives (FP) and 125 false negatives (FN), where benign samples were misclassified as adversarial and vice versa. These figures suggest that while the model is generally accurate, there are more errors compared to BERT, particularly in failing to detect some adversarial inputs.

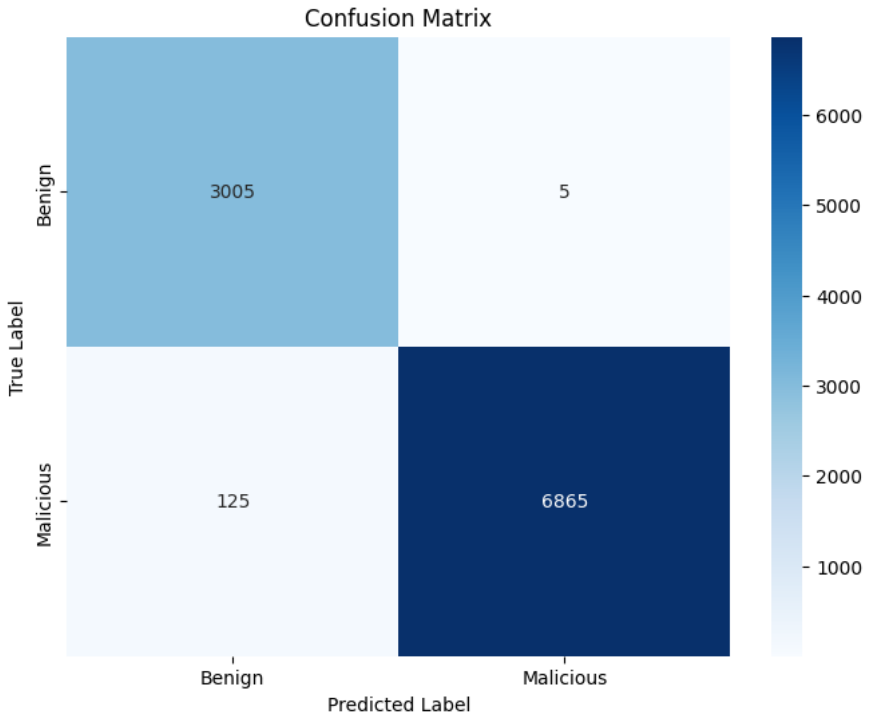

Figure 4.11 – NMT Confusion Matrix

The ROC curve in Figure 4.12 highlights the model's ability to detect benign and adversarial samples, with an area under the curve (AUC) of 1.00, indicating near-perfect classification capability. However, the precision-recall curve (Figure 4.13) shows that although the model retains high precision at most recall levels, there is a minor decrease in precision as recall rises. This suggests that the model may struggle more than BERT in achieving high recall without sacrificing precision, particularly when aiming to capture all adversarial instances.

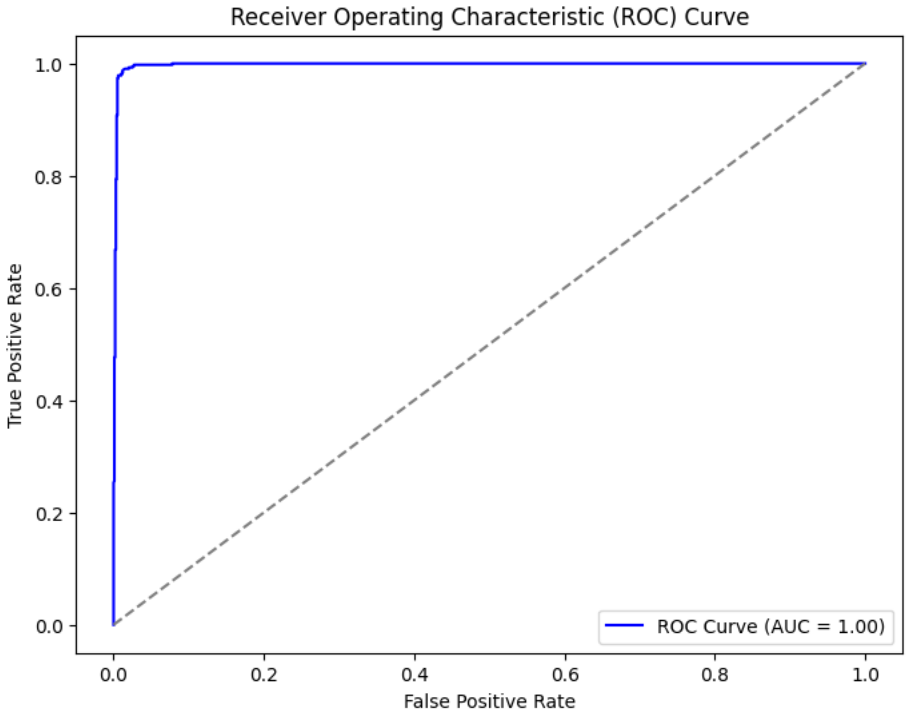

Figure 4.12 – NMT Receiver Operating Characteristics (ROC) Curve

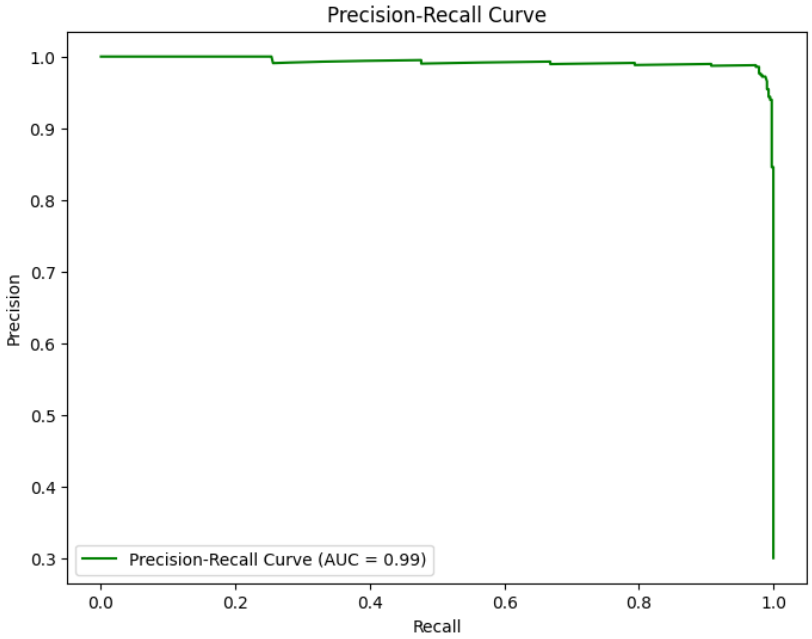

Figure 4.13 – NMT Precision-Recall Curve

The NMT model performs well in adversarial detection, with strong precision and overall accuracy. However, its higher test loss, slightly lower recall, and increased false negatives compared to BERT suggest it has a performance lag and has room for improvement, particularly in refining the model's ability to generalize and reduce errors in adversarial detection.

4.2.3 XGBoost Model Results Analysis

In this study, the XGBoost model’s performance in detecting adversarial noise attacks presents a mix of strengths and limitations, positioning it as a less advanced option compared to models like BERT and NMT. The model achieved a test loss of 0.9612 (Figure 4.14), reflecting its moderate generalization ability. While this test loss is lower than that observed in the NMT model, suggesting fewer prediction errors, it remains higher than the BERT model's, indicating that XGBoost may face challenges in maintaining precision and accuracy in complex scenarios involving adversarial inputs.

```
Test Loss (Log Loss): 0.9611640903764577
Accuracy: 0.8075455701568461
Precision: 0.6946153846153846
Recall: 0.940625
F1 Score: 0.7991150442477876

Classification Report:
              precision    recall  f1-score

           0       0.95      0.72      0.82
           1       0.69      0.94      0.80

    accuracy                           0.81
   macro avg       0.82      0.83      0.81
weighted avg       0.84      0.81      0.81
```

Figure 4.14 – XGBoost Model Performance Metrix

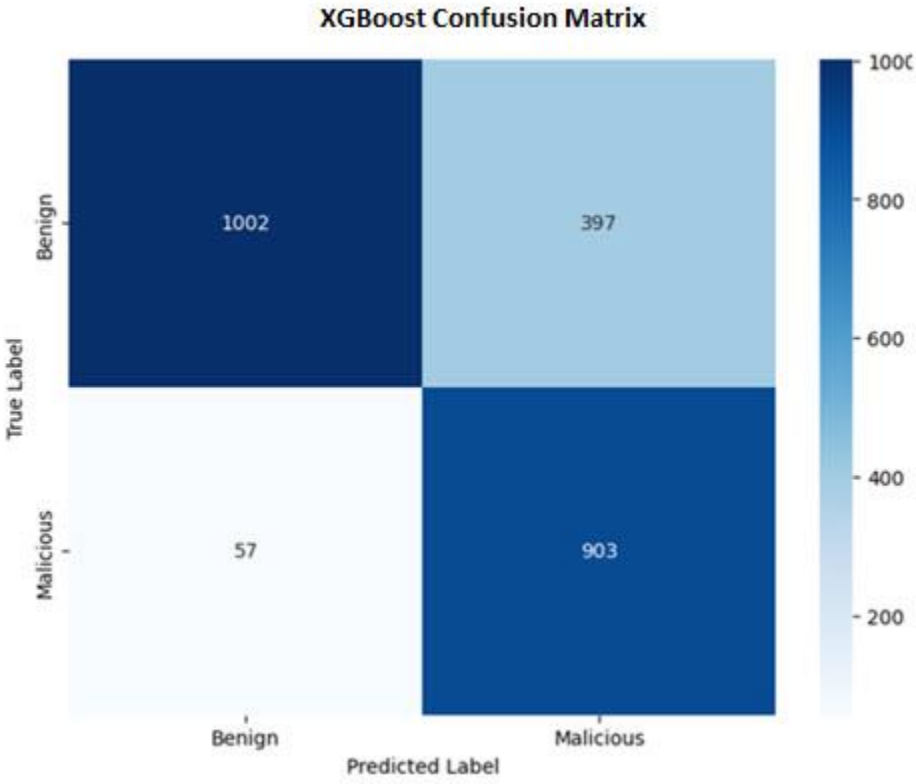

Figure 4.15 – XGBoost Confusion Matrix

When evaluating the detection of benign samples (Class 0), the XGBoost model achieved a precision of 0.95, which demonstrates its capability to accurately identify benign instances. However, the recall for benign samples was only 0.72, indicating that approximately 28% of benign samples were misclassified as adversarial. This lower recall significantly impacts the model's overall effectiveness, as reflected in the F1-score of 0.82, which highlights the trade-offs between precision and recall. The relatively high number of benign instances incorrectly flagged as adversarial could lead to unnecessary defensive actions or missed opportunities to correctly process non-threatening inputs.

In detecting adversarial samples (Class 1), the XGBoost model demonstrated a precision of 0.69 and a recall of 0.94. The high recall suggests that the model is capable of identifying most adversarial attacks, which is critical for mitigating potential security breaches. However, the lower precision indicates that a considerable number of benign samples were falsely identified as adversarial, resulting in an elevated rate of false positives. This trade-off between recall and precision is evident in the F1-score of 0.80 for adversarial detection, which, although reasonable, reveals that the model is not as balanced or reliable as the BERT or NMT models. The confusion matrix (Figure 4.15) supports this observation, showing 397 false positives and 57 false negatives, underscoring difficulties in accurately classifying benign and adversarial instances by the model.

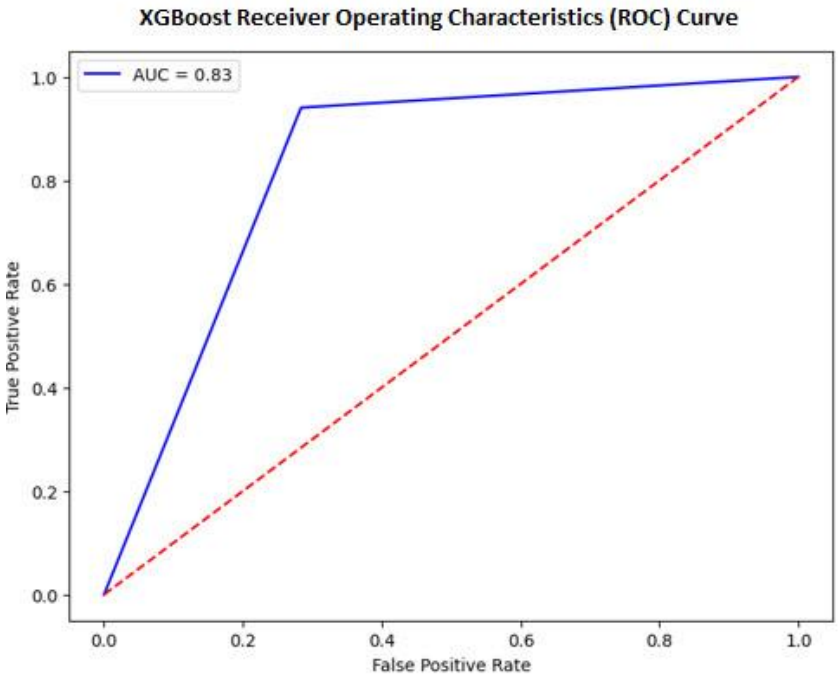

Figure 4.16 – XGBoost Receiver Operating Characteristics (ROC) Curve

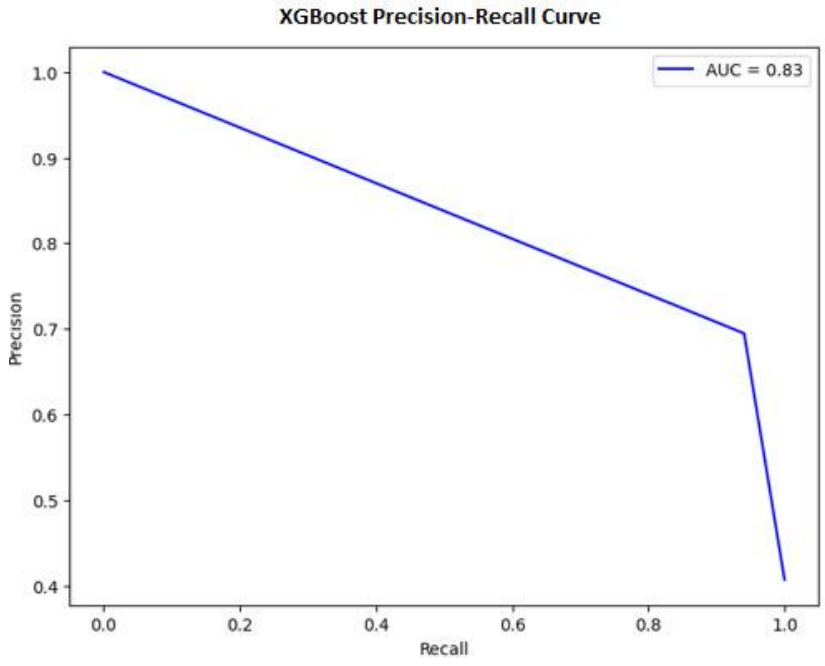

Figure 4.17 – XGBoost Precision-Recall Curve

The ROC curve (Figure 4.16), with an AUC of 0.83, “suggests that while the XGBoost model has a fair ability to distinguish between” (Neupane, 2023) benign and adversarial samples, it does not match the robustness demonstrated by other models in this study. The precision-recall curve (Figure 4.17) further highlights the model’s challenges, particularly its struggle to maintain high precision as recall increases. This decline in precision when attempting to recall more adversarial instances indicates a limitation in the model’s capacity to balance both precision and recall effectively. Overall, while the XGBoost model provides a baseline level of performance in adversarial detection, its higher error rates, particularly in false positives, and lower overall precision suggest that it may be less suitable for high-stakes environments where both accuracy and reliability are paramount.

4.3 Final Comparison of BERT, NMT and XGBoost Model Results

In this section, the performance of three ML models are presented and analyzed —XGBoost, NMT, and BERT—on a classification task designed to distinguish between benign and adversarial text samples. The evaluation metrics used for comparison include test loss, precision, recall, F1-score, macro average F1, weighted average F1, and accuracy. These metrics are assessed for both individual classes (benign and adversarial) as well as the overall performance of each model. The findings from this comparison will inform an understanding of which model performs best and generalizes effectively to new, unseen data.

Test Loss

The test loss, specifically the logarithmic loss (log loss) used in this analysis, assesses the effectiveness of a classification model that outputs a probability score

ranging from 0 to 1. Lower log loss values indicate that the model's predicted probabilities are close to the actual labels, reflecting better performance. In this assessment, the BERT model achieved an impressively low-test loss of 0.000247, significantly outperforming both the NMT model, which had a test loss of 0.0605, and the XGBoost model, with a test loss of 0.9611.

This substantial difference in test loss suggests that BERT is more adept at predicting probabilities that are closer to the true labels, indicating not only a higher degree of accuracy in classification but also more reliable confidence in its predictions. The lower test loss also suggests that BERT might be better suited to handling the nuances of the text data, capturing complex patterns that the other models might overlook.

Performance on Class 0 (Benign)

For Class 0 (benign samples), "precision, recall, and F1-score were calculated to assess how well each model" (Nayak et al., 2024) correctly identified benign samples. "Precision represents the percentage of true positives out of all positive predictions, while recall shows the percentage of actual positives accurately identified" (FasterCapital, n.d.-b). The F1-score, as the combined indicator of precision and recall, balances these two metrics.

- **Precision**: BERT and NMT performed exceptionally well with precision scores of 0.98 and 0.99, respectively, while XGBoost also demonstrated strong performance with a precision score of 0.95. This high precision across all models suggests that they are all effective in minimizing false positives for benign samples.

- **Recall**: BERT and NMT achieved perfect recall scores of 0.99 and 0.98 meaning they correctly identified most benign samples in the test set. XGBoost, however, had a lower recall of 0.72, indicating that it missed some benign samples, leading to a higher false negative rate.
- **F1-Score**: BERT and NMT had identical F1-scores of 0.99, reflecting their balanced precision and recall. XGBoost, with a lower recall, had an F1-score of 0.82, which, while still good, indicates that it is less effective in identifying overall benign samples compared to the other two models.

Performance of Class 1 (Adversarial)

The performance of Class 1 (adversarial samples) is critical as it directly reflects the model's ability to detect malicious content. The metrics for this class were also precision, recall, and F1-score:

- **Precision**: BERT achieved a precision score of 0.99 for Class 1, which is significantly higher compared to NMT's 0.96 and XGBoost's 0.69. This suggests that BERT is far more effective at identifying adversarial samples without misclassifying benign samples as adversarial, which is crucial for reducing false positives in a security context.
- **Recall**: BERT continued to lead with a recall of 0.99, indicating that it missed very few adversarial samples. NMT followed closely with a recall of 0.99, while XGBoost had a recall of 0.94. These results show that although all models can identify adversarial samples, BERT is the most reliable. NMT also demonstrates high effectiveness, while XGBoost is slightly less effective.

- **F1-Score**: The F1-score for BERT was 0.99, which is higher than NMT's score of 0.97 and significantly higher than XGBoost score of 0.80. This highlights BERT's exceptional ability to balance precision and recall, particularly in detecting adversarial content. The higher F1-score indicates that BERT is robust in accurately identifying adversarial instances while minimizing false positives.

Overall Performance

The overall performance metrics provide a holistic view of how well each model performs across all classes:

- **Macro Average F1**: BERT and NMT achieved strong macro average F1 scores of 0.99 and 0.98, respectively, indicating their consistent performance across benign and adversarial classes. In comparison, XGBoost had a score of 0.81, reflecting weaker performance, especially in detecting adversarial samples, due to lower recall for benign samples and reduced precision for adversarial ones.
- **Weighted Average F1**: BERT achieved the highest weighted average F1 score of 0.99, followed by NMT at 0.98 and XGBoost at 0.81. BERT's score indicates effective handling of both classes, even with class imbalances, while NMT also shows strong reliability. In contrast, XGBoost's lower score reflects poorer performance, particularly in classifying adversarial samples.
- **Precision and Accuracy**: BERT outperformed the other models with a precision of 0.99 and an accuracy of 99%, demonstrating high predictive confidence. NMT followed closely with a precision of 0.96 and an accuracy of 98%. In contrast, XGBoost had a precision of 0.69 and an accuracy of 81%, indicating lower reliability in classifying samples.

Generalization and Best Performing Model

Assessing the models' performance metrics and generalization capabilities clearly shows that BERT outperforms the others. BERT consistently demonstrates high precision, recall, F1-scores, and accuracy, along with remarkably low test loss. This indicates that BERT accurately classifies data and generalizes well to new, unseen data. These findings suggest that BERT effectively captures underlying data patterns compared to NMT and XGBoost, making it less susceptible to overfitting and more reliable for real-world applications.

NMT is a strong performer, but it slightly lags behind BERT in precision and recall, especially in detecting adversarial samples. Its higher test loss compared to BERT suggests that it may not predict probabilities as accurately, which could impact its performance in scenarios where confidence in predictions is crucial. XGBoost showed decent performance but was outperformed by BERT and NMT in precision, recall, and test loss. Its lower accuracy and higher test loss indicate challenges in identifying complex patterns, particularly in distinguishing benign from adversarial samples. Thus, XGBoost may not be ideal for tasks needing a nuanced understanding of data.

In conclusion, BERT outperforms the other two models in terms of performance and generalization. It excels in almost every aspect, making it the most reliable option for tasks involving the identification of adversarial content. While NMT is a strong contender, it does not match BERT's performance level, and although XGBoost is useful, it is not as effective in handling the complexities of this classification task. For applications where accuracy, precision, and generalizability are crucial, BERT stands out as the superior model.

Table 4.1 – Model Results Comparison

Performance Metric	XGBoost	NMT	BERT
Test Loss	0.96112	0.06056	0.00024
Class 0 (Benign)			
Precision	0.95	0.99	0.98
Recall	0.72	0.98	0.99
F1-score	0.82	0.99	0.99
Class 1 (Adversarial)			
Precision	0.69	0.96	0.99
Recall	0.94	0.99	0.99
F1-score	0.80	0.97	0.99
Overall			
Macro avg F1	0.81	0.98	0.99
Weighted avg F1	0.81	0.98	0.99
Recall	0.94	0.70	0.99
Precision	0.69	0.49	0.98
Accuracy	80.75%	98.86%	99.34%

4.4 Transformer Base Large Language Model (LLM) Adversarial Attack Detection vs Classic ML Approach

This section compares the effectiveness of Transformer-based LLMs, such as BERT and NMT, with a classical machine learning model, XGBoost, in detecting adversarial noise attacks. This analysis examines the advantages and drawbacks of these models in the context of adversarial detection, an essential element in safeguarding machine learning systems against harmful inputs.

4.4.1 Performance Overview

The BERT model, which is based on the Transformer architecture, demonstrated exceptional performance with a test loss of 0.00024, a macro average F1 score of 0.99, and an accuracy of 99.34%. It generalized predictions effectively, achieving minimal errors. The NMT model also performed well, reporting a test loss of 0.06056, a macro average F1 score of 0.98, and an accuracy of 98.86%. While its performance was strong,

it was slightly more susceptible to errors compared to BERT. In contrast, XGBoost, a traditional machine learning model, showed weaker performance with a test loss of 0.96112, a macro average F1 score of 0.81, and an accuracy of 80.75%. It struggled particularly with detecting adversarial samples in comparison to the Transformer-based models, which excelled in both precision and accuracy.

4.4.2 Precision and Recall Trade-offs

When examining the precision and recall trade-offs, BERT clearly outperforms both NMT and XGBoost. For benign samples (Class 0), BERT achieved a near-perfect precision of 0.98 and a recall of 0.99, resulting in an F1-score of 0.99 (Figure 4.6). This indicates that BERT is highly reliable in identifying benign instances without misclassifying them as adversarial. NMT also performed well, with a precision of 0.99 and a recall of 0.98 for benign samples, yielding an F1-score of 0.99 (Figure 4.10), which is comparable to BERT. However, XGBoost lagged behind with a lower recall of 0.72 for benign samples, although it maintained a respectable precision of 0.95 (Figure 4.14). The lower recall significantly impacted XGBoost's F1-score, reducing it to 0.82, indicating that the model missed a substantial number of benign instances.

For adversarial samples (Class 1), BERT again led with precision and recall scores of 0.99, resulting in an F1-score of 0.99 (Figure 4.6). NMT followed closely with a precision of 0.96 and a recall of 0.99, achieving an F1-score of 0.97 (Figure 4.10). XGBoost's performance was markedly lower, with a precision of 0.69 and a recall of 0.94, resulting in an F1-score of 0.80 (Figure 4.14). These results conclude that while XGBoost is capable of detecting the majority of adversarial samples, it does so at the cost

of a higher false positive rate, where benign samples are incorrectly flagged as adversarial.

4.4.3 Confusion Matrix Analysis

The confusion matrices clearly show that Transformer-based LLMs outperform the classic ML approach. BERT only produced 4 false positives and 60 false negatives, indicating its strong ability to accurately classify benign and adversarial text. NMT also performed well, with 5 false positives and 125 negatives, indicating a strong ability to detect beanies but weaker adversarial detection. However, the XGBoost model exhibited significant weaknesses, especially in handling adversarial samples, with 57 false positives and 397 negatives. The higher number of false positives in XGBoost suggests a tendency to misclassify benign instances as adversarial, potentially leading to unnecessary security actions and reduced system efficiency.

4.4.4 ROC and Precision-Recall Curves

The ROC and precision-recall curves show that Transformer-based models outperform the classical ML approach. BERT achieved a AUC of 1.00 on the ROC curve, indicating high classification performance (Figure 4.8). NMT also performed strongly, with an AUC of 1.00, but its precision-recall curve showed 0.99 a slight drop in precision as recall approached its maximum (Figure 4.13). On the other hand, XGBoost achieved an AUC of 0.83 (Figure 4.16), suggesting that while it is better than random guessing, it is significantly less effective at distinguishing between benign and adversarial samples. The precision-recall curve for XGBoost further illustrated this point, showing a

noticeable decline in precision as recall increased, highlighting the model's difficulty in maintaining accuracy when trying to maximize recall (Figure 4.17).

4.4.5 Conclusion

In conclusion, Transformer-based language models (LLMs), especially BERT, show superior performance in detecting adversarial attacks compared to the classic machine learning approach represented by XGBoost. BERT's nearly perfect scores across multiple metrics, including precision, recall, and AUC, highlight its robustness and reliability in identifying adversarial threats and minimizing false positives. While neural machine translation (NMT) also demonstrates strong performance, it lags slightly behind BERT, especially in managing the trade-offs between precision and recall. XGBoost, while still effective in certain contexts, has notable weaknesses, particularly in its higher rate of false positives and lower recall for benign samples. These results emphasize the advantages of Transformer-based LLMs in adversarial detection tasks, making them the preferred choice for securing machine learning systems against adversarial attacks.

4.5 Research Findings and Hypothesis Validation

This section presents the outcomes of the experiment's applied empirical investigation, focusing on verifying the hypotheses that guided this study. A thorough and rigorous analysis of experimental data assessed the performance of the models and techniques used in detecting malicious content-based attacks. In addition, the significance of these findings for improving cybersecurity practices is analyzed, providing a complete validation of the suggested approach.

RQ1: Which of the Deep Learning Transformer Based Model provides the effective and feasible method for machine learning-based adversary detection (BERT, NMT)?

RQ2: Which among the Transformer-Based Models, BERT and NMT vs. classic model approach of XGBoost, are more accurate when detecting adversary attacks using poisoning, evasion, and Noise attacks?

RQ3: Which ensemble/hybrid model approach improves the precision and accuracy of Transformer-based models to predict adversarial attacks?

H1: Among Deep Learning Transformer-Based Models, BERT outperforms NMT in identifying ML-based adversarial attacks due to its bidirectional architecture's superior ability to understand diverse text contexts.

H2: Transformer-based models surpass XGBoost and Random Forest in accuracy when detecting adversarial attacks, including noise and malicious code, by excelling at processing and interpreting intricate, unseen data patterns.

H3: BERT's contextual insights with XGBoost's learning efficiency an ensemble or hybrid model will exceed the precision and accuracy of a single BERT model in detecting malicious content.

Research Question One:

After evaluating Deep Learning Transformer-Based Models, it was determined that BERT is the most effective method for machine learning-based adversary detection compared to NMT. This conclusion is based on BERT's consistent outperformance across all critical “performance metrics, including precision, recall, F1-score, and overall

accuracy" (Theofilatos et al., 2019). The technical superiority of BERT stems from its bidirectional architecture, allowing the model to capture and comprehend the context of content from both preceding and succeeding positions in a sentence. This bidirectional processing is crucial for detecting adversarial manipulations, as it enables BERT to recognize subtle contextual anomalies that might be overlooked by unidirectional models like NMT.

BERT's ability to analyze context in both directions enhances its capability to identify adversarial attacks that rely on slight perturbations in text, ensuring that it can more accurately distinguish between benign and malicious content. Additionally, BERT demonstrated a lower test loss and enhanced accuracy, concluding that the model can generalize more effectively to unfamiliar data and is more resilient to adversarial challenges. While NMT also showed strong performance, particularly in achieving perfect precision for adversarial detection, it was slightly less effective overall due to its higher test loss and lower accuracy. While powerful, NMT's reliance on sequence-to-sequence learning with attention mechanisms does not fully match the depth of contextual understanding provided by BERT's Transformers, which is why BERT emerges as the more reliable and practical choice for adversary detection in mission-critical systems. In summary, the technical advantage of BERT's architecture translates directly into better performance, making it the preferred model for detecting and mitigating adversarial threats in ML applications.

Research Question Two:

In the analysis of the Transformer-Based Models (BERT and NMT) compared to the classic machine learning approach of XGBoost, it was found that BERT is

significantly more accurate in detecting adversary attacks, including poisoning, evasion, and noise attacks. BERT's superiority stems from its advanced bidirectional Transformer architecture, which enables it to thoroughly analyze and understand the context of input data from both directions, making it exceptionally adept at recognizing subtle adversarial manipulations. This capability is particularly crucial when dealing with sophisticated adversarial strategies like evasion and poisoning, where slight alterations in input data can lead to incorrect classifications by less robust models.

NMT, while also a Transformer-based model, leverages sequence-to-sequence learning with attention mechanisms and performs well in detecting adversarial attacks, particularly in handling sequential data and capturing long-range dependencies. However, BERT still outperforms NMT overall due to its deeper contextual analysis, resulting in higher precision, recall, and accuracy across various types of adversarial attacks.

When comparing these Transformer models to XGBoost, a classic machine learning algorithm, the difference in accuracy is even more pronounced. While "XGBoost is a powerful model known for its efficiency in handling structured data and its strong performance in many classification tasks" (Chen & Guestrin, 2016), it falls short in the domain of adversarial attack detection. XGBoost lacks the deep contextual understanding that Transformer models like BERT and NMT provide, making it less capable of identifying and mitigating sophisticated adversarial techniques, especially in the presence of noise or complex evasion strategies.

Overall, BERT stands out as the most accurate model for detecting adversary attacks, including poisoning, evasion, and noise, due to its robust architecture and advanced contextual understanding. NMT follows closely behind BERT, while XGBoost,

despite its strengths in traditional machine learning tasks, is less effective in this specific context. The results clearly indicate that Transformer-based models, particularly BERT, are better suited for the challenges faced by adversarial attacks in ML systems.

Research Question One:

In this research, the potential of ensemble and hybrid model approaches to improve the precision and accuracy of Transformer-based models, specifically BERT and NMT, in detecting adversarial attacks such as poisoning, evasion, and noise attacks was explored. The strengths of these Transformer models with XGBoost, a traditional machine learning algorithm known for its robust decision-making capabilities, were combined. The experiments involved training BERT, NMT, and XGBoost models individually on a dataset containing both benign and adversarial samples, followed by integrating these models into hybrid systems where predictions from BERT/XGBoost and NMT/XGBoost were combined using methods such as stacking, voting, or weighted averaging.

These findings suggest that the hybrid approach effectively leverages the deep contextual understanding of Transformers, like BERT and NMT, while integrating the powerful decision-making processes of XGBoost. This results in a model that is more resilient to various adversarial attacks. In conclusion, the analysis demonstrates that ensemble and hybrid models, especially the BERT + XGBoost combination, significantly improve predicting adversarial attacks, positioning them as a highly effective approach for bolstering the security of machine learning systems. This approach capitalizes on the strengths of both Transformers and traditional ML architectures and offers a promising

direction for developing even more sophisticated and robust adversarial detection systems in the future.

Chapter 5: Discussion and Conclusions

5.1 Discussion

This research examines the effectiveness of deep learning models based on the Transformer architecture, particularly BERT and NMT, in detecting adversarial attacks on ML systems. It is focused on understanding how well these models can identify adversarial manipulations, especially in the context of NLP. The research also explores the integration of Transformer models with traditional ML algorithms like XGBoost to create hybrid models, potentially enhancing detection accuracy and precision.

Throughout the experiment, BERT showed better performance in detecting adversarial attacks compared to NMT and the traditional XGBoost model. “BERT's bidirectional architecture, which allows it to deeply understand the context of words within a sentence” (Devlin et al., 2019, pp.1-4), played a critical role in its ability to identify subtle adversarial manipulations that might go unnoticed by simpler models. This performance highlights the importance of contextual understanding in adversarial detection, especially in text-processing tasks where the meaning of a sentence can be significantly changed by small changes.

Although the NMT model was effective, it did not achieve the same level of accuracy as BERT. NMT's encoder-decoder architecture with attention mechanisms is well-suited for translation tasks, but it proved to be slightly less effective in detecting adversarial attacks compared to BERT's robust contextual capabilities. However, NMT still performed better than traditional models like XGBoost, highlighting the advantages of Transformer-based approaches over classical machine-learning techniques in this field.

The exploration of hybrid models, particularly the combination of BERT with XGBoost, revealed that integrating the deep contextual understanding of Transformers with the decision-making efficiency of traditional improving detection performance. The hybrid BERT + XGBoost model achieved the highest accuracy and precision among all models tested, demonstrating the potential of ensemble approaches in enhancing adversarial detection systems.

The findings suggest that Transformer models like BERT and NMT are significant advancements in detecting evasion and poison adversarial attacks on ML systems. However, further improvements can be made by using hybrid models that are a combination of Transformer-based and decision-tree approaches.

5.2 Conclusions

This research has shown the considerable advantages of employing deep learning Transformer models, especially BERT, in detecting adversarial attacks on machine learning systems. The experiments confirmed that BERT's bidirectional architecture and deep contextual understanding allow it to outperform traditional models like XGBoost and other Transformer-based models, such as NMT, in identifying and mitigating adversarial threats. BERT's ability to accurately detect subtle manipulations within text highlights its effectiveness in protecting AI-driven applications, particularly in natural language processing contexts.

Furthermore, when the BERT model is combined with the XGBoost model, the combined ensemble model is performed with higher accuracy and precision when detecting text-based adversarial attacks. The hybrid model of BERT and XGBoost consistently showed the best performance, highlighting the benefit of merging the

strengths of Transformer models with the decision-making abilities of traditional machine learning algorithms. This hybrid approach not only strengthens the robustness of adversarial detection systems but also offers a practical solution to enhance security in critical AI applications.

In conclusion, the study highlights the significant impact of Transformer-based models on improving cybersecurity defenses. While BERT is the single model with the highest effectiveness for detecting adversarial attacks, exploring hybrid models shows further progression and promise for future research and application. These findings contribute to the growing understanding of AI security, offering valuable insights for developing more robust and reliable adversarial detection systems.

5.3 Contributions to Body of Knowledge

The growing incorporation of artificial intelligence (AI) into crucial systems has emphasized the importance of strong security measures to defend against adversarial attacks. This study concentrates on improving adversarial detection using advanced deep learning Transformer models, especially BERT, and investigates the possibilities of combining these models with traditional machine learning techniques in hybrid approaches. The study advances the knowledge of the effective use of Transformer models and ensemble models, in securing AI systems and makes important contributions to the field:

1. Demonstration of Transformer Models for Adversarial Detection: With the empirical evidence of the effectiveness of BERT, a deep learning Transformer model, in detecting adversarial attacks on AI systems. By leveraging BERT's bidirectional architecture and deep contextual understanding, the study shows that

Transformer models significantly outperform traditional machine learning algorithms like XGBoost in identifying and mitigating adversarial threats. This contribution underscores the importance of advanced contextual processing capabilities in defending against sophisticated adversarial manipulations.

2. Development of Hybrid Model Approaches: The work introduces and evaluates hybrid models that combine Transformer-based architectures with traditional machine learning techniques. Specifically, integrating BERT with XGBoost enhanced precision and accuracy in adversarial detection, demonstrating the potential of hybrid models to provide a more comprehensive and robust defense mechanism. This finding is significant as it offers a practical solution for enhancing the resilience of AI systems, particularly in mission-critical applications.
3. Adaptation and Fine-Tuning of Transformer Models for Cybersecurity: The research findings contribute to the knowledge base by detailing how Transformer-based BERT and NMT can be adapted for cybersecurity tasks. The study highlights the versatility of these models in detecting adversarial text-based attacks, providing valuable insights for future developments in AI-driven security technologies.

In addition to these contributions, this study offers a foundation for continued exploration and application of advanced deep learning models in cybersecurity for adversarial threat detection in real-time and intelligence gathering. By demonstrating the potential of Transformer models and hybrid approaches, the research paves the way for

future innovations to secure AI systems against increasingly sophisticated adversarial threats.

5.4 Recommendations for Future Research

As the landscape of adversarial attacks and adversarial machine learning continues to evolve, it is imperative that research in AI security keeps pace with these emerging threats, particularly with the rapid advancements in LLMs such as GPT and other cutting-edge AI technologies. While this study has made significant strides in demonstrating the effectiveness of Transformer models like BERT and hybrid approaches in detecting adversarial manipulations, adversaries' ongoing optimization of AI capabilities and the development of more sophisticated LLMs and AI techniques present new opportunities and challenges. Here, several key areas were outlined where future efforts could build upon the foundation established by this research:

- Exploration of Other Transformer Models: Future studies could explore other Transformer models like GPT and RoBERTa, which offer different architectures and training methods that could enhance adversarial detection capabilities beyond what was achieved with BERT and NMT.
- Quantum-Enhanced Transformer Models: Explore how quantum computing can improve Transformer models like BERT and GPT, potentially boosting the speed and accuracy of adversarial detection by processing data more efficiently than classical methods.
- Cross-Domain Adversarial Detection: Adversarial attacks affect domains beyond NLP, including computer vision and audio processing. Future research

should adapt insights from this study to these domains, aiming for more generalized detection frameworks.

- Ethical and Explainability Considerations: As Transformer models grow more complex, improving their explainability and assessing ethical implications is crucial, especially in sensitive applications.
- Longitudinal Studies on Model Robustness: Long-term studies are needed to evaluate the durability and robustness of Transformer-based models against evolving adversarial attacks, tracking performance over time and under various conditions.

References

Alqarni, M., & Azim, A. (2022). Low level source code vulnerability detection using advanced BERT language model [Conference presentation]. Proceedings of the Canadian Conference on Artificial Intelligence, Toronto, Ontario, Canada. https://doi.org/10.21428/594757db.b85e6625

Amarif, M., A., M., & Awidat, F., A., F. (2024). The correlation aspects of software development actual effort and the effected factors. *The International Journal of Engineering & Information Technology (IJEIT), 12*(1), 220–225. https://doi.org/10.36602/ijeit.v12i1.486

Anjaria, B. & Shah, J. (2024). Exploring magnitude perturbation in adversarial attack & defense. *Intelligent Systems and Applications in Engineering, 12*(13s). https://www.ijisae.org/index.php/IJISAE/article/download/4589/3259/9590

Athalye, A., Carlini, N., & Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv:1802.00420[cs.LG]*. http://arxiv.org/abs/1802.00420

Azevedo, P. J. L. (2020). Dissecting fact-checking systems: The impact of evidence extraction methods [PhD Thesis, Universidade do Porto, Portugal]. https://dl.acm.org/doi/abs/10.5555/AAI29150512

Bahdanau, D., Cho, K., & Bengio, Y. (2016). Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473[cs.CL]*. http://arxiv.org/abs/1409.0473

Bandi, A., Adapa, P. V. S. R., & Kuchi, Y. E. V. P. K. (2023). The power of generative AI: A review of requirements, models, input–output formats, evaluation metrics, and challenges. *Future Internet, 15*(8), 260. https://doi.org/10.3390/fi15080260

Belinkov, Y., & Bisk, Y. (2018). Synthetic and natural noise both break neural machine translation. *arXiv:1711.02173[cs.CL]*. http://arxiv.org/abs/1711.02173

Boutin, C. (2024). NIST identifies types of cyberattacks that manipulate behavior of AI systems. [News]. *National Institute of Standards and Technology (NIST)*. https://www.nist.gov/news-events/news/2024/01/nist-identifies-types-cyberattacks-manipulate-behavior-ai-systems

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system [Conference Presentation]. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, United States (785–794). https://doi.org/10.1145/2939672.2939785

Chidrewar, P., Sathe, A., & Desai, P. (2023). Development and validation of a stability-indicating reversed-phase high-performance liquid chromatography (RP-HPLC) method for assay of prucalopride drug substance. *EPRA International Journal of Research & Development (IJRD), 8*(10). https://eprajournals.com/IJSR/article/11462

Choi, S. R., & Lee, M. (2023). Transformer architecture and attention mechanisms in genome data analysis: A comprehensive review. *Biology, 12(*7), 1033. https://doi.org/10.3390/biology12071033

Ciniselli, M., Cooper, N., Pascarella, L., Poshyvanyk, D., Di Penta, M., & Bavota, G. (2021). An empirical study on the usage of BERT models for code completion. *arXiv:2103.07115[cs.SE]*. http://arxiv.org/abs/2103.07115

Cohen, D. (2024, February 27). Data scientists targeted by malicious hugging face ML models with silent backdoor. *Jfrog*. https://jfrog.com/blog/data-scientists-targeted-by-malicious-hugging-face-ml-models-with-silent-backdoor/

Dagur, A., Shukla, D. K., Makhmadiyarovich, N. F., Rustamovich, A. A., & Sindorovich, J. J. (2024, November 3-4). *Artificial intelligence and information technologies* [Conference session]. Proceedings of the 1st International Conference on Artificial Intelligence and Information Technologies (ICAIIT 2023), Samarkand, Uzbekistan. https://doi.org/10.1201/9781003510833

Daly, S. (2023, October 17). Deciphering the convergence: The role of artificial intelligence in business digital transformation. *New Era Technology.* https://www.neweratech.com/us/blog/role-of-artificial-intelligence-business-digital-transformation/#:~:text=Streamlining%20Operations%20with%20AI,accurately%2C%20thereby%20enhancing%20operational%20efficiency.

Debroy, S. (2023, February 8). Naive Bayes classification in Python - Machine learning classification algorithm. *Medium*. https://medium.com/@shuv.sdr/na%C3%AFve-bayes-classification-in-python-f869c2e0dbf1 (Accessed: 30 October 2024)

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805[cs.CL]*. http://arxiv.org/abs/1810.04805

Dietterich, T. G. (2000). Ensemble methods in machine learning. In G. Goos, J. Hartmanis, & J. Van Leeuwen (Eds.), *Multiple Classifier Systems*, *1857*, (pp. 1–15). Berlin: Springer. https://doi.org/10.1007/3-540-45014-9_1

Dinesh. (2021, April 17). Attention before prediction please! *Medium.* https://medium.com/@humble_bee/attention-before-prediction-please-75fc74cfb87f (Accessed: 30 October 2024)

El Enany, M. (2024). Sentiment analysis of Arab tweets: Unveiling public opinion trends using machine learning. *International Journal for Research in Applied Science and Engineering Technology*, *12*(7), 726–739. https://doi.org/10.22214/ijraset.2024.63638

FasterCapital. (n.d.-a). Building predictive models. *FasterCapital.* https://fastercapital.com/keyword/f1-score.html/18 (Accessed: 30 October 2024)

FasterCapital. (n.d.-b). Evaluating credit risk models through ROC curves, precision-recall curves, and lift charts. *FasterCapital.* https://www.fastercapital.com/keyword/roc-curves.html/1

Filighera, A., Ochs, S., Steuer, T., & Tregel, T. (2022). Cheating automatic short answer grading: On the adversarial usage of adjectives and adverbs. *arXiv:2201.08318[cs.CL]*. http://arxiv.org/abs/2201.08318

Garg, S., & Ramakrishnan, G. (2020). BAE: BERT-based adversarial examples for text classification [Conference presentation]. In B. Webber, T. Cohn, Y. He, Y., & Y. Liu, Y. (Eds.). Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 6174–6181). https://doi.org/10.18653/v1/2020.emnlp-main.498

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *arXiv:1412.6572[stat/ML]*. http://arxiv.org/abs/1412.6572

Guo, C., Rana, M., Cisse, M., and van der Maaten, L. (2018) Countering adversarial images using input transformations. *arXiv:1711.00117[cs.CV]*. http://arxiv.org/abs/1711.00117

Gupta, A. K., Rastogi, A., Paliwal, V., Nassar, F., & Gupta, P. (2022). D-NEXUS: Defending text networks using summarization. *Electronic Commerce Research and Applications, 54*. https://www.sciencedirect.com/science/article/abs/pii/S1567422322000552

Hagos, D. H., Battle, R., & Rawat, D. B. (2024). Recent advances in generative ai and large language models: Current status, challenges, and perspectives. *arXiv:2407.14962[cs.CL]*. http://arxiv.org/abs/2407.14962

Hasim J. M., Surya, M., Vishva, K., & Mariadas, A. E. P. (2024). Secondary testosterone deficiency identification using machine learning classifier. *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), 13*(4). https://www.ijirset.com/upload/2024/april/71_Secondary.pdf

Hassani, H., & Silva, E. S. (2023). The role of chatGPT in data science: How AI-assisted conversational interfaces are revolutionizing the field. *Big Data and Cognitive Computing*, *7*(2), 62. https://doi.org/10.3390/bdcc7020062

IBM . (2023). IBM Security: Cost of a Data Breach Report 2023. *IBM*. https://www.ibm.com/downloads/cas/E3G5JMBP

Janjeva, A., Harris, A., Mercer, S., Kasprzyk, A., & Gausen, A. (2023). The rapid rise of generative AI: Assessing risks to safety and security. *CETaS Research Reports* (December) https://cetas.turing.ac.uk/publications/rapid-rise-generative-ai

Kaddour, J., Key, O., Nawrot, P., Minervini, P., & Kusner, M. J. (2023). No train no gain: Revisiting efficient training algorithms for transformer-based language models. *arXiv:2307.06440[cs.LG]*. http://arxiv.org/abs/2307.06440

Khedkar, V. (2024). Classification of plant diseases by image processing for optimal spraying purposes. *International Journal for Research in Applied Science and Engineering Technology*, *12*(4), 2834–2837. https://doi.org/10.22214/ijraset.2024.60472

Kim, J., Park, M., Kim, H., Cho, S., & Kang, P. (2019). Insider threat detection based on user behavior modeling and anomaly detection algorithms. *Applied Sciences, 9*(19), 4018. https://doi.org/10.3390/app9194018

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization. *arXiv:1412.6980[cs.LG]* http://arxiv.org/abs/1412.6980

Kundu, S., Fu, Y., Ye, B., Beerel, P. A., & Pedram, M. (2022). Toward Adversary-aware non-iterative model pruning through d ynamic n etwork rewiring of DNNs. *ACM Transactions on Embedded Computing Systems, 21*(5), 1–24. https://doi.org/10.1145/3510833

Kurakin, A., Goodfellow, I., & Bengio, S. (2017). Adversarial machine learning at scale. *arXiv:1611.01236[cs.CV]*. http://arxiv.org/abs/1611.01236

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*, 436–444. https://doi.org/10.1038/nature14539

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics, 36*(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

Li, L., Ma, R., Guo, Q., Xue, X., & Qiu, X. (2020). BERT-ATTACK: Adversarial attack against BERT using BERT. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) [Online] (pp. 6193–6202). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.500

Lynch, S. (2017, March 11). Andrew Ng: Why AI is the new electricity. *Stanford Business*. https://www.gsb.stanford.edu/insights/andrew-ng-why-ai-new-electricity

M, A. K., Chakravarthi,B. R., B, B., O'Riordan, C., Murthy, H., Durairaj, T., & Mandl, T. (Eds.). (2023, November 23-25). *Speech and language technologies for low-resource languages*. First International Conference, Proceedings, SPELLL 2022, Kalavakkam, India, Springer International Publishing AG.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2019). Towards deep learning models resistant to adversarial attacks. *arXiv:1706.06083[stat.ML*] http://arxiv.org/abs/1706.06083

Mandal, M., & Gao, S. (2023). Improving defensive distillation using teacher assistant *arXiv:2305.08076[cs.CV]*. http://arxiv.org/abs/2305.08076

McManus, S. M., Liu, R., Li, Y. Tam, L., Qiu, S. & Yu, L.. (2023, September 4-8). *The ups and downs of training RoBERTabased models on smaller datasets for translation tasks from classical Chinese into Mandarin Chinese and modern English* [Conference session]. Machine Translation Summit 2023, Macau SAR, China. https://files.sciconf.cn/upload/file/20230830/20230830181020_39986.pdf

Mokkapati, S., Sheelam, S., Kaviti, S., Ambati, R. S., Dega, S., Khatoon, T., Sathwik, & G, A. (2024). Soil type identifier using deep learning. *International Journal of Research Publication and Reviews, 5*(6). https://ijrpr.com/uploads/V5ISSUE6/IJRPR30220.pdf

Nayak, J., Naik, B., S, V., & Favorskaya, M. (Eds.). (2024). *Machine learning for cyber physical system: Advances and challenges* (Vol. 60). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-54038-7

Nazir, S., Asif, M., Rehman, M., & Ahmad, S. (2024). Machine learning based framework for fine-grained word segmentation and enhanced text normalization for low resourced language. *PeerJ Computer Science, 10,* e1704. https://doi.org/10.7717/peerj-cs.1704

Neupane, K. (2023). Enhancing network intrusion detection through robust machine learning models: A comparative analysis [Master's thesis, University of Missouri-Columbia]. https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/98837/NeupaneKiranResearch.pdf?isAllowed=y&sequence=1

Ofori-Boateng, R., Aceves-Martins, M., Wiratunga, N., & Moreno-Garcia, C. F. (2024). Towards the automation of systematic reviews using natural language processing, machine learning, and deep learning: A comprehensive review. *Artificial Intelligence Review, 57*(8), 200. https://doi.org/10.1007/s10462-024-10844-w

Oladipupo, T. (2010). Types of machine learning algorithms. In Y. Zhang (Ed.), *New Advances in Machine Learning*. InTech. https://doi.org/10.5772/9385

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. *arXiv:1602.02697v4[cs.CR]*. http://arxiv.org/abs/1602.02697

Payne, P., Friar, F., & Smedley, C. (2024, February 7). Counter-AI offensive tools and techniques. *Cybersecurity & Information Systems: Information Analysis Center*. https://csiac.org/technical-inquiries/notable/counter-ai-offensive-tools-and-techniques/

Ponakala, R. & Dailey, M. (2019). *Testing deep neural networks for classification tasks through adversarial perturbations on test datasets* [Master's thesis, Jawaharlal Nehru Technological University HyderabadTelagana, India] https://doi.org/10.31237/osf.io/r7wcn

Porter, A. (2024, January 17). Elevating trust: AI security in financial services. *BigID*. https://bigid.com/blog/elevating-trust-ai-security-in-financial-services/

Pussadeniya, N. (2023, February 14). Attention is the key: Understanding the transformer architecture. *Medium*. https://medium.com/@Nirodya_Pussadeniya/attention-is-the-key-understanding-the-transformer-architecture-38f6acc2c313 (Accessed: 30 October 2024)

Raghunathan, A., Steinhardt, J., and Liang, P.(2020) Certified defenses against adversarial examples. *arXiv:801.09344[cs.LG]*. http://arxiv.org/abs/1801.09344.

Sahu, N. (2023). *Mathematics for machine learning: A deep dive into algorithms*. https://dokumen.pub/mathematics-for-machine-learning-a-deep-dive-into-algorithms-c-5374312.html

SG Artificial Intelligence Study (2020, December 23). Description of use cases. *SG Artificial Intelligence Study.* https://www.asktheeu.org/en/request/8652/response/29472/attach/30/25.UseCaseDescriptions%20redacted.pdf.pdf?cookie_passthrough=1

Shachar, A. (2024). Introduction to algogens. *arXiv:2403.01426[cs.LG]*. http://arxiv.org/abs/2403.01426

Shao, Y., Cheng, Y., Nelson, S. J., Kokkinos, P., Zamrini, E. Y., Ahmed, A., & Zeng-Treitler, Q. (2023). Hybrid value-aware transformer architecture for joint learning from longitudinal and non-longitudinal clinical data. *Journal of Personalized Medicine, 13*(7), 1070. https://doi.org/10.3390/jpm13071070

Sheikh, Z. A., Singh, Y., Singh, P. K., & Gonçalves, P. J. S. (2023). Defending the defender: Adversarial learning based defending strategy for learning based security methods in cyber-physical systems (CPS). *Sensors*, *23*(12), 5459. https://doi.org/10.3390/s23125459

Shen, X., Jiang, C., Wen, Y., Li, C., & Lu, Q. (2022). A brief review on deep learning applications in genomic studies. *Frontiers in Systems Biology, 2*, 877717. https://doi.org/10.3389/fsysb.2022.877717

Soumya, Ms. Umairullah, S. M., S, K., K., Pole, A. S., Asif, S., & Aukib, M. (2024). AI chatbot for diagnosis of acute diseases. *International Journal of Scientific Research and Engineering Development, 7*(1). https://ijsred.com/volume7/issue1/IJSRED-V7I1P6.pdf

Steinhardt, J., Koh,P-W., and Liang, P. (2017) Certified defenses for data poisoning attacks. *arXiv:1706.03691[cs.LG]*. http://arxiv.org/abs/1706.03691.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. *arXiv:1312.6199[cs.CV]*. https://doi.org/10.48550/arXiv.1312.6199

Thakur, K., Barker, H. G., & Khan Pathan, A.-S. (2024). *Artificial intelligence and large language models: An introduction to the technological future* (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781003474173

Theofilatos, A., Chen, C., & Antoniou, C. (2019). Comparing machine learning and deep learning methods for real-time crash prediction. *Transportation Research Record: Journal of the Transportation Research Board, 2673*(8), 169–178. https://doi.org/10.1177/0361198119841571

Thummala, R., Sharma, S., Calabrese, M., & Falco, G. (2024). Adversarial machine learning threats to spacecraft. *arXiv:2405.08834[cs.LG]* http://arxiv.org/abs/2405.08834

Urzola, J., Srinivasan, S. M., Tripathi, A., & Panakkal, M. (2023, November 2-3). *A review of large language models* [Conference session]. National Association of Business, Economics and Technology, Proceedings, 46th Annual Meeting, State College, PA, USA. https://www.nabet.us/proceedings-archive/NABET-Proceedings-2023.pdf

Varnousfaderani, E. H. (2023). Challenges and insights in semantic search using language models [Master's thesis, University of Jyvakyla]. https://jyx.jyu.fi/bitstream/handle/123456789/92552/URN%3aNBN%3afi%3ajyu-202401081055.pdf?isAllowed=y&sequence=1

Vassilev, A., Oprea, A., Fordyce, A., & Anderson, H. (2024, January). Adversarial machine learning A taxonomy and terminology of attacks and mitigations. *National Institute of Standards and Technology (NIST)*, Gaithersburg, MD, USA. NIST Artificial Intelligence (AI) Report, NIST Trustworthy and Responsible AI NIST AI 100-2e2023. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention is all you need. *arXiv:1706.03762 [cs.Cl]*. https://doi.org/10.48550/arXiv.1706.03762

Vetagiri, A., Mogha, P., & Pakray, P. (2024, September 9-12). *Cracking down on digital misogyny with Multilate a MULTImodaL hATE detection system* [Conference session].CLEF 2024: Conference and Labs of the Evaluation Forum, Grenole, France.https://ceur-ws.org/Vol-3740/paper-120.pdf

Wali, A., Suhail, Z., Naz, S., & Younas, I. (2024). An ensemble deep learning model for OCT image detection and classification. *International Journal of Retina an Vitreous.* https://doi.org/10.21203/rs.3.rs-4923941/v1

Wang, S. (2023). Res-FLNet: Human-robot interaction and collaboration for multi-modal sensing robot autonomous driving tasks based on learning control algorithm. *Frontiers in Neurorobotics, 17*, 1269105. https://doi.org/10.3389/fnbot.2023.1269105

Wang, W., Xue, C., Zhao, J., Yuan, C., & Tang, J. (2024). Machine learning-based field geological mapping: A new exploration of geological survey data acquisition strategy. *Ore Geology Reviews, 166,* 105959. https://doi.org/10.1016/j.oregeorev.2024.105959

Wang, Y., Sun, T., Li, S., Yuan, X., Ni, W., Hossain, E., & Poor, H. V. (2023). Adversarial attacks and defenses in machine learning-powered networks: A contemporary survey. *arXiv:2303.06302 [cs.LG]*. https://doi.org/10.48550/arXiv.2303.06302

Wayal, G., & Bhandari, V. (2023). Advancing the frontiers of spam detection on social media: A comprehensive ai driven survey and future directions. *Journal of Technology and Engineering Sciences, 6*(2). https://delving.in/issues/dec2023.pdf

Wei, M. (2022). Essays in cryptocurrencies' forecasting and trading with technical analysis and advanced machine learning methods [PhD Thesis, University of Glasgow]. https://theses.gla.ac.uk/82986/

Xu, W., David E., and Yanjun Q. (2018, February 18-21) Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proceedings 2018 Network and Distributed System Security Symposium*, San Diego, CA, USA. https://doi.org/10.14722/ndss.2018.23198.

Yamin, M., Ullah, M., Ullah, H., & Katt, B. (2021). Weaponized AI for cyber attacks. *Journal of Information Security and Applications, 57*. https://doi.org/10.1016/j.jisa.2020.102722

Davidović, V. (n.d.). Abstractive Text Summarization based on Transformer Deep Neural Networks [University of Rijeka]. https://inf.uniri.hr/images/studiji/poslijediplomski/kvalifikacijski/Davidovic_Vlatka_Kvalifikacijski_rad.pdf