Thesis for the Degree of BSc in Computer Science, Forensics and Cybersecurity

Exploring Adversarial Machine Learning

Kyla Jade Franks

Department of Computer Science and Mathematics
Graduate School
South East Technological University
Waterford, Ireland

October, 2024

Exploring Adversarial Machine Learning

Kyla Jade Franks

Department of Computer Science and Mathematics
Graduate School
South East Technological University
Waterford, Ireland

October, 2024

Exploring Adversarial Machine Learning

by

Kyla Jade Franks

Advised by

Dr. Bernard Butler

Submitted to the Department of Computer Science and Mathematics
and the Faculty of the Graduate School of
South East Technological University in partial fullfillment
of the requirements for degree of
BSc in Computer Science, Forensics and Cybersecurity

Contents

List of Figures

1 Introduction

AI security is an interdisciplinary field focused on preventing accidents, misuse, or harmful consequences arising from artificial intelligence systems. These systems range from reactive machines to self-aware models. However, they face significant risks, particularly in the area of data safety, which is crucial for machine learning.
The swift proliferation of artificial intelligence (AI) technology has transformed the real world and industries, from autonomous vehicles and technology to healthcare and more. However, the ever-growing dependence on machine learning (ML) models creates a litany of vulnerabilities to adversarial risks, where malicious adversaries or actors are able to exploit and manipulate input data that is in training, in order to mislead and deceive the systems. Such a vulnerability highlights the importance for Adversarial Machine Learning (AML), a field that focused on finding through investigation any of these threats, to mitigate them and create more robust and reliable systems.
AML explores adversarial attacks—data poisoning, evasions, or model extraction, and how these exploit the vulnerabilities of ML models, and how to further protect the systems provenance and integrity. Attacks such as these could lead to many consequential consequences, especially in sensitive cases where it is vital to keep the systems secure and reliable, an example being medical diagnostics being misclassified. This report divulges into the variety of methods and concepts surround AML, the role it has in bettering AI security, the evolution of AML, and more, all aiming to mitigate the threats of adversaries in this ML-revolutionised world.
As AML evolves and develops, the contributions it has are pivotal when it comes to guaranteeing AI systems are to remain robust and dependable. The nuances of adversarial defences and attacks are investigated further through various scholar-published papers and reports, all aiding in the goal to create a report that provides well-developed insights into the challenges at present, and the potential future plans in AML research. Therefore, highlighting the significance in the wider expanse of AI security and integrity.
[1, 2, 3, 4, 5].

2 Research Questions

1. What is Adversarial Learning?

- Adversary: A rival, opponent, or contestant.
- Adversarial: Two sides who are oppositions, two opponents.
- Learning: Acquisition of knowledge or skills.

2. What is the problem, and why is it important? The problem with adversarial ML is that this could lead to the following.

- Data poisoning attacks
- Evasion attacks

- Data breaches, privacy, misuse

3. Can this be used to our benefit? Is there a side beyond the negative?

4. How can we protect ourselves against AML attacks? What are these defence mechanisms?

2.1 Backup Questions

- What are the standard practices or criterion that should be implemented to best protect from these attacks?
- How to best protect from these attacks in more high-stake situations?
- Legal and ethical issues associated with AML?
- Can we effectively anticipate future adversarial threats or attacks?
- Is AML applied beyond text or image data? If so, how?

3 Hypothesis

4 Methodology

Throughout this report a variety of methodologies were implemented, for purposes from organisation to research. The methodologies employed comprise of an organised and structured approach to assist in exploring AML, focusing on theoretical analysis, with a small amount of practical implementation. Below is an elaboration on the methodology tools and types that were occupied.

4.1 Agile

Agile frameworks were primarily used for management over this report. Agile is a term that covers methodologies, frameworks, and practices mainly used in software development and project management. This system lays emphasis on flexibility, iterative progression over periods knowns as “sprints”, and collaboration across projects. Agile as a methodology was highly useful to keep on track and visually communicate the progress of this report, without losing sight on tasks that took priority.

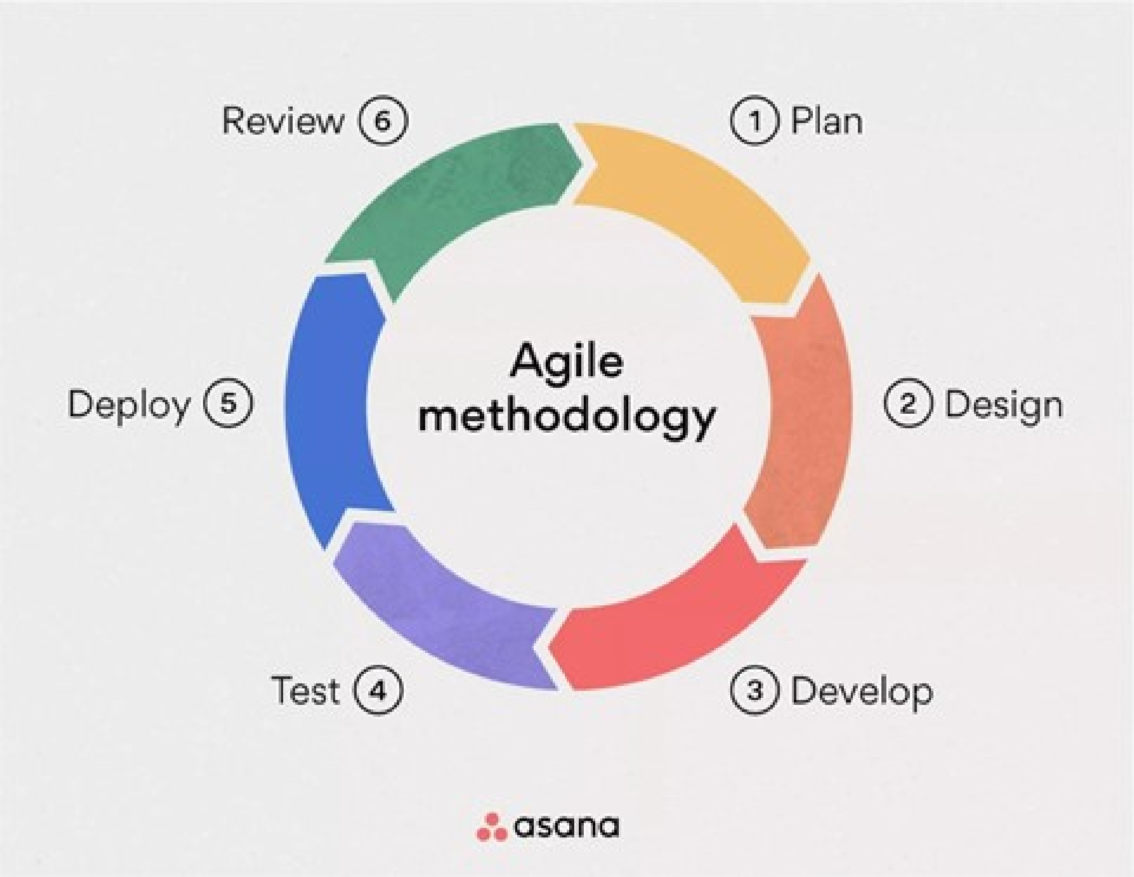
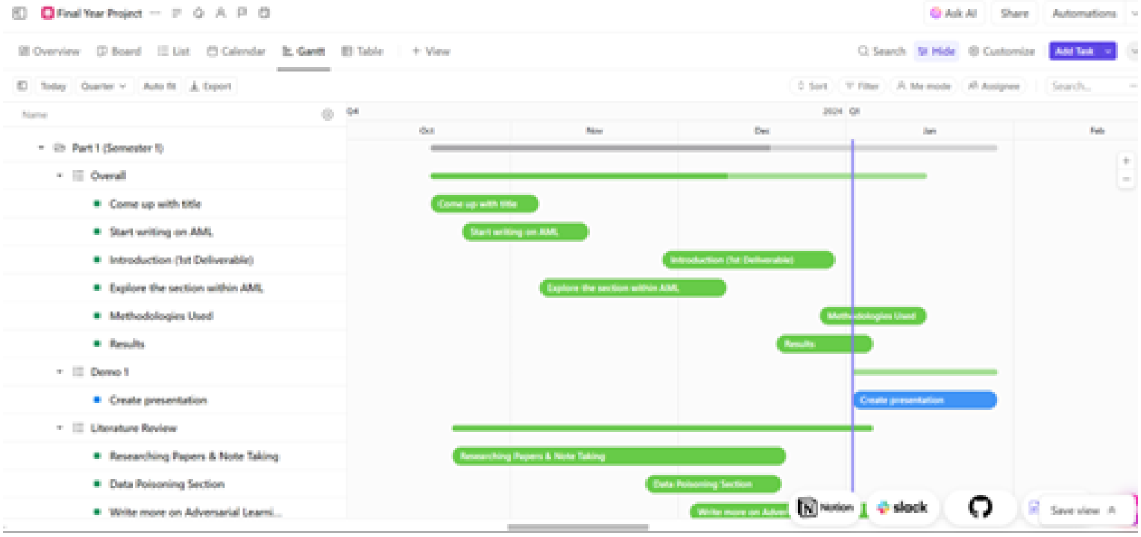


Figure 1: Agile Systems

4.1.1 Gannt Diagram

A Gannt Diagram was used to plan out the timeline of tasks that need to be completed. This visual methodology was proficient as it created an easy-to-read timeline comprised from the umbrella tasks and phases, aiding in the assurance of milestones being met.



Part 1 (Semester 1)
Overall
Come up with title
Start writing on AML
Introduction (1st Deliverable)
Explore the section within AML
Methodologies Used
Results
Demo 1
Create presentation
Literature Review
Researching Papers & Note Taking
Data Poisoning Section
Write more on Adversarial Learni...

Figure 2: Gantt Diagram for Report

4.1.2 Kanban

Kanban is another visual management method that assists the user to focus on continuous delivery and completion of their tasks. This was created through the use of Trello, a simple application that allows users to create kanban boards. Using Kanban was highly beneficial, not only due to prior knowledge of this methodology from my certificate in such, but mainly because it was flexible being able to organise and adjust all the tasks in a clean and organised manner.

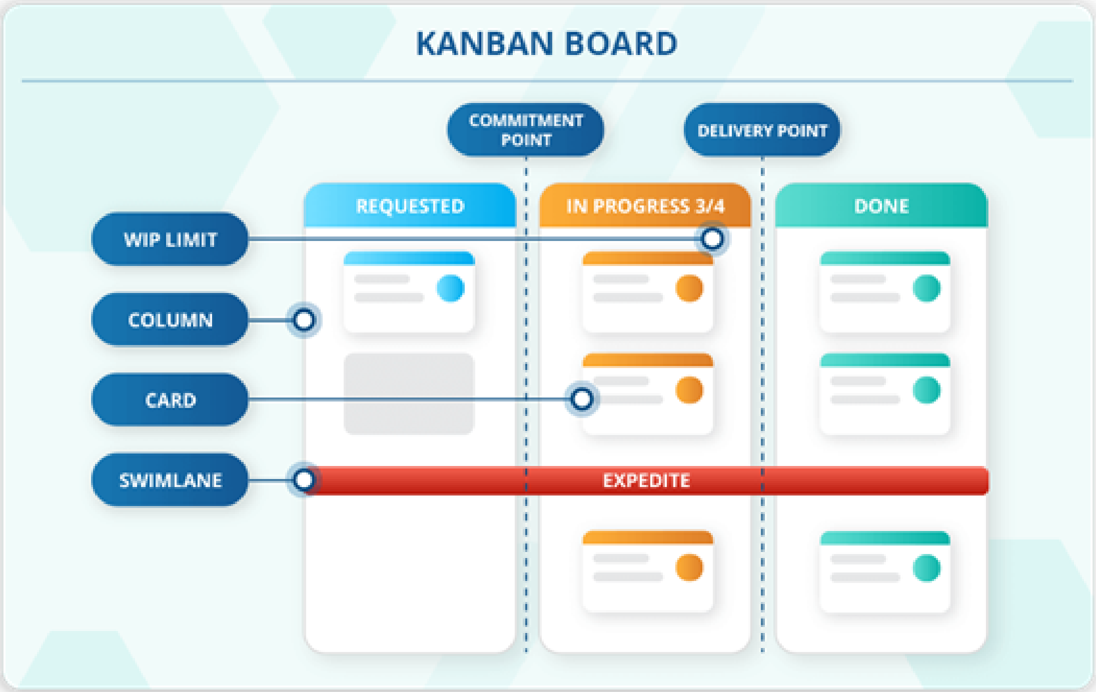
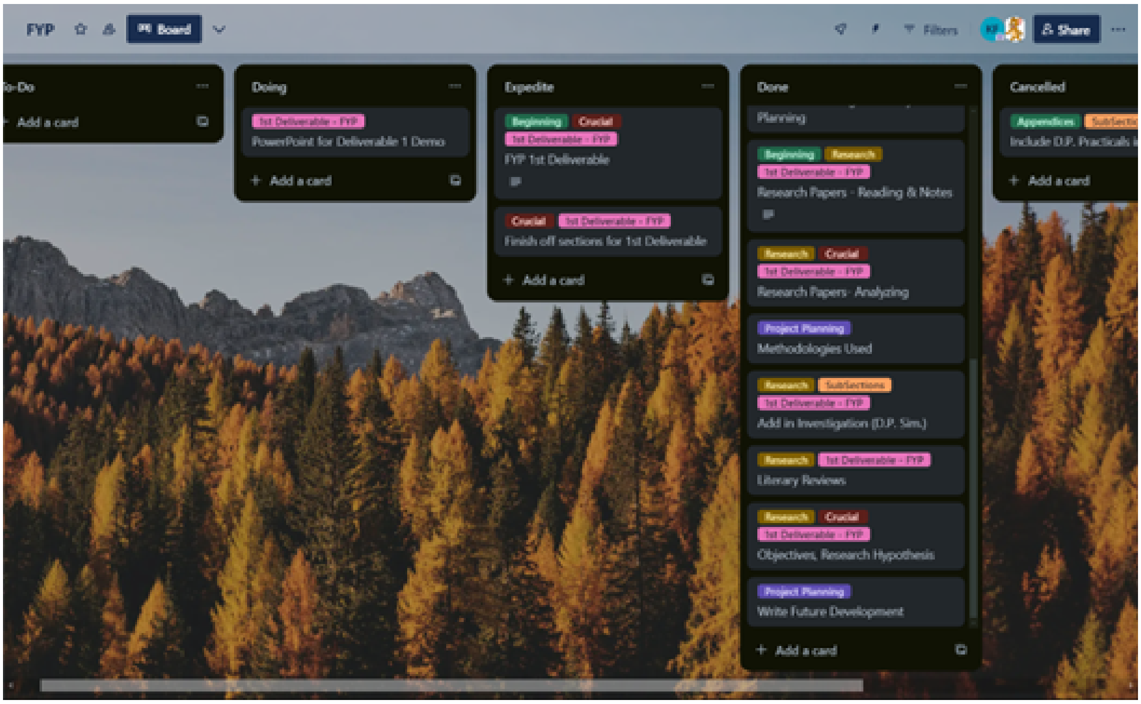


Figure 3: Kanban Systems

4.1.3 Trello



o-Do
Add a card

Doing
1st Deliverable - FYP
PowerPoint for Deliverable 1 Demo
Add a card

Expedite
Beginning Crucial 1st Deliverable - FYP
FYP 1st Deliverable
Crucial 1st Deliverable - FYP
Finish off sections for 1st Deliverable
Add a card

Done
Planning
Beginning Research 1st Deliverable - FYP
Research Papers - Reading & Notes
Research Crucial 1st Deliverable - FYP
Research Papers - Analyzing
Project Planning
Methodologies Used
Research SubSections 1st Deliverable - FYP
Add in Investigation (D.P. Sim.)
Research 1st Deliverable - FYP
Literary Reviews
Research Crucial 1st Deliverable - FYP
Objectives, Research Hypothesis
Project Planning
Write Future Development
Add a card

Cancelled
Appendices SubSectio
Include D.P. Practicals i
Add a card

Figure 4: Kanban Board for report, planned on Trello

4.2 Research-Oriented Tools

4.2.1 Zotero

Zotero is a research assistant tool that allows users to save papers for further organisation, with features allowing annotations, organisational features, and more, all ensuring a comprehensive and encompassing literature review.

4.2.2 Google Scholar

Google Scholar is main source for finding scholar-published papers and research reports.

4.2.3 Overleaf

This is a tool used for structuring content into academic reports, with correct citations and formatting.

4.3 Experimental Implementation

4.3.1 Python notebooks

Python notebooks was used for the label-flipping simulation. It was used as a testing ground for the practical analysis.

5 Literature Review

5.1 What is Adversarial Learning?

Adversarial Machine learning (AML)—and training, is a concept of a machine learning (ML) paradigm in which digital models and systems are trained to perform and carry out certain tasks and functions, that will ameliorate the performance and robustness of these models when in the presence of adversaries and their conditions. Adversarial training is frequently connected to models and systems that are exposed to the manipulation of attackers and adversaries, such as malicious inputs that are created to utilise their vulnerabilities and weaknesses for exploitation. The primary goal of adversarial training is to design machine models and systems to become more robust and resilient to such exploits and attacks, so that in these instances they will react resolutely. [6]

5.1.1 Key Concepts of Adversarial Learning

Adversarial Training

Adversarial training is a method conveyed to enhance the strength of the ML model and system. This is done through the approach of supplementing data training sets with examples that may be augmented by adversaries. The ML model will then not be completely vulnerable to these examples in real world situations; thus, the model may be able to respond robustly and have the ability to correctly classify such adversarial examples and instances—challenging the attacker as this makes it more difficult for the adversary to succeed. [7]

Adversarial Examples

Adversarial examples consist of deliberate inputs that are designed to mislead and corrupt the ML model. These are created specifically to cause the models and systems to make a mistake and perturb the model. An example of this could start from a simple methodical attack where noise is added to an image, misleading the neural network, and causing misclassification. This will be discussed in more detail further on. [8]

Attacks methods

These are the methodologies and techniques implemented to carry out AML attacks and create adversarial examples. These are used to test the models, as well as train the systems to be better equipped to handle such attacks when they are faced with them in real-world-scenarios. Examples of these vary from Fast Gradient Sign Method (FGSM) to Projected Gradient Descent (PGD).

Defence Methods

These are the strategies put in place—as well as the techniques developed, to build ML model resilience towards attacks conveyed by adversaries, id est adversarial training, input preprocessing, and robust optimisation. [9]

Generative Adversarial Networks (GANs)

Generative Adversarial Networks is an application of adversarial learning that comprises of two types of networks: a discriminator and a generator.

The Generator attempts to create the data that will mimic the actual genuine data, while the discriminator's role is to attempt to make distinguished decisions between the genuine and generated data. This AML structure enhances the generated data quality. [10]

5.1.2 Adversarial Training Methods

Adversarial Training

This training method, as aforementioned, involves the dataset in training to be augmented with the use of adversarial examples; thus, sanctioning the ML model to be able to resist ad recognise potential manipulations methods, and exploitation. [11]

Defensive Distillation

Defensive distillation is a technique that trains the ML model to output the soft labels, or class probabilities, instead of the hard decisions, which even the decision boundaries of the model; thus, making it more resilient to adversarial manipulations and such perturbations. [12]

Ensemble Methods

Ensemble methods is a method that utilizes various different models in order to conduct predictions that will enhance the robustness. [13]

Gradient Masking

Gradient Masking is a defence method that is used in adversarial ML to try and conceal certain gradients that attackers may try to exploit in order to create adversarial attacks and examples. The defence is carried out by obscuring the gradients to obstruct the attackers ability to create effectively perturbed estimations that would potentially mislead the model. [14]

5.2 Types and Strategies of AML Attacks

5.2.1 Evasion Attacks

In evasion attacks the adversary modifies the data that is inputted during the model's training phase to manipulate the model to misclassify data and lead to erroneous and imprecise results. Such attacks take place after the ML model has been trained, when the poisoned and tampered data is used. [11]

5.2.2 Poisoning Attacks

Data Poisoning attacks involve malicious data being injected into the model when it is being trained. The aim is to corrupt the ML model's learning process to manipulate the outputs and leads to degradation in performance and introduce vulnerabilities during the deployment. [11]

5.2.3 Model Extraction Attacks

Model extraction attacks—also known as model stealing, are attacks that aim to replicate a target ML model by exploiting the outputs. The adversaries use the models that are duplicated without accessing the original training data. [13]

5.2.4 Inference Attacks

Inference attacks aim to deduce the sensitive data and information that regards the model's data training from the model's outputs, thus potentially affecting and potentially compromising the data privacy. [15]

5.3 Relationship between AML & AI Security

AML being the specialized area that it is within the AI cohort, analyses how AI models may be intentionally attacked and manipulated by adversaries, as well as how to defend these models and systems from such threats. The relationship between AML and AI security is complex and intricate, due to AML presenting the risks and vulnerabilities that are inherent in these AI systems, and further seeking to amplify their dependability, integrity, and robustness.

5.3.1 The convergence of AML and AI Security

AI security encompasses a plethora of measures and strategies applied to assure reliability and protect the systems from threats. AML is a crucial constituent of AI Security; due to the insight it provides around AI models being compromised or manipulated and offers methodologies to strengthen them against adversarial threats. By mitigating and understanding adversarial attacks, AML may contribute towards the advancement of secured and more resilient AI systems.

5.3.2 Literary Review on AML and AI Security

There is a litany of professional scholarly papers that have been produced around the topic of adversarial machine learning, as well as AI security, below I will give an overview of a few of such.

“Adversarial Machine Learning and Cybersecurity” Through this report, the scholars delve into the threats and risks that are creates by adversarial attacks on AI systems, as well as the difficulties and challenges we have in defending models from such attacks. Additionally, it discusses the legal considerations that may be involved, while emphasizing the significance that AML possesses in the wider expanse of AI security. [16]

“A Systematic Review of Adversarial Machine Learning Attacks, Defensive Controls, and Technologies” This paper supplies an extensive review of copious AML attacks and the defensive controls and countermeasures that were implemented to fortify the systems against the threats of adversaries and attackers. Furthermore, the authors discuss elements around the ongoing challenges faced during securing AI systems, highlighting aspects such as formal verification, privacy parameters, the MITRE ATLAS framework—”Adversarial Tactics, Techniques, and Common Knowledge for Machine Learning”, the main difficulties faced in research around AML, and more; all underpinning the limitations that surround AML. [17]

“Artificial intelligence (AI) cybersecurity dimensions: a comprehensive framework for understanding adversarial and offensive AI” Through this paper, aspects of AI are examined through a cybersecurity lens, including the role that AML possesses in identifying and mitigating adversarial threats, as well as discussing the implications this may have for AI security, elaborated below. [18]

5.4 Implications for AI Security

Through the integration of AML into AI security, there are diverse strategies that are quintessential for the security and integrity of the systems, as well as several other reasons.

5.4.1 Legality and Ethics

AML has a pivotal role shaping ethical and legal frameworks for AI, through identifying threats that would have manipulated results, breaching intellectual property, or exploiting vulnerabilities in systems and data. The research conducted in this topic only goes to further enhance the legal governance of AI and ML models by identifying these vulnerabilities as what they are and creating defences to mitigate them. Understanding adversarial attacks assists in informing the legal and ethical frameworks required to address such misuse of AI systems and technologies. With regards to ethics, AML advocates for privacy protection and transparency in AI systems. Biases are mitigated during adversarial attacks, which further safeguards the highly sensitive data from any potential exploitation and inference threats. This is in alignment with the EUs GDPR (General Data Protection

Regulation) as well as the US AI Bill of Rights, highlighting the significant part that AML has in not only security, but also the integrity of AI systems. [19]

5.4.2 Advancing Model Validity

AML is a primary contribution towards improving the robustness of AI models, through vulnerability identification, and then further addressing them. This involves various methodologies and implications. Through further analysis around adversarial attacks, AML contributes to improving models to be more robust and resilient when faced with threats and potential manipulation, improving the overall reliability. Additionally, AML enhances AI models robustness by recognising the inherent risks and vulnerabilities, further aiming to address such through implementation of specific strategies, such as defensive distillation, or adversarial training. As a result, the models become more resilient, the systems are more reliable, increased scalability, and the models are more robust against potential threats, which is especially beneficial for sensitive data and critical fields such as healthcare.

5.4.3 Informing Robust Security Policies

Studies in AML provide essential insights that are used to guide policies and practices to further protect AI systems from potential adversarial threats. Additionally, it aids in data security, threat modelling, and assists in creating reactive measures to combat adversarial attacks.

Overall, adversarial machine learning has a pivotal part in the security of artificial intelligence, whether it be threat identification, building better defence techniques, or guiding the legal and ethical frameworks to secure AI systems and ML models from potential adversarial threats. It is critical for research to be further developed if there are hopes in advancement in the security and provenance of such technologies.

5.5 Evolution of Adversarial Machine Learning

The evolution of Adversarial Machine Learning has significantly grown and evolved since when it was founded. With ML model vulnerabilities being addressed, researchers are able to further investigate adversarial manipulations and the threats this has.

The origin of AML can be seen to go back to the early 2000s, showing significant developments in intrusion detection systems and spam filtering. In 2004, researchers Nilesh Dalvi as well as others highlighted linear classifiers susceptibility to "evasion attacks", where the adversary inserts "good words" into the system to bypass detection. Thus, underscoring the necessity for more robust ML models that would be able to resist adversarial inputs and attacks, and be more resilient towards such. [11]

AML as a term was introduced in 2011 through the paper "Adversarial Machine Learning", by scholars Huang et al. Defined as the "study of effective machine learning techniques" when faced with an adversary, this paper classified attack methods, which cultivated a discussion around the related risks and potential mitigation strategies for AML. [20]

Between 2012 and 2014 was an important period for AML research due to the significant advance in studies. The paper: “Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning”, displays the first gradient-based attacks on ML models, divulging the vulnerabilities exposed to non-linear classifiers, some of which include neural networks and Support Vector Machines (SVMs)—discussed below. [21] Simultaneously, the paper: “Explaining and Harnessing Adversarial Examples” highlighted deep neural networks being able to be deceived by adversarial tampering through gradient-based methods, stressing better security methods as a necessity in deep learning algorithms. [22]

AML has continued to grow over the past decade to encompass an array of defence strategies and attack vectors; spanning cybersecurity, autonomous systems, and more. Researchers and scholars have developed numerous strategies to support the robusticity, such as adversarial training, ensemble methods, defensive distillation and more. However, despite these improvements, there are still many challenges and threats that persist, underscoring the essential need for more resilient models. [21]

Conclusively, AML’s evolution thus far reflects a potent cooperation between finding vulnerabilities in ML models as well as developing advanced defensive mechanisms. Especially as AI systems become more integral in critical cases such as healthcare and finances, the significance of AML maintaining reliability and security of AI systems is more and more recognised.

5.6 Why is AML is critical in present day AI Security?

Adversarial Machine Learning is critical in AI security attributable to its focus on vulnerability identification and the mitigation of risks and potential threats that ML models may be exposed to. This is especially due to the security concerns from AI systems becoming more integrated in different sectors, with a higher chance of risk from adversaries.

AML manages potential challenges by investigating how the adversaries and malicious attackers try to exploit or deceive the models, and further try implanting strategies to protect the systems. An example may be seen through input data being exposed to adversarial manipulation to mislead the models and outputs. This could, if in real-world cases, have severe consequences, especially when we are faced with autonomous vehicles and healthcare systems. [17]

Additionally, AML adds further to ML model resilience through defence advancement. Techniques like adversarial training and defensive distillation are used, as aforementioned, to provide security and resilience against adversarial manipulation. All crucial if there are hopes in maintaining provenance and integrity of ML models and AI systems. [23]

5.7 Data Poisoning

As data poisoning is a subsector attack of AML, that focuses specifically on strategies to exploit and manipulate ML systems I order to achieve the adversary's malicious intents and goal. Below I will be taking a deeper look at data poisoning, and what this specific attack method entails.

5.7.1 What is Data Poisoning?

Data Poisoning is a category of attacks on Machine Learning (ML) or Artificial Intelligence (AI) models, where an adversary deliberately introduces manipulated, malicious, or tampered data into the training set, in hopes to compromise the model's performance or behaviour. [24] These attacks aim to corrupt the model by tampering with the data the machine learns from. The outcomes of this range from biased, or incorrect results, to even harmful and influenced outcomes, when the model's deployed to the public. [25]

As Data Poisoning can be such a sophisticated adversarial attack, targeting not only Artificial Intelligence systems, but also ML (Machine Learning) models, it is quintessential that we find methods and strategies that may be implemented to deter these attacks as much as possible. In said attacks, the adversary's goal to obtain is to compromise the model's behaviour, performance, and integrity. They do this through a litany of measures, but most with the similarity of injections of malicious data into the training datasets that are occupied by ML models; thus, the imminent and pressing threats to the reliability and security of AI systems, and cybersecurity.

The foremost objective of data poisoning is the adversaries aim to influence the model's performance and behaviour, for their own benefit. An example is, in a system that aims to detect spam, the attacker may aim to manipulate the data in training, to ensure their spam emails bypass detection. Juxtaposed with backdoor attacks, where the models are trained to perform as they do normally, until certain triggers are introduced, in which they produce the output the attacker wants.

5.7.2 Types of Data Poisoning Attacks

[26]

Label Flipping

- This is when the labels of certain data points are altered on purpose, usually maliciously, for the intention to confuse the model, and cause misclassification.
- Scenario: Image classification systems, such as those detecting the difference between dogs and cats, the attacker may change and swap labels, i.e. label an image of a dog to "cat," so that this would purposefully confuse the model to misclassify the images during output. [27]

Data Injection

- This is when malicious data is introduced into the system and training set, with the intention and aim to influence the model behaviour. Such injection points are created to affect the behaviour of the model in a certain way.
- Scenario: A system purposed to detect spam, as aforementioned. [28]

Backdoor Attack

- A backdoor attack happens when triggers are embedded within the training data and leads to misclassification from the model when certain triggers are used. This is discussed further below.
- Scenario: Though this may be a widespread problem, it is also a possible example of a backdoor attack. In facial recognition systems, the attacker may aim to train the model to wrongly identify normal users as target individuals, when wearing a certain accessory—a hat or glasses. [29]

Data Modification

- This occurs when an attacker alters the characteristics or content of the existing data within the training set, in order to mislead the training process.
- Scenario: In a system made to detect financial fraud, an adversary may alter transaction records and details to look legitimate, with the intentions that the system misclassifies the fraudulent transactions as normal. [30]

Clean-Label Attacks

- In these types of attacks, the adversary injects malicious data points that come across as benign or correctly labelled, which can make detection of such a type incredibly challenging.
- Scenario: A high-risk example would be an attacker adding noise-patterned pictures of traffic signs to training datasets, with the correct sign labels to interpret from. Therefore, causing the model to misclassify a sign with noise as a sign it is not, leading to potential lethal outcomes. [31]

Targeted Poisoning Attacks

- Targeted poisoning attacks occur when the attacker's goal is to upkeep the overall performance of other data, while misclassifying certain inputs.
- Scenario: The training data is altered and poisoned, so that only specific faces are misclassified as the wrong person, while others are not. [32]

Availability Poisoning Attacks

- The objective for this attack is to degrade the performance of the system and model, so that it left untrustworthy, unreliable, and unusable.
- Scenario: Spam emails may bypass the detection, if a large amount of mislabelled data points is injected, leading to the filter being rendered ineffective. [33]

Federated Learning Poisoning

- When a model is trained in collaboration with multiple clients, this is known as federated learning. An adversary may poison the data locally, so that it is therefore influencing the global model.
- Scenario: Local data could be poisoned so that the global model is made to be biased, and generates certain phrases, especially by a malicious adversary in a federated learning setup for a model for predictive text. [34]

More attacks:

- Collusion Attacks
- Targeted Attacks with Poison Capsule
- Triggerless Attacks
- Error-Amplification Attacks
- Availability Attacks

If there is any hope to build robust defences or maintain the integrity of our systems, it is of the utmost importance to understand the diverse types of possible attacks.

5.7.3 Machine Learning Models

Support Vector Machines (SVMs) [27]

Why are these models targeted? Support Vector Machines are targeted due to their use and how intelligible they are in the preliminary stages of ML research. SVM models held primary attention for understanding data poisoning attacks.

What are the mechanics of the attack? Usually, the attacker modifies the training data points with the aim to alter the decision limits of the SVM, which would lead to classifications being incorrect. Scenario: Similar to the case above, in a spam detection model that uses SVMs, the attacker could inject spam emails and confuse the training data by titling them as "not spam," altering the decision boundary, and allowing the spam mail to go through undetected.

Federated Learning Models [35]

Why are these models targeted? Federated Learning Models are targeted due to the federated learning aggregates updates from devices that are decentralized, which proves to be more difficult to detect poisoned contributions, allowing attackers to be able to inject adversarial updates and be more susceptible to data poisoning.

What are the mechanics of the attack? The attacker or malicious participant could choose to skew the global model by tampering with the updates of the local model.

Scenario: As aforementioned, in a federated text system, a user with malevolent intentions could potentially bias the federal learning model to output certain data, creating the spread of misinformation.

Deep Neural Networks (DNNs) [28]

Why are these models targeted? DNNs are target because they are considered "high value" targets. This is due to their power applications such as language processing, image recognition, and autonomous systems.

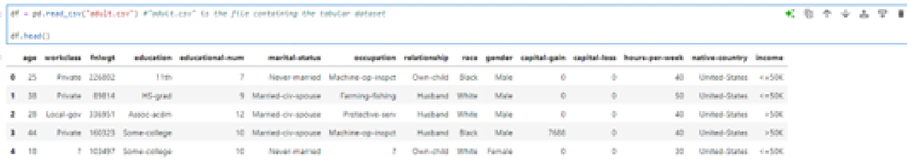
What are the mechanics of the attack? The problem with models that operate on large datasets, such as DNNs, is that attackers have the opportunity to manipulate a small size of the data, and this will lead to the learning process being affected immensely.

Scenario: An example of this is adding adversarial images that consist of undetected noise to the data training set, and thus, causing consequential misclassification when made to create an output.

6 Investigation

6.1 Label-flipping Attack

This is an example of a label-flipping attack I simulated for another module. I thought it appropriate to reference it and use it as a small-scale example of data poisoning. I first started off by downloading a tabular dataset for individuals annual incomes, than have many different variables, from Kaggle. [36] I started this attack off by first reading in the "adult.csv" dataset, and processing this.



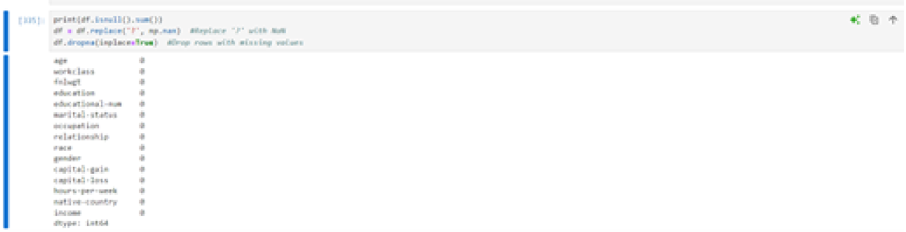
```
df = pd.read_csv("adult.csv") #"adult.csv" is the file containing the tabular dataset

df.head()
```

	age	workclass	fnlwgt	education	educational-num	marital-status	occupation	relationship	race	gender	capital-gain	capital-loss	hours-per-week	native-country	income
0	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black	Male	0	0	40	United-States	<=50K
1	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White	Male	0	0	50	United-States	<=50K
2	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White	Male	0	0	40	United-States	>50K
3	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	7688	0	40	United-States	>50K
4	18	?	103497	Some-college	10	Never-married	?	Own-child	White	Female	0	0	30	United-States	<=50K

Figure 5: Loading the Dataset

I then checked for and inconsistencies and missing values within the dataset so that they could be fixed. [37] As you can see in figure two below, there are no missing values and inconsistencies. [38]



```
[335]: print(df.isnull().sum())
       df = df.replace('?', np.nan)  #Replace '?' with NaN
       df.dropna(inplace=True)  #Drop rows with missing values

age                0
workclass          0
fnlwgt             0
education          0
educational-num    0
marital-status     0
occupation         0
relationship       0
race               0
gender             0
capital-gain       0
capital-loss       0
hours-per-week     0
native-country     0
income             0
dtype: int64
```

Figure 6: Checking for missing values or inconsistencies

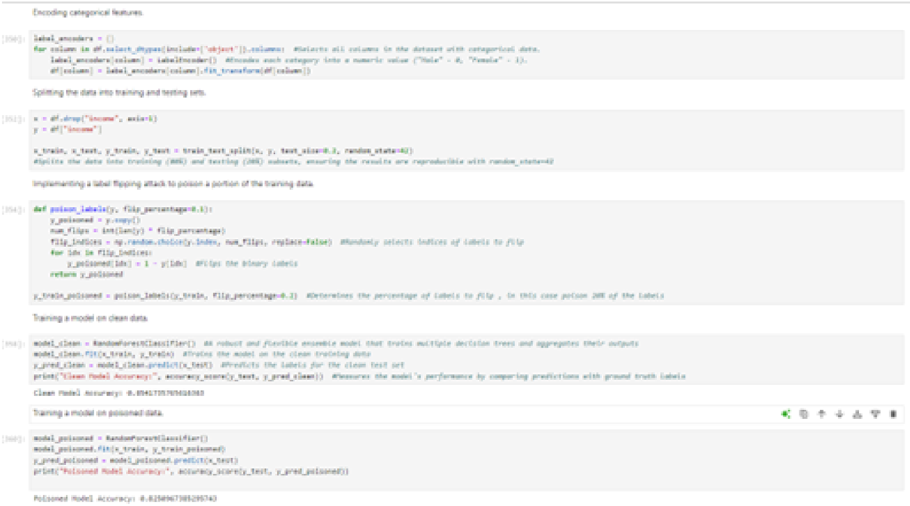
In figure 3, I am encoding categorical features [39] , by converting the categorical variables within the dataset into numerical values as ML models cannot directly work with non-numerical data.

Next, I split the dataset into two sets: training and test, for the model evaluation. In this scenario, x is the independent variable - features, and y is dependent - income.

Next, I am implementing a simulation of a label-flipping attack, by flipping the percentage of the training dataset's labels. The result of this is that it will make a poisoned version of the training [40] labs, "y_train_poisoned," where 20 percent of the dataset labels have been flipped. [27]

Once the labels have been flipped, I trained the model on clean data, so that I could evaluate a Random Forest model using the clean dataset. Underneath this cell is the result output, which is 0.8541, this shows the accuracy of the model. [41]

Below this, I do the same thing, except this time I train a model using poisoned data instead. This is similar to the clean data model above but used "y_train_poisoned" instead of "y_train." The labels that have been poisoned introduced incorrect information into the models training process. The accuracy output is measured using the set of the clean test to determine how much the label-flipping attack has lowered the performance. The poisoned model accuracy is 0.8251, which shows the accuracy of the poisoned model, and shows a slight drop in the performance due to this attack.



Encoding categorical features.

```
label_encoders = {}
for column in df.select_dtypes(include=['object']).columns:  #Selects all columns in the dataset with categorical data.
    label_encoders[column] = LabelEncoder()  #Encodes each category into a numeric value ("Male" - 0, "Female" - 1).
    df[column] = label_encoders[column].fit_transform(df[column])
```

Splitting the data into training and testing sets.

```
x = df.drop("income", axis=1)
y = df["income"]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
#Splits the data into training (80%) and testing (20%) subsets, ensuring the results are reproducible with random_state=42
```

Implementing a label flipping attack to poison a portion of the training data.

```
def poison_labels(y, flip_percentage=0.1):
    y_poisoned = y.copy()
    num_flips = int(len(y) * flip_percentage)
    flip_indices = np.random.choice(y.index, num_flips, replace=False)  #Randomly selects indices of labels to flip
    for idx in flip_indices:
        y_poisoned[idx] = 1 - y[idx]  #Flips the binary labels
    return y_poisoned

y_train_poisoned = poison_labels(y_train, flip_percentage=0.2)  #Determines the percentage of labels to flip , in this case poison 20% of the labels
```

Training a model on clean data.

```
model_clean = RandomForestClassifier()  #A robust and flexible ensemble model that trains multiple decision trees and aggregates their outputs
model_clean.fit(x_train, y_train)  #Trains the model on the clean training data
y_pred_clean = model_clean.predict(x_test)  #Predicts the labels for the clean test set
print("Clean Model Accuracy:", accuracy_score(y_test, y_pred_clean))  #Measures the model's performance by comparing predictions with ground truth labels
```

```
Clean Model Accuracy: 0.8541735765616383
```

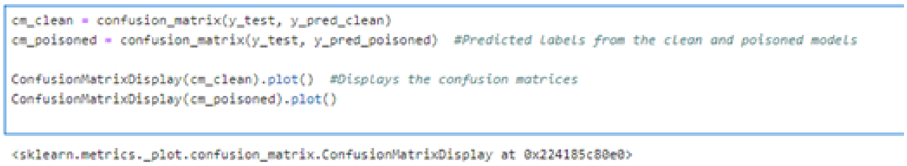
Training a model on poisoned data.

```
model_poisoned = RandomForestClassifier()
model_poisoned.fit(x_train, y_train_poisoned)
y_pred_poisoned = model_poisoned.predict(x_test)
print("Poisoned Model Accuracy:", accuracy_score(y_test, y_pred_poisoned))
```

```
Poisoned Model Accuracy: 0.8250967385295743
```

Figure 7: Label-Flipping steps, with coded explanations

In figure 4, the confusion matrices are being calculated. The matrices are being computed for the clean (cm_clean) and poisoned (cm_poisoned) models.



```
cm_clean = confusion_matrix(y_test, y_pred_clean)
cm_poisoned = confusion_matrix(y_test, y_pred_poisoned)  #Predicted labels from the clean and poisoned models

ConfusionMatrixDisplay(cm_clean).plot()  #Displays the confusion matrices
ConfusionMatrixDisplay(cm_poisoned).plot()
```

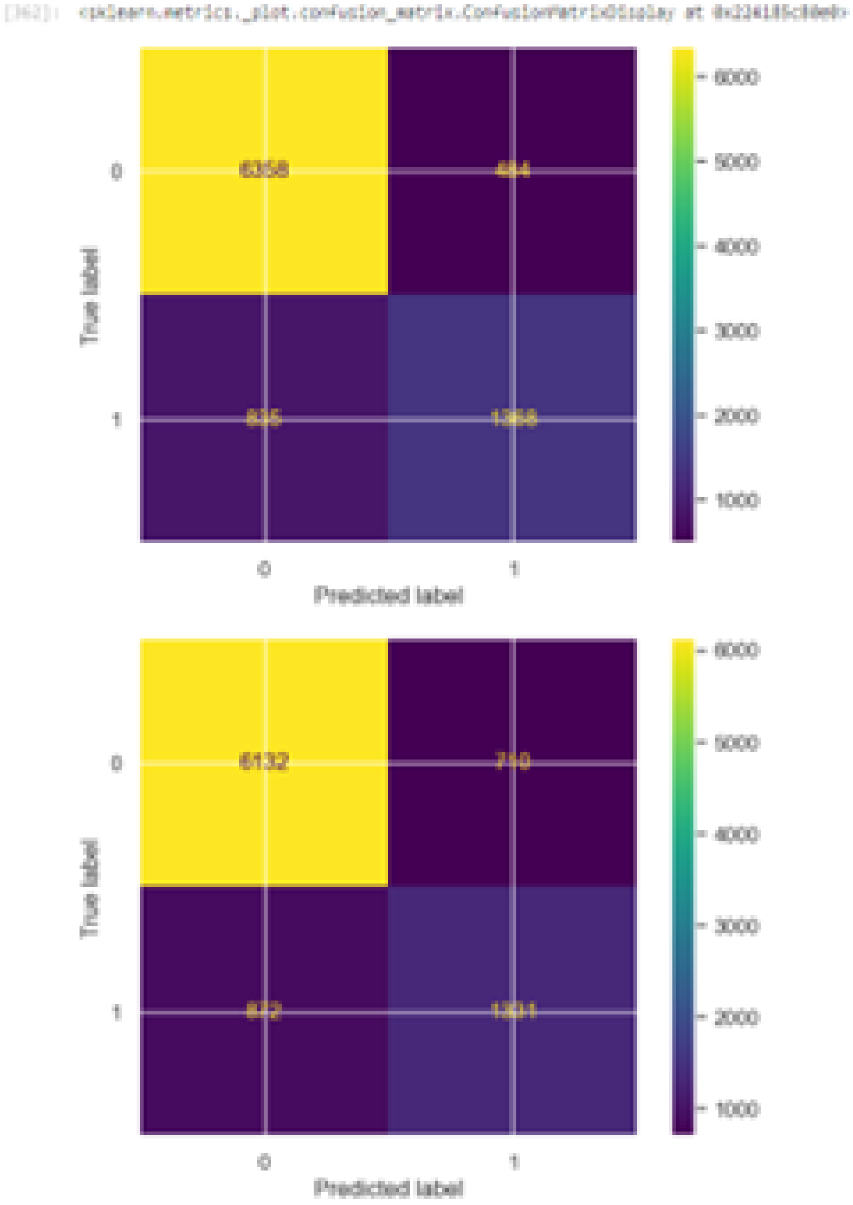
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x224185c80e0>
```

Figure 8: Comparing metrics between the clean and poisoned datasets. In the plots in figure 5, it can be seen that there are counts for the true positive and negative, as well as the false positive and negative

In the plots in figure 5, it can be seen that there are counts for the true positive and negative, as well as the false positive and negative. [42]

In the first confusion matrix for the clean model, there is powerful performance demonstrated, with quite low misclassification rates. The matrix correctly identifies 0 (low income) for 6,358 samples (True Negatives) and predicts correctly 1 (high income) for 1,368 samples (True Positives). This model creates 484 False Positive errors, where it incorrectly predicts 1 instead of 0. There are 835 False Negative errors, where it predicts 0 instead of 1. Therefore, the model has effectively classified most of the samples, which demonstrates reliable and robust accuracy.

In comparison to this is the second confusion matrix, the poisoned model. This displays a decline in the performance, when contrasted to the clean model, due to the increase in misclassification rates. It correctly predicts 0 (True Negatives) for 6,132 samples, and 1 (True Positive) for 1,331 samples. Both are lower than the clean model. However, the False Positives, where 1 is predicted instead of 0, increased to 710; and the False Negatives—predicting 0 instead of 1, increases to 872. The increases in error are indicative of the label-flipping attack negatively effecting the model's ability to classify samples correctly. Thus, introducing a bias towards misclassification.



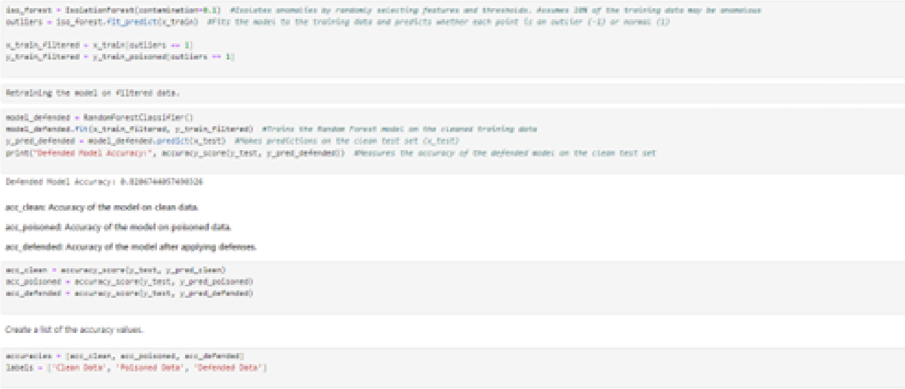
```
[362]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x224185c38e0>
```

Figure 9: Resulted Plots of the different metrics like accuracy, precision, recall, and confusion matrices

In figure 6 below, I have implemented defences, by applying outlier detection to filter any suspicious, and potentially poisoned, data points in the training set. This works through the use of a ML algorithm isolating the anomalies, by randomly choosing thresholds and features. It assumes that there are a few of the anomalies—in this case 10 percent, and that they are different from most of the data. [43]

The dataset is then filtered, through the creation of a filtered training set: "x_train_filtered." This works by keeping data points that are classified as normal (1) and discarding any regarded as suspected anomalies (-1).

Next, I retrained the model using filtered data. The accuracy of the defended model was 0.8207, which indicates it has recovered some of the performance compared to the poisoned model.



```
iso_forest = IsolationForest(contamination=0.1)  #Isolates anomalies by randomly selecting features and thresholds. Assumes 10% of the training data may be anomalous
outliers = iso_forest.fit_predict(x_train)  #Fits the model to the training data and predicts whether each point is an outlier (-1) or normal (1)

x_train_filtered = x_train[outliers == 1]
y_train_filtered = y_train_poisoned[outliers == 1]
```

Retraining the model on filtered data.

```
model_defended = RandomForestClassifier()
model_defended.fit(x_train_filtered, y_train_filtered)  #Trains the Random Forest model on the cleaned training data
y_pred_defended = model_defended.predict(x_test)  #Makes predictions on the clean test set (x_test)
print("Defended Model Accuracy:", accuracy_score(y_test, y_pred_defended))  #Measures the accuracy of the defended model on the clean test set
```

```
Defended Model Accuracy: 0.8206744057490326
```

acc_clean: Accuracy of the model on clean data.

acc_poisoned: Accuracy of the model on poisoned data.

acc_defended: Accuracy of the model after applying defenses.

```
acc_clean = accuracy_score(y_test, y_pred_clean)
acc_poisoned = accuracy_score(y_test, y_pred_poisoned)
acc_defended = accuracy_score(y_test, y_pred_defended)
```

Create a list of the accuracy values.

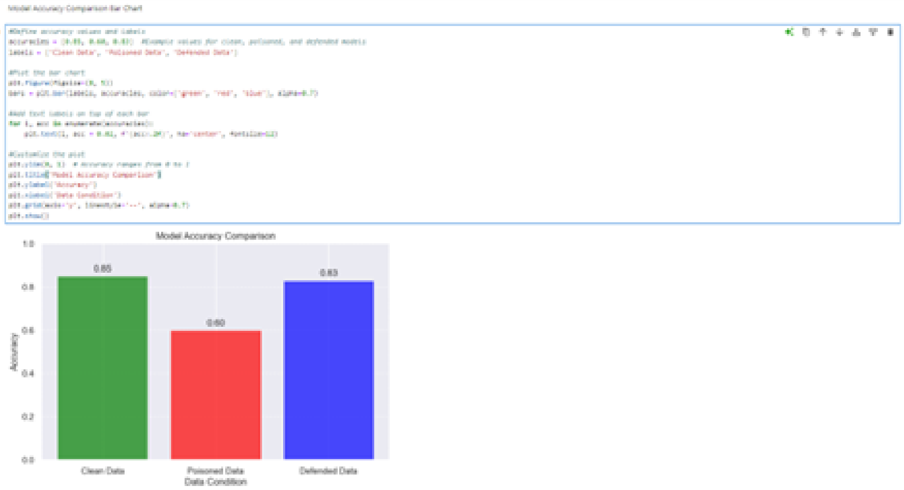
```
accuracies = [acc_clean, acc_poisoned, acc_defended]
labels = ['Clean Data', 'Poisoned Data', 'Defended Data']
```

Figure 10: Implementing defence mechanism against the label-flipping attack using outlier detection, and more

In figure 7, is a bar chart comparison of the three models, highlighting the impact of data poisoning on the model's performance, as well as evaluating the effectiveness of the defences implemented to mitigate the attack.

The results are:

- 0.85 for the accuracy of the model trained on the clean data. This achieves the highest accuracy score, being the baseline for contrast.
- 0.60 for the accuracy of the model trained on the poisoned data, (label-flipping attack). The accuracy drops to 60 percent, indicating that data poisoning has a significant impact on the reliability of the model.
- 0.83 for the accuracy of the model trained on the data after applying the defences, (Outlier detection using Isolation Forest). The accuracy then improves to 83 percent, showing that the defence mitigated much of the attacks impact, however, did not fully restore to the level of the clean data accuracy.



Model Accuracy Comparison Bar Chart

```
#Define accuracy values and labels
accuracies = [0.85, 0.60, 0.83]  #Example values for clean, poisoned, and defended models
labels = ['Clean Data', 'Poisoned Data', 'Defended Data']

#Plot the bar chart
plt.figure(figsize=(8, 5))
bars = plt.bar(labels, accuracies, color=['green', 'red', 'blue'], alpha=0.7)

#Add text labels on top of each bar
for i, acc in enumerate(accuracies):
    plt.text(i, acc + 0.01, f'{acc:.2f}', ha='center', fontsize=12)

#Customize the plot
plt.ylim(0, 1)  # Accuracy ranges from 0 to 1
plt.title('Model Accuracy Comparison')
plt.ylabel('Accuracy')
plt.xlabel('Data Condition')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Figure 11: Model Accuracy Comparison Bar Chart

7 Future Development

The future development for this research paper will be to focus on the domain of AML, the opportunities that AML models have, with further analysis on the exploitation and potential areas of advancement. The primary direction for the second deliverable of this report will include:

- Exploring the sophistication of adversarial attacks, and potential strategies and mechanisms used to mitigate them.
- Investigating the real-world applications of AML, as well as the consequences of malicious adversaries.
- A wider scope of research seeking a broader array of attack taxonomies.
- The direction and steps for future protection of AML models and AI systems, and implementations for provenance security as well as maintaining integrity.
- Further analysis on adversarial attacks across ML models.

References

[1] Donald E. Knuth. *The TEX Book*. Addison-Wesley Professional, 1986.

[2] Frank Mittelbach, Michel Gossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LATEX Companion*. Addison-Wesley Professional, 2 edition, 2004.

[3] Leslie Lamport. *LATEX: a Document Preparation System*. Addison Wesley, Massachusetts, 2 edition, 1994.

[4] Donald E. Knuth. Literate programming. *The Computer Journal*, 27(2):97–111, 1984.

[5] Michael Lesk and Brian Kernighan. Computer typesetting of technical journals on UNIX. In *Proceedings of American Federation of Information Processing Societies: 1977 National Computer Conference*, pages 879–888, Dallas, Texas, 1977.

[6] Jagsir Singh and Jaswinder Singh. Adversarial machine learning. *Science Direct*, 2021.

[7] Daniel Lowd and Christopher Meek. Adversarial learning. *ACM Digital Library*, 2005.

[8] Ian Goodfellow. Adversarial examples and adversarial training. *Stanford University*, 2017.

[9] Rahul Holla. Adversarial machine learning: Techniques and defenses. *Medium*, 2024.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Cornell University*, 2014.

[11] Unknown Authors. Adversarial machine learning. *Wikipedia*.

[12] Staff. What is adversarial machine learning? *Coursera*, 2024.

[13] Matt Duffin. 7 types of adversarial machine learning attacks. *rareconnections*, 2024.

[14] Franziska Boenisch, Philip Sperl, and Konstantin Böttinger. Gradient masking and the underestimated robustness threats of differential privacy in deep learning. *Cornell University*, 2021.

[15] Manpreet Dash. Understanding types of ai attacks. *ai-infrastructure*, 2023.

[16] Micah Musser, Andrew Lohn, James X. Dempsey, Jonathan Spring, Ram Shankar Siva Kumar, Brenda Leong, Christina Liaghati, Cindy Martinez, Crystal D. Grant, Daniel Rohrer, Heather Frase, Jonathan Elliott, John Bansemer, Mikel Rodriguez, Mitt Regan, Rumman Chowdhury, and Stefan Hermanek. Adversarial machine learning and cybersecurity. *Cornell University*, 2023.

[17] Jasmita Malik, Raja Muthalagu, and Pranav M. Pawar. A systematic review of adversarial machine learning attacks, defensive controls, and technologies. *ieeexplore*, 2024.

[18] Masike Malatji and Alaa Tolah. Artifcial intelligence (ai) cybersecurity dimensions: a comprehensive framework for understanding adversarial and offensive ai. *Springer Nature Link*, 2024.

[19] Lance B. Eliot. Legal and ethical ai in adversarial exemplar attacks of machine learning. *Stanford Law School*, 2022.

[20] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I. P. Rubinstein, and J. D. Tygar. Adversarial machine learning. *Berkeley*, 2011.

[21] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of. *Cornell University*, 2018.

[22] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *Cornell University*, 2015.

[23] Yulong Wang, Tong Sun, Shenghong Li, Xin Yuan, Wei Ni, Ekram Hossain, and H. Vincent Poor. Adversarial attacks and defenses in machine learning-powered networks: A contemporary survey. *Cornell University*, 2023.

[24] Data poisoning. *Nightfall AI.*

[25] Bart Lenaerts-Bergman. Data poisoning: The exploitation of generative ai. *CrowdStrike*, 2024.

[26] Neil Lawrence. Proceedings of machine learning research. *Proceedings of Machine Learning Research.*

[27] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *ACM Digital Library*, 2012.

[28] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. *Cornell University*, 2018.

[29] Tianyu Gu, Brendan Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *Semantic Scholar*, 2017.

[30] Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector. *Chair of IT Security.*

[31] Ali Shafahi, W. R. Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and T. Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Semantic Scholar*, 2018.

[32] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *Proceedings of Machine Learning Research*, 2017.

[33] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. *Cornell University*, 2017.

[34] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *Proceedings of Machine Learning Research*, 2020.

[35] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against. *Cornell University*, 2020.

[36] Adult income dataset. *Kaggle*.

[37] pandas.dataframe.replace. *pandas*.

[38] pandas.dataframe.dropna. *pandas*.

[39] Labelencoder. *skikit learn*.

[40] numpy.random.choice. *NumPy*.

[41] Randomforestclassifier. *skikit learn*.

[42] Confusion matrix. *scikit learn*.

[43] Isolationforest. *scikit learn*.