A Behaviour-Informed Approach to Mitigate Data Poisoning Attacks for Machine Learning

BIRMINGHAM CITY University

By

Anum Paracha

Supervised by:

Prof. Junaid Arshad, Dr Mohamed Ben Farah and Dr Khalid Ismail

A thesis submitted in partial fulfillment of the requirements of Birmingham City University for the degree of Doctor of Philosophy

Faculty of Computing, Engineering and Build Environment

October 2025

Abstract

The widespread deployment of machine learning (ML) across critical domains such as healthcare, transport, and smart grids has increased dependence on automated decision-making, exposing these systems to potential exploitation by adversaries. ML models are highly susceptible to adversarial perturbations, including intentional input manipulations that can alter model performance. Among these, data poisoning attacks are particularly evolving as there are enormous ways to corrupt training data to distort underlying behaviour and undermine system reliability. Furthermore, the risks of data poisoning attacks increase with the dependence on public datasets.

Data poisoning attacks have been extensively explored in the context of deep learning (DL) models; however, traditional ML, especially multiclass models, remains underexplored in assessing vulnerabilities and defences. Consequently, most mitigation strategies are limited to DL and are designed for specific algorithms or attack models. For example, adversarial training is effective for gradient-based models but less effective for traditional models as they do not rely on gradient optimisation. These limitations enable adversaries to exploit defences through new attack vectors, thereby complicating the security of ML systems. Moreover, limited defences for traditional ML keep these models vulnerable to such attacks.

This thesis analysed the security of traditional ML under data poisoning attacks implemented with limited adversarial capabilities and knowledge and analysed limitations of existing defences, subsequently introducing an enhanced mitigation strategy. The manipulations to training datasets are analysed through comprehensive deep behavioural analysis, identifying the change in model characteristics, the impact of increasing poisoning levels and their relationships. Furthermore, a new multiclass poisoning attack is proposed by exploiting a common outlier characteristic of ML models, called Outlier-Oriented Poisoning (OOP) attack. This attack leveraged the examination of multiclass ML under limited adversarial capabilities. These studies revealed how data poisoning alters the learning dynamics of the model and its characteristics. Insights from this analysis informed the development of SecureLearn, a behaviour-informed, attack-agnostic mitigation solution combining enhanced data sanitisation with a novel feature-oriented adversarial training (FORT) approach to improve model resilience against data poisoning.

This thesis examined SecureLearn by proposing a 3D evaluation matrix. Experimental results of this study demonstrated that SecureLearn effectively enhanced the security and robustness of multiclass ML across random forest (RF), decision tree (DT), gaussian naive bayes (GNB) and neural networks, confirming its generalisability beyond algorithm-specific defences. SecureLearn consistently maintained accuracy above 90%, recall and f1-score above 75%, and reduced the false discovery rate to 0.06 across all evaluated models against three distinct poisoning attacks. For RF models, Secure-

Learn maintained a minimum recall of 84.19% and f1-score of 81.54% at 20% poisoning level with the OOP attack. For DT models, the minimum recall is 78.20% and f1-score is 7.80%. However, it is observed that SecureLearn is less effective in enhancing the resilience of GNB models trained with the MNIST dataset. GNB models trained with the MNIST dataset, SecureLearn maintained the recall at a minimum of 57% with f1-score of 56%. In the context of neural networks, SecureLearn achieved at least 97% recall and f1-score against all selected poisoning attacks. The adversarial robustness of models, trained with SecureLearn, improved with an average accuracy trade-off of only 3%.

Declaration of Authorship

I, Anum Paracha, declare that this thesis titled, "A Behaviour-Informed Approach to Mitigate Poisoning Attacks for Machine Learning" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Anum Paracha

October 2025

Publications

- **Paracha, A.**, Arshad, J., Farah, M. and Ismail, K. "Machine learning security and privacy: a review of threats and countermeasures". EURASIP Journal on Information Security (JINS), 2024.
- **Paracha, A.** and Arshad, J. "A bibliometric study toward quantitative research assessment of security of machine learning". Information Discovery and Delivery (IDD), 2024.
- **Paracha, A.**, Arshad, J., Farah, M. and Ismail, K. "Deep behavioral analysis of machine learning algorithms against data poisoning". International Journal of Information Security, 2025.
- **Paracha, A.**, Arshad, J., Farah, M. and Ismail, K. "Outlier-oriented poisoning attack: a grey-box approach to disturb decision boundaries by perturbing outliers in multiclass learning". International Journal of Information Security, 2025.
- **Paracha, A.**, Arshad, J., Farah, M. and Ismail, K. "Exploring Data Poisoning Attacks Against Adversarially Trained Skin Cancer Diagnostics". 2024 IEEE/ACM 17th International Conference on Utility and Cloud Computing (UCC), 2024.
- **Paracha, A.**, Baiyekusi, O., Arshad, J., Mi, D., Lu, C., Zhang, Y., Wang, F., Chen, L., Zhang, J. "Exploring Multiclass Data Poisoning within an Industrial 5G Private Network". 2025 IEEE 101st Vehicular Technology Conference (VTC2025-Spring), 2025.
- **Paracha, A.**, Arshad, J., Farah, M. and Ismail, K. "SecureLearn - An Attack-agnostic Defense for Multiclass Machine Learning Against Data Poisoning Attacks". Submitted to IEEE Transactions on Information Forensics and Security, 2025.

Acknowledgments

Firstly, I am thankful to the Almighty for the blessings and ability to complete this PhD study.

Secondly, I would like to thank my Director of Studies, Prof. Junaid Arshad, for his endless support, invaluable discussions and guidance provided during this study. I am also greatly thankful to my second supervisors, Dr Mohamed Ben Farah and Dr Khalid Ismail, for their continuous support, discussions and guidance throughout this journey.

I would like to thank all fellows, friends and colleagues for the encouragement and every help provided during this time.

I am greatly thankful to my family, especially to my dad, Mr. Atique Ahmed Paracha and mum, Mrs. Amna Sadia Paracha, for all their love, endless support and prayers during this complete journey. Without their support, I would not be able to pursue this degree.

Special thanks to the Department of Computing, Engineering and Built Environment and Doctoral Research College (DRC) for their support and guidance.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

The challenge of extracting knowledge and patterns from the data is older than the developed field of machine learning (ML) (Rosenbloom et al., 1985). However, with the evolution of time, the complexities of the data have increased with more dimensions and variability. In response to these complexities, researchers have developed algorithms to autonomously identify intricate patterns and correlations within data rather than relying only on human brains and explicit programming. Further improvements to these autonomous algorithms allow digital systems to learn independently for decision making and have developed a field called ML (Cohen, 2025).

Such advances in automating digital systems are especially relevant in fields requiring high accuracy and fairness, such as medical diagnostic applications (Murugan et al., 2021), autonomous vehicle systems (Sasmono et al., 2021), cybersecurity solutions (Chalé and Bastian, 2022), and financial systems (Qiao and Beling, 2016), making ML algorithms a more suitable solution. It is a powerful tool to analyse complex data and enable businesses and researchers to make accurate predictions, leading to innovative solutions and increased efficiency by identifying hidden patterns in the data. Its growing importance stems from the availability of computational resources, large datasets, and efficient algorithms.

The fundamental ML algorithms are known as *traditional ML*, which includes the baseline algorithms, such as support vector machines (SVM), random forest (RF) and neural networks (NN) as perceptron. However, the increasing complexities of the problems and further advancements in ML have developed modern ML called deep learning (DL), which typically extends NN to advanced algorithms such as convolutional neural networks (CNN) and recurrent neural networks (RNN) (Wang et al., 2021b).

Irrespective of the algorithms, the ML model can be developed in two settings: *binary* and *multiclass* models. When the application problem is defined across two groups, the model is developed in binary settings. Conversely, when the prediction problem is defined across more than two groups, the model is developed in multiclass settings. Multiclass models are useful in various applications, such as healthcare diagnostics and autonomous driving. However, they are associated with certain challenges that need to be addressed for successful application development. One of the major challenges in this regard is the equal distribution of data within different groups in the given dataset, which enables the model to learn each group equally, where an unequal distribution can introduce unfairness in their predictions (Del Moral et al., 2022). These complexities should be addressed to develop a robust and fair ML model.

1.1 Machine Learning Security

The prevalent use of ML in almost every digital application makes it an attractive target for adversaries to attack these models and harm the integrated systems. Attacks on the security of ML are called *adversarial attacks*. Unfortunately, the literature has highlighted many adversarial ways of attacking the model security at different development phases and breaching ML systems (Hu and Hu, 2020). Adversaries can harm it in various aspects, including exploiting privacy (Shafee and Awaad, 2021), security (Goldblum et al., 2020), robustness (Dunn et al., 2020), fairness (Xue et al., 2023), and accountability (Ghosh et al., 2022) of the model. The study of these attacks and countermeasures against these attacks is called *adversarial machine learning (AML)*.

Adversarial attacks that manipulate training data of the ML model development are known as *data poisoning attacks*. With the increasing complexities of the autonomous systems, the need for large training data is becoming a significant challenge, and one way to address this challenge is to use publicly available datasets (Zhou et al., 2017). However, reliance on public datasets increases the risk of adversarial manipulations, which can compromise the security of the model. One common form of data poisoning is the label-flipping attack (Yerlikaya and Şerif Bahtiyar, 2022), (Paudice et al., 2018b), where the labels of randomly selected data points are altered to degrade model integrity. For instance, in an intrusion detection system (IDS) trained on the public CIC-IDS-2017 dataset, an adversary may change the labels of 10 percent of the data points, causing the model to misclassify certain intrusive instances as normal. Additionally, a substantial number of mislabeled data points can induce overfitting, resulting in a breach of availability. Another significant threat is the breach of confidentiality. During model training, an adversary may introduce targeted data points that the model memorises, enabling the extraction of confidential information during inference. These considerations underscore the necessity to enhance ML security to ensure resilience against data poisoning attacks. The details of these attacks and their types are discussed in Section 2.2 in Chapter 2. Irrespective of attack types and approaches that adversaries applied to ML model development, these attacks mainly harm model performance in one of the five aspects highlighted in Fig. 1.1.

The first aspect is *privacy*, where the adversary tries to extract confidential information about the training data or learned parameters from the trained model (Shi et al., 2020). Privacy attacks can be mitigated with privacy preservation (Truex et al., 2019) and differential privacy (Zhang et al., 2020) solutions. However, these solutions imply various performance tradeoffs, such as adding noise to the training samples, which may decrease the model's performance. Conversely, various adversarial attacks successfully exploited these mitigations (Paracha et al., 2024b). This is an active research area in the domain of AML, where researchers are developing solutions, toolkits, and processes to enhance the privacy preservation of ML while preserving their performance.

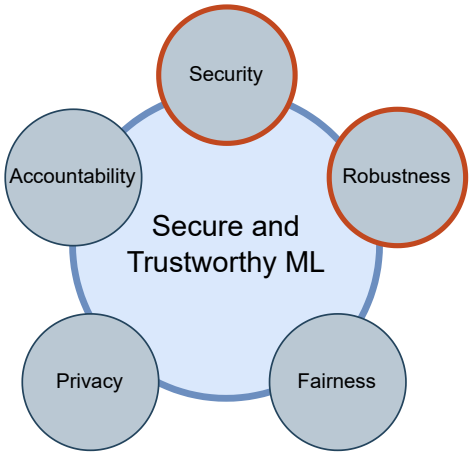


Figure 1.1: Fundamental aspects of secure and trustworthy machine learning

The next important aspects are the *security* and *robustness* of ML (Tian et al., 2024), where the adversary can manipulate the training or testing datasets to undermine the decision mechanism of the model. For example, the adversary can manipulate the dataset labels (Paracha et al., 2024a), which may confuse the model to misinterpret the dataset and reduce the expected performance. Other than that, manipulated data points can be injected to disturb the data distributions (Jagielski et al., 2021) or some important dataset features can be perturbed, which can shift the decision boundaries (Paracha et al., 2025a) of the model and ultimately harm the confidence. Defending against these attacks is a complex challenge, since distinguishing adversarial perturbations from legitimate variations is inherently difficult.

Accountability and *fairness* can also be exploited with adversarial attacks (Mehrabi et al., 2021). For example, autonomous medical diagnostic applications should be fair in their decisions and not discriminate against specific responses or data groups. The adversarial attack can change the data distribution, making it imbalanced between different subsets or groups. This biased data can train biased and unfair models, which can reflect in their predictions. Various studies highlighted that adversarial attacks have successfully exploited the accountability of ML models (Farinu, 2025).

Of these aspects, this thesis focuses on the security and robustness of ML models against data poisoning attacks. The security of ML aims to protect the model from malicious activities and external inference that compromises its confidentiality, integrity and availability (Sagar and Keke, 2021). The robustness of ML aims to overcome the adversarial noise or errors generated in the dataset, such as misplaced features or missing feature values, and how they can impact the statistical computation of the model performance (Yu et al., 2025). Numerous studies highlighted successful poisoning attacks on secured ML models (Lu et al., 2024), (Das et al., 2024) that breached the security or robustness

of ML models. Achieving adversarial security and robustness is a significant challenge. To achieve security and robustness from adversarial attacks, there is a pressing need to understand adversarial perturbations and distinguish them from normal data. It is formulated with similar objectives to model generalisation across a given domain, which can be achieved by ignoring irrelevant features. For example, receiver email in spam detection classifiers or source ports in IDS. Thus, the aim is to generalise over the underrepresented classes or invariant data. Developing strong resilience is still a challenge as the attack surface is evolving. There is a need to think about the next level of understanding algorithm architectures and training processes, rather than only generalising over the domains.

To achieve the aim of secure and robust ML, this thesis focuses on classification models, namely support vector machines (SVM), random forest (RF), decision tree (DT), gaussian naive bayes (GNB), k-nearest neighbors (KNN) and neural networks with perceptron and multilayer perceptron (MLP). These models are developed under binary and multiclass settings. Furthermore, the scope of the thesis is defined for the grey-box data poisoning attacks, following the assumptions that training datasets can be publicly accessible and an adversary can manipulate datasets and develop surrogate models; however, no model settings are known to the adversary.

1.2 Problem Background

The existing literature has predominantly studied data poisoning attacks and their mitigations in deep learning (DL) models; however, few studies have explored data poisoning in traditional ML, particularly in multiclass models. Traditional multiclass ML is significantly useful for developing multimodal applications with predefined feature datasets and limited computational resources, such as intrusion detection systems (IDS), healthcare diagnostics, and cyber threat intelligence applications. Therefore, the analysis and strengthening of the security of such models is important to defend integrated systems. Studying multiclass ML alongside DL and binary models provides an in-depth understanding of ML security against data poisoning attacks.

Furthermore, existing studies have explored data poisoning impact, considering performance metrics including accuracy, precision, recall and f1-score. However, it is also important to analyse the disruption in the underlying decision mechanism of the model when trained with a poisoned dataset. Understanding these changes helps develop strengthened and attack-agnostic mitigations against such attacks, which may defend ML from various data poisoning attacks.

1.3 Problem Statement

Data poisoning attacks manipulate the training dataset of ML models, leading to degraded model performance (Chillara et al., 2024) or enabling backdoors (Saha et al., 2020). These manipulations

can be achieved by perturbing dataset labels, poisoning dataset features, or injecting poisoned data points into the dataset. For example, metapoison (Huang et al., 2020), class-oriented poisoning attack (Zhao and Lao, 2022b), and clean-label poisoning attack (Zhang et al., 2023) successfully poisoned ML systems. Though various mitigation solutions are provided in the literature to secure ML, new attacks have successfully exploited such mitigations Paracha et al. (2024b), Koh et al. (2021), Kuppa and Le-Khac (2020). These successful exploitations highlight the complex problem of securing ML models from data poisoning attacks.

The core challenge behind this problem is to analyse vulnerabilities in ML models. The prevalence of ML models opens doors for adversaries to manipulate them in various ways, which makes it difficult to determine to what extent the outcomes of ML should be trusted. Various research studies proposed data poisoning against deep neural networks, specifically in the computer vision domain Ahmed et al. (2022), Salama et al. (2023), Wei et al. (2022), Su et al. (2019); however, traditional models are explored to a limited extent, particularly multiclass models, which are overlooked in the literature. Moreover, existing analysis techniques do not completely identify these underlying vulnerabilities. Existing studies assess data poisoning attacks from the performance aspects of the models, which do not answer the underlying changes in their training and decisions. Studying the underlying changes against poisoned training provides an understanding of model vulnerabilities and identifies relationships between model characteristics and the impact of data poisoning attacks. These understandings help develop strengthened defences against such attacks. Therefore, there is an urgent need to develop enhanced analysis processes for the detailed ML examination in various settings. These analyses should focus on analysing the behavioural changes in the ML model and their impact on the performance of the model.

Additionally, most data poisoning attacks are developed under the assumption of full adversarial capabilities and complete knowledge of the target system Zhang et al. (2023), Zhao and Lao (2022b), Zhu et al. (2023b). Evaluating the effectiveness of these attacks under limited adversarial knowledge and capabilities is essential for assessing their practicality against real-world systems, which are often not fully accessible to adversaries. The real-world impact of attacks under such constraints is still a complex challenge.

Following the challenges mentioned above, the unified problem is to develop attack-agnostic and security-by-design solutions that secure ML not only from specifically identified poison but from evolving threats in this domain. Current solutions are often tailored to specific attacks or algorithms Xu et al. (2021), You et al. (2019), Ma et al. (2021b), which limits their applicability to novel poisoning methods. For example, adversarial training (Tu et al., 2021) is designed for neural networks that utilise gradient learning and do not generalise to traditional ML models. Therefore, there is a need to investigate and analyse how poisoned models behave under limited adversarial capabilities. Such

analysis will inform the development of effective mitigation strategies that secure ML models against evolving poisoning attacks across diverse applications.

1.4 Research Aim and Objectives

To address challenges outlined in the problem statement, this thesis aims to develop a robust behaviour-informed solution to mitigate data poisoning attacks in multiclass ML and improve the generalisability and robustness of the model. The intended objectives to fulfill the aim of this research are given as follows.

- To explore the impact of data poisoning attacks under limited knowledge and adversarial capabilities. This exploratory analysis is conducted against six classification models, in binary and multiclass settings. This study analyses the impact of poisoning and investigates the limitations of existing mitigations against these attacks.
- To develop an improved defence solution to enhance the security of the multiclass ML against data poisoning attacks. The proposed solution is an attack-agnostic mitigation that secures the model by understanding underlying behaviours and decision mechanisms, irrespective of attack vectors. .
- To develop the evaluation mechanism to assess the effectiveness of the proposed security solution for ML. The proposed mechanism evaluates the mitigation solution by comparing its effectiveness against existing solutions and data poisoning attacks for various ML models.

1.5 Research Questions

Following the above research objectives, this study considers several factors to understand the security of ML by answering the following research questions (RQ).

- What are the security vulnerabilities in ML algorithms that are exploited by adversarial attacks and their impact on real-world applications? It is important to understand the realistic threats to ML models. A model being attacked likely provides strong incentives to the adversary. However, it is crucial to understand what capabilities of the attacker are required to devise an attack when limited knowledge and access to the model are provided.
- What are the potential risks, impact and consequences of data poisoning attacks? Poisoning impact varies based on the domain of applications and adversarial capabilities. Following question 1, there is a pressing need to understand how successful and impactful the attack is under limited adversarial capabilities for cybersecurity and healthcare applications. Suppose the targeted application is developed for the learning of primary school students. In that case, the

impact of poisoning is considerably lower. However, if the application is developed for cancer diagnosis, it may result in loss of human life, and thus represent a high impact of poisoning attacks.

- What are the limitations of existing solutions in mitigating data poisoning attacks? The third question addresses the limitations of existing mitigations against data poisoning attacks by analysing them across various attack vectors for traditional models. It also examines the impact of data poisoning on changing the underlying decision mechanism and characteristics of binary and multiclass models and assesses whether existing solutions address these behavioural changes.

1.6 Thesis Contributions

Following the above research questions, the contributions of this thesis are as follows:

- This study conducts a deep behavioural analysis of poisoned binary and multiclass models to analyse the change in their decision capabilities and identify relationships between poisoning impact and individual characteristics of models. These characteristics include the number of support vectors in SVM, the number of trees in RF, and the changing dataset distributions. This analysis also identified a range of poisoning levels that have a high impact on degrading the model's performance. (Paracha et al., 2024a)

- This study analyses the effectiveness of multiclass poisoning attacks under limited adversarial capabilities. To analyse multiclass poisoning, a novel outlier-oriented poisoning (OOP) attack is proposed that disturbs the decision boundaries of the model by perturbing outliers. Following the OOP attack, this analysis has also strengthened the results of the deep behavioural analysis by strengthening its outcomes of the identified poisoning levels and highlighting the impact of various dataset structures on multiclass poisoning. Moreover, the efficacy of the OOP attack is examined against a well-known mitigation solution called adversarial training, highlighting its ineffectiveness in securing traditional multiclass models; however, it has improved the resilience of neural networks. (Paracha et al., 2025a), (Paracha et al., 2024b), (Paracha et al., 2025c)

- This study proposes an attack-agnostic two-layer defence, SecureLearn, to secure multiclass models against data poisoning attacks by understanding their training fundamentals. SecureLearn proposes an improvised data sanitisation with an additional layer of feature-oriented adversarial training (FORT), to secure ML reactively and proactively against poisoning attacks. (Paracha et al., 2025b)

- This study proposes a 3D evaluation matrix to evaluate SecureLearn from three orthogonal dimensions comprehensively. The 3D evaluation matrix assesses the solution against various

data poisoning attacks and compares it with existing mitigation solutions, including multiple data sanitisation and adversarial training mechanisms.

The conceptual alignment of research objectives, research questions and thesis contributions is given in Fig. 1.2.

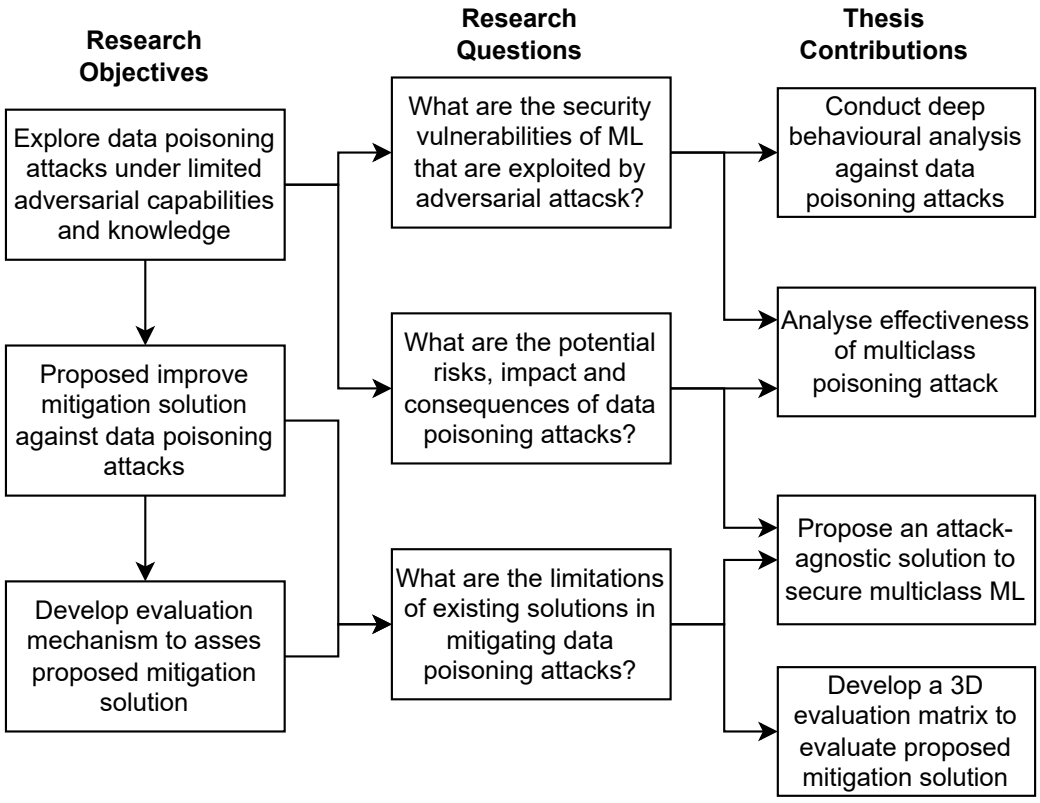


Figure 1.2: The conceptual alignment of research objectives, research questions and thesis contributions

1.7 Thesis Organisation

Following the above-mentioned contributions, the organisation of this thesis is given as follows.

Chapter 2 In this chapter, a detailed background of supervised ML, concepts of adversarial ML and its attack types is provided. Following the background, a comprehensive state-of-the-art analysis is conducted with a quantitative review to understand the threat landscape of adversarial ML, followed by a comprehensive literature review built on the adversarial attack types. This background and literature review form the basis of all the subsequent chapters. This work is discussed in Paracha and Arshad (2024), Paracha et al. (2024c).

Chapter 3 In this chapter, deep behavioural analysis of poisoned models is conducted under limited

adversarial capabilities and knowledge. This analysis studies how poisoned training shifts their decision boundary and how poison impacts individual characteristics of each algorithm. For example, how does the feature importance score change with data poisoning, or what is the relationship between the number of trees in the RF algorithm and increasing poisoning level? This analysis provides a detailed understanding of how data poisoning attacks manipulate the decision mechanism of ML models, providing insights into enhancing defence mechanisms against these attacks. This work is presented in (Paracha et al., 2024a).

Chapter 4 In this chapter, a new data poisoning attack is introduced that manipulates the feature space of multiclass models by perturbing outliers. This attack is implemented with limited adversarial capabilities and knowledge to examine the efficacy of new attacks under limited adversarial capabilities and knowledge. This chapter also examines the effectiveness of existing mitigation solutions to secure traditional multiclass models. This work is published in Paracha et al. (2025a), Paracha et al. (2024b), Paracha et al. (2025c).

Chapter 5 This chapter proposes an attack-agnostic two-layer defence called SecureLearn to secure multiclass models from data poisoning attacks. It presented a new formulation of data sanitisation and FORT adversarial training that enhances the adversarial robustness of classification models. Following that, a 3D evaluation matrix is proposed to analyse the efficacy of SecureLearn and existing similar solutions from three different aspects. This work is submitted to IEEE Transactions on Information Forensics and Security.

Chapter 6 This chapter concludes all the previous chapters, summarises the key achievements and contributions to defence in each chapter.

Chapter 2. Machine Learning Security - Background and State-of-the-Art Attacks and Countermeasures

2.1 Introduction

This chapter introduces the background and key concepts related to supervised ML algorithms, adversarial ML, and adversarial attack types, followed by a state-of-the-art analysis of adversarial attacks and their countermeasures. This chapter addresses RQ 1 by offering a comprehensive analysis of the severity, impact, and consequences of adversarial attacks. It also provides insights to guide future research. The objective 1 of the thesis is fulfilled from a theoretical perspective by examining adversarial attacks and mitigation techniques. The analysis considers adversarial knowledge, goals, and capabilities. It classifies and compares attack types and highlights the limitations of current solutions. The background section describes fundamental characteristics and training mechanisms of supervised ML algorithms. It discusses the formation of decision boundaries in both binary and multiclass contexts. This section also introduces adversarial machine learning and outlines the primary attack types. It connects these attacks to various phases of machine learning development and attack surfaces. The attack surface is defined as a set of points, features, or components that the attacker can exploit and harm the system (Zeng et al., 2019). This section establishes the conceptual foundation for the following chapters.

Next, this chapter presents the state-of-the-art analysis of adversarial attacks and their countermeasures in ML. The analysis examines the literature using a two-step approach based on prior work Paracha et al. (2024c), Paracha and Arshad (2024). First, a bibliometric analysis is conducted to analyse research trends across four dimensions of publication types and languages, publication citations, annual publications and keywords analysis, followed by a qualitative literature review based on a comprehensive methodology. This bibliometric analysis provides insights into the evolution of adversarial ML and evaluates its research impact on various domains.

Following the bibliometric analysis, a qualitative literature review critically evaluates adversarial attacks and their countermeasures to identify research gaps. This review is organised around four types of adversarial attacks: poisoning, evasion, model inversion, and membership inference. These correspond to the training, testing, and deployment phases of model development. The analysis assesses the feasibility of attacks across eight dimensions, such as machine learning algorithm, exploited vulnerability, and attack type. Mitigation strategies are also examined for each attack category.

2.2 Background

In this section, the training process of each selected algorithm is discussed to understand their decision mechanisms. This understanding will later help determine the adversarial attack methods and how these attacks can disturb the ML decision processes and affect their performance.

2.2.1 Machine Learning

Machine learning refers to a mathematical model that learns patterns from the given data to perform tasks autonomously without providing explicit input. The core goal is to approximate the mapping function $f : X \rightarrow Y$ by minimising the loss $\mathcal{L}(f(x), l)$ to predict the correct output l where $l \in Y$. Generally, larger data improves performance, as the model can capture complex patterns (Taherdoost, 2022). ML is categorised into three main types, discussed below.

Supervised learning is a type of ML development where the dataset is labeled with a predefined number of groups (Taherdoost, 2022). This type of learning is useful for task-driven problems where the ground truth is available in the form of dataset labels. The dataset is divided into training and testing instances. The training dataset is used to train the model, which is trained to recognise features and their corresponding labeled groups. Then the test dataset is used to assess the predictions of the trained model by comparing predicted labels with the test dataset labels. Supervised learning is categorised into two types: classification and regression. Supervised classification learning is used for categorical predictions with a predefined number of groups in an application, and regression learning predicts continuous values as its outcomes.

The next ML type is *unsupervised learning*, where the unlabeled dataset is provided for the model development (Balevi and Gitlin, 2017). This learning is aligned with the data-driven tasks for which no fixed number of groups or labels is provided in the dataset. In this type of learning, the model identifies features in the dataset to determine patterns. Clustering is the most common type of unsupervised learning, where the training process identifies unique clusters within the dataset.

The third type of ML development is *reinforcement learning*, also known as agent-based learning (Nguyen and Han, 2023). This type of learning emphasises feedback from humans, following rewards and punishments. In this learning, the model is iteratively trained and assessed to reach a predefined goal by continuing good practices and receiving penalties for incorrect predictions. Agent learning is a common type of reinforcement learning, used for training robots. Of these approaches, this thesis focuses on the security of supervised classification ML against data poisoning attacks, following two settings: binary and multiclass classifications, and it interchangeably uses supervised ML for classification learning throughout the thesis.

2.2.2 Supervised Machine Learning

In classification ML, the problem involves organising the data into distinct categories or outputs. The model learns patterns from the dataset to distinguish specific entities and predict their classes or labels. Consider a training dataset $D = \{(x_i, l_i)\}_{i=1}^{n}$ where x_i is the input feature of the dataset, known as an independent variable, and l_i is the target label, which is the dependent variable.
During training, the model learns patterns from the dataset D and associates them with their labels or classes. Once trained, it is tested on a previously unseen dataset. The model predicts the correct labels, which are later evaluated against the ground truth (Gil-Fournier and Parikka, 2021). A fundamental consideration in model development is that the training and testing sets are assumed to be drawn from the same underlying data distribution, meaning they share similar statistical properties. If the model demonstrates expected results during testing, as measured by performance metrics such as accuracy, recall, and f1-score, it indicates that the model effectively generalises over patterns in the data. Following successful testing, the model can be deployed in real-world applications. The development process of the supervised ML model is illustrated in Fig. 2.1. The supervised classifiers are additionally divided into parametric and non-parametric models based on how they learn their mapping functions.

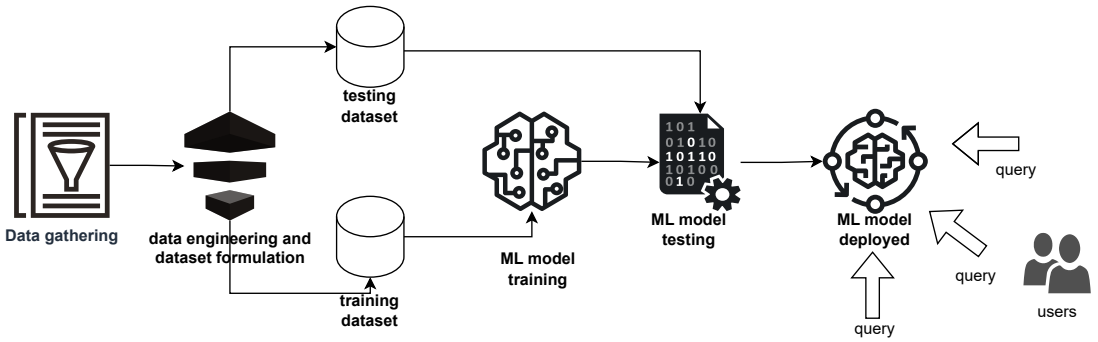


Figure 2.1: The development process of a supervised machine learning model, illustrating all phases of the development, starting from data gathering to model deployment

Parametric models follow a training process that develops their decision boundaries by learning a fixed, pre-defined number of configuration parameters θ that characterise the model's complexity (Guého et al., 2020). This process aims to learn underlying patterns from training examples until the configuration parameters are estimated in a way that reduces classification errors. Therefore, the trained model can classify new data in the relevant class. On the other hand, *non-parametric* models develop their decision mechanisms by capturing the relationship between various dataset features. These models rely on the structure of the training data at inference time rather than summarising it into a fixed number of parameters. These models predict without making strong assumptions about

specific data distribution. Conclusively, parametric models are computationally efficient but may underfit to evolving distributions or complex patterns. Conversely, non-parametric models are highly flexible to capture dynamic patterns within the provided data (Shi and Wang, 2021).
This thesis selected six supervised parametric and non-parametric algorithms, i.e., Support Vector Machines (SVM), Random Forest (RF), Decision Tree (DT), Gaussian Naive Bayes (GNB), K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). Next, the binary and multiclass classification settings of the model development are explained in Sections 2.2.2.1 and 2.2.2.2, followed by the development process for each of the selected algorithms.

2.2.2.1 Binary Classification The simplest classification type in supervised ML is binary classification, where the model categorises instances into two classes. In this type of classification, the labels are typically encoded as 0 (false) or 1 (true). For example, in a spam detection classifier, an instance is either classified as spam or not spam. Similarly, in a skin cancer diagnostics application, a skin lesion is either detected as cancerous (true) or benign (false). Let $D = \{(x_i, l_i)\}_{i=1}^{n}$ be a dataset consisting of two classes where dataset features are denoted with $x \in X$ and labels are denoted as $l \in \{0, 1\}$. The model learns with a mapping function $f : X \to Y$ to separate classes in a given dataset. The trained model uses the learned function to classify new instances in a class. If the predicted probability exceeds 0.5, the instance is assigned to the positive class.

2.2.2.2 Multiclass Classification Multiclass classification assigns each instance to one of m possible classes, where $m > 2$. For instance, in a skin cancer diagnostics application, the model may classify lesions as melanoma, melanocytic nevus, or basal cell carcinoma, rather than simply distinguishing between cancerous and benign cases. Multiclass classifiers accommodate multinomial categorisation by supporting more than two classes as needed. For example, in the same diagnostic context, if a lesion exhibits characteristics that overlap between benign and melanoma, an additional category, such as intermediate melanoma, can be adjusted to enable more nuanced diagnostic decisions and treatment. The classifier is trained on a multiclass dataset $D = \{(x_i, l_i)\}_{i=1}^{n}$, where $x \in X$ and $l \in \{0, 1, 2, ..., m-1\}$. The mapping function f for multiclass classification is defined in Eq. 2.1.

$$l_i = \arg\max\ f(x_i) \tag{2.1}$$

2.2.3 Support Vector Machines

Support Vector Machine is a non-parametric algorithm that aims to construct optimal decision boundaries, known as hyperplanes, to separate dataset classes. Considering a training dataset $D_t = \{(x_i, l_i)\}_{i=1}^{n}$ where $x \in X$ are dataset features and $l \in Y$ are the class labels. To train an SVM model on the given dataset, the decision function is optimised to identify decision boundaries.

The general formulation of the decision function is given in Eq. 2.2.

$$f(x) = sign(w^T.\phi(x) + b) \tag{2.2}$$

where w are weight vectors, $\phi(x)$ denotes the feature mapping function that maximises the margin between classes by mapping features to an n-dimensional space, and b is the bias. The goal is to find the optimal value of w and b to identify optimal decision boundaries. The data points nearest to decision boundaries are known as support vectors. Support vectors are crucial for defining decision boundaries, as they determine the position of the hyperplane to separate classes. This algorithm is inherently a binary classifier, where $l \in \{0, 1\}$; however, it can be extended to multiclass classification. To handle the separation between non-linear data, SVM includes a soft margin controlled by the regularisation parameter, allowing flexibility to some misclassification while maximising the margin. There are two approaches for developing a multiclass SVM: the one-vs-one (OvO) and the one-vs-rest (OvR) approach. One-vs-one develops a binary classifier between each pair of classes, and one-vs-rest develops a binary classifier for each class versus the rest. The SVM models are widely used in developing cybersecurity applications such as intrusion detection and spam classification due to their effectiveness in handling high-dimensional data. A three-class SVM is illustrated in Fig. 2.2, demonstrating the decision boundaries between classes with support vectors highlighted in red.

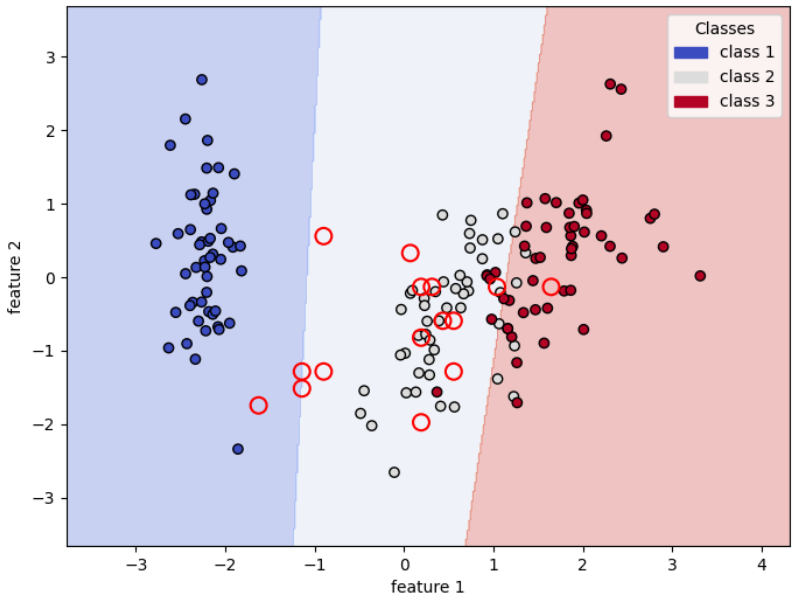


Figure 2.2: A trained support vector machines model developed with a dataset comprises three classes. The support vectors of this model are highlighted in red, which influenced the decision boundaries

2.2.4 Decision Tree

Decision Tree is a non-parametric algorithm used for both binary and multiclass classification. It develops a hierarchical tree-like structure that separates the feature space into regions associated with dataset classes. The algorithm follows a greedy, top-down approach to build a classification tree by selecting features that maximise the class homogeneity at each tree node. The tree recursively splits the dataset using a divide-and-conquer learning mechanism to minimise the impurity. The optimal split is estimated with an impurity measure such as the Gini Index, Log Loss, or Entropy. Given a training dataset $D_t = \{(x_i, l_i)\}_{i=1}^{n}$ where x represent features and l are the dataset labels, the objective at each node N is to identify the optimal feature that minimises impurity at child nodes, given a threshold t. Let the data at node m be denoted by N_m with n_m data points. Mathematically, the data is partitioned into N_m^{left} and N_m^{right} at each node $\theta(f, t_m)$ consists of a feature f and threshold t_m, such that:

$$\begin{aligned} N_m^{left}(\theta) &= \{(x, l) | x_f \leq t_m\} \\ N_m^{right}(\theta) &= N_m \backslash N_m^{left} \end{aligned} \tag{2.3}$$

Then, the quality of each split is calculated with the impurity (loss) measure $\mathcal{L}$, given in Eq. 2.4.

$$G(N_m, \theta) = \frac{n_m^{left}}{n_m} \mathcal{L}(N_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} \mathcal{L}(N_m^{right}(\theta)) \tag{2.4}$$

such that, the optimal loss is given in Eq. 2.5 until the maximum allowable depth is reached where $n_m < min\, datapoints$ or $n_m = 1$.

$$\theta^* = argmin\, G(N_m, \theta) \tag{2.5}$$

Following training, the effectiveness of the model is evaluated by testing it on the classification of unseen data. While DT is capable of handling complex, non-linear data, it is prone to overfitting when developed in full depth. Fig. 2.3 highlights the decision surface in a decision tree.

2.2.5 Random Forest

Random Forest is a non-parametric ensemble learning algorithm that combines multiple decision trees. Each tree is developed following bootstrap samples from the dataset, using the underlying split criteria of the decision tree algorithm. These bootstrapping and random subsampling mechanisms help reduce the correlation and improve generalisation in the model. At each node, a random subset of features is selected to determine the best split and introduce diversity in the RF model. This process improves the performance and controls overfitting by aggregating the predictions of all trees. In RF, each tree is constructed using the same splitting criteria described in Section 2.2.4. Let $T_1, T_2, \ldots T_n$ represent decision trees that are aggregated to develop an RF classifier. For classifying the i^{th} instance

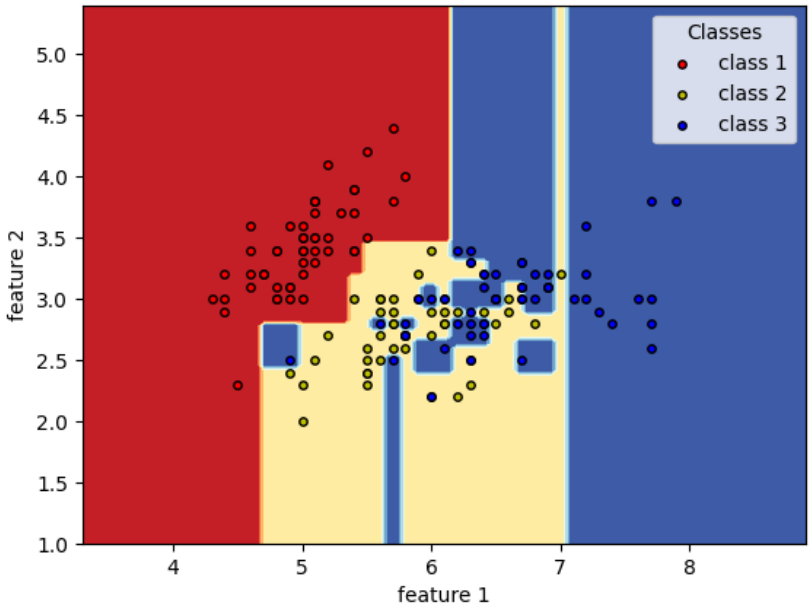


Figure 2.3: A trained decision tree model developed with a dataset comprises three classes. The decision surface of the decision tree is illustrated between three classes of the dataset with some overfitting between class 2 and class 3

x_i, the function to predict its class label is given in Eq. 2.6. The class label is determined through majority voting.

$$f(x) = mode(T_1(x), T_2(x), ..., T_n(x)) \tag{2.6}$$

where the mode returns the most frequent class among all trees. By combining predictions from decorrelated trees, this process controls the overfitting of the model.

RF inherently supports the development of both binary and multiclass classification. In binary classification, the model predicts between two classes following the decision function $f(x) \rightarrow l \in \{0, 1\}$ and for multiclass classification, it follows the decision function $f(x) \rightarrow l \in \{0, 1, 2, ..., m-1\}$ across more than two classes. Fig. 2.4 illustrates the decision boundaries created by an RF model trained on a three-class dataset, highlighting how RF splits the feature surface.

2.2.6 K-Nearest Neighbor

K-Nearest Neighbors is a non-parametric, lazy-learning algorithm that classifies based on the majority votes of the nearest data points in the training dataset. As a lazy-learning algorithm, it stores the complete dataset without performing any explicit model training and computes predictions at inference time. KNN follows the principle of the nearest neighbors and finds a predefined number k of nearest neighbors to predict the class of the test instance (Zhang et al., 2017). The model calculates the distance between the test instance and training data to identify the k nearest neighbors. This

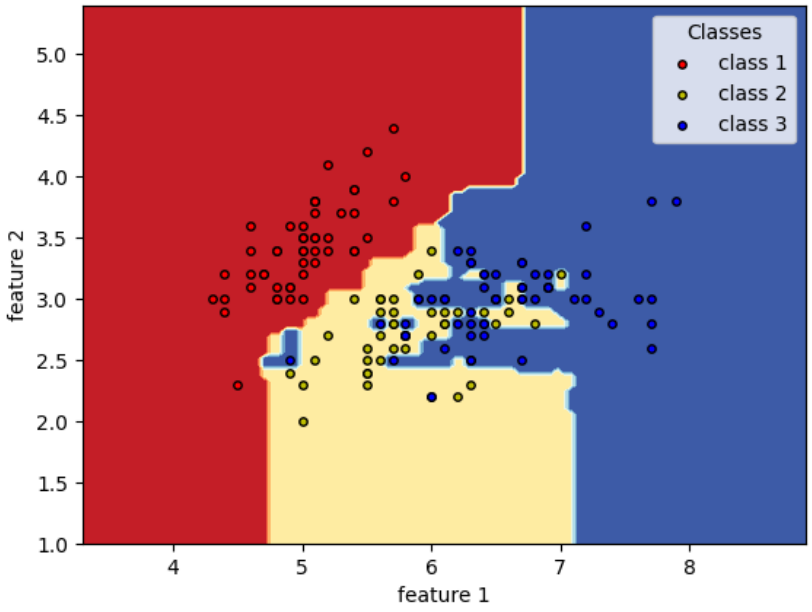


Figure 2.4: A trained random forest model developed with a dataset of three classes. The decision surface between three classes is illustrated

distance is measured using a distance algorithm such as Euclidean, Manhattan, or Minkowski, with the Euclidean distance as the default method. The voting process of KNN is sensitive to the noisy data points, which can shift the predictions. The majority voting can be inaccurate due to mislabeled data points among k-neighbors, particularly when k is small, which introduces high variance. The prediction mechanism is given in Eq. 2.7.

$$f(x) = argmax \sum_{i=1}^{k} \mathbb{I}(l_i = c) \tag{2.7}$$

where I is the indicator function, c is the possible class label of the test instance and y_i is the class label from i^{th} neighbors. Fig. 2.5 highlights a decision boundary of a KNN applied to a dataset with three classes.

2.2.7 Gaussian Naive Bayes

Gaussian Naive Bayes is a parametric algorithm that applies a naive assumption of conditional independence of all features in the dataset given a class. It also applies bayes theorem that calculates the posterior probability of a class label, for a data point, given specific feature values. Extending naive bayes with the assumption of conditional independence and probability estimations of features, GNB assumes that the dataset features follow a gaussian distribution given a class. GNB is inherently applicable to multiclass settings. Therefore, GNB applies to both binary and multiclass classification.

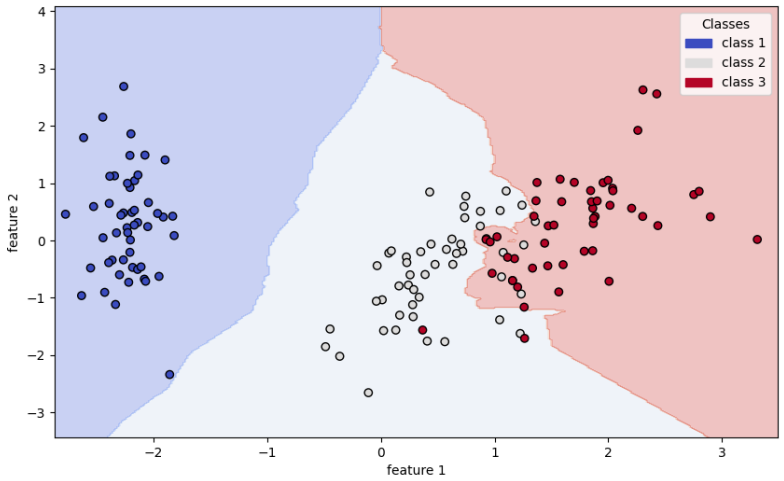


Figure 2.5: A trained k-nearest neighbors model developed with a dataset of three classes is given, showing the decision surface of each class

In supervised ML, let l_i be the class to be predicted for x_i instance given its features in n-dimensional vector $(v_1, v_2, \dots v_n)$ such that naive bayes probability estimation is given in Eq. 2.8.

$$\hat{l} = argmax\ P(l)\Pi_{j=1}^{n} P(x_j|l) \tag{2.8}$$

For x_i, the class probability $f(x)$ is predicted with Eq. 2.9.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma_{l_i}^2}} exp(-\frac{(x_j - \mu_{l_i})^2}{2\sigma_{l_i}^2}) \tag{2.9}$$

Fig. 2.6 illustrates the conditional distribution of GNB on a three-class dataset, estimated with Eq. 2.9.

2.2.8 Neural Networks

Neural networks is a parametric algorithm inspired by the human brain to learn complex, non-linear patterns in the provided data. In supervised ML, NN learns training data by approximating the mapping function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ where n denotes the dimensionality of the input features and m represents the number of output classes. This algorithm is designed in layers that comprise neurons, and each neuron applies a linear transformation followed by a non-linear activation function. Given $x \in \mathbb{R}^n$ as an input, the mapping function with weights w and bias b, given in Eq. 2.10 to compute the output.

$$z = actf(w.x + b) \tag{2.10}$$

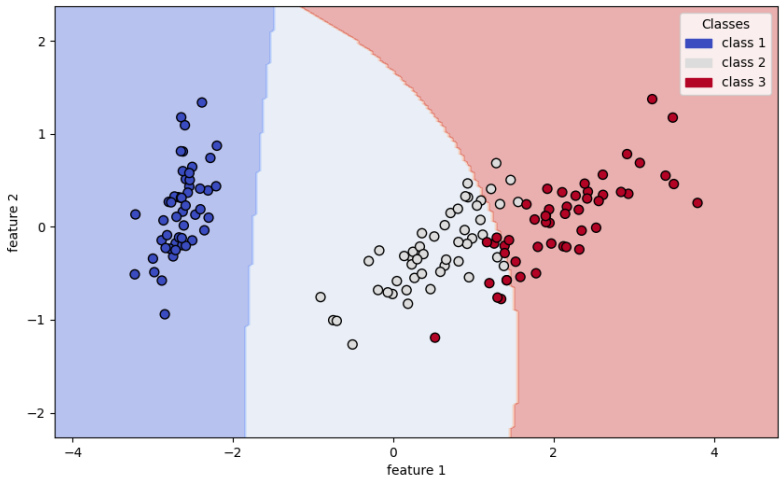


Figure 2.6: A trained gaussian naive bayes model developed with a dataset of three classes showing the decision boundaries of each class with some overlapping between class 2 and class 3

where z denotes output of a hidden layer, f denotes predictions at an output layer, and $actf$ is the activation function. The general structure of NN is shown in Fig. 2.7, illustrating an interconnected network of layers where preceding layers influence each layer.

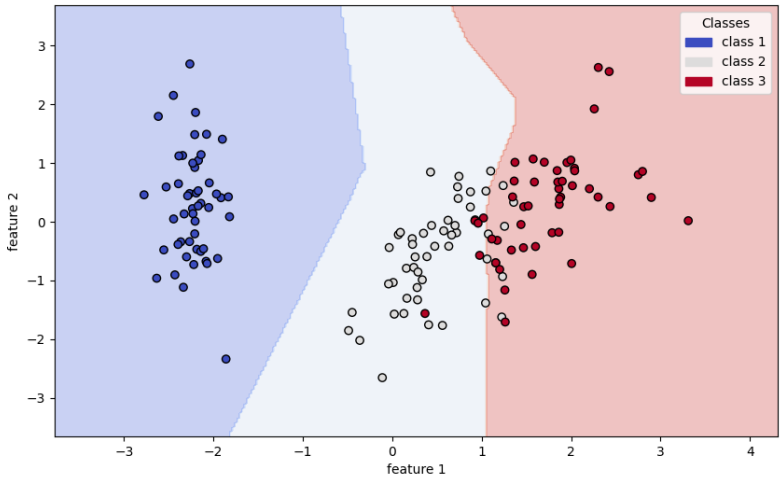


Figure 2.7: A trained multilayer perceptron model developed with a dataset of three classes showing the decision surface of the classes

A perceptron is the simplest NN form with one layer, used for binary classification, where a step function is employed as $actf$ to predict $l \in \{0, 1\}$. Further, an NN can be extended to multiclass classification by developing a multilayer perceptron (MLP), which comprises more than one layer in the network. MLP employs a non-linear activation function, such as ReLU, in intermediate layers, followed by the softmax function $actf$ at the output layer. These activation functions calculate pre-

diction confidence and output the class label with the highest confidence. These non-linear functions allow MLP to learn non-linear patterns in the data.

2.2.9 Adversarial Machine Learning

Adversarial machine learning is the study of attacks in ML and the countermeasures against them. The attacks against these models are implemented with maliciously crafted inputs called **adversarial examples** which mislead the performance of the model. Formally stating AML attacks in supervised ML as: *Supervised ML is about teaching machines to solve classification problems defined with a mapping function* $f : X \rightarrow Y$ *where in AML, adversarial examples* x' *are developed by perturbing* x *to generate manipulated output* l'*. Given a perturbed dataset* $D' = (x'_i, l'_i)_{i=1}^{n}$*, the classification function* $f(x') = l'$ *and* $l' \neq l$ *such that* x' *is generated as* $x + \epsilon = x'$ *where* ϵ *is the perturbation.*

The concept of adversarial attacks was first defined by Biggio et al. (2011) and Vidnerová and Neruda (2016), explaining these attacks on an SVM classifier. With this concept, the development pipeline of ML is found to be vulnerable to adversarial attacks, which can be attributed to various methods such as flipping labels, augmenting additional features in the dataset, and with the recent advancements in generative AI, adversarial queries can also be developed to manipulate ML/AI models. Since their discovery in 2011 to date, researchers have developed various types of adversarial exploits that have shown successful attacks on training (Chen et al., 2021), testing (Merzouk et al., 2022) and deployment (Zhu et al., 2023a) phases of the model.

To better understand adversarial attack mechanisms, a taxonomy is defined as shown in Fig. 2.8, of adversarial attacks taking into account the development phases of ML. Adversarial attacks are mainly divided into four types of data poisoning, evasion, inversion and inference attacks. This chapter will go through each attack type, explaining it in detail with its mechanism, evolution and formulation in subsequent sections.

In an orthogonal dimension, various adversarial capabilities or settings are studied in designing these attacks based on their knowledge. The highest level of capabilities is provided with complete knowledge of the ML model to the adversary, known as the white-box attack (Patterson et al., 2022). The second level of capabilities is provided with the grey-box attack (Wang et al., 2021a), where some knowledge of the model or dataset is provided to the adversary. For example, the adversary only knows the modality or the distribution of the dataset to devise an attack; however, the underlying settings of the target system are unknown to the adversary. And the lowest level of capabilities is provided with a black-box attack (Yu and Sun, 2022) where the target system is completely unknown to the adversary or may only know the name of the algorithm or dataset used to develop the system. For example, in an attack development, only the algorithm is known, but the dataset and target settings are completely unknown to the adversary. It is important to note that black-box and grey-box

capabilities overlap and are used interchangeably (Vivek et al., 2018). The next section will explain each attack type, explaining its concept and development in the context of supervised ML.

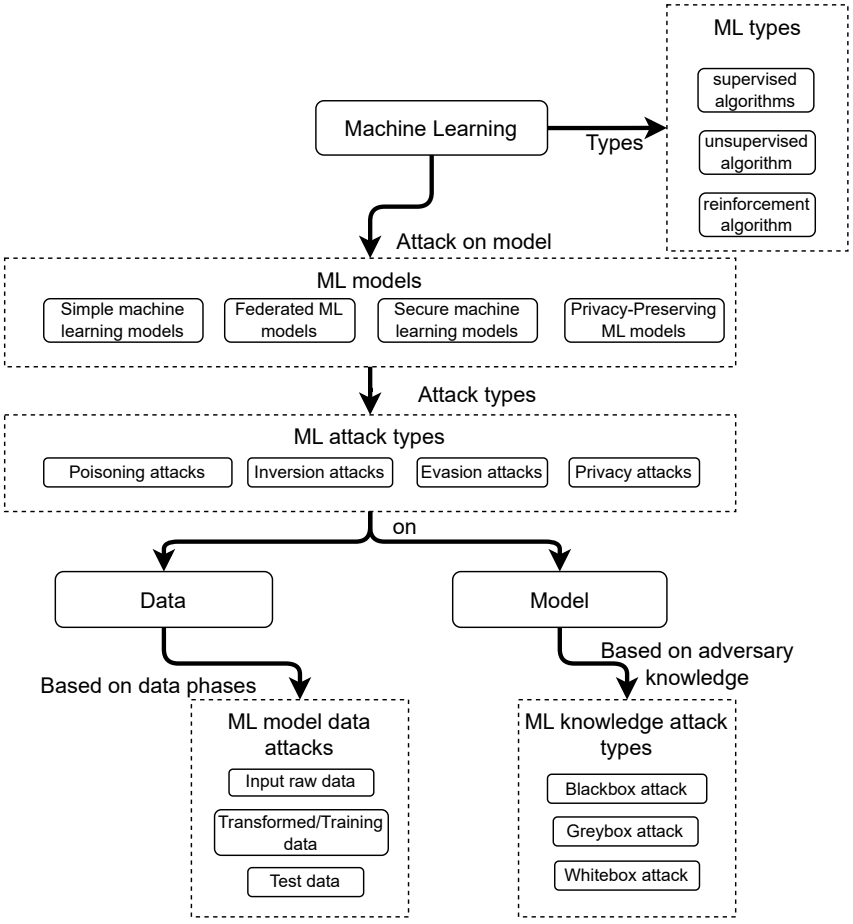


Figure 2.8: Taxonomy of adversarial machine learning attack types

2.2.9.1 Data Poisoning Attack A data poisoning attack is an adversarial attack that manipulates the training dataset of the model to corrupt the development process (Baker et al., 2024). The attack surface of the data poisoning attack is shown in Fig. 2.9. A data poisoning attack can be devised either to degrade the overall model performance, called an availability attack (Zhao and Lao, 2022a), or to alter the classification at specific inputs, called an integrity attack (Carlini and Terzis, 2021). The attacker can carry out data poisoning attacks in two ways. First, the poison may manipulate labels in the dataset, detaching features from their respective classes, known as label poisoning (Gupta et al., 2023b). Second, the poison may perturb features of the dataset, leaving labels in their original form, known as clean label poisoning (Jagielski et al., 2021). Consider $D = (x_i, l_i)_{i=1}^{n}$ as the clean dataset that is manipulated with malicious input and modified to $D' = (x'_i, l'_i)_{i=1}^{n}$ to maximise the classification errors of the decision function $C(.)$ at test time. One crucial consideration in data

poisoning attacks is that poisoned instances in the dataset closely resemble clean instances to avoid detection.

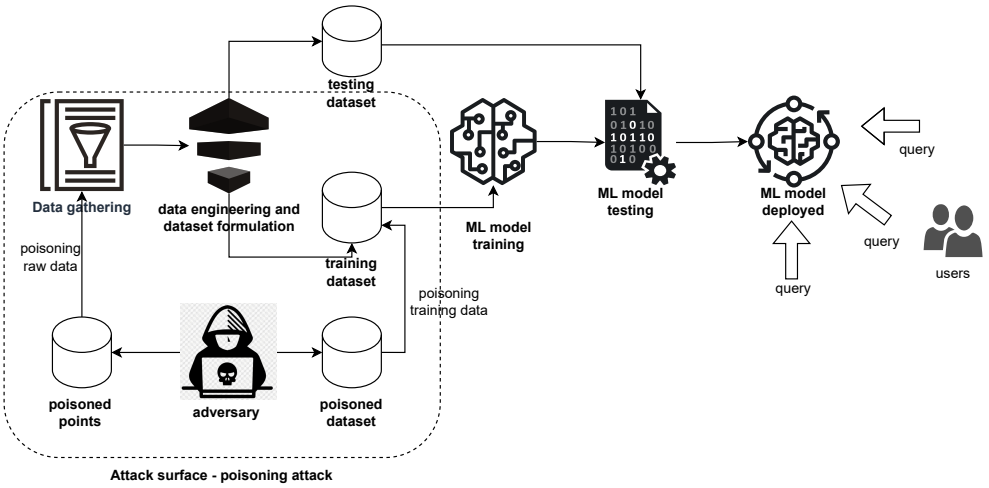


Figure 2.9: Poisoning attack surface in machine learning model development process - Poisoning machine learning attack

2.2.9.2 Evasion Attack An evasion attack is an adversarial attack that perturbs test inputs to manipulate the classification results during testing (Biggio et al., 2013). The attack surface for evasion attack is shown in Fig. 2.10. An evasion attack aims to manipulate the model's sensitivity to perturbations in test inputs by identifying a path that maximises loss in the input space. In this way, it forces the model to generate incorrect classifications while leaving the trained model and the training dataset intact. Consider a dataset $D = \{(x_i, l_i)\}_{i=1}^{n}$ that is divided into training dataset D_t and test dataset D_{test} that is manipulated with perturbation $x' = x+\epsilon$ such that $D'_{test} = \{(x'_i, l'_i)\}_{i=1}^{n}$, so that the perturbed test input x' is classified with an incorrect output $l' \neq l$.

2.2.9.3 Model Inversion Attack A model inversion attack is an adversarial attack in which the adversary steals or learns the confidential configuration parameters of the trained model to uncover the private information or replicate the model (Usynin et al., 2023). For example, in a facial recognition system, an attacker provides a face image to the model to assess its output confidence for a specific class or label associated with that image. In this way, the attacker reconstructs the image for a targeted label or identifies the model parameters by interpreting the output confidence. The attack surface for the inversion attack is given in Fig. 2.11. Let $C(.)$ be the decision function that provides the class label l, where, with an inversion attack, the attacker calculates loss in the confidence predicting y. Mathematically, it is given in Eq. 2.11.

$$argmin\, \mathcal{L}(C(x), y) \tag{2.11}$$

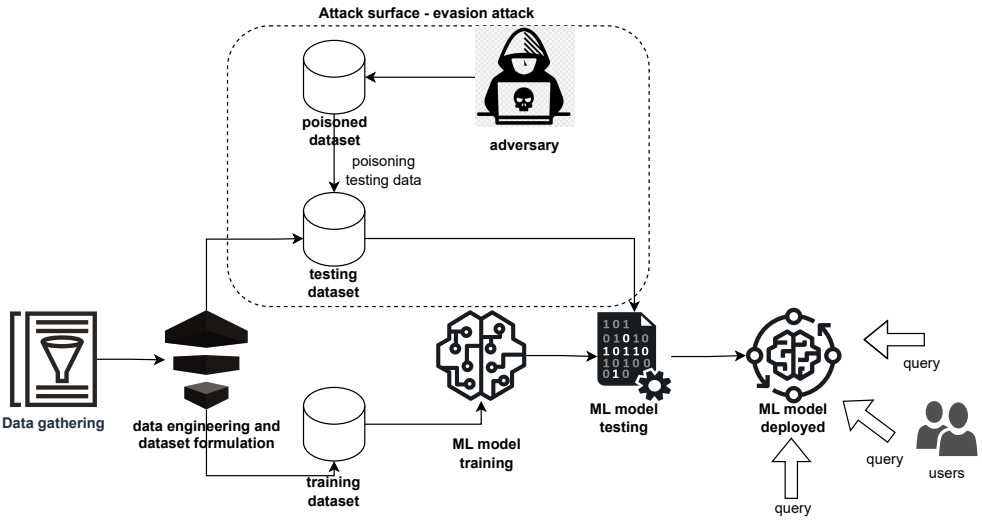


Figure 2.10: Evasion attack surface in machine learning model development process - Evasion machine learning attack

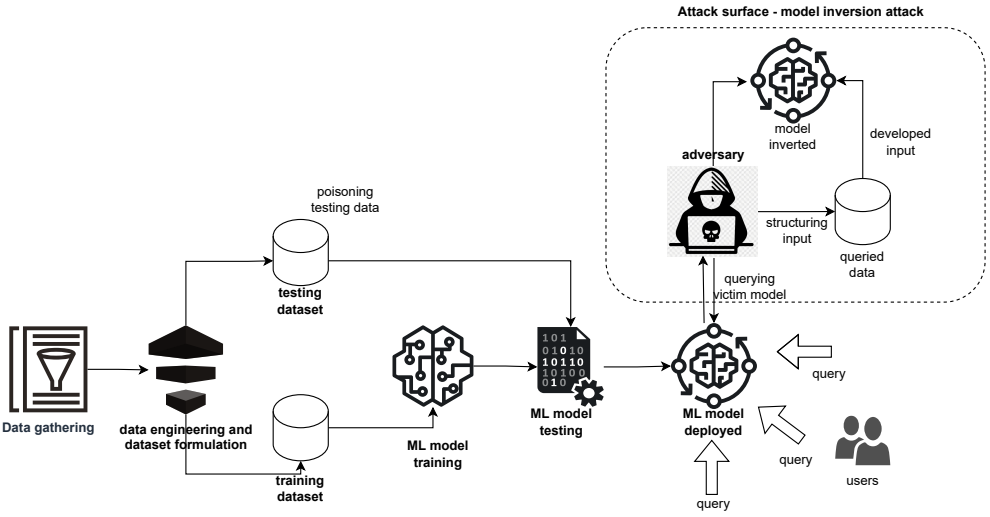


Figure 2.11: Model inversion attack surface in machine learning model development process - Model inversion machine learning attack

2.2.9.4 Membership Inference Attack A membership inference attack (Qiu et al., 2020) is an adversarial attack in which the attacker seeks to answer the question:

Is the input $\hat{x}$, given to classification function $f(\hat{x}) = y$, a member of the training dataset for the targeted model $\mathcal{M}$?

Generally, an ML model behaves differently on the training dataset compared to unseen data, which is why a difference in training accuracy and test accuracy is observed. Fig. 2.12 illustrates the attack

surface for membership inference attacks. Considering this behaviour, the primary objective of a membership inference attack is to exploit differences in model predictions and identify whether x is part of the training dataset. In this way, the attacker can infer whether specific inputs were part of the training data for the targeted model. Let $f(.)$ be the classification function, where $\hat{x}$ is the input and y is the output. Then the objective is to reconstruct $D = \{(\hat{x}, y)\}_{i=1}^{n}$.

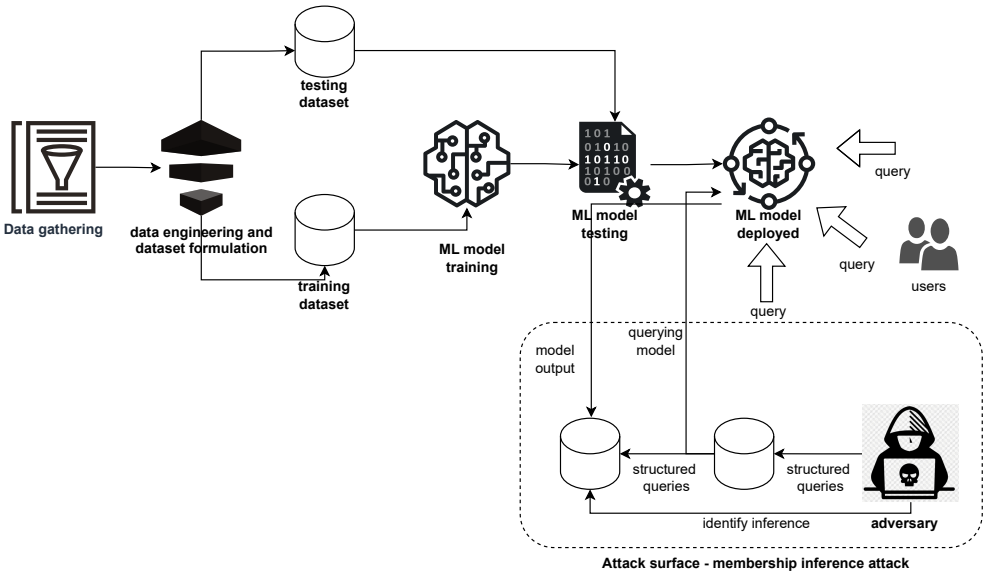


Figure 2.12: Membership inference attack surface in machine learning model development process - Membership inference machine learning attack

2.3 State-of-the-Art Analysis

2.3.1 Methodology

A detailed literature review is conducted following eight dimensions to analyse and compare adversarial attacks. The methodology for the state-of-the-art analysis is given in Fig. 2.13. The peer-reviewed

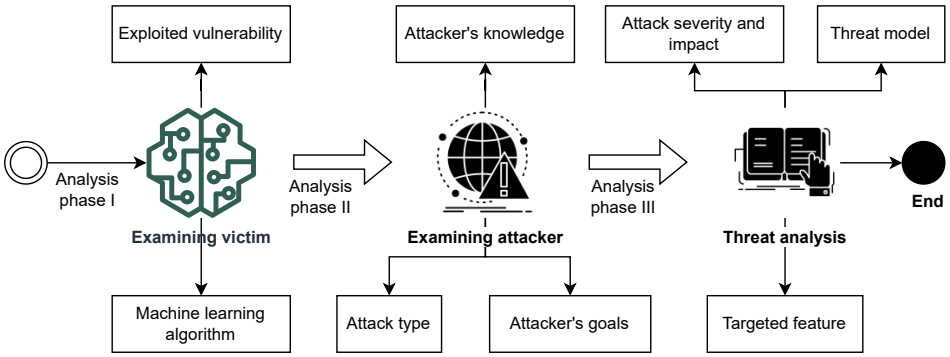


Figure 2.13: Research methodology of the state-of-the-art analysis

and conference papers, published in or after 2017, are selected for this literature review to assess adversarial attacks on the latest ML models and systems, developed with updated technologies and practices. The papers selected either focus on AML attacks or mitigation solutions against AML attacks to provide technical insights into the attack development.

Machine learning model/algorithm In this literature review, firstly, the ML algorithm/model included in the selected papers is analysed to study technical interpretations of the attack design. It is considered an influential factor in identifying the design and complexity of adversarial attacks. It also highlights the vulnerabilities of individual ML algorithms.

Exploited vulnerability Next, the exploited ML vulnerabilities are examined in detail, which helped develop the attack vector to manipulate ML models. This dimension technically assesses the attack success against the targeted system.

Attack type Further to the analysis of the ML algorithm/model and the exploited ML vulnerability, the adversarial attack type is examined to assess the attack surface of adversarial attacks. The attack types included are poisoning, evasion, model inversion, and membership inference attacks. Analysing existing studies based on these attack types explains the practicality, implications, and comparison of these attack types against ML systems.

Attacker's knowledge The literature review also explored the attacker's knowledge to deepen the understanding and examine the impact of the knowledge and capabilities of the adversary in developing adversarial attacks. The knowledge levels can be set between zero, partial, and complete knowledge of the targeted system. It helps analyse the impact of attacks from existing studies and compare the complexity and implications of each adversarial attack, as it is considered an important benchmark when designing these adversarial attacks.

Attacker's goals The next significant dimension is the detailed synopsis of the adversary's goals and objectives set with the devised attack. The consequences of the adversarial attack can be understood by analysing the intention and goals of the adversary.

Threat model The threat model is analysed under which the attack has been developed. The adversarial capabilities and access to the target system have been studied to analyse the practical feasibility of the attack.

Attack severity and impact After analysing adversarial attacks with the above dimensions, the attack severity is determined by studying the outcomes, as discussed in publications. Analysing the attack severity will provide us with grounds to study the complexities and practical implications of adversarial attack types.

Exploited feature The exploited feature of the model is determined for each attack to understand the vulnerabilities in ML models and their development processes.

2.3.2 Literature Review

This chapter proposed an in-depth and comparative analysis based on four major adversarial attack types on ML. To analyse various adversarial attack vectors in detail, each attack vector is examined based on attack type, exploited features, adversary, its capability and knowledge, and the impact of the attack vector on the victim model or algorithm. This study is the first literature analysis that examines existing works according to eight dimensions. A hierarchical summary of articles studied for attack analysis is given in Fig. 2.14. A detailed analysis of the examined attacks is provided in Sections 2.3.2.1 to 2.3.2.4. Attack vectors are analysed following eight dimensions, given in the methodology.

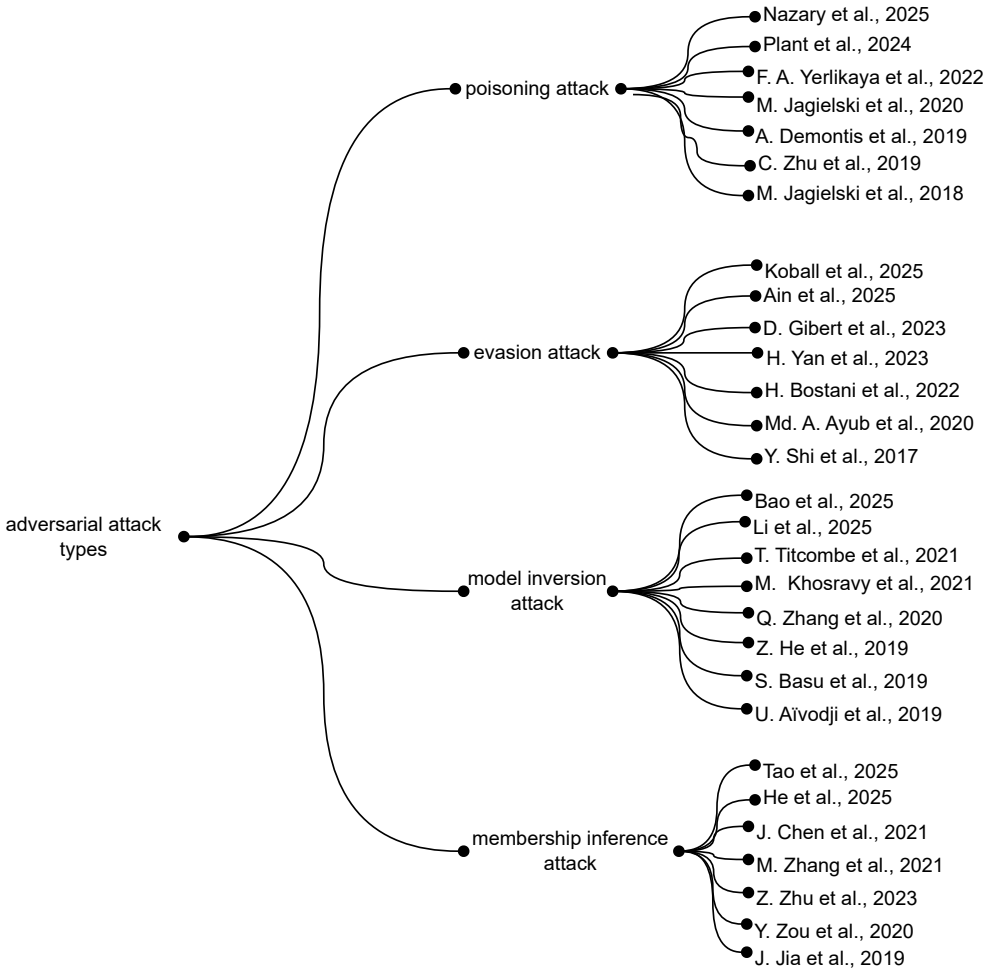


Figure 2.14: State-of-the-art - AML attack types

2.3.2.1 Data Poisoning Attacks Poisoning the dataset is possible in two approaches. The first approach is to disrupt the labeling strategy of the victim model, known as the label poisoning attack (Gupta et al., 2023a) and the second is the perturbation of features, where the adversary manipulates

the features of the dataset, leaving the label integrated, known as a clean-label poisoning attack (Zhao and Lao, 2022a). The existing literature provides several approaches to implement data poisoning attacks, considering the attack surface defined for data poisoning in ML development. The comparison of existing studies is given in Table 2.1. Yerlikaya and Şerif Bahtiyar (2022) designed two

Table 2.1: Analysis of poisoning attacks

Reference	Machine learning model/ algorithm	Attack type	Exploited vulnerability	Attacker's knowledge	Attacker's goals	Attack severity and impact	Defined threat model	Targeted feature
Nazary et al. (2025)	OpenAI LLM	Emotional attack, neighbor borrowing, chain attack	inject poisoned words to change emotional intensity	white-box attack	promote and demote recommendations	no	model performance	
Plant et al. (2024)	BERT, RoBERTa, ALBERT, DistilBERT, BERTeus, EIBERTheu, IXAmBERT, RobertaEUSCrawl	label-flipping attack	probabilistic approach to change class labels	black-box attack	reduce model performance	large pretrained models may be more robust than ML models	no	model performance
Yerlikaya and Şerif Bahtiyar (2022)	SVM, SGD, Logistic regression, Random forest, Gaussian NB, K-NN	random label and distance-based label flipping attacks	poisoning dataset by changing class labels with two effective strategies	white-box attack	reduce performance (accuracy) of the system	KNN and Random forest algorithms are not much affected by label poisoning attacks	no	model accuracy
Jagielski et al. (2021)	Convolution neural networks	subpopulation attack	poisoned cluster is integrated as subproportion of training dataset	gray-box attack	misclassification targeted attack	Subpopulation attacks are difficult to detect and mitigate specifically in non-linear models	yes	test time prediction
Demontis et al. (2018)	SVM classifier, Logistic, Ridge, SVM-RBF	training time poisoning attack	reduced gradient loss with poisoned data points in transferable setting	white-box, black-box attacks	violate model's integrity and availability	poisoning attacks are more effective on models with large gradient space and high complexity	yes	model availability
Zhu et al. (2019b)	Deep Neural networks	feature collision attack, convex polytope attack	feature space with perturbed training samples	gray-box attack	over fit target classifier with poisoned dataset	Turning dropout during training with poisoned data enhance transferability of poisoning attack in deep neural networks	yes	test time misclassification
Jagielski et al. (2018)	Linear regression	statistically based regression points poisoning generation with flipped labels	distinguishing legitimate and poisoned regression points with minimal gradient loss	mean and co-variance dependent gray-box attack	misclassification of the system	residual filtrating mitigates poisoning attack on Linear regression	yes	model accuracy

label-flipping attacks to perturb six ML algorithms with four datasets. The attacks were designed to poison binary classifiers and reduce model performance. Jagielski et al. (2021) proposed an ML poisoning attack called the Subpopulation attack. This attack injected a poisoned cluster into the

dataset to evade detection of poisoned data points. Existing security techniques, such as training a regression model (TRIM), activation clustering, reject on negative impact (RONI), and spectral signatures, were found to be ineffective against a subpopulation data poisoning attack. In the study (Demontis et al., 2018), the poisoning and evasion attacks on ML are designed to highlight their transferability during both the training and testing phases of model development. Demontis et al. (2018) highlighted the risk of transferring poison from the surrogate to the victim model. A gradient-based optimisation framework is developed to transfer the poison that alters the gradient of input samples in both training and testing datasets. It empirically analysed the security vulnerabilities in transfer learning and identified major factors that breach integrity, making poisoning and evasion attacks successful in transferring between surrogate and victim ML models. These factors are the attacker's optimisation objectives, gradient alignment of surrogate and target models, and model complexity.

Zhu et al. (2019b) also demonstrated the transferability of poisoning attacks in ML by implementing polytope attacks in deep neural networks. They explained the impact of clean-label poisoning attacks. They highlighted the successful poisoning with 50% performance degradation with 1% poisoning of the training dataset. The convex polytope attack is implemented on various deep neural networks as case studies in this research, highlighting the efficacy and consequences of data poisoning attacks in transfer learning. Their study confirmed the reliability and effectiveness of a convex polytope attack, comparing it with a feature collision attack. It also demonstrated the successful transferability of the convex polytope attack in a black-box setting where the adversary does not know the dataset of the victim model and still achieves almost the same results as when the adversary has a 50% overlap with the target dataset. In conclusion, the research discussed above has formulated improvements in the transferability of poisoning attacks by turning on the dropout rate and implementing convex polytope objectives in multiple layers of neural networks. This research underscores the need to secure ML, specifically neural networks, from poisoning attacks in various adversarial settings.

The research by Jagielski et al. (2018) particularly focuses on the security vulnerabilities and defense solutions related to linear regression. It focuses on poisoning linear regression models with gradient-based optimisation and statistical attack strategies. Their study introduces a new optimisation framework to poison linear regression in a gray-box attack setting, evaluating the limitations of existing attacks. It also proposed a statistical poisoning attack to maximise the loss by introducing poisoned points at the edges of the decision boundary. This attack exploits secure regression models. However, TRIM has been proposed, proving to be more effective in mitigating poisoning attacks in the linear regression model but ineffective against subpopulation attacks, thus proving the severity of the poisoning attack in adversarial settings.

2.3.2.2 Evasion Attacks Attacking the ML model at test time is known as an evasion attack. This attack aims to mislead the model by perturbing the test data (Ayub et al., 2020). The ultimate objective of this attack is to manipulate the testing input to undermine the integrity of models during testing. Malware Generative Recurrent Neural Network (MalRNN) is a deep learning-based approach designed to conduct evasion attacks on ML-based malware detection systems (Ebrahimi et al., 2020). MalRNN evades three malware detection systems, which demonstrates the effectiveness of evasion attacks. Moreover, these attacks highlight the importance of reliable security solutions to mitigate vulnerabilities in ML against evasion attacks. A comparison of evasion attacks is given in Table 2.2. Malware classifiers are also vulnerable to adversarial attacks. In the study by Bostani and Moonsamy (2021), a test-time attack is developed on an Android malware classifier to disrupt its classification outcome. The attack presented in this paper was a black-box attack that extracts opcodes using the n-grams strategy from disassembled Android application packages (APKs) and manipulates benign samples into malicious ones through a random search technique. This attack was tested on five malware detectors. It demonstrated the effectiveness of a test-time attack that evades the ML model and caused misclassification during testing. As a result, ML-based malware detectors, including Drebin, Detection malware in Android (MaMaDriod), with an accuracy of 81% and 75%, respectively, and others failed to detect malicious Android applications.

Similarly, the Jacobian-based saliency map attack (JSMA) also demonstrated the stealthiness of the evasion attack. JSMA was developed using a multi-layer perceptron for IDS. The goal is to achieve targeted misclassification, where the adversary intends to classify malware traffic in network intrusion detection systems (NIDS) as benign. The experimental analysis followed a white-box setting to devise this evasion attack, achieving a maximum accuracy drop of approximately 29% using the TRabID 2017 dataset. Hence, it demonstrated the malicious approach to threatening ML applications in cybersecurity, subsequently highlighting the test-time security vulnerabilities in neural networks.

The sensitivity of evasion and causative attacks is examined against DL models, proposed in (Shi and Sagduyu, 2017). This examination helps understanding the security vulnerabilities in deep learning models. This research devised an adversarial perturbation approach and tested it with text and image datasets. Initially, an evasion attack was conducted, followed by an exploratory attack aimed at inferring the trained classification model and extracting its private tuning parameters. The exploratory attack was a black-box query-based attack that replicates the victim model following the obtained query outputs. In the replicated model, this attack was extended by poisoning test labels to mislead the deep learning model through an evasion attack.

2.3.2.3 Model Inversion Attacks The objective of this attack is to disrupt the privacy of ML. Model inversion attack is a type of attack in which an adversary tries to steal the developed ML model

Table 2.2: Analysis of evasion attacks

Reference	Machine learning model/ algorithm	Attack type	Exploited vulnerability	Attacker's knowledge	Attacker's goals	Attack severity and impact	Defined threat model	Targeted feature
Koball et al. (2025)	decision tree, random forest, adaptive boosting, isolation forest	label perturbation to mislead classification	white-box attack	test-time performance decrease	yes	leaf nodes of the model		
Ain et al. (2025)	Fused Truncated DenseNet, Fused Swish-ReLU Efficient-Net, Efficient-capsule Net, Regularized Forensic Efficient Net	statistical distraction pipeline	gradient perturbation	black-box attack	evade deepfake detector	deepfake detector are susceptible to evasion attack	no	brightness of the localised masked area
Gibert et al. (2023)	Generative adversarial networks	query-free feature-based attack	perturbed features in executable	black-box attack	evade ML detector with malicious executable	ML-detectors are vulnerable to be evaded with query-free attacks	no	victim detection decision
Yan et al. (2023)	Logistic regression, SVM, NB, Decision tree, RF, xgBoost, ANN, Ensemble model	label-based evasion attack	poisoned labeled samples	black-box attack	transfer adversarially crafted samples to evade	transfer-based evasion attack is a serious threat to ML and DL	no	test time precision
Bostani and Moonsamy (2021)	ML-based malware detector	n-gram based attack on malware classifier	transform malware samples into benign with n-gram based incremental strategy	black-box attack with model query access	misclassification of android malware detector	DNN are more affected by evading surrogate models comparing to linear SVM classifier	yes	test time prediction
Ayub et al. (2020)	multi-layer perceptron network	Jacobian-based saliency map attack	iterative approach to insert perturbation near sensitive feature of benign samples	white box attack	misclassify malicious sample as benign in IDS	multi-layer perceptron can be exploited with evasion attack with minimal model's knowledge	no	test time prediction
Shi and Sagduyu (2017)	naïve bayes classifier	evasion attack with feed-forward neural networks	feed poisoned samples with DL score under computed attack region	exploratory black-box attack	misclassify test data samples	controlled perturbations to labels and classification boundary may limit adversarial impact on DL	yes	model availability

by replicating its underlying behaviour, querying it with different datasets. An adversary extracts the baseline model representation through a model inversion attack and can regenerate the training data to replicate the model. The comparison of the selected studies is provided in Table 2.3. Usynin et al. (2023) designed a framework for a model inversion attack on a collaborative ML model and demonstrated its effectiveness. Their study also highlights the impact of model inversion attacks on transfer learning models.

The research paper (He et al., 2019) experimentally demonstrated a privacy attack during inference

Table 2.3: Analysis of model inversion attacks

Reference	Machine learning model/ algorithm	Attack type	Exploited vulnerability	Attacker's knowledge	Attacker's goals	Attack severity and impact	Defined threat model	Targeted feature
Li et al. (2025)	VGG16, ResNet50, IR50-SE, AlexNet	gradient perturbation on entropy loss	gradient of images	white-box attack	GAN transfer learning	model inversion attacks have limited efficacy against GAN models	no	dataset samples privacy
Bao et al. (2025)	FaceNet64, ResNet-152, VGG16	generative MI attack, knowledge-enriched distributional MI attack, reinforcement-learning-based black-box MI attack, boundary-repelling MI attack	latent probability distribution	black-box attack, white-box attack	identify private data within optimal latent probability distribution	identifies model private training data samples	no	training dataset samples
Titcombe et al. (2021)	split neural networks	model inversion attack on distributed ML	steal intermediate/distributed data from nodes in transfer learning	black box attack	invert intermediate stolen data into input format	Model inversion attacks are effective and dependent on input dataset	yes	model interception
Khosravy et al. (2021)	deep neural networks	images reconstruction with MIA	regenerate model by intercepting private data of victim model by gathering output	gray-box attack	inverted model and developed duplicate	ML is under serious threat of MIA attack with partial knowledge of system	no	model privacy
Zhang et al. (2020)	deep neural networks	stealing victim's model classes	sample regeneration helps to determine private data of victim's model classes	white box attack	developed surrogate model similar to the target	ML model can be inverted even if secured with differential privacy	yes	model privacy
He et al. (2019)	deep neural networks	Inverse-Network attack strategy	used untrusted participant in collaborative system	black box, white box and query-free inversion attacks	extract inference data with an un-trusted adversarial participant in collaborative network	Privacy-preservation is challenging to achieve in split DNN	yes	model privacy
Basu et al. (2019)	deep neural networks	generative adversarial network approach	extracted output from targeted network with generative inference details	white-box attack	extract model class/inference details by replicating generative adversarial network	machine learning can be inverted with generative samples	no	model accuracy
Aïvodji et al. (2019)	deep neural networks	query based generative adversarial network	extract model details by interpreting queried outputs	black box attack	Breach privacy of Convolutional neural networks (CNN)	differential privacy is not much effective to mitigate MIA on machine learning	no	model privacy

in collaborative ML and argued that a single malicious participant could infer the target system and steal its confidential information. This attack is successful in all three settings of complete knowledge, zero knowledge, and a query-free attack setting. The confidential tuning parameters were extracted using a regularised maximum likelihood estimation technique in which the adversary followed the Euclidean distance estimation and identified the optimal sample with the least variation. In conclusion, this research highlighted the potential of inference attacks that require attention to be mitigated to ensure privacy preservation of deep learning. Basu et al. (2019) demonstrated privacy issues in ML algorithms by inverting a deep neural network (DNN) with a model inversion attack. This research study implemented the model inversion attack on a facial recognition system and extracted the class representation of the model. The attack developed in this research had only baseline knowledge of the target system. The attacker employed a generative adversarial network to create input samples and invert the victim model, highlighting the effectiveness of generative AI in inverting the model. Another framework called generative adversarial model inversion (GAMIN), developed by U. Aivodji and others (Aïvodji et al., 2019), was also based on generative adversarial networks. It created adversarial images to query the targeted model and extract its details through comparative output resemblance. The threat disclosed with adversarial networks is that even without prior knowledge of the system under attack, the adversary can extract its confidential parameters and reconstruct the model. Khosravy et al. (2022) also developed a model inversion attack on a deep neural network-based face recognition system. It was a gray-box attack in which the adversary had partial knowledge of the system under attack, including model structure and parameters. This attack extracted the model configurations by reconstructing images based on the confidence scores provided by the targeted model, hence inverting the targeted CNN model. In conclusion, all the mentioned attacks emphasise the privacy-preservation of ML, which is a primary consideration in constructing trustworthy and resilient AI/ML that resists adversarial attacks.

2.3.2.4 Membership Inference Attacks A membership inference attack is another privacy attack that infers the victim model and extracts its training data, privacy settings, and model parameters. In this type of attack, the adversary has access to query the victim model under attack and can analyse the output gathered from the queried results. The adversary can regenerate the training dataset of the targeted adversarial ML model by analysing the gathered queried results. The Membership inference attack (MIA) is another privacy risk to ML and DL models. The comparison of selected MIA attacks is given in Table 2.4. Zou et al. (2020) comprehensively studied membership inference attacks in deep learning models under transfer learning and achieved 95% accuracy in determining whether the input instance belonged to the training dataset. These attacks were developed for three different transfer learning modes as part of this research. When the adversary had access to

Table 2.4: Analysis of membership inference attacks

Reference	Machine learning model/ algorithm	Attack type	Exploited vulnerability	Attacker's knowledge	Attacker's goals	Attack severity and impact	Defined threat model	Targeted feature
He et al. (2025)	LLaMA2-13B, Falcon-7B, Pythia-6.9B, OPT-6.7B	label-only MIA attack	token-level semantic similarity	black-box attack	label-only MIA attacks are designed for the fine-tuning phase of LLM	label-only MIA attacks are highly effective in inferring fine-tuning LLMs	yes	personalisation of LLMs with fine-tuning tokens
Tao and Shokri (2025)	Multilayer Perceptron	hyperparameters range following the reference model	black-box attack	identify training dataset	MIA attack infer accurate data within a defined range	no	hyperparameter range	
Zhu et al. (2023d)	multi-layer perceptron	MIA on sequential recommendation system	surrogate and shadow models are designed to extract recommendations	black-box attack	infer user recommendations	inferring sequential recommendations leads to provide personalised details	yes	dataset inference
Chen et al. (2020)	Lasso regression, CNN	MIA with shadow model	shadow model is used to mimic ground truth	white box attack	retrieve confidential details of target model	differential privacy mitigates MIA compromising accuracy of model	no	model inference
Zhang et al. (2021)	neural networks-based recommendation system	inference attack to extract user-level details	adversarial model is developed with theft users' private data	black box attack	retrieve private details of victim model	Popularity randomisation is effective against MIA in recommender system	yes	model privacy
Zou et al. (2020)	deep neural networks	transfer learning based black-box attack	no privacy-preserved in transfer learning model	black box attack	infer training model details with three formulated attacks	transfer learning is at serious threat of MIA	yes	model inference
Jia et al. (2019)	neural network	MIA against binary classifier	interpret output confidence score to manipulate model details	black-box attack	retrieve private training data of classifier	existing solutions are subject to the dataset used in the classifier	no	dataset inference

the teacher model, they targeted the trained student model and inferred the teacher model's dataset using the student model. Zou et al. (2020) implemented a surrogate model based on ResNet20 convolutional neural networks with derived and student datasets to infer the membership inference of the victim model. This attack vector was effective in demonstrating the capability of membership inference attacks to compromise privacy, even with limited access or information about the victim model. Another potential privacy attack is mentioned in (Zhang et al., 2021), where the attacker targeted an automated recommender system using a membership inference attack. This is a zero knowledge attack. Their attack posed a serious privacy threat to the recommender system's sensitive user data, which adversaries can reveal using a query-based attack. In this context, the inference attack is characterised by three recommender algorithms: item-based collaborative filtering, a latent factor model, and neural collaborative filtering. A shadow model was implemented to mimic the victim's training dataset, which ultimately puts its privacy at risk.

Various mitigation techniques are also proposed to secure ML models from the adversarial attacks mentioned above. Many existing solutions are primarily attack-agnostic or algorithm-focused. While they may secure models from specific attacks, they might fail against new attacks. The proposed security solutions in the literature have various limitations that must be addressed to maintain the integrity of ML, ensuring that AI/ML remains secure and trustworthy. A hierarchical description of the mitigation techniques analysed in this study, based on adversarial attack types, is presented in Fig. 2.15. A detailed analysis of existing security solutions based on adversarial attack types is given as follows:

2.3.2.5 Mitigating Data Poisoning Attacks

Data Sanitisation Data sanitisation is one of the prominent mitigation techniques against data poisoning attacks, which preprocesses training datasets and removes erroneous or poisoned data points. Nevertheless, this process may reduce the size of the dataset, leading to underfitting in model development. Venkatesan et al. (2021) proposed a solution to overcome the limitations of data sanitisation by creating random training data subsets to train an ensemble of ten classifiers to balance the poisoning effect and dataset size. This mechanism reduces poisoning effects in NIDS by 30%. Similarly, another data sanitisation derivative is applied to malware detection systems to mitigate clean-label poisoning attacks (Ho et al., 2022b). This approach is an enhancement provided in (Venkatesan et al., 2021). Further, the study (Paudice et al., 2018b) proposed another approach to label sanitisation to reduce the impact of overfitting and underfitting. However, P. W. Koh and others (Koh et al., 2021) proposed three sophisticated poisoning attacks by introducing cluster-based poisoning that breached the sanitisation solutions highlighted above.

RONI is also a derivation of data sanitisation proposed by Chan et al. (2018a), which removes poi-

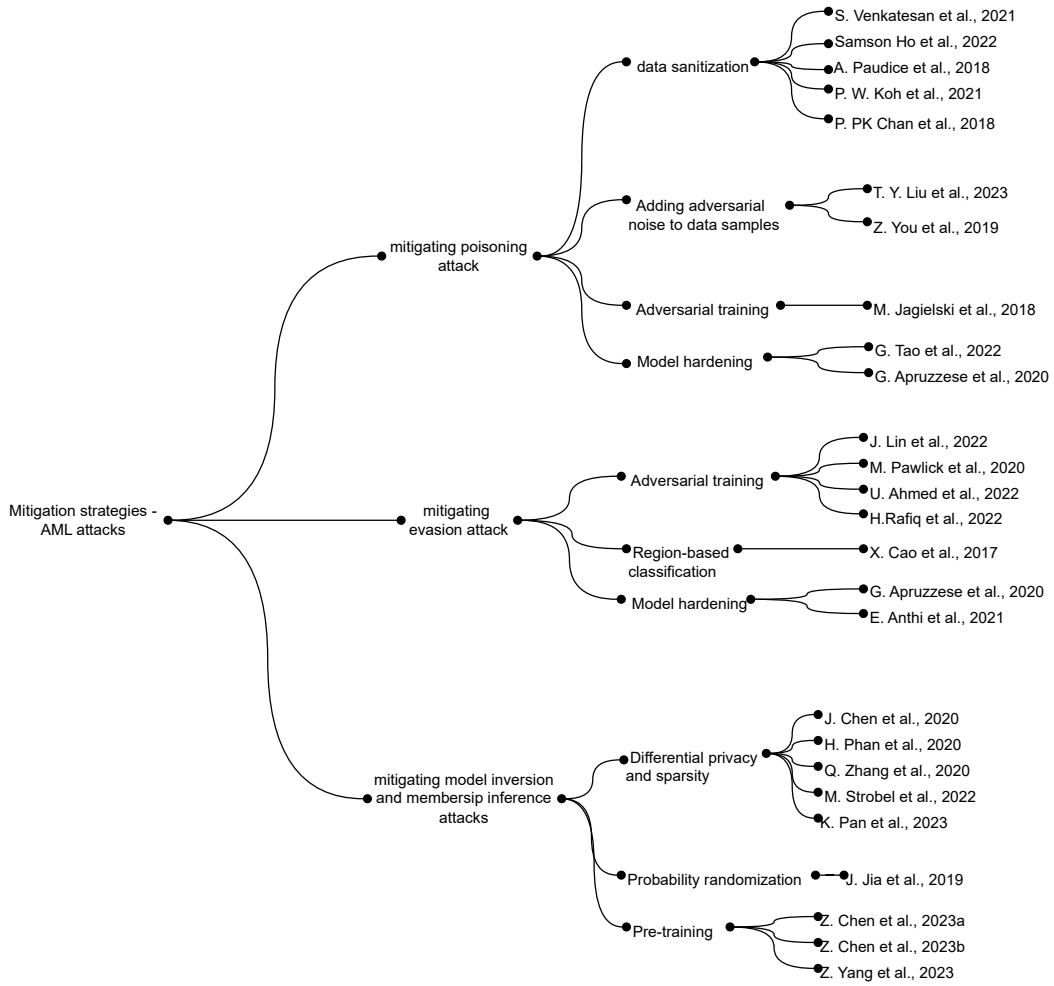


Figure 2.15: State-of-the-art of mitigation techniques - AML attack types

soned data samples by analysing their negative impact on classification accuracy. However, it also leads to underfitting issues that reduce the flexibility and increase false negatives at test time.

Adding Adversarial Perturbation Training the ML model with an adversarially developed dataset allows the trained model to identify poisoned samples at test time. Liu et al. (2023) have boosted the immunity of the model by adding specifically crafted noise samples in the dataset during training, which is effective against bulls-eye polytope, gradient masking and sleeper agent attacks. Another study (You et al., 2019) has introduced adversarial noise into the intermediate layer of CNN to mitigate FGSM attacks.

Adversarial Training Training an ML model with adversarial data samples allows it to be resilient against poisoning attacks. TRIM is one of the techniques used to adversarially train models with a residual subset of a dataset with a minimum error rate. Jagielski et al. (2018) have designed and experimented with this TRIM algorithm against adversarial poisoning attacks against the linear regression algorithm to solve optimisation problems. This approach has reduced the error rate to

approximately 6%. It performs robustly compared to Random sample consensus (RANSAC), a data sanitisation derivative. Whereas, TRIM and RONI security techniques failed against the subpopulation attack developed in (Jagielski et al., 2021).

Model Hardening Another innovative technique to mitigate poisoning attacks is model hardening, in which the model is trained until it leads to large class distances where it should not accept outliers. This technique makes it challenging for an adversary to poison the model. Tao et al. (2022a) proposed a model hardening mechanism with additional training to increase the class distances and challenge the label-flipping attack. The study (Apruzzese et al., 2020a) hardens the random forest algorithm to mitigate the poisoning impact on an IDS. Moreover, it can also help mitigate backdoor attacks against neural networks. It reduces misclassification up to 80%, but it is still only effective against label-flipping backdoor attacks.

2.3.2.6 Mitigating Evasion Attacks

Adversarial Training Adversarial training is a prominent mechanism to mitigate evasion attacks in ML. A particular dataset part is intentionally poisoned to lessen the test time evasion and make the model adversarially robust (Pawlicki et al., 2020). It allows the victim to be aware of adversarial samples if injected at test time to detect and defend itself if attacked by an adversary. Ahmed et al. (2022) proposed adversarial training by classifying adversarial and normal data samples, followed by centroid-based clustering of features and calculating the cosine similarity and centroid of the image vector. The research (Rafiq et al., 2023) trains independent models to reduce fabricated classification attacks and (Lin et al., 2022) secures against Carlini and Wagner and FGSM attacks.

Model Hardening The hardening ML model also applies to developing a wall of security in ML against adversarial attacks at test time. Evasion attacks are also mitigated with the help of a training model until they reach the state of hardening, which activates the model to evade adversaries and mitigate attack impact. Adversarially crafted samples are intentionally injected during the ML model training to evade the system until it reaches the state of hardening, making the victim model resilient and robust. These poisoned input data samples evade the system and are then marked as poisoned in the system to identify similar patterns if injected by the adversary at test time. Apruzzese et al. (2020b) have introduced a similar strategy to mitigate evasion attacks in botnet detection systems by deep reinforcement learning. They have developed an agent based on deep reinforcement learning capable of generating adversarial samples to evade the targeted botnet. Then, including these adversarially generated samples into the targeted system marked as malicious to make the model understand the pattern of adversarial samples if attacked during test time, whereas the research study (Anthi et al., 2021) used model hardening to secure an ML-based IoT system. A threshold is specified that trains the model properly with the legitimate and illegitimate datasets, which makes the botnet

detector robust against evasion attacks.

Region-based Classification Cao and Gong (2017) have designed a classification mechanism based on region rather than individual sample points. The researchers provided this technique based on the assumption that the adversarial points lie near the classification boundary. A hypercube-centered classification approach is determined by omitting single-point-based classification at test time to reduce the impact of adversarial points.

2.3.2.7 Mitigating Model Inversion and Membership Inference Attacks

Differential Privacy and Sparsity To preserve the privacy of ML models, one of the profound solutions is differential privacy. It makes it difficult for the adversary to analyse the output and extract the victim's confidential information. Chen et al. (2020) have used differential privacy applied with stochastic gradient descent on Lasso and CNN neural networks to preserve genomic data privacy. Phan et al. (2020) improve DNN robustness by implementing differential privacy with the logarithmic relation between the privacy budget and the accuracy of the targeted model. They have empirically analysed genomic data for phenotype prediction with a white-box attack. However, Zhang et al. (2020) improves differential privacy by implementing it at the class and sub-class level, proving the minimal probability of model inversion attack at the dataset only. Class and sub-class level differential privacy is more effective and robust than simple record-level differential privacy, providing more Euclidean distance between original and inverted data samples. However, it is tested with neural networks only with the Face24 and MNIST datasets. Also, this type of differential privacy requires high computational resources, whereas the study (Strobel and Shokri, 2022) highlights trade-offs of data privacy and assuring its trustworthiness. Pan et al. (2023) implemented differential privacy to mitigate privacy attacks and data leaks against generative adversarial networks. Whereas, the floating-point attack mentioned in (Jin et al., 2022) has invalidated differential privacy implemented to preserve the privacy of ML models.

Probability Randomisation Adversarial privacy attacks, specifically membership inference attacks, target ML classifiers and infer input datasets by interpreting the confidence score and probability of the queried output. Adding noise to the output or intentionally interrupting the confidence probability score leads to the privacy preservation of ML, preventing adversaries from inferring confidential details of the victim model. Membership inference guard (MemGuard) (Jia et al., 2019) is one of the solutions designed to preserve the privacy of ML models against membership inference attacks by adding randomised noise to each of the score vectors with a specified probability of accuracy loss, and makes ML-based binary classifiers resilient to mitigate membership inference attacks. However, the solution is only tested for securing neural networks under the black-box attack settings.

Pretraining Chen et al. (2023) have proposed a model-preserving framework to preserve the security

of deep learning models while training models by combining model parameters and training data. Chen and Pattabiraman (2023) have introduced a new framework to pre-train an ML-based model to preserve privacy by enforcing less confidence in the queried results between members and non-members. Z. Yang and others (Yang et al., 2023) have introduced another model to statistically in-distinguish the confidence scores of members and non-members.

2.4 Research Gaps in the State-of-the-Art

This state-of-the-art study reveals that data poisoning attacks are primarily examined in computer vision tasks using image datasets Ghosh et al. (2022), Zhong et al. (2020), Su et al. (2019). However, analysing poisoning impact in other domains focusing on traditional models such as RF, DT, and GNB is also important as these models are used in various real-world applications and often preferred when working with limited or unprocessed datasets Dixit et al. (2018), Sasmono et al. (2021), Kamath et al. (2018). Also, these models are a baseline for the advanced ML, such as neural networks for DL; therefore, understanding their poisoned behaviours helps improve mitigations for both ML and DL models.

In particular, traditional multiclass models have received limited attention in studies of data poisoning attacks Paracha et al. (2025a), Paracha et al. (2024b). Multiclass models play a critical role in addressing complex, multifaceted problems Lee et al. (2021), Rahman et al. (2023), Adarsh and Jeyakumari (2013). Analysing data poisoning attacks against these models provides a comprehensive understanding of poisoning impact under various development settings.

In addition, existing literature highlighted that most studies evaluate data poisoning attacks using standard performance metrics, including accuracy, recall, f1-score, and adversarial success rate to assess their impact Chen et al. (2021), Lu et al. (2024), Jagielski et al. (2018). However, it is crucial to understand how data poisoning alters the underlying decision mechanisms of these models, which is a limitation of existing work. Such deep analysis provides an understanding of the changing characteristics of the model, which helps improve and strengthen security-by-design solutions. Another identified limitation is that most studies have developed data poisoning attacks with full knowledge and capabilities of adversaries, whereas understanding their feasibility under constrained adversarial capabilities highlights their effectiveness and practicality against real-world applications.

Conversely, this analysis identifies several limitations in existing mitigation strategies for data poisoning attacks. A primary issue is that most solutions are either attack-specific or algorithm-specific. For instance, adversarial training (Ho et al., 2022a) is predominantly applied to deep neural networks, leveraging gradient loss during training, which restricts its applicability to models such as RF and GNB that do not utilise gradient-based optimisation. Evaluating the effectiveness of such methods remains unexplored in traditional models. Similarly, data sanitisation (Venkatesan et al., 2021) was

developed for network intrusion detection systems (NIDS), but its dependence on domain-specific thresholds limits its generalisability.

This thesis seeks to address these research gaps with a systematic evaluation of traditional ML models by first proposing deep behavioural analysis in Chapter 3, followed by evaluating the effectiveness of multiclass data poisoning attacks under limited adversarial capabilities in Chapter 4. Following this structural analysis, this thesis proposes an attack-agnostic and security-by-design solution in Chapter 5 to enhance the resilience and reliability of multiclass models.

2.5 Summary

This chapter presented a comprehensive state-of-the-art analysis of existing efforts in the domain of ML security. Following a detailed literature analysis, this chapter attempted to provide a good reference to researchers to gain insights into the domain of AML and understand the limitations of existing mitigation techniques against adversarial attacks. This chapter provided a systematic understanding of vulnerabilities across various threats, forming the baseline for evolving attacks and solutions. This chapter answered the RQ 1 from the theoretical aspects with this state-of-the-art analysis. Furthermore, this chapter analysed the security vulnerabilities in ML models under different adversarial capabilities and threat models to highlight the significance of such attacks. For example, data poisoning attacks are identified as a significant threat to training pipelines that can also be extended to initiate other attacks, such as inference attacks that leverage data poisoning. Moreover, existing mitigations are reviewed, revealing that most of the solutions are attack-specific solutions that are designed for specific attack vectors or applicable to limited datasets, domains, or systems. These solutions secure models from one attack, but adversaries can attack the model using another technique. The answer to RQ 1 is extended in the next chapter to practically analyse the security vulnerabilities of data poisoning attacks with limited adversarial knowledge and capabilities. .

Chapter 3. Investigating Machine Learning Behaviour Against Data Poisoning Attacks

3.1 Introduction

Chapter 2 covers the foundational concepts of ML and state-of-the-art adversarial attacks, including poisoning, evasion, inversion, and inference attacks. It highlights that ML models are inherently flexible and adaptive, making them vulnerable to adversarial manipulations. Among these attacks, data poisoning stands out for its practicality and prevalence. This chapter investigates fundamental threats posed by data poisoning attacks. While existing studies have typically explored these attacks with performance metrics, such as reduced accuracy or increased adversarial success rates, this study conducted a deep behavioural analysis. This study examined changes in learning dynamics, including altered model characteristics and shifted decision boundaries, to reveal how poisoning affects the decision mechanisms of individual models. This chapter answers RQ 1 by analysing how the underlying behaviour of ML models changes with the poisoned training. It also begins to answer RQ 2 by analysing the severity and impact of data poisoning attacks under limited adversarial capabilities. This chapter proposes a behavioural analysis to assess these changes and practically fulfill the objective 1 of the research. It interprets the variance in the ML model with poisoned training. It also compares the change in individual characteristics of models at varying poisoning levels to analyse their impact on the model training.

During the model training, the ML model develops a dynamic classification mechanism that leverages these models to understand the nature of new data and classify it. Such proliferation of ML models and their dynamic classification mechanism render their security fundamentals to the security of systems underpinned by them. Several poisoning techniques are available in the literature, such as (Geiping et al., 2020; Koh et al., 2022). Further, various sophisticated attacks are formulated to poison ML algorithms. Some of its successful examples are convex-polytope (Zhu et al., 2019b), label-flipping (Yerlikaya and Şerif Bahtiyar, 2022), bullseye-polytope (Aghakhani et al., 2021) and poison frog (Shafahi et al., 2018).

These data poisoning attacks manipulate training datasets in two ways. Firstly, the adversary can perturb features of the training dataset so that ML models misinterpret this data and wrongly develop their classification mechanism. Secondly, data poisoning attacks manipulate the classification labels of the training dataset, which blurs the decision boundaries of models. Data poisoning attack has a serious impact on various real-life applications, such as deceiving breast cancer diagnosis applications (Das et al., 2024), skin cancer diagnosis applications (Paracha et al., 2024b), and IoT-enabled smart city systems (Zhu et al., 2023c). To better mitigate these data poisoning attacks, it is significant to

understand the manipulated behaviour of these poisoned ML models.

Security analysis of ML algorithms has received significant attention in recent years, focusing on poisoning and other adversarial attempts. Specifically on data poisoning attacks, various research papers have conducted an empirical analysis of the performance of ML algorithms (Yerlikaya and Şerif Bahtiyar, 2022; Aryal et al., 2022). But these attempts do not provide the technical understanding of change in their classification behaviour and why data poisoning attacks are successful in corrupting their decision mechanisms, which is attempted for the first time in this research. The contributions of this chapter are given as follows:

- This chapter conducts a thorough behavioural analysis of ML classification algorithms against a label-flipping attack to analyse the correlation between the percentage of poisoned data and the impact on classification accuracy.
- This chapter analyses the impact of data poisoning on the design of ML algorithms and highlights sensitive factors for each algorithm.
- This chapter analyses the behaviour and impact of data poisoning on classification results with a noisy dataset. This chapter subsequently studied the impact of anisotropic features and imbalanced dataset distribution.

3.2 Related Work

This chapter first understands the fundamentals of data poisoning attacks, existing attack vectors, mitigating solutions, and the limitations of these solutions. Keeping ML models secure and integrated is a potential and active research challenge. Poisoning attacks, which involve contaminating datasets, pose significant risks to the integrity and confidentiality of ML systems. Notable examples of poisoning attacks are rethinking Label-Flipping (Xu et al., 2022), geometric algorithms for KNN poisoning (Centurion et al., 2023), and subpopulation data poisoning attack (Jagielski et al., 2021). These attacks have effectively compromised the integrity and intruded into the targeted models. However, there is a pressing need to understand the differential responses of these models to poisoning techniques.

Poisoning attacks are increasingly endangering the reliability and accessibility of ML models. Mayerhofer and Mayer (2022) have polluted a featured-extraction system developed with convolutional neural networks (CNN) and evaluated the effectiveness of poisoning attacks against it. Zhang et al. (2023) have fooled deep neural networks (DNN) by integrating manipulated features into them. Further studies, such as (Zhu et al., 2019b; Shafahi et al., 2018) manipulated ML with features set perturbation while Zhu et al. (2022) showed how specifically altered pixels could mislead classifiers. In another study (Chen et al., 2022), perturbed labels are proved to help insert a backdoor in ML

with a success rate of 98%. A brief comparison of existing similar work with this study is given in Table 3.1. Further Table 3.2 extends the analysis presented in Table 3.1, including a comparative analysis of this approach with existing data poisoning studies, highlighting the impact on performance of individual algorithms. This work aims to interpret the analytical reasoning of different supervised ML algorithms and their response to various poison levels. Although some research studies, such as (Yerlikaya and Şerif Bahtiyar, 2022; Aryal et al., 2022; Dunn et al., 2020), have conducted empirical analysis to show the performance degradation of ML, they fall short in explaining why models trained on identical datasets with the same poisoning rates exhibit divergent performance behaviours. Addressing this gap forms a central part of the contribution. Many existing research studies investigated the integrity violation and shift in ML decision-making, either deterministically or indiscriminately. However, the urge is to study the underlying model design that reveals poison differently for each ML algorithm and so to interpret its explainability against poisoning.

To distill ML models from the detrimental effects of poisoning attacks, a variety of mitigation strate-

Table 3.1: Comparative analysis with existing similar studies that provides an empirical analysis of poisoning attacks on machine learning along with their objectives/focus to conduct empirical analysis

S.No.	Source	Datasets used	Attack settings	Focus
1	Yerlikaya and Şerif Bahtiyar (2022)	Instagram fake spammer genuine accounts, botnet detection, android malware/benign permissions, breast cancer wisconsin	White box attack	Practical implication of data poisoning
2	Aryal et al. (2022)	Malware detection	Black box attack	Impact of data poisoning on Malware detection
3	Dunn et al. (2020)	ToN-IoT, UNSW-NB15	-	Impact of data poisoning on smart IoT systems
4	**This Behavioural Analysis**	BotDroid, CIC-IDS-2017, CTU-13, UNSW-NB15	Black box attack	Factual study to identify factors affecting the performance of each ML algorithm with data poisoning

Table 3.2: Performance analysis with existing similar studies analysing the impact of poisoning attacks on machine learning models

S.No.	Source	Performance Metrix	Model performance					
			SVM	RF	DT	KNN	GNB	Perceptron
1	Yerlikaya and Şerif Bahtiyar (2022)	Accuracy	69.33	68.16	-	74.08	56.83	-
		Precision	-	-	-	-	-	-
		F1-score	71.33	67.34	-	74.2	51.65	-
2	Aryal et al. (2022)	Accuracy	78.58	96.54	96.54	87.41	-	75.16
		Precision	74.45	93.04	93.54	82.48	-	68.58
		F1-score	73.51	95.9	95.88	85.12	-	72.57
3	Dunn et al. (2020)	Accuracy	-	79.22	-	-	73.11	76.91
		Precision	-	79.22	-	-	74.9	70.01
		F1-score	-	-	-	-	-	-
4	**This Behavioural Analysis**	Accuracy	48.5	77.02	62.5	64.56	47.5	71.2
		Precision	48.3	58.79	58.56	63.29	62.1	71.2
		F1-score	61.5	60	57.4	61.04	61.04	62.79

gies have been developed. A data sanitisation against an adversarial label-flipping attack is proposed in (Chan et al., 2018a). This method begins with random dataset clustering to train the model on

subsets of a dataset and then sanitises data points, taking into account the complexity of the data and its impact on model performance. Adversarial training Geiping et al. (2021), Tao et al. (2021), Qin et al. (2023) is also an effective technique to allow ML models to proactively learn poisoned patterns to guard against poison if injected. A brief overview of the discussed mitigation techniques is given in Table 3.3.

Where Drews et al. (2020) proposed an antidote to verify the robustness of a decision tree against

Table 3.3: Detailed synopsis of relevant existing defence solutions against data poisoning attacks along with the parameter targeted to develop mitigation strategy and algorithms on which the respected proposed solutions are experimented

Research paper	Mitigation technique	Exploited feature	Experimented algorithm
Chan et al. (2018a)	Data sanitisation with cluster training	Dataset features complexity vector	SVM
Baracaldo et al. (2017)	Data segmentation and filtering using provenance feature	Provenance feature	SVM, Logistic Regression
Paudice et al. (2018a)	Training classifiers for individual dataset classes and calculate outliers with Empirical Cumulative Distribution Function	Dataset outliers	Linear classifiers
Geiping et al. (2021)	Adversarial training to maximise adversarial loss to detect poisonous data points	No explicit feature is exploited	Neural networks
Tao et al. (2021)	Adversarial training to mitigate delusive attacks with different data distributions	Robust and non-robust features in data distributions	Neural Networks
Qin et al. (2023)	Adversarial training with data augmentation in unlearnable examples to maximise error	No explicit feature is exploited	Neural Networks

data poisoning attacks. Although these solutions are effective, they lack generalisability. In particular, the main contribution of this research study is to provide a rigorous examination of supervised ML behaviour against poisoning attacks. On a large scale, none of the studies have yet highlighted the reasons behind the different responses of ML algorithms when trained with the same poison levels.

3.3 Behavioural Analysis

This chapter proposes a behavioural analysis as the study of understanding the change in the classification behaviour of supervised models by implementing a data poisoning attack. The poison is implemented at various levels in order to analyse the behavioural changes at certain classification metrics such as feature importance score, decision boundary, k-neighbors, and individual algorithm properties. This deep behavioural analysis is conducted for six ML algorithms, analysing poisoning impact and correlation between poisoning levels and classification accuracy. Adopting an empirical approach, this analysis highlightes the practical feasibility of data poisoning, comprehensively analysing factors of individual algorithms affected by poisoning.

Real-World Detection Applications Although various mitigations have been proposed, this chapter discussed their limitations in Section 3.2. Given these limitations, this chapter conducts a be-

havioural analysis of poisoned intrusion and botnet detection systems in binary classification settings to understand the change in behaviour of detection mechanisms with poisoned training.

As digital systems evolve, cybersecurity applications have become essential in protecting data and infrastructure. These applications support various security-critical scenarios such as malware detection (Brown et al., 2024), intrusion detection (Wang et al., 2020), automated firewalls (Al-Haijaa and Ishtaiwia, 2021), and biometric recognition (Jhong et al., 2020). Among these, an intrusion detection system (IDS) monitors system behaviour or network traffic to detect anomalies caused by cyber attacks. It helps strengthen system security by detecting anomalous traffic and notifying the administration of potential threats. However, if adversaries evade IDS, attacks may compromise systems undetected.

The widespread use of open standards such as TCP/IP, ethernet, and web technologies allows digital systems to connect with enterprise networks, enabling interoperability and external connectivity. Nonetheless, this expands the threat surface, increasing the number of attacks and their sophistication. Significant incidents, such as the Mirai attack (Sinanović and Mrdovic, 2017), have been reported that use robot networks, known as botnets, to launch more sophisticated intrusions. Some prominent methods are distributed denial-of-service (DDoS) attacks, spam distribution, and data theft, known as botnet attacks.

Intrusion and botnet detection systems can be implemented using various methods, of which the simplest is the signature-based approach. It compares incoming requests to the database of known attacks, though it may fail to detect previously unseen attacks. To overcome this limitation, a data-driven approach is followed that derives rules by observing behaviour, for example, data mining techniques (Awajan, 2023).

Instead of developing signatures, the patterns in the given data are modelled to identify deviations from normal behaviour by using statistical techniques. These are known as prediction-based approaches, which can be implemented with ML algorithms. Training these detection systems with ML algorithms makes them adaptive to predict complex structured intrusions and zero-day attacks.

The training process of ML algorithms is vulnerable to data poisoning attacks, which can be extended to these systems. The first step is to understand the fundamentals of datasets to enable the behavioural analysis of these systems.

3.4 Fundamentals of Training Datasets

This section discusses fundamental concepts of ML training that are considered in the following behavioural analysis. In supervised ML, a dataset $D = \{(x_i, l_i)\}_{i=1}^{n}$ is a collection of structured data where each data instance is known as a data point. Each data point consists of a set of features x_i, representing measurable attributes from real-world observations and a corresponding label l_i. For

example, in an intrusion detection application, the features can be the source of the input, contents, and port, and the label indicates if an email is an intrusion(1) or not an intrusion(0). A dataset D is characterised by various properties that describe its complexity, structure, interconnections between features, class alignment, and outliers, described as follows:

3.4.1 Dataset Size and Distribution

The dataset size, represented by n samples in a dataset D, influences the learning capability and generalisability of an ML model. It allows the model to learn the underlying data distribution. According to the law of large numbers, the empirical risk $P_n(D)$ converges to the expected loss as the dataset size n increases, given in Eq. 3.1.

$$P_n(D) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), l_i) \tag{3.1}$$

where $\mathcal{L}(.,.)$ is the loss function. In an ideal dataset, all the classes or labeled data points are of equal size. However, when an adversary perturbs the dataset through label poisoning or altering certain features, these manipulations often change the data distribution. The changed data distribution enables the model to learn classes of variable size, which may lead to model bias. For example, consider training a binary classifier for IDS where the adversary manipulates the labels of 10% of intrusion (1) data points, changing them to normal (0). Therefore, the model may fail to learn intrusion characteristics completely, resulting in an increased false negative rate for various intrusions. However, it is interesting that the behavioural analysis observed different effects of the same data poisoning attack on different IDS. Also, the behavioural analysis observed a non-linear correlation between various poisoning levels and their impact on the targeted system. Thus, it is crucial to investigate the correlation between dataset size and data poisoning.

3.4.2 Feature Correlation

In supervised ML, feature correlation refers to a statistical relationship between two features in a dataset, usually measured with Pearson correlation for a linear relationship or monotonic relationship with Spearman correlation. High feature correlations can introduce redundancy, potentially increase model complexity, and impair feature importance scores. For example, it may degrade the performance of GNB, which assumes conditional independence of features, and such correlations may undermine the posterior probabilities. In hierarchical models like DT, strong feature correlation may lead to redundant splits, potentially increasing model variance and risk of overfitting. While low feature correlation is generally beneficial in linear and probabilistic models, the model should be evaluated under the specific context of the use case.

For example, consider training an IDS with a decision tree algorithm on a dataset where two features, x_i and x_j, are highly correlated. Since a decision tree uses greedy split based on information gain, it may select one over the other and learn the noise and specifics of training data rather than understanding underlying patterns that may introduce overfitting. In adversarial settings, data poisoning may disrupt feature correlations to manipulate the structure of the dataset. Thus, understanding the disruption introduced by data poisoning helps strengthen mitigations against it. On the other hand, it is also important to examine how different dataset structures and feature correlations influence data poisoning, which also helps improve mitigation solutions.

3.4.3 Dataset Outliers

Dataset instances that deviate significantly from the original distribution are known as outliers. Outliers may introduce training errors and increase classification loss. It particularly impacts distance-based models, such as KNN and SVM, which classify data based on distances. For example, it can shift the decision boundary between two classes in SVM, which relies on data points, known as support vectors, that are closest to the margin. When an outlier lies near the margin, it becomes a support vector that can distort the hyperplane between two classes and reduce the model's generalisability. Conversely, it minimally impacts hierarchical-based models, such as decision trees and random forests, because these models follow greedy splits based on feature importance scores rather than calculating distances between features. Biggio et al. (2011) disrupted the dataset features and generated outliers in it, thereby degrading the model's performance. Therefore, it is essential to understand how data poisoning generates outliers in training data and how these outliers affect models with various poisoning attacks under similar adversarial settings.

3.5 Approach to Behavioural Analysis

This approach is formulated on an untargeted attack with an underlying assumption that the adversary does not know any internal model settings and data distribution. The label-flipping attack is implemented because it is the simplest and popular attack, used in many research papers(Jebreel et al., 2024),(Mengara, 2024), in data poisoning and to keep the attack settings simple to understand models' behaviour clearly.

The attacking procedure is developed on a practical examination of data poisoning on ML classification models. An experimental study is conducted to analyse the impact of data poisoning on individual ML algorithms. The goal is to highlight optimal levels of poisoning that are difficult to interpret and lead to significant performance degradation in classifiers. Data poisoning is implemented by flipping labels randomly from 5%-25% of the dataset. The dataset is split with 75%-25%, where 75% of the data is used for training and 25% of the data is used for testing, as the ML model

accurately learn with atleast 75% of the dataset (Song et al., 2017). This analysis is significant in understanding the poisoning behaviour for developing profound mitigation solutions that proactively secure ML models from existing and novel attacks.

This chapter aimed to specifically study the correlation between the granular impact of poisoning on the performance of ML algorithms. It also studied the correlation between data poisoning and feature relations in the dataset. This analysis employs six supervised ML algorithms: Support Vector Machines (SVM), Decision Tree (DT), K-nearest neighbors (KNN), Random Forest (RF), Gaussian naive Bayes (GNB), and Perceptron. These algorithms are selected to cover all the baseline classification methods in ML. This helps analyse different classification behaviours in ML with four distinct datasets (BotDroid (Seraj, 2022), UNSW-NB15 (David, 2018), CTU-13 (Malik, 2022), and CIC-IDS-2017 (Chenthan, 2023)). These datasets are selected as they distinct in their structure, size and feature correlation.

3.5.1 Attack Method

The fundamentals of the attack method have been interpreted to poison binary ML classifiers in black-box attack settings. Definitions of the attack method and evaluation metrics to measure the poisoning effects and analyse behaviours of ML algorithms are also provided. The architectural overview of the attack model is given in Fig. 3.1, representing the attacked and clean model development and defining the analysis criteria. Steps 1-9 are given in Fig. 3.1, highlighting the attack development

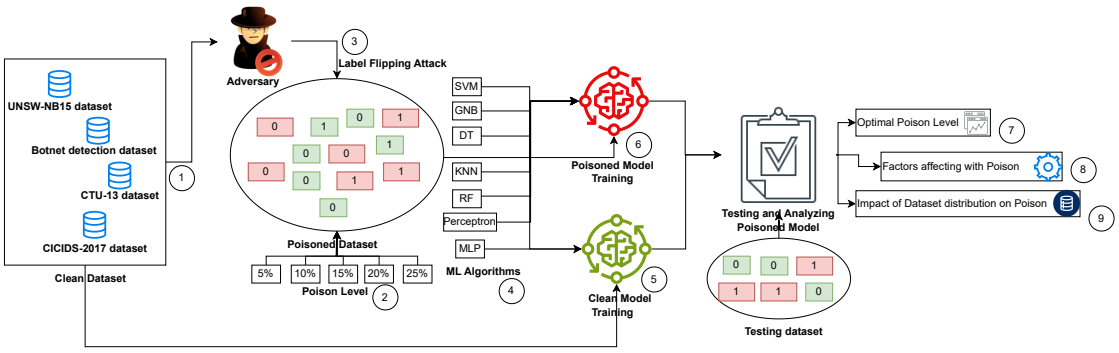


Figure 3.1: Architectural overview of the attack model, developed for conducting behavioural analysis

flow. Step 1 is the dataset selection, step 2 defines the poisoning levels to poison datasets, step 3 is the implementation of the label flipping attack on datasets with the defined poisoning levels, and step 4 defines algorithms to be considered for analysis. Steps 5 and 6 are the development of poisoned and cleaned models subsequently. Steps 7-9 define the analysis criteria. The objective of this research study is the behavioural analysis of poisoned ML algorithms. Binary classifiers are intended to be poisoned following the algorithm described in Algorithm 1. Each algorithm is audited to identify individual factors that are affected by dataset manipulation. The attack is developed in

black-box settings with a notion of generalisation to poison labeled datasets to degrade the classifier's performance. The sub-definitions of the attack are given as follow.

Algorithm 1 Poisoning Algorithm

```
Datasets used are: UNSW-NB15, BotDroid, CTU-13, CIC-IDS-2017
D_t ← Training dataset
D' ← Poisoned dataset = []
D_r ← subset of Training dataset
Record ← D_t(x_i, l_i) is the instance in dataset D_t with features x an label l at position i
ΔL ← Poisoning level = [0%, 5%, 10%, 15%, 20%, 25%]
for poison ∈ ΔL do
    Select D_r of length poison from D_t
    for Record ∈ D_r do
        if Record not in D' then
            if l_i equals 0 then
                l_i = 1
            else
                l_i = 0
            end if
        end if
        D' ← Record
    end for
end for
```

Poison Penetration Considering poison $L = [5\%,10\%,15\%,20\%,25\%]$, of training dataset, developed with algorithm A_{RFlip} as given in Alg 1 to manipulate data points x_i, by changing its label and generating poisoned data points x'_i, to disrupt ML model M is given in Eq 4.7.

$$M'_\theta = training(M(x'), x' = A_{RFlip}(D_o, \Delta L)) \tag{3.2}$$

which allows us to factorise poisoned model for behavioural analysis as given in Eq 3.3.

$$P_{Pen} = factorizing(M'_\theta) \tag{3.3}$$

where $M'_\theta \in$ [SVM, DF, RF, KNN, GNB, Perceptron] and *factorizing* is the interpretation method of analysing individual parameters affected with poison.

Performance Degradation Here, the test time performance statistics are calculated at each poisoning level to statistically analyse degradation as given in Eq. 3.4.

$$\begin{aligned} P_{Deg} &= \sum_{i}^{n=1} L_{i\to j}(A_{Acc}(M'_j) < A_{Acc}(M'_i)) \\ and\, P_{Deg} &\propto \frac{1}{A_{Acc}(M'_j)} \propto FPR(M'_j) \end{aligned} \tag{3.4}$$

where P_{Deg} is the performance degradation, injected with poison $L_i \to j$ at each incremental level from i to j. A_{Acc} is accumulated as the accuracy of the poisoned model M' whereas FPR is the

rate at which the poisoned model predicted falsified positive outcomes. The poison is injected with a minimum 5% poisoning rate, leading to a maximum of 25%, at a scale of 5. Binary classifiers are developed with four benchmarked datasets consisting of network traffic, described in Section 3.5.2, following the purpose of providing a precise and granular study, to put forward analytical reasons of how and why the performance of every algorithm is affected differently. For example, the performance (accuracy, precision, recall, f1-score) of DT, KNN, and RF is degraded linearly. Whereas the performances of neural networks are continuously fluctuating. The ablation study is given in Section 3.6.

3.5.2 Experimental Datasets

A rigorous ML analysis is conducted, developing intrusion detection classifiers with four benchmarked datasets, namely UNSW-NB15 (David, 2018), CIC-IDS-2017 (Chenthan, 2023), CTU-13 (Malik, 2022), and BotDroid (Seraj, 2022). UNSW-NB15 is based on captured network traffic with nine attacks included in it. It contains a total of 48 features containing packet-level information that enables us to conduct a detailed analysis for intrusion detection. Aligning it for binary classification, the records are classified into benign and malignant. BotDroid is the next dataset included, which comprises 45 features in total, and is encoded with one-hot encoding. CTU-13 is also included to develop a binary IDS classifier in this study which is developed to capture real-time traffic in thirteen different scenarios. and CIC-IDS-2017 is the last dataset included as part of this analysis. A brief description of all datasets is mentioned in Table 3.4. For the appropriate visualisation and features correlation in the dataset, the dataset complexity has been reduced to N dimensions with PCA, as given in Eq. 3.5.

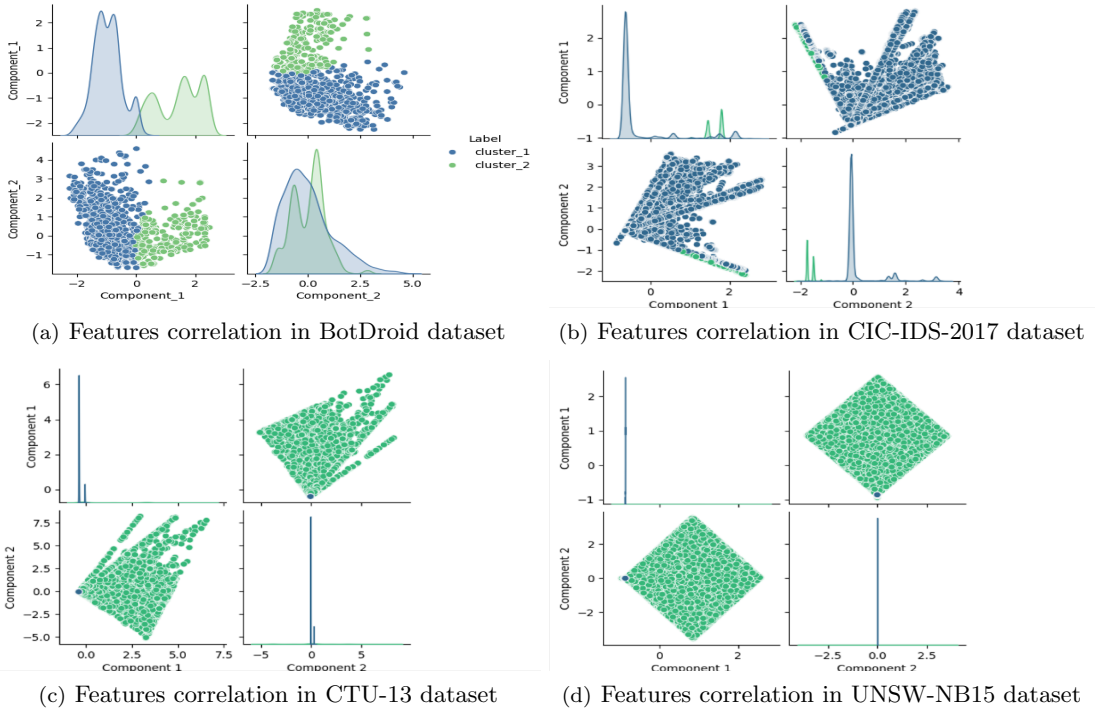
$$Dimentionality\_Reduction = PCA(n\_components = N) \tag{3.5}$$

And, for features structure and understanding, Gaussian Mixture Models(GMM) have been used.

Table 3.4: Dataset description used to perform data poisoning on machine learning algorithms

S.No.	Dataset	No. of features	No. of instances	Description
1	UNSW-NB15	48	82332	Dataset comprises of nine network intrusion attacks
2	BotDroid	45	1367	Dataset with limited instances, focusing BotDroid
3	CTU-13	57	92212	Captured with real time network traffic for BotDroid
4	CIC-IDS-2017	78	692703	Dataset consists of network packets

The visual dataset representation can be seen in Fig. 3.2. BotDroid dataset is the most appropriately distributed dataset with aligned features and minimal distortion(outliers) as shown in Fig. 3.2(a). Whereas UNSW-NB15 contains anisotropic features with asymmetric relation for which an uneven and elongated distribution is highlighted with sharp peaks at one axis in Fig. 3.2(d). However,



(a) Features correlation in BotDroid dataset

(b) Features correlation in CIC-IDS-2017 dataset

(c) Features correlation in CTU-13 dataset

(d) Features correlation in UNSW-NB15 dataset

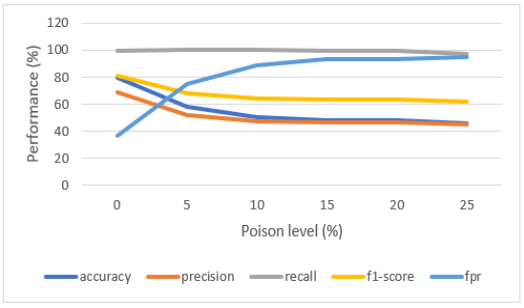
Figure 3.2: Gaussian mixture model visualisation of features relationship in dataset with principal component analysis reduction. Legend shown in (a) applies to all

Fig. 3.2(b) shows a symmetric correlation between some features, including a few with anisotropic covariance, with some overlapping outliers for CIC-IDS-2017. Fig. 3.2(c) although reflecting an uneven distribution, also contains some symmetric features with isotropic covariance. Conclusively, datasets with different characteristics help us in providing a profound behavioural analysis of ML against a data poisoning attack.

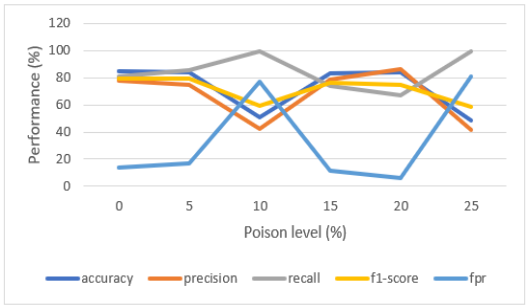
The individual characteristics of baseline ML models are assessed with cybersecurity benchmarked datasets in binary classification data poisoning settings. This study selected four cybersecurity application datasets for analysing data poisoning against security-critical applications. Also, these datasets consist of various features, sizes, and features correlation that help understand the classifiers' behaviour from various perspectives.

3.6 Experimentation Results and Analysis

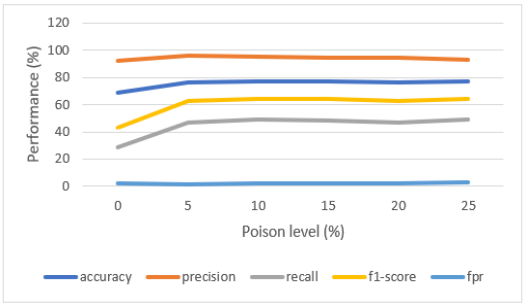
Following this approach and attack method, a deep behavioural analysis of six ML algorithms is conducted, trained with four datasets of varying sizes and structures. The accuracy, precision, recall, f1-score, and false positive rate are plotted against incrementing poisoning levels, as shown in Fig. 3.3 to Fig. 3.8. The experimentation results and analysis are given as follows.



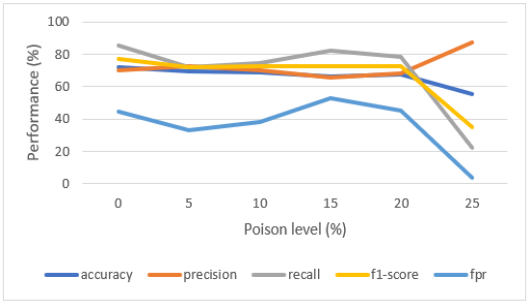
(a) Poisoning support vector machines with BotDroid dataset



(b) Poisoning support vector machines with CIC-IDS-2017 dataset

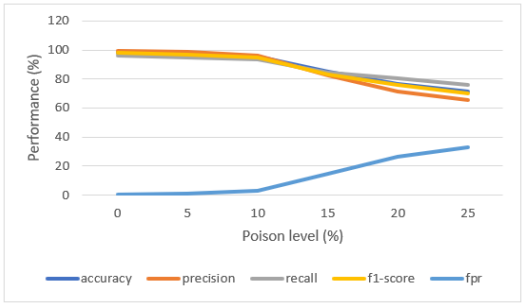


(c) Poisoning support vector machines with CTU-13 dataset

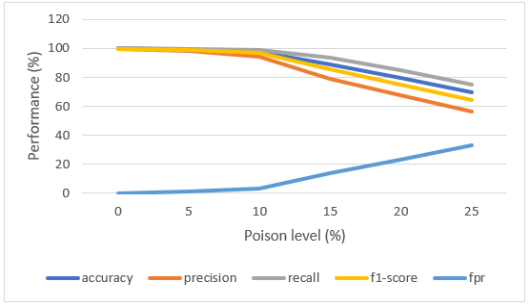


(d) Poisoning support vector machines with UNSW-NB15 dataset

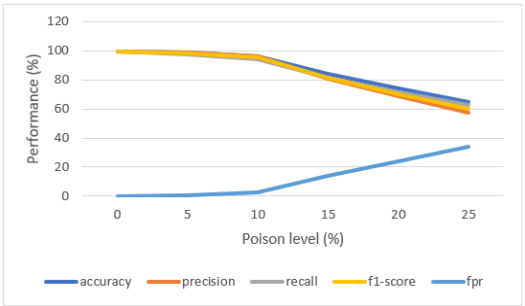
Figure 3.3: Performance analysis of support vector machines with consistent poisoning



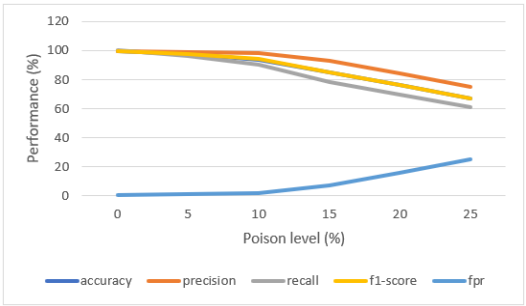
(a) Poisoning random forest with BotDroid dataset



(b) Poisoning random forest with CIC-IDS-2017 dataset



(c) Poisoning random forest with CTU-13 dataset

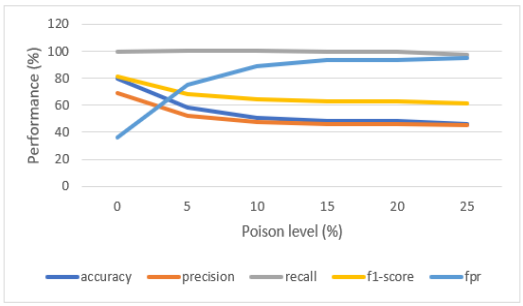


(d) Poisoning random forest with UNSW-NB15 dataset

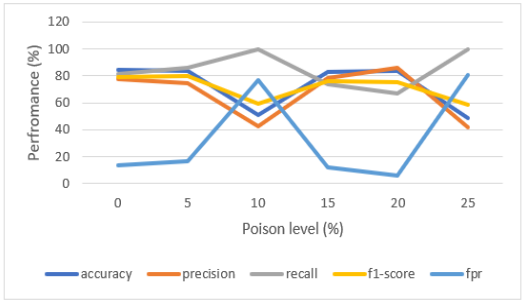
Figure 3.4: Performance analysis of random forest with consistent poisoning

3.6.1 Optimal Poisoning Level

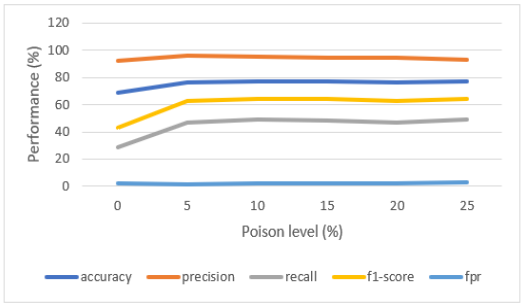
The results of the behavioural analysis first used to examined the impact of various poisoning levels on the underlying decision mechanisms of the models. This examination enable us to identify limits



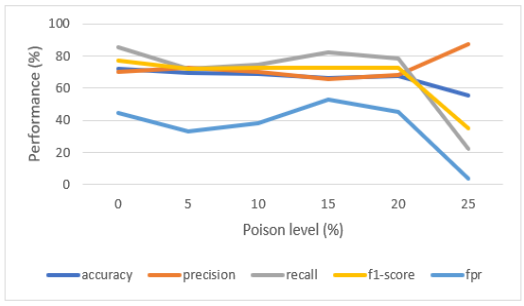
(a) Poisoning gaussian naive bayes with BotDroid dataset



(b) Poisoning gaussian naive bayes with CIC-IDS-2017 dataset

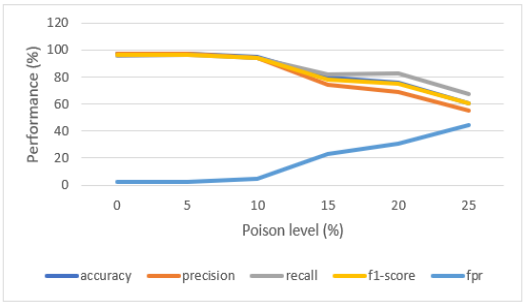


(c) Poisoning gaussian naive bayes with CTU-13 dataset

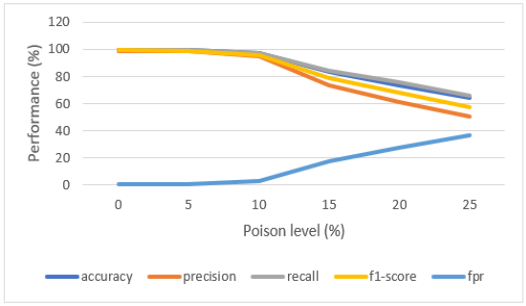


(d) Poisoning gaussian naive bayes with UNSW-NB15 dataset

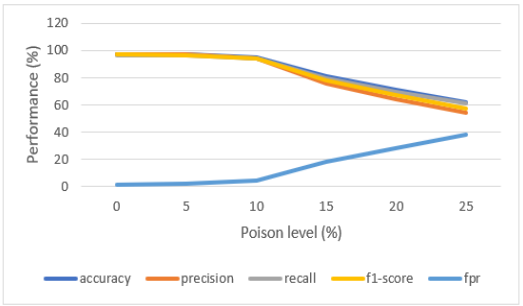
Figure 3.5: Performance analysis of gaussian naive bayes with consistent poisoning



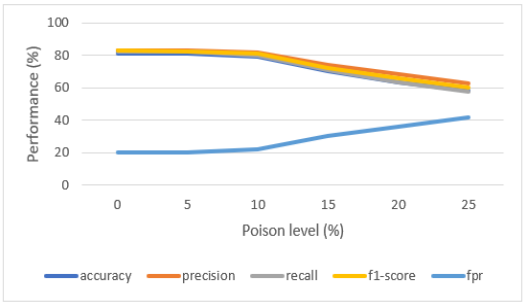
(a) Poisoning k-nearest neighbors with BotDroid dataset



(b) Poisoning k-nearest neighbors with CIC-IDS-2017 dataset



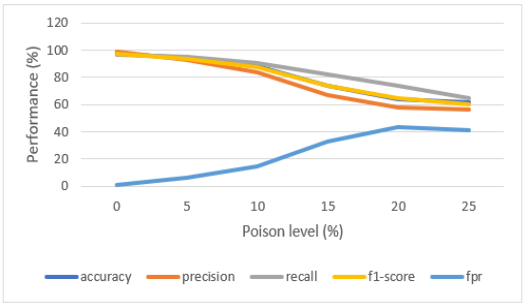
(c) Poisoning k-nearest neighbors with CTU-13 dataset



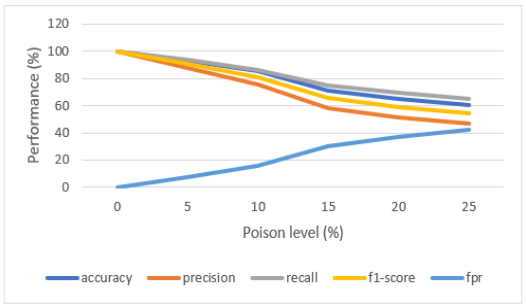
(d) Poisoning k-nearest neighbors with UNSW-NB15 dataset

Figure 3.6: Performance analysis of k-nearest neighbors with consistent poisoning

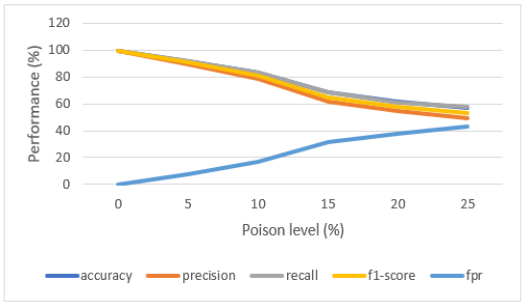
of the optimal poisoning levels. This identification will contribute to further research in enhancing testing scenarios, threat modelling and mitigation solutions against data poisoning attacks. The



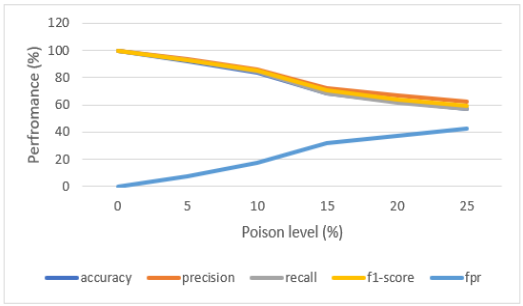
(a) Poisoning decision tree with BotDroid dataset



(b) Poisoning decision tree with CIC-IDS-2017 dataset

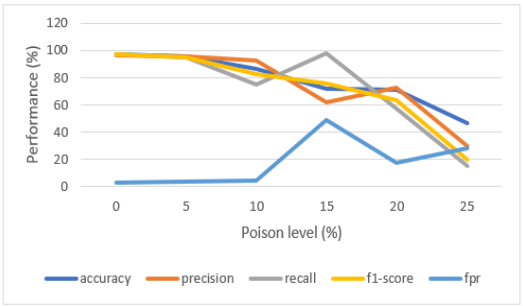


(c) Poisoning decision tree with CTU-13 dataset

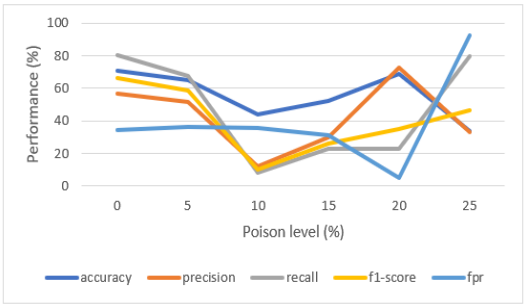


(d) Poisoning decision tree with UNSW-NB15 dataset

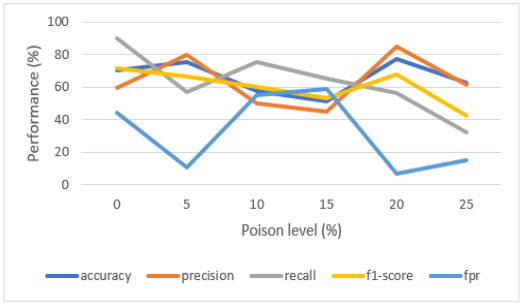
Figure 3.7: Performance analysis of decision tree with consistent poisoning



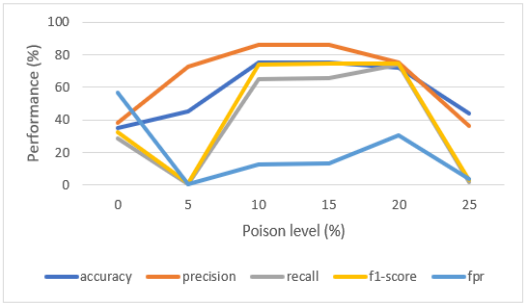
(a) Poisoning perceptron with BotDroid dataset



(b) Poisoning perceptron with CIC-IDS-2017 dataset



(c) Poisoning perceptron with CTU-13 dataset



(d) Poisoning perceptron with UNSW-NB15 dataset

Figure 3.8: Performance analysis of perceptron with consistent poisoning

analysis is given as follows.

Data poisoning of 10-15% is identified as optimal poisoning levels whilst analysing the models' performances. From the results, a sudden drift is visible between $10\% \leq \Delta L \leq 15\%$ of data poisoning

whereas $\Delta L = 5\%$ has put a negligible impact and $\Delta L > 15\%$ make algorithms over-fit. KNN trained with BotDroid dataset has the highest accuracy decrease of 15.48%, as shown in Fig. 3.6. Following the fact that it classifies based on the principle of information gain analysing maximum features similarities from the closest points. The attack randomly poisoned data points, which manipulate the relationship of the underlying features in the KNN model, making its decision boundary rough and irregular at $\Delta L = 15\%$, as shown in Fig. 3.9. However, DT is the most affected algorithm, with an

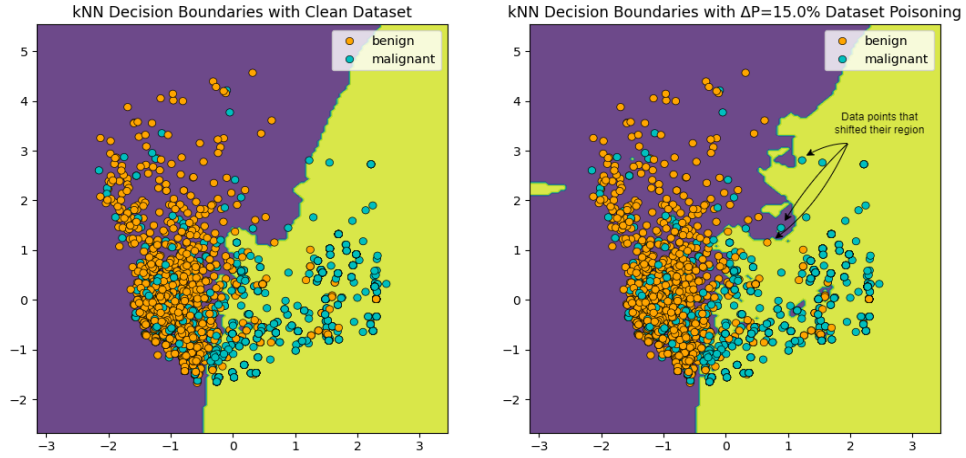


Figure 3.9: Change in decision boundary of k-nearest neighbors with 15% dataset poisoning

average accuracy degradation of 14.42% and degradation score of 4.33% at $\Delta L = 25\%$, irrespective of datasets. Table 3.5 presents the change in feature importance score, for poisoned DT, making important features anomalous, degrading its performance, specifically for less noisy BotDroid dataset.

Table 3.5: Features importance score - Decision tree where $\Delta L = (0\%, 10\%, 15\%)$

	Clean Dataset			Poisoned Dataset $\Delta L = 10\%$			Poisoned Dataset $\Delta L = 15\%$		
Dataset	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3
BotDroid	0.90	0.07	0.02	0.61	0.20	0.18	0.60	0.22	0.18
UNSW-NB15	0.43	0.29	0.22	0.33	0.34	0.32	0.33	0.34	0.32
CTU-13	0.19	0.32	0.49	0.26	0.34	0.40	0.26	0.34	0.39
CIC-IDS-2017	0.40	0.33	0.26	0.36	0.33	0.31	0.36	0.33	0.31

Table 3.6: Features importance score - Support vector machines where $\Delta L = (0\%, 10\%, 15\%)$

	Clean Dataset			Poisoned Dataset $\Delta L = 10\%$			Poisoned Dataset $\Delta L = 15\%$		
Dataset	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3
BotDroid	0.53	0.80	0.23	0.42	0.80	0.40	0.18	0.98	0.03
UNSW-NB15	0.81	0.56	0.10	0.72	0.32	0.61	0.30	0.72	0.61
CTU-13	0.13	0.50	0.85	0.06	0.52	0.84	0.38	0.21	0.89
CIC-IDS-2017	0.07	0.99	0.11	0.53	0.69	0.47	0.45	0.79	0.39

Further analysis interpreted that parametric algorithms including SVM, GNB, and Perceptron, are less affected by data poisoning when dataset features are symmetric and become saturated when $\Delta L \geq$

15%. But these algorithms become sensitive and over-fitted when dataset features are asymmetric and $\Delta L \geq 15\%$. For example, the margin score of SVM is minimally changed from 0.0017 to 0.0013 when ΔL=10% and 15% respectively. Features importance scores of SVM are also minimally affected by poisoning as given in Table 3.6 where features space is reduced following Eq 3.5. So poisoned and cleaned SVM classifiers follow the same importance of features for making classification decisions except for the UNSW-NB-15 dataset for which Feature1 with the highest importance score has become an anomaly at $\Delta L = 15\%$, due to irregular features relations. Explaining further, SVM in Fig. 3.3(b) and GNB in Fig. 3.5(b) shows a high decrease of 35.14% in accuracy with 74% of increment in false positive rate but simultaneously, recall also increases to 33.1% between poisoning rate of 20-25%, highlighting over-fitting of the models. This study also calculated the class probabilities of poisoned GNB to interpret their classification decision as given in Table 3.7 which has a minimal impact on its classes, leading to no change in its decisions. Conclusively, Perceptron losses its accuracy to 50.98%

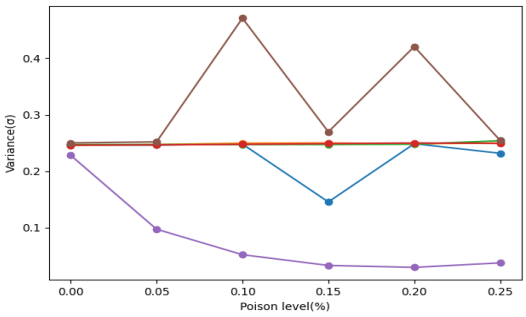
Table 3.7: Analysing class probabilities of gaussian naive bayes with poisoned dataset

Dataset	Clean Dataset		$\Delta L = 10\%$		$\Delta L = 15\%$	
	Class0	Class1	Class0	Class1	Class0	Class1
BotDroid	0.55	0.45	0.54	0.46	0.52	0.48
UNSW-NB15	0.45	0.55	0.46	0.54	0.47	0.53
CTU-13	0.57	0.43	0.56	0.44	0.54	0.46
CIC-IDS-2017	0.64	0.36	0.61	0.39	0.58	0.42

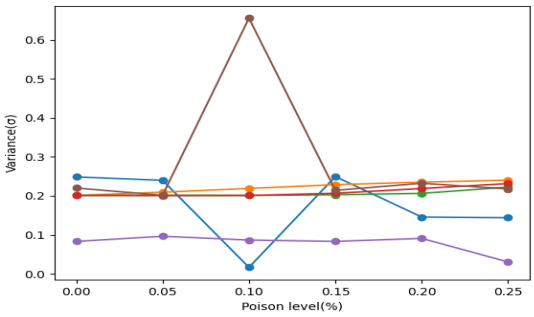
with the BotDroid dataset at $\Delta L = 25\%$ followed by DT, whose accuracy is reduced to 42.74% with the UNSW-NB15 dataset. Also, an interesting relation between poison and dataset noise is revealed where $\Delta L \ \frac{1}{\propto} \ Dataset \ Noise$ and working as a catalyst at $\Delta L = 25\%$. So, ΔL between 10%-15% are identified as optimal and effective poisoning rates, particularly affecting KNN followed by DT however minimally penetrating SVM and GNB. The variance of the trained models is given in Fig. 3.10(a) to 3.10(d). Whereas, the adversarial success rate (ASR) of these poisoned models is given in Tables 3.8 to 3.11.

Table 3.8: Adversarial success rate with BotDroid dataset

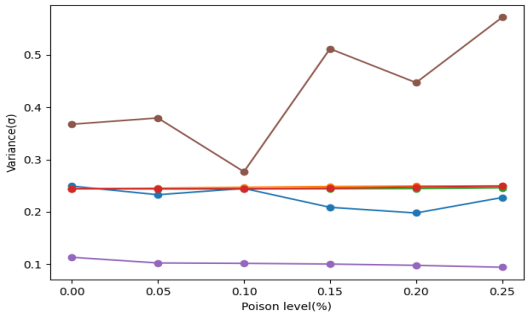
Algorithm	$\Delta L = 5\%$	$\Delta L = 10\%$	$\Delta L = 15\%$	$\Delta L = 20\%$	$\Delta L = 25\%$
SVM	0.2	0.29	0.31	0.31	0.33
DT	0.097	0.13	0.22	0.35	0.33
RF	0.025	0.065	0.15	0.21	0.22
KNN	0.01	0.021	0.16	0.19	0.33
GNB	0.2	0.29	0.31	0.31	0.33
Perceptron	0.055	0.11	0.24	0.25	0.52



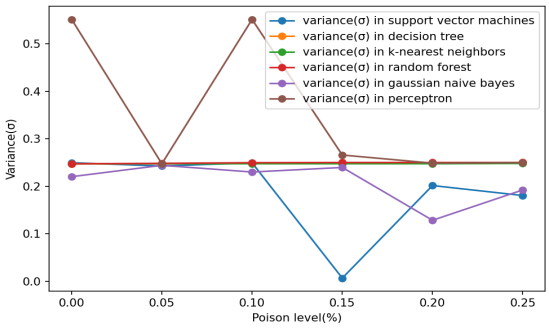
(a) Variance of poisoned models with BotDroid dataset



(b) Variance of poisoned models with CIC-IDS-2017 dataset



(c) Variance of poisoned models with CTU-13 dataset



(d) Variance of poisoned models with UNSW-NB15 dataset

Figure 3.10: Variance analysis of trained models with consistent poisoning. Legend shown in (d) applies to all

Table 3.9: Adversarial success rate with UNSW-NB15 dataset

Algorithm	$\Delta L = 5\%$	$\Delta L = 10\%$	$\Delta L = 15\%$	$\Delta L = 20\%$	$\Delta L = 25\%$
SVM	0.034	0.071	0.10	0.12	0.15
DT	0.12	0.16	0.27	0.35	0.39
RF	0.03	0.06	0.11	0.22	0.28
KNN	0.0074	0.02	0.1	0.17	0.22
GNB	0.034	0.071	0.10	0.12	0.15
Perceptron	0.063	0.4	0.39	0.33	0.08

3.6.2 Vulnerable Parameters

Following the analysis of optimal poisoning levels, this behavioural analysis contributes to analyse vulnerable parameters of individual ML algorithms against data poisoning attacks. Understanding the relationship between data poisoning and individual algorithms help develop strong resilience against such attacks which are independent of attack.

This behavioural analysis study individual factors affecting the selected poisoning attack to study the behaviour of individual algorithms. Overall, DT and KNN are prone to data poisoning. DT comprises a tree data structure following a probabilistic approach to structure decision nodes, whereas this attack initiates a wrong features split that supports misleading classification by manipulating feature importance as shown in Table 3.5. The feature space of datasets has been reduced for simplicity

Table 3.10: Adversarial success rate with CTU-13 dataset

Algorithm	$\Delta L = 5\%$	$\Delta L = 10\%$	$\Delta L = 15\%$	$\Delta L = 20\%$	$\Delta L = 25\%$
SVM	0.06	0.08	0.07	0.09	0.06
DT	0.07	0.16	0.28	0.36	0.39
RF	0.01	0.03	0.18	0.21	0.35
KNN	0.009	0.02	0.16	0.26	0.35
GNB	0.06	0.08	0.07	0.09	0.06
Perceptron	0.07	0.12	0.19	0.09	0.03

Table 3.11: Adversarial success rate with CIC-IDS-2017 dataset

Algorithm	$\Delta L = 5\%$	$\Delta L = 10\%$	$\Delta L = 15\%$	$\Delta L = 20\%$	$\Delta L = 25\%$
SVM	0.03	0.33	0.03	0.007	0.36
DT	0.06	0.14	0.30	0.36	0.39
RF	0.001	0.02	0.08	0.20	0.27
KNN	0.003	0.02	0.18	0.29	0.33
GNB	0.03	0.33	0.03	0.007	0.36
Perceptron	0.03	0.26	0.17	0.02	0.34

with PCA reduction following Eq. 3.5. Also, a direct relation between poisoning and performance degradation has been analysed. KNN is non-parametric and groups data based on feature similarities, making it highly susceptible to poisoning. This attack changes the feature space of the part of the dataset, which disrupts the decision boundary and degrades its performance. Whereas, an inverse relation is identified between the number of neighbors and poisoning level, where increasing k-neighbors normalises the poisoning effects on the KNN classifier, given in Table 3.12.

Table 3.12: Analysing k-neighbors affecting k-nearest neighbors accuracy with $\Delta L = (0, 10, 15, 25)\%$

	Poison Level	k=3	k=5	k=10	k=15
BotDroid	$\Delta L = 0\%$	97.54	96.80	96.92	96.68
	$\Delta L = 10\%$	95.94	96.80	96.31	96.68
	$\Delta L = 15\%$	83.78	90.05	95.57	94.47
	$\Delta L = 25\%$	69.28	70.76	87.22	88.69
CTU-13	$\Delta L = 0\%$	97.80	97.55	96.94	96.55
	$\Delta L = 10\%$	95.06	96.52	96.78	96.50
	$\Delta L = 15\%$	87.13	90.90	94.54	95.95
	$\Delta L = 25\%$	72.0	76.14	83.68	87.52
UNSW-NB15	$\Delta L = 0\%$	81.57	81.17	80.92	80.71
	$\Delta L = 10\%$	79.85	80.36	80.63	80.63
	$\Delta L = 15\%$	74.17	76.46	79.06	80.04
	$\Delta L = 25\%$	64.48	66.85	70.19	74.62
CIC-IDS-2017	$\Delta L = 0\%$	99.61	99.57	99.49	99.42
	$\Delta L = 10\%$	96.21	98.84	99.42	99.41
	$\Delta L = 15\%$	88.70	92.93	96.79	98.91
	$\Delta L = 25\%$	73.49	78.06	87.09	90.43

Because by inverting data labels, decision boundaries are getting blurred resulting in disjunctive classes split and hence more affected with poison. Whereas, SVM and GNB follow parametric learning to develop an optimal hyperplane and the parameters values and then become saturated, which makes it independent of dataset size and robust to poisoned data when $\Delta L \geq 15\%$. In Table 3.13, the

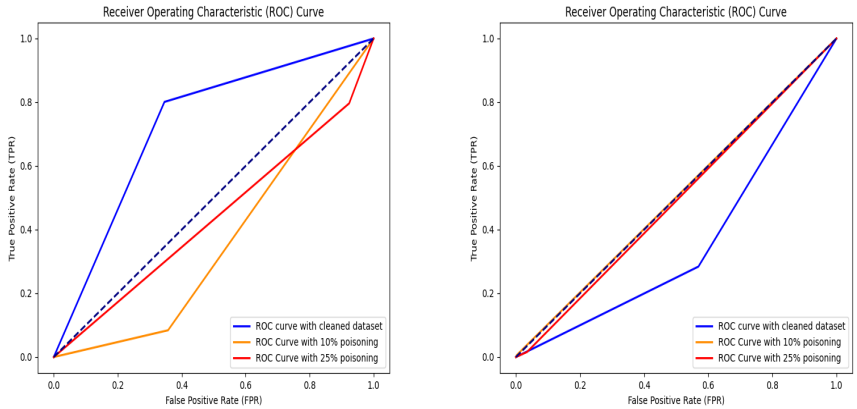
Table 3.13: Analysing support vector machines margin score for different datasets with $\Delta L = (0, 10, 15)\%$

Dataset	$\Delta L = 0\%$	$\Delta L = 10\%$	$\Delta L = 15\%$
BotDroid	0.008	0.00047	0.006
CTU-13	0.000007	0.00002	0.0000003
UNSW-NB15	0.00010	0.00015	0.000019
CIC-IDS-2017	0.00003	0.000015	0.000018

Table 3.14: Analysing one-to-one relation between poison and various parameters of ML algorithms

Algorithm	Algorithmic Parameters	Relation to ΔL
SVM	Margin score	Minimal impact
	Decision boundary	Minimal impact
	Features importance score	Minimal impact
DT	Features importance score	High impact
	Asymmetric features space	High impact
KNN	Decision boundary	High impact
	k-neighbors	Inverse impact
GNB	Decision boundary	Minimal impact
	Class probabilities	Minimal impact
RF	No. of trees	Inverse impact
Perceptron	Weights	High impact

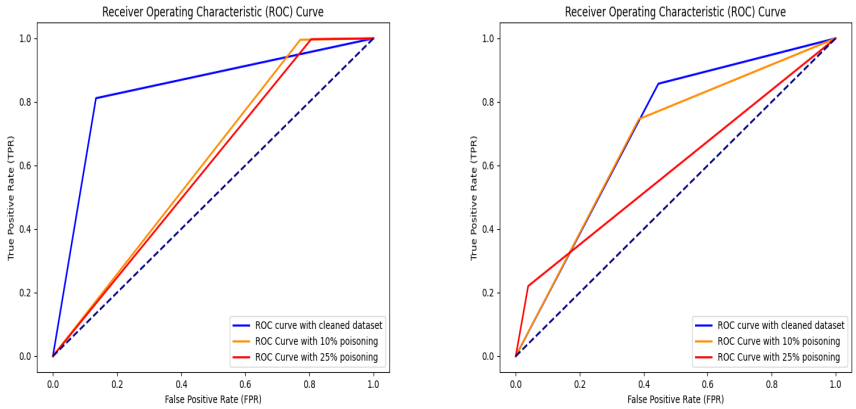
minimum margin scores of cleaned and poisoned SVM classifiers have been calculated to analyse the change in their decision boundary, which shows minimal change with little to no impact on their decision formulation. Also, looking at Fig. 3.3(a) and Fig. 3.3(c), SVM classifiers performance becomes saturated at $\Delta L \geq 15\%$ for BotDroid and CTU-13 datasets. But for UNSW-NB15 and CIC-IDS-2017 which comprises anisotropic features, SVM becomes sensitive and over-fit when $\Delta L \geq 15\%$, leading to inappropriate and continuous fluctuations in their learning parameters as shown in Fig. 3.3(b) and Fig. 3.3(d). A very similar behaviour is visible for GNB and Perceptron can be seen in Fig. 3.5 and Fig. 3.8.



(a) Poisoning perceptron with CIC-IDS-2017 dataset

(b) Poisoning perceptron with UNSW-NB15 dataset

Figure 3.11: ROC curve of perceptron with consistent poisoning



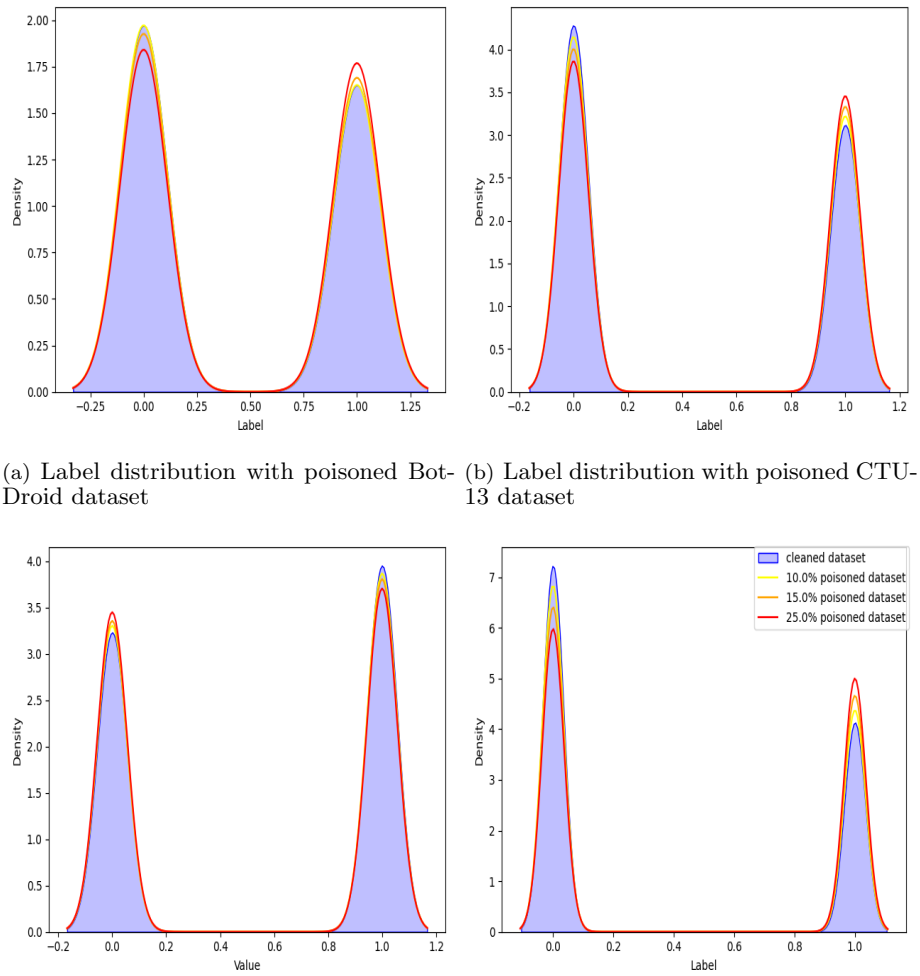
(a) Poisoning gaussian naive bayes with CIC-IDS-2017 dataset

(b) Poisoning gaussian naive bayes with UNSW-NB15 dataset

Figure 3.12: ROC curve of gaussian naive bayes with consistent poisoning

3.6.3 Imbalanced Data Distribution

The behavioural analysis also determine the link between various structures of the datasets and the impact of data poisoning attacks on the poisoned model. This understanding allows strengthen threat models and testing and analysing ML against data poisoning attacks. Data poisoning is highly impacted on DT followed by KNN, irrespective of dataset noise and imbalanced data distribution as shown in Fig. 3.7 and Fig. 3.6, respectively. Whereas, interpreting Fig. 3.4, Random Forest (RF) is found to be most effective and resilient against data poisoning attacks although a non-parametric algorithm. Because RF makes decisions calculating the mean from all of its trees, which normalises the data poisoning effects. However, particularly for intrusion detection, KNN is immune to poison specifically at 10% although a high drift in performance can be seen at 25% dataset poisoning. Fig. 3.13 provides an overview of the change in dataset distribution when labels are attacked with the label-flipping attack at different poisoning levels. Looking at classification performances in Fig. 3.11(a) and Fig. 3.11(b), dataset noise and asymmetric features correlating with poison making Perceptron sensitive to classify intrusions resulting in an unstable classifier. For parametric algorithms i-e. SVM, GNB, and Perceptron, poison in an erroneous dataset works as a catalyst for performance disruption. For example, Fig. 3.12(a), Fig. 3.12(b) and Fig. 3.11(a) shows a significant decrease in TPR concurrently increasing FPR with 10% poisoned dataset for GNB and Perceptron, respectively. From this analysis, Table 3.14 provides a relative impact of ΔL on individual parameters of ML algorithms. Implications of the behavioural analysis and its future directions are given in Section 3.7.



(a) Label distribution with poisoned BotDroid dataset

(b) Label distribution with poisoned CTU-13 dataset

(c) Label distribution with poisoned UNSW-NB15 dataset

(d) Label distribution with poisoned CIC-IDS-2017 dataset

Figure 3.13: Analysing change in dataset distribution D_{dis} with label poisoning using *kernel density estimation* function at an incremental poison levels ΔL=(0%, 10%, 15%, 25%). Legend shown in (d) applies to all

3.7 Implications of Behavioural Analysis

This behavioural analysis helped strengthen the explainability of the model by understanding the change in behaviour of the model when trained with a poisoned dataset. The shifts in decision boundaries of the models are analysed, the change in their individual characteristics with varying poisoning levels and the impact of various features correlations in poisoning models. The implications of the deep behavioural analysis are provided as follows:

3.7.1 Improving Mitigations

Extending Table 3.3, data sanitisation and adversarial training are the two most effective techniques to safeguard against data poisoning attacks. Adversarial training is developed with the gradient of the data point which is effective for neural networks only. It can not enhance the security of

baseline ML models including SVM, DT, RF, GNB, and KNN. However, data sanitisation filters the dataset's ambiguity to clean the classifier's training by removing doubtful dataset features or outliers. These sanitisation techniques are limited to be developed on the dataset features only. None of the techniques focus on the classifiers' inherent parameters and their development mechanisms. Furthermore, Cinà et al. (2023) highlighted the limitations of existing mitigation techniques and the importance of the generalisability of ML models. Another research study (Paracha et al., 2024b) experimentally proved the limitations of adversarial training in security baseline models. Considering these limitations, it is important to understand the underlying decision model development and the impact of poisoned data points on it.

To enhance the security of ML models and their generalisability against data poisoning, it is crucial to study the behaviour of the inherent characteristics of the models. The results highlighted potentially vulnerable parameters of individual algorithms and their susceptibility to poisoning at various levels. Also, highlighting the most pervasive poisoning levels allows the development of potential security solutions in this regard. Including this behavioural analysis in data sanitisation and in setting model parameters will better generalise the model. Also, this behavioural analysis will help strengthen the explainability of the model's decision. Considering future research directions, the behavioural analysis will be highly effective in developing adversarial training techniques for baseline ML models other than perceptron. Conclusively, the results of this chapter are discussed in Section 3.8.

3.8 Discussion and Limitations

There are various solutions proposed in existing literature to mitigate data poisoning attacks such as data sanitisation (Chan et al., 2018a), data filtration based provenance features to remove poisoned data points (Baracaldo et al., 2017), or mitigating poisoning impact by detecting outliers (Paudice et al., 2018a). Almost every solutions designed mechanism to improve the performance of the given model though none has provided a mechanism to understand hopw poisoned data points are changing the underlying classification mechanisms of the model. To develop strong and resilience mitigation mechanisms, it is important to understand the changing behaviour and capabilities of ML models with various types of data poisoning attacks. The deep behavioural analysis provides an in-depth understanding of how poisoned training can change the underlying decision mechanisms of ML models. This research provides a further detailed synopsis of technical uncertainties in the model's decisions created with a poisoned dataset to help strengthen mitigation solutions. As the model's decision mechanisms are underexplored in existing solutions, ML models are still susceptible to new data poisoning attacks that can breach existing mitigations such as data sanitisation or adversarial training, as highlighted in Section 3.7.

To understand the changing behaviour of ML models trained with poisoned datasets, a deep be-

havioural analysis is conducted to analyse the impact of poisoned data on the decision mechanisms of the models. This analysed the change in the decision boundaries of the model, identified how the poisoned data is impacting individual characteristics of the model and what are the optimal poisoning levels up to which the poison has a high impact and is difficult to detect. This study also analysed the relationship between data poisoning attacks and various structures of the datasets to understand the impact of poison with varying datasets.

The results highlight certain affected factors with label poisoning, analysing individual ML algorithms. Such as segregating nodes hierarchy development in DT and RF, misplacing data points in KNN and SVM, and probability misinterpretation in KNN and GNB. Overall RF is resilient to data poisoning because it may create more hierarchical nodes within its trees with poisoned data points, averaging their prediction probabilities dilutes misclassification. Although DT is the most affected algorithm because flipped labels make it difficult to disjoint nodes considering features. Also, the results showed that a 10%-15% poisoned dataset is more impacting whereas, after 25% poisoning, the sensitivity of a model is getting increased.

These results highlight that noise in the dataset works as a catalyst for data poisoning. The more noisy the dataset, the more adverse impact can be achieved in performance degradation against classification results because it filters poison (intentional noise) from dataset noise. Also, non-linear features in the dataset help poison to better penetrate the classifiers. Whereas, cleaning datasets and feature engineering can help mitigate data poisoning in this scenario.

Based on the results and implications of the deep behavioural analysis, this study has been extended to further understand the adversarial impact of data poisoning attacks with novel data poisoning attacks. A new data poisoning attack has been proposed, which exploits a significant characteristic of ML models called outliers. This deep behavioural analysis helped define a threat model with limited adversarial capabilities to experiment the outlier poisoning attack in Chapter 4 then extended it to identify the impact of the attack on security-sensitive applications.

This thesis has conducted a deep behavioural analysis of supervised ML models; however, this analysis can be applied to assess other ML and DL models where practitioners need to analyse the resilience of ML against data poisoning attacks. It is also helpful in auditing training time mitigations against data poisoning attacks.

This behavioural analysis is limited to the practical implication and analysis of supervised classification algorithms for binary classifiers. This limitation helps to provide a detailed and thorough technical study of supervised ML classifiers and their parameters that are affected by poisoning. Unsupervised ML can not be covered in this research as this study focused on label poisoning and has contributed to analysing decision boundaries of classifiers and resultant metrics, including accuracy, precision, f1-score, and recall. For the in-depth analysis, the variance and ASR of the baseline models

are also examined and compared with the benign models to understand the disrupted behaviours. This chapter only focused on the baseline ML models, which are the foundation of complex deep neural networks and advanced models. Regression algorithms are also out of the scope of this study. However, this behavioural analysis can be extended to multiclass classifiers, which is a limitation of this chapter. Furthermore, it is important to conduct this behavioural analysis in security-sensitive applications with real-world case studies for a rigorous analysis. Chapter 4 addresses this limitation and conducts a detailed study of analysing data poisoning impact in multiclass classification models.

3.9 Summary

This chapter highlighted an important aspect of understanding the impact of data poisoning attacks on the decision mechanisms of the ML model. For achieving this purpose, a deep behvaioural analysis of six classification algorithms is conducted to analyse poisoning impact on the classification behaviour of these models and their characteristics. The results of this chapter strengthen the answer to RQ 1 by providing practical understanding to analyse security vulnerabilities of ML exploited by data poisoning attacks. This chapter extended the answer to RQ 1 by experimenting data poisoning attacks in realistic attack settings by providing minimal knowledge of the targeted system to the adversary. This chapter also addressed RQ 2 by highlighting potential risks of data poisoning to ML applications with limited adversarial capabilities and the impact of poisoning in such scenarios. The deep behavioural analysis identified the optimal poisoning levels that have a high impact on the model and are difficult to detect. Furthermore, the relationships between varying data structures in the dataset, the correlation of features in the dataset and the impact of data poisoning are also determined. The technical insights are provided that 10%-15% poisoning to the dataset is optimal whereas less than 10% has minimal impact and more than 25% introduces sensitivity and 50% of data poisoning leads to overfitting. Different behaviours towards performance degradation of ML algorithms are also highlighted and put forward the facts that technically interpret why DT is the most affected algorithm against poisoning whereas RF is resilient to it. The impact of data poisoning when the training dataset is imbalanced and its features are anisotropic is also examined, and is concluded that anisotropic or asymmetric features serve as catalysts to data poisoning between 10%-20% of poisoning level, whereas they increased the sensitivity of the models, specifically of neural networks. Chapter 4 continues to answer RQ 2, to assess data poisoning impact, severity and consequences in security-sensitive applications and extend the behavioural analysis in such applications. These insights will help develop strong mitigation mechanisms and security by design solutions to secure ML models from existing data poisoning attacks as well as from the evolving threats and new data poisoning attacks.

Chapter 4. Further Exploring Novel Outlier-Oriented Poisoning (OOP) Attack in Multiclass Machine Learning

4.1 Introduction

This chapter proposes an Outlier-Oriented Poisoning (OOP) attack to analyse the implications of new data poisoning attacks under limited adversarial capabilities. The OOP attack perturbs a common characteristic of ML called outliers to manipulate the feature space of multiclass models to analyse changes in learning dynamics. The previous chapter conducted behavioural analysis of binary classifiers, which is now extended to multiclass classifiers in this chapter. This chapter completes the answer to RQ 2 to analyse the impact and consequences of data poisoning attacks in real-world applications. With the development of the OOP attack, the objective 2 of designing and developing new data poisoning attacks is achieved. This chapter also begins to answer RQ 3 by analysing the limitations of existing mitigation solutions and exploiting adversarially trained models with OOP poisoning.

Prior research, such as Baker et al. (2024), Das et al. (2024), Tian et al. (2024), mostly focuses on poisoning availability and integrity attacks against DL and binary classification models. However, data poisoning attacks against traditional multiclass classifiers have been explored to a limited extent. In general, there are three approaches to data poisoning attacks. Firstly, label poisoning Liu et al. (2022), Shahid et al. (2022), Aryal et al. (2022) perturbs the labels of the dataset to manipulate training datasets. Secondly, clean-label poisoning Zhu et al. (2019a), Aghakhani et al. (2021) is typically generated by solving one or more optimisation problems, such as bi-level optimisation Ma et al. (2021a), Russo and Proutiere (2021) or gradient descent optimisation (Sanchez Vicarte et al., 2020) to craft and inject poisoned data points into the model. Thirdly, the existing dataset can be manipulated with feature perturbation. However, multiclass poisoning attacks are explored mostly against DL models.

MetaPoison (Huang et al., 2020) solves a bi-level optimisation problem with meta-learning to craft poison against neural networks. It is practically implemented against the Google Cloud AutoML API and extended for experimentation on multiclass neural networks. While MetaPoison demonstrates effective poisoning in neural networks, its reliance on meta-learning makes it less generalisable to traditional ML classifiers. Subpopulation data poisoning (Jagielski et al., 2021) injects a perturbed cluster into the dataset. Its efficacy is highlighted with a variety of neural networks with multiple datasets. Another research study (Muñoz-González et al., 2017) proposed a gradient poisoning at-

tack and extended their experimentation from binary classification to multiclass classification. This research focuses on experiments with the poisoning availability of the targeted subclass in neural networks.

Furthermore, limited work is provided in the literature that analysed the poisoning impact on real-world applications, developed with traditional ML classifiers. Such limitations are highlighted in Section 2.4 in Chapter 2. To address these limitations, this chapter conducted two case studies of analysing the poisoning impact with a novel data poisoning attack in real-world applications. The experimentation results of these case studies provide us with an understanding of how data poisoning impacts such applications and their consequences and real-life impacts.

Given the limitations, this chapter assesses the effectiveness of novel data poisoning attacks in multiclass settings to inform the development of improved mitigation strategies. While most existing research on multiclass poisoning focuses on neural networks, this study examines six supervised machine learning algorithms: SVM, DT, RF, KNN, GNB, and Neural Networks using a Multilayer Perceptron (MLP). These algorithms represent a comprehensive baseline of classification methods. Poisoning levels are set ranging from 5% to 25% in 5% increments to assess model behaviour under varying attack intensities, following the approach in (Paracha et al., 2024a). The analysis identifies key parameters of each algorithm that are sensitive to poisoning, determines optimal poisoning rates, and quantifies performance degradation in terms of accuracy and model-specific characteristics. The main contributions of this chapter are outlined below.

- A new OOP attack is developed, as a novel label poisoning attack to introduce misclassification in multiclass ML. This attack is formulated based on the label perturbation of the most distant data points from the decision boundaries of the multiclass classifier.
- A thorough behavioural analysis of multiclass classifiers is performed, analysing the correlation between different poisoning levels and the performance degradation of classifiers.
- The OOP attack is implemented against real-world applications, analysing the impact of poisoning in such systems. Following the poisoning of these applications highlighted the limitations of existing security techniques in mitigating data poisoning attacks, particularly focusing on traditional multiclass models.

4.2 Related Work

Existing literature highlights a significant number of poisoning attacks that harm the integrity and availability of ML models. Such as Zhao and Lao (2022b) proposed a class-oriented poisoning attack to introduce misclassification for a targeted dataset class. Similarly, Carlini and Terzis (2021) highlighted a security threat of poisoning and backdoor attacks against multiclass ML with only 0.0001%

of data poisoning. They have introduced misclassification during model training to overfit, which increases errors during testing. Alarab and Prakoonwit (2023) have developed a poisoning attack using a Monte-Carlo algorithm against deep learning multiclass models to analyse their classification uncertainties. Pantelakis et al. (2023) have evaluated the performance disruption of IoT-based multiclass models against JSMA, FGSM, and DeepFool attacks, highlighting the effectiveness of these attacks in poisoning multiclass models. Other prominent poisoning attacks are Suya et al. (2021), Mayerhofer and Mayer (2022), Saha et al. (2020). Table 4.1 highlights existing studies that experimented with data poisoning attacks with various DL and ML models. It is crucial to understand the behaviour of the underlying baseline models and their sensitivity against poisoning attacks. This investigation helps us better mitigate poisoning, not only focusing on their performance but also on their underlying classification mechanisms. Following the discussed attacks, this chapter focuses on manipulating outliers to disrupt the feature spaces of the multiclass models, discussed in Section 4.5.1. This study has shown the efficacy and effectiveness of the attack on six ML algorithms at various poisoning levels.

Table 4.1: Analysing existing studies against the behavioural analysis with outlier-oriented poisoning attack

Research paper	ML model	Dataset	Effective poisoning level	Model degradation and variance at:	
				Various poisoning levels	Various classes
Zhao and Lao (2022b)	LeNet-5, Vgg-9, ResNet-50	MNIST, CIFAR-10, ImageNet	✗	✗	✗
Carlini and Terzis (2021)	ResNet-50, Transformer language model	Conceptual Captions	✗	✗	✗
Alarab and Prakoonwit (2023)	LEConv, CNN	Cora, MNIST	✗	✗	✗
Pantelakis et al. (2023)	DT, RF, KNN, MLP	IoTID20	✗	✗	✗
OOP Attack	SVM, DT, RF, GNB, KNN, MLP	IRIS, MNIST, ISIC	✓	✓	✓

Limited techniques are provided in the literature that are generalisable and effective in mitigating poisoning effects against traditional multiclass models. McCarthy et al. (2023) proposed a hierarchical learning mechanism to secure the network traffic attack classification model. Hossain and Oates (2024) developed a solution to detect backdoor poison in deep neural networks by extracting, relabeling, and classifying features with a tensor decomposition method. They have experimented with their mitigation solution on MNIST, CIFAR-10, and TrojAI datasets. Curie (Laishram and Phoha, 2016) is the method proposed to mitigate poisoning attacks against SVM. They introduced an additional feature dimension to map labels with features that help segregate poisoned data points with flipped labels from the normal data points.

Melacci et al. (2021) have experimented with the effectiveness of incorporating domain knowledge into the neural networks, in detecting adversarial data points added in the model training. They have experimented with their solution on neural networks with CIFAR-100, ANIMALS, and PASCAL-Part datasets. None of the above solutions is generalisable to both ML and DL models. To strengthen mitigation solutions, it is important to understand how poisoning affects the underlying classification behaviours of these models. Therefore, this chapter assesses multiclass classifiers under poisoning attacks to identify their key characteristics affected by poison and reveal their relationships with injected poison.

4.3 Threat Model

A grey-box threat model is developed to evaluate the OOP attack in real-world applications. In this threat model, the goals of poisoning multiclass ML models are defined under the limited knowledge and capabilities of the adversary. This study aims to design the OOP attack that manipulates the decision boundaries of multiclass classifiers and to evaluate their susceptibility to data poisoning attacks. The analysis is further extended to assess the effects of this novel poisoning technique in real-world applications that incorporate adversarial training defense. The limitations of existing adversarial training are described in Chapter 2, noting that these approaches have primarily been applied to DL models and require evaluation on traditional classification models. The targeted applications in this research include an image classification system, a skin cancer diagnostics system, and an industrial 5G network system. The following metrics are used for comprehensive evaluation and analysis.

- **Baseline metrics**: Accuracy, precision, recall, f1-score, and false positive rate
- **Robustness metrics**: Model variance and adversarial success rate
- **Feature-sensitivity metrics**: Feature importance score
- **Algorithm-specific metrics**: KNN k neighbors, SVM margin, and GNB class probabilities

4.3.1 Attack Surface and Goals

The attack surface considered is the training pipeline, where the adversary aims to corrupt the feature space of the model by injecting perturbed outliers. The goal is to analyse the susceptible characteristics and impact of novel data poisoning against a benign and adversarially trained model, and the impact of data poisoning in real-world applications. By disturbing the training data, the adversary seeks to induce systematic misclassification and degrade model reliability in practical deployments.

4.3.2 Attacker Knowledge

A realistic scenario is developed to analyse model susceptibility and impact against data poisoning attacks. This scenario considers a grey-box knowledge attack, where only the dataset and algorithm family are known to the adversary. The model is inaccessible with no known configurations or parameters. This assumption is considered following the training of ML models with publicly available datasets, which can be accessible to the adversary.

4.3.3 Attacker Capability

This threat model follows limited adversarial capabilities where the target model is inaccessible to the adversary. In this way, the adversary develops surrogate models to implement data poisoning attacks and manipulate training datasets. To implement the threat model, three scenarios were developed to study data poisoning against multiclass models in supervised ML. The details of these scenarios are given in Section 4.3.5 and Section 4.3.6.

4.3.4 Outlier-Oriented Poisoning Attack

This study adopts a practical approach to design the attack strategy under the assumption that the adversary A_{dv} does not know the underlying settings of the targeted model M and the dataset distribution. The OOP attack mechanism is illustrated in Fig. 4.1. The OOP attack develops surrogate models to identify and perturb the most distant data points in each class of the model. After identification, the OOP attack changes the class of the selected data points to manipulate the feature surface of the classes in the ML model to misalign decision boundaries and decrease classification performance.

Three datasets—IRIS, MNIST, and ISIC—each with three, ten, and four classes, respectively, were used to assess this attack. The OOP attack is formulated as an end-to-end poisoned training setting, where only the datasets are known to the adversary. Surrogate models M_s are developed and trained to craft poisoned data points x' with perturbed class labels l' at different poisoning levels ΔL. Poisoning levels ranged from 5% to 25% at a scale of 5%. The OOP attack initiates multiclass poisoning by calculating the data points at maximum distance from hyperplane and changing their classes l'.

With this outlier perturbation, benign feature spaces were manipulated by misplacing outliers in them. The goals are to assess the performance degradation of individual algorithms and to analyse the behaviour of multiclass models under the OOP attack. For the dataset manipulation with the OOP attack, let $x \in D$ be the distanced data point that is perturbed by manipulating its label l to increase the loss of the model $\mathcal{L}$ with change γ in multiclass decision boundaries b_c following model

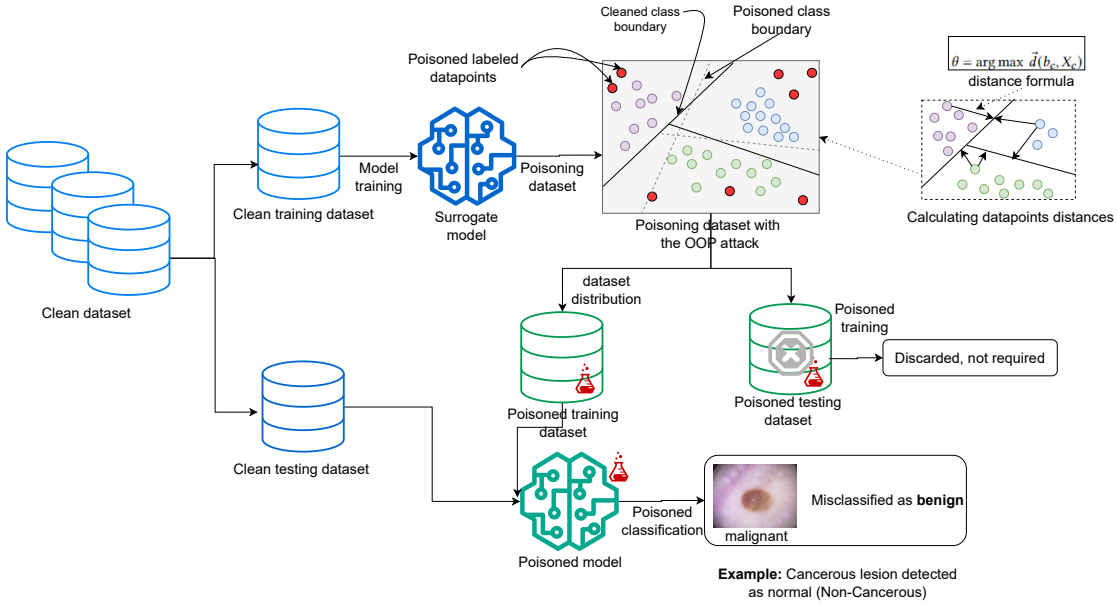


Figure 4.1: The architectural overview of Outlier-oriented poisoning attack

training $\mathcal{T}$ as:

$$\mathcal{L}(M, D') = \gamma = \Delta b_c(\mathcal{T}(M, D')) \tag{4.1}$$

4.3.5 OOP Attack Against Adversarially Trained Skin Cancer Diagnostics

The OOP attack is extended under the assumptions of a weak adversary A_{dv} to deceive an adversarially trained skin cancer diagnostic application. The application is trained using the public ProveAI version of the ISIC dataset, which contains four classes. The OOP attack poisons this public dataset with $\Delta L = 10\%$, which is then used to train the diagnostic application.

The diagnostic is secured with adversarial training and implemented with the FGSM attack. Further details of the training approach are given in Section 4.7.1. This attack scenario is novel in two aspects. Firstly, the OOP poisoning against secured traditional models is analysed in non-complex settings. Previously, the adversarial training is implemented and tested primarily on neural networks. Secondly, it analyses the effects of dataset poisoning on the diagnostic model, implemented with three ML algorithms in multiclass settings.

Consider an ML model $M(D;\theta)$ with a classification skin lesion dataset D and θ as configuration parameters. This model $M(D;\theta)$ is trained to make it resilient against data poisoning attacks, using 75% dataset for training and 25% for testing. Mathematical notation of the secured training of a diagnostic application is given in Eq. 4.2.

$$M_{adv}(D;\theta) = minMSE_M[max\mathcal{L}(f(x+\delta), l)] \tag{4.2}$$

Table 4.2: Configurations of the models

S.No.	Model	Model Configurations
1	SVM	decision_function_shape='ovo', kernel='poly', degree=3, C=1
2	RF	criterian='log_loss', n_estimators=9, max_depth=4
3	MLP	penalty='elasticnet', alpha=0.0001, max_iter=1000

where $M_{adv}(D;\theta)$ is the adversarially trained model, MSE_M is the mean square error of the model under testing, $\mathcal{L}$ is the loss of the model, δ is the FGSM-based perturbation added to individual data sample x and l is the original label of the data sample x. The OOP attack generates poisoned data D' to corrupt the model training $\mathcal{T}$ of $M_{adv}(D';\theta)$. The mathematical notation of poisoned training is given in Eq. 4.3.

$$M'_{adv} = \mathcal{T}(M(D';\theta)) \tag{4.3}$$

where, $\mathcal{T}$ is the training process of the model and D' is the poisoned dataset. The model configurations θ for all the algorithms are given in Table 4.13.

4.3.6 OOP Attack Against Industrial 5G Networks

The adversary A_{dv} is considered as weak, meaning the network configurations and parameters are unknown to it. ML models are developed to manage the spectrum sharing between 5G and 802.11ax networks using a data-driven approach. By targeting the training dataset D of these ML models, the spectrum sharing process is poisoned. The OOP attack is implemented to poison the spectrum dataset. The OOP attack is implemented with poisoning levels $\Delta L = 10\%, 15\% and 20\%$. Following this threat model, the impact of data poisoning in a time-sensitive spectrum sharing application is analysed as a diverse industrial use case.

4.4 Case Study Selection Rationale

The selection of skin cancer diagnostic and industrial 5G private network applications aligns with the objective to assess the impact of data poisoning in real-world applications. These case studies were chosen based on the criticality of decision outcomes, susceptibility to poisoned training and availability of datasets and deployment environments to enable practical assessment. Following the research gaps of the thesis, given in Section 2.4 in Chapter 2, these assessments address the limitations of assessing data poisoning in traditional multiclass ML.

The adversarially trained skin cancer diagnostic application utilises the **SkinCheck** application, enabling assessment of the OOP attack in a practical context. This study specifically addresses the research gap of evaluating the effectiveness of adversarial training in securing traditional ML models.

This case study highlights the risks associated with data poisoning, particularly its impact on missed diagnoses of actual skin cancer cases.

The second case study examines an industrial 5G network in collaboration with the 5G networks team at **Birmingham City University**. This network, developed for a material recycling facility, is used to optimise network resources and assess vulnerability to OOP attacks. While data poisoning has been studied in the image domain, it is still underexplored in networking applications, particularly in Industry 4.0, associated with multiclass ML. To respond to research gaps, this study leverages the analysis of risks in high-yield production environments and evaluates how data poisoning influences latency, throughput, and contention windows in demanding industrial processes.

4.5 Approach to OOP Attack

This section details the approach of designing and evaluating the OOP attack. It first discussed the attack method, followed by describing datasets, attack settings and evaluation metrics.

4.5.1 Attack Method

Instinctively, the training dataset is poisoned with the OOP attack to disrupt ML performance at validation. The OOP attack algorithm is given in Alg. 2. This attack manipulates the class labels of the most distant data points from their class boundaries, manipulating the feature space of the classes and misleading classification predictions. To perturb the training dataset, the adversary develops the surrogate model, as no access to the targeted model is provided. The OOP attack is implemented using six surrogate multiclass models: SVM, RF, DT, GNB, KNN and MLP, following default configurations for non-linear datasets. The SVM is developed using a polynomial kernel with the default degree and regularization parameters of 3, and the MLP is trained using the RELU activation. However, the other four models support multiclass non-linear classification in default configurations. The class probability is the model confidence in classifying an instance against each class in GNB model. The surrogate models are developed to calculate outliers following the Alg. 3. The Alg. 4 distinctly calculates the farthest outliers from decision boundaries for the given ML model to maximise classification errors. To identify decision boundaries for individual models, the algorithm is described in Alg. 4.

4.5.2 Experimental Datasets

The OOP attack is implemented on multiclass classifiers using three multiclass datasets: IRIS, MNIST, and ISIC. The reason for selecting these datasets is their varying sizes and structures, where IRIS is a small dataset, MNIST is very large, and ISIC is medium-sized with varying feature correlation and number of classes. The dataset characteristics are provided in Table 4.3. By employing

Algorithm 2 OOP Poisoned Model Generation

Datasets: IRIS, MNIST, ISIC, datasets
Inputs: Training Dataset D, Poison level ΔL
Outputs: Poisoned Model M'_c
Initialise: $D \leftarrow$ Training dataset
$\Delta L \leftarrow$ Poisoning level $\in [0\%, 5\%, 10\%, 15\%, 20\%, 25\%]$
$M_{conf} \leftarrow [SVM, DT, RF, GNB, KNN, MLP]$
$D' \leftarrow$ Poisoned dataset = []
$D_{dist} \leftarrow$ subset of Training dataset
while $len(D') \leq \Delta L$ **do**
  Set index i = max(D_{dist})
  Set data point $x_i = D[i]$
  if x_i not in D' **then**
    Set l = Class(x_i)
    Update $l = l_i$; where $l_i \neq c$
    Update Class(x') = l_i
  end if
  $D' \leftarrow (x', l')$
  Set $D_{dist}[i] = 0$
end while
D'_{ctrain} = split(D', 0.75)
M' = train(M_{conf}, D'_{train})
return M'

Algorithm 3 Surrogate Model Development

Datasets: IRIS, MNIST, ISIC datasets
Inputs: Training Dataset D, Model Configuration M_{conf}
Outputs: Surrogate Trained Model M_{surr}
Initialise: $D \leftarrow$ Training dataset
$M_{conf} \leftarrow$ [Support Vector Machines (SVM) = Config(kernel='poly', degree of polynomial function=3, regularisation parameter=3),
Decision Tree(DT) = Config(criterion='gini', splitter='best')
Random Forest(RF) = Config(n_estimators=3, criterion='gini')
K-Nearest Neighbors (kNN) = Config(n_neighbors=5, weights='uniform')
Gaussian Naive Bayes (GNB) = Config(var_smoothing=$1 * 10^{-9}$)
Multilayer Perceptron (MLP) = Config(activation='relu', solver='adam')]

for config in M_{conf} **do**
  $M_{surr}(config) = initialise(M_{surr}, config)$
  $M_{surr}(config) = training(M_{surr}(config), D)$
end for
return $M_{surr}(config)$

the OOP attack across datasets with differing structures, a comprehensive analysis is conducted of how data poisoning influences feature correlations, class numbers, and dataset sizes within multiclass contexts. The visual datasets representation with the GMM is given in Fig. 4.3, highlighting their features correlation. Fig. 4.3(a) illustrates that certain features within the IRIS dataset are strongly interdependent, whereas the complete dataset is not in a linear relation. However, MNIST is found to be a highly dense dataset with strong features relations as visualised in Fig. 4.3(b). The ISIC dataset, shown in Fig. 4.3(c), displays a non-linear relationship with significant outliers, indicative of substantial noise levels. The statistical correlations of datasets are highlighted in Table 4.4. Features

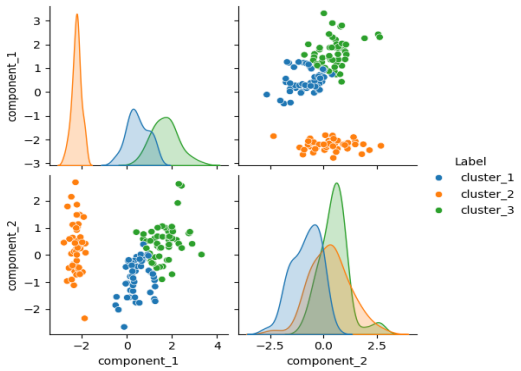
Algorithm 4 Calculating Distances from Decision Boundaries

Inputs: Surrogate Models M_{surr}, Training Dataset D
Outputs: Calculated distances of Models $dist_M$
Initialise: $dist_M = [dist_{SVM}, dist_{DT}, dist_{RF}, dist_{GNB}, dist_{KNN}, dist_{MLP}]$
$x \leftarrow$ Model data points
$M_{surr} = [M_{SVM}, M_{DT}, M_{RF}, M_{GNB}, M_{KNN}, M_{MLP}]$
if $M_{surr} == M_{SVM}$ **then**
  for $x_i \in M_{SVM}$ **do**
    $dist_{SVM}[x_i] \leftarrow decision_{function}(x_i, M_{SVM})$
  end for$dist_M[SVM] = x_i$
end if
if $M_{surr} == M_{DT}$ **then**
  $Clf_{tree} = M_{DT}.tree\_$
  for $x_i \in D$ **do**
    $dist[x_i] \leftarrow calculate\_depth(x_i, Clf_{tree})$
  end for
  $dist[DT] = x_i$
end if
if $M_{surr} == M_{RF}$ **then**
  for $clf_x \in M_{RF}$ **do**
    $Clf_{tree} = clf_x.tree\_$
    for $x_i \in D$ **do**
      $dist[x_i]_D = calculate\_depth(x_i, Clf_{tree})$
    end for
  end for
  $dist_M[RF] = avg(dist[x_1]_D, dist[x_2]_D, ...dist[x_n]_D)$
end if
if $M_{surr} == M_{KNN}$ **then**
  for $x_i \in D$ **do**
    $dist(x_i)_{neighbors} = M_{KNN}.kneighbors\_$
    $dist_M[KNN] \leftarrow \arg\max(distance(x_i)_{neighbors})$
  end for
end if
if $M_{surr} == M_{GNB}$ **then**
  $D_a, D_b = split(D, 2)$
  for $i \in [D_{c_a}, D_{c_b}]$ **do**
    $j = -i + 1$
    for $x_i \in D[i]$ **do**
      $Class(x_i) = predict_probability(D[j], M_{GNB})$
      $log_likelihood \leftarrow log(Class(x_i))$
      $distance(x_i) \leftarrow distance(\arg\max(Class(x_i), axis = 1))$
    end for
  end for$dist_M[GNB] = distance(x_i)$
end if
if $M_{surr} == M_{MLP}$ **then**
  for $x_i \in M_{MLP}$ **do**
    $dist_M[MLP] \leftarrow decision_function(x_i, M_{MLP})$
  end for
end if
return $dist_M$

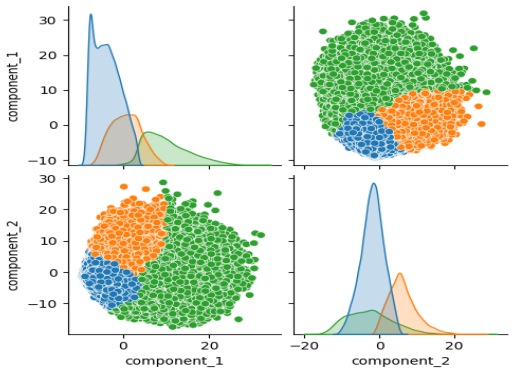
in the MNIST dataset are highly associated with a p-value of 0.0141, highlighting direct proportionality between its features. A low statistical significance is shown in the IRIS datasets with a p-value of 0.07, and the p-value of the ISIC dataset is 0.2396. In contrast, a negative Spearman correlation coefficient highlights a negative linear correlation between its features with a high noise ratio. Further

Table 4.3: Dataset description used for experimentation with outlier-oriented poisoning attack

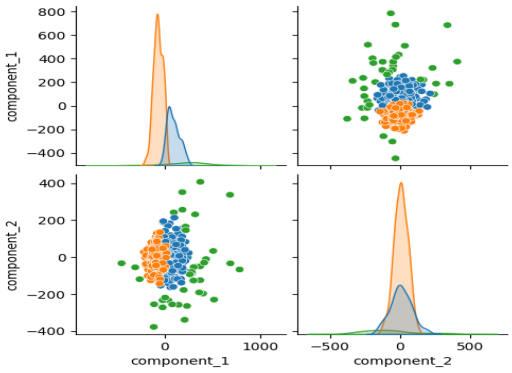
S.No.	Dataset	No. of features	No. of Classes	No. of instances
1	IRIS	4	3	170
2	MNIST	784	10	70,000
3	ISIC	20	4	603



(a) Features correlation in IRIS dataset



(b) Features correlation in MNIST dataset



(c) Features correlation in ISIC dataset

Figure 4.2: Gaussian mixture model visualisation of features relationship in the dataset with PCA reduction

analysis of the importance of features correlation and the impact of dataset noise for the OOP attack is given in Section 4.6.

Table 4.4: Statistical correlation of features in the dataset

S.No.	Dataset	Spearman Correlation	p-value
1	IRIS	0.123888	0.0791
2	MNIST	0.009282	0.0141
3	ISIC	-0.014311	0.2396

4.5.3 Attack Settings

Surrogate models for each algorithm is initialised and trained them with selected datasets. The distances of each data point are calculated from the decision boundary for each class to manipulate those far from the decision boundaries. Consider $\mathcal{T}$ is the model training process with poisoned

dataset D' and $\mathcal{P}_{\mathcal{M}}$ is the function of performance measure, the objective function of the attack method is given in Eq. 4.4 whereas, θ is the measure of distance of data points from the decision boundaries. Mathematical notation of θ is given in Eq. 4.5.

$$\arg\min \mathcal{P}_{\mathcal{M}}(\mathcal{M}(x', l'); \theta)) \tag{4.4}$$

$$\textbf{s.t. } \theta = \arg\max \vec{d}(b_c, (x, l)) \tag{4.5}$$

Also, D' is the poisoned dataset manipulated at various poisoning levels ΔL where the notation of dataset poisoning is given in Equation 4.6.

$$\begin{gathered} D' = \sum_{i=1}^{n \to \Delta L} f(D(x_i, l_i), \Delta L) \\ \textbf{where; } X_{l_c} \neq X'_{l_c} \end{gathered} \tag{4.6}$$

where, f is the function of manipulating labels, (x, l) is the clean data point, (x', l') is the poisoned data point with l'.

Let poisoning levels ΔL = [5%,10%,15%,20%,25%] manipulate model training by disturbing class-level decision boundaries b_c with notation given in Alg 2. Let f be the function to poison the dataset D' at ΔL poisoning level. M' is the poisoned model trained with a dataset having manipulated data points (x', l') as given in Eq. 4.7. This allows us to analyse the model behaviour and change in decision boundaries as given in Eq. 4.8

$$\begin{gathered} M' = \mathcal{T}(M, D') \\ \textbf{where; } D' = f(D(x, l), \Delta L) \end{gathered} \tag{4.7}$$

$$Mod_{Dis} = \Delta b(M') \tag{4.8}$$

where M' is the poisoned model developed for algorithms [SVM, DF, RF, KNN, GNB, MLP] and Δb is the change in decision boundaries. To conduct a statistical analysis of the performance degradation of multiclass models and the variance in test-time classification across different poisoning levels, the correct classification rate is defined in Eq. 4.9.

$$\begin{gathered} C_{CR} = \frac{\sum_{i=1}^{n} f(N_c, C_{c_i}(M'_{c_i}(D_t(x_i, l_i))))}{\sum_{i=0}^{n=1} N_c} \\ and\ f(N_c, C(D_t(x, l))) = \begin{cases} true & \text{if } (x, l) \in Class\ c \\ false & \text{otherwise} \end{cases} \end{gathered} \tag{4.9}$$

where, f is the function of classification, X_t is the data point from the validation dataset D_t, N_c is the total number of data points in Class c, $C(.)$ is the class estimator and C_{CR} is the non poisoned classification rate.

The OOP attack is conducted against multiclass classification models to assess the poisoning impact on multiclass ML performance. For the performance evaluation and analysis of the impact of poisoning availability attacks in multiclass models, poisoned models are evaluated by analysing how many outliers successfully intrude themselves in wrong classes, which is the **False Positive Rate(FPR)** of the model where the model fails in classifying the correct class. However, where the poisoned outliers remain disjointed in the incorrect classification classes and model availability is intact, is the **Accuracy(Acc)**, and where the outliers are unsuccessful in intruding the multiclass decision boundaries is the **Precision** of the model against OOP attack. **Recall** in the evaluation is the quantification where a model can segregate dataset classes and keep the decision boundaries intact. **Variance(Var)** reflects how the model's behaviour changes when its parameters or dataset change. Considering $f(N_c, C((x,l)))$ is the classification function as given in Eq 4.10, the evaluation metrics are mentioned in Eq 4.11, 5.17, 4.13, 5.18 and 4.15.

$$f(N_c, C((x,l))) = \begin{cases} true & \text{if } (x,l) \in Class\ c \\ false & \text{otherwise} \end{cases} \tag{4.10}$$

$$FPR = \frac{\sum_{i=0}^{n} f_{tr}(N_c, C(x_i', l_i'))}{\sum_{i=0}^{n} f(N_c, C(x_i', l_i')) \wedge \sum_{i=0}^{n} f(N_c, C(x_i, l_i))}$$

$$\begin{aligned} where\ f_{tr}(N_c, C(x_i', l_i')) &\in D' \\ and\ f(N_c, C(x_i', l_i')) &\in D' \\ and\ f(N_c, C(x, l)) &\in D \end{aligned} \tag{4.11}$$

where D is the clean dataset, N_c is the total number of data points in Class c, and D' is the poisoned dataset with changed class labels of the farthest data points. $f_{tr}(N_c, C(x', l'))$ are poisoned data points with perturbed labels and classified as false positives(FP) and $f_{fs}(N_c, C(X'_{tr_i}))$ are false negative(FN) data points.

$$Acc = \frac{\sum_{i=0}^{n} f_{fs}(N_c, C(x_i, l_i)) \wedge \sum_{i=0}^{n} f_{tr}(N_c, C(x_i, l_i))}{((x,l) \in D) \wedge ((x_i', l_i') \in D')} \tag{4.12}$$

$$Prn = \frac{\sum_{i=0}^{n} f_{tr}(N_c, C(x_i, l_i))}{\sum_{i=0}^{n} f_{tr}(N_c, C(x_i, l_i)) \wedge \sum_{i=0}^{n} f_{tr}(N_c, C(x'_i, l'_i))} \tag{4.13}$$

$$where\ f_{tr}(N_c, C(x'_i, l'_i)) \in D'$$

$$Rcl = \frac{\sum_{i=0}^{n} f_{tr}(N_c, C(x_i, l_i))}{\sum_{i=0}^{n} (f_{tr}(N_c, C(x_i, l_i))) \wedge \sum_{i=0}^{n} (f_{fs}(N_c, C(x'_i, l'_i)))} \tag{4.14}$$

$$where\ f_{fs}(N_c, C(x'_i, l'_i)) \in D'$$

$$Variance(\sigma) = \frac{1}{N_c} \sum_{i=0}^{n} (f(N_c, C(x_i, l_i)) - \mu(f(N_c, C(x'_i, l'_i))))^2 \tag{4.15}$$

4.6 Experimentation Results and Analysis

The objective is to analyse the behaviour of multiclass models and answer questions about how the characteristics of these models are affected and what their relationship is with the poison. What are the optimal poisoning levels ΔL and the effects of changing poisoned data distributions? What is the effectiveness and persistence of data poisoning with the OOP attack and its impact on model validation performance (specifically accuracy)? And quantifying and analysing model variance σ at test-time classification at different poisoning levels ΔL.

4.6.1 Effects on Multiclass Classification

The baseline results of the OOP attack are given in Fig. 4.3 to Fig. 4.8, where validation accuracy, precision, recall, f1-score and FPR are plotted against poisoned training with maximum poisoning level $\Delta L = 25\%$. These results indicate that the KNN algorithm is particularly vulnerable, experiencing the most significant accuracy disruption with a maximum decrease in accuracy $(\lambda) = 40.35$ at $\Delta L = 25\%$ with an increase in FPR=31.6% from FPR=2.7%, shown in Fig. 4.6(a). This vulnerability stems from KNN being a non-parametric algorithm that relies on the proximity of data points to determine class features. Table 4.5 highlight that the number of nearest neighbors found to be inversely proportional to ΔL, reducing the attack success rate 15.79% to 2.76% for the IRIS dataset by changing k=3 to k=15. Fig. 4.6(c) demonstrates high ASR when KNN is trained with the ISIC dataset, decreasing its validation accuracy to 63% with FPR=28.25%. From Table 4.5, increasing the number of nearest neighbors decreases ASR from 3.97% to 3.31% with $\Delta L = 25\%$.

The GNB is the second most affected algorithm with a decrease in validation accuracy from 92.98% to 56.14% and an increase in FPR from 5.68% to 32.49% at $0\% \leq \Delta L \leq 25\%$, for the IRIS dataset, given in Fig 4.5. Interestingly, the GNB model is failing with the OOP attack at $\Delta L = 15\%$ where

it's *precision* ≤ 0 where a lower impact can be seen with MNIST and ISIC datasets. Further analysis reveals the change in the importance of classes, leading to misclassification, with changing class probabilities at poisoning levels, given in Table 4.6, whereas the class probability is the model confidence in classifying an instance against each class.

The attack manipulates the gaussian probability measures, making the highest probability class an anomaly and vice versa for the IRIS dataset. However, minor changes are visible for MNIST and ISIC datasets with no change in classes ranking at $0\% \leq \Delta L \leq 15\%$. This analysis also highlights that GNB is the most affected algorithm when trained with a dataset with fewer classes.

Table 4.5: Analysing k-neighbors affecting k-nearest neighbors accuracy with $\Delta L = (0, 10, 15, 25)\%$

Dataset	Poison Level	k=3	k=5	k=10	k=15
IRIS	$\Delta L = 0\%$	94.73	97.50	97.36	97.36
	$\Delta L = 10\%$	89.47	97.36	97.30	94.73
	$\Delta L = 15\%$	81.57	92.10	94.73	92.10
	$\Delta L = 20\%$	78.94	84.21	94.60	94.60
MNIST	$\Delta L = 0\%$	98.16	97.55	96.94	96.55
	$\Delta L = 10\%$	92.41	96.52	96.78	96.50
	$\Delta L = 15\%$	89.44	90.90	94.54	95.95
	$\Delta L = 25\%$	85.34	76.14	83.68	87.52
ISIC	$\Delta L = 0\%$	80.79	82.11	70.19	77.48
	$\Delta L = 10\%$	77.48	77.48	66.88	74.17
	$\Delta L = 15\%$	76.15	74.17	68.87	76.13
	$\Delta L = 25\%$	76.82	74.07	64.90	74.17

Table 4.6: Analysing class probabilities of gaussian naive bayes with the poisoned dataset

Dataset	Dataset Class	Clean Dataset	$\Delta L = 10\%$	$\Delta L = 15\%$
IRIS	Class 0	0.33	0.36	0.38
	Class 1	0.35	0.25	0.33
	Class 2	0.31	0.37	0.27
MNIST	Class 0	0.09	0.09	0.09
	Class 1	0.11	0.11	0.11
	Class 2	0.09	0.09	0.09
	Class 3	0.10	0.10	0.10
	Class 4	0.09	0.10	0.09
	Class 5	0.08	0.09	0.09
	Class 6	0.09	0.09	0.09
	Class 7	0.10	0.10	0.10
	Class 8	0.09	0.09	0.09
	Class 9	0.09	0.10	0.10
ISIC	Class 0	0.76	0.69	0.64
	Class 1	0.05	0.08	0.10
	Class 2	0.02	0.04	0.07
	Class 3	0.14	0.17	0.17

Whereas the OOP attack has minimally disrupted DT, resulting in λ values of 31.6 for IRIS, 15.18 for MNIST, and 17.88 for ISIC at $\Delta L = 25\%$. Table 4.7 demonstrates the change in features importance scores with dataset poisoning, where feature1 scores (0.90, 0.39) remain highest for IRIS

and MNIST. But feature1 (0.36) with the highest importance score for ISIC becomes anomalous, making anomaly feature2 (0.37) the most important feature at $\Delta L = 15\%$, degrading its classification. The RF algorithm demonstrates relative robustness, with its FPR converge to $\approx 2\%$ with an overall accuracy decrease to 61.25% from 87% for the ISIC dataset and FPR converge to $\approx$ 9% for the MNIST dataset with accuracy of 82.38% at $\Delta L = 25\%$ as shown in Fig. 4.4. Because RF follows the ensemble approach and classifies averaging decisions from all of its trees, which normalises the poisoning effects in this case. The change in features importance scores for RF is given in Table 4.10, where features ranks remain the same for IRIS and MNIST, but for ISIC highest ranked feature dropped to rank two at $\Delta L = 15\%$ poisoning. Lastly, SVM and MLP are also not found to be very sensitive to the OOP attack. For SVM, features ranks remain intact, given in Table 4.8, except for ISIC, where feature3 (0.39) importance score reduces to (0.33) at $\Delta L = 15\%$, making it an anomaly. A lower impact is visible on MLP from Fig 4.8, with this attack, except at $\Delta L = 15\%$ where it is failing for the IRIS dataset.

Table 4.7: Features importance score - Decision tree where $\Delta L = (0\%, 10\%, 15\%)$

	Clean Dataset			Poisoned Dataset $\Delta L = 10\%$			Poisoned Dataset $\Delta L = 15\%$		
Dataset	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3
IRIS	0.90	0.00	0.02	0.87	0.008	0.11	0.79	0.07	0.12
MNIST	0.39	0.34	0.26	0.39	0.33	0.27	0.39	0.32	0.28
ISIC	0.36	0.28	0.35	0.28	0.38	0.32	0.32	0.37	0.30

Table 4.8: Features importance score - Support vector machines where $\Delta L = (0\%, 10\%, 15\%)$

	Clean Dataset			Poisoned Dataset $\Delta L = 10\%$			Poisoned Dataset $\Delta L = 15\%$		
Dataset	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3
IRIS	0.90	0.02	0.08	0.78	0.05	0.15	0.86	0.10	0.02
MNIST	0.40	0.16	0.43	0.34	0.23	0.42	0.36	0.21	0.42
ISIC	0.33	0.27	0.39	0.30	0.22	0.47	0.32	0.33	0.33

Table 4.9:
Analysing support vector machines margin score for different datasets with $\Delta L = (0, 10, 15)\%$

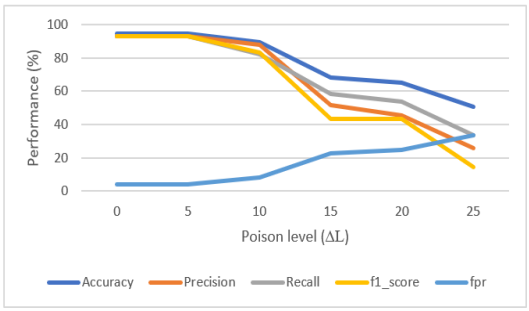
Dataset	$\Delta L = 0\%$	$\Delta L = 10\%$	$\Delta L = 15\%$
IRIS	0.005	0.01	0.001
MNIST	0.0000011	0.00000022	0.00000027
ISIC	0.01	0.003	0.003

4.6.2 Effects of Poisoning Rates

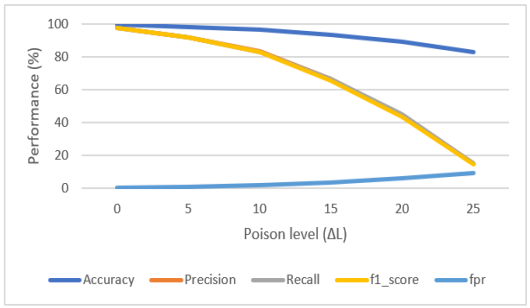
The analysis is extended to study the effects of consistently increasing poisoning rates on multiclass models with the OOP attack. The aggregated results, given in Fig. 4.3 to Fig. 4.8, show over-fitting? **No**. These results demonstrated that the classification accuracy of multiclass classifiers has maximum disruption when the training dataset is poisoned with the OOP attack at $\Delta L = 10\%$, irrespective of

Table 4.10:
Features importance score - Random forest where $\Delta L = (0\%, 10\%, 15\%)$

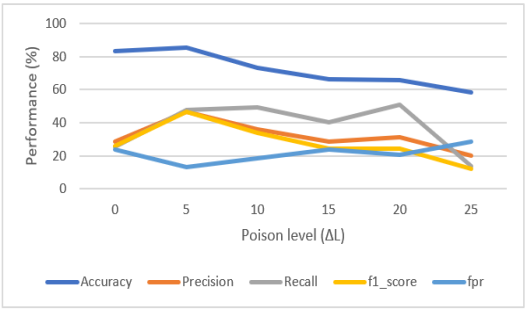
Dataset	Clean Dataset			Poisoned Dataset $\Delta L = 10\%$			Poisoned Dataset $\Delta L = 15\%$		
	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3	Feature1	Feature2	Feature3
IRIS	0.66	0.15	0.17	0.58	0.19	0.22	0.52	0.22	0.25
MNIST	0.39	0.34	0.26	0.39	0.33	0.27	0.39	0.32	0.27
ISIC	0.31	0.35	0.34	0.31	0.36	0.32	0.34	0.33	0.32



(a) Poisoning support vectore machines with IRIS dataset



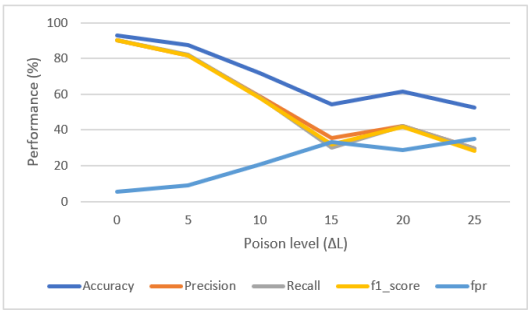
(b) Poisoning support vector machines with MNIST dataset



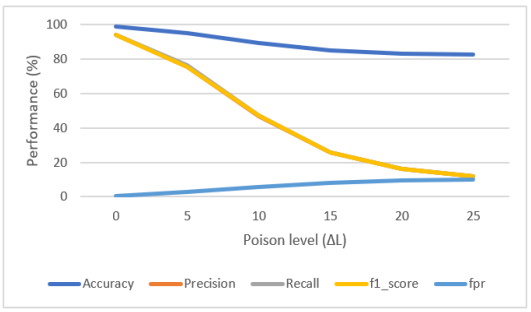
(c) Poisoning support vector machines with ISIC dataset

Figure 4.3: Performance analysis of support vector machines with consistent poisoning

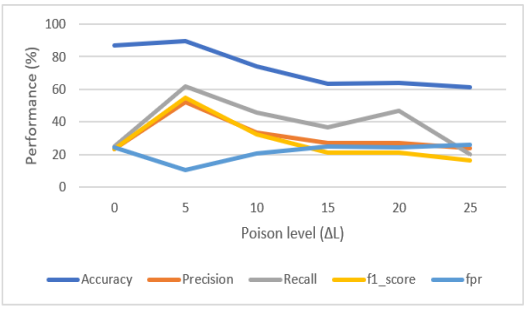
datasets. An inverse relationship is observed between the number of classes in the dataset and the rate of performance degradation. For the MNIST dataset, from Fig. 4.3(b) to Fig. 4.8(b), the ten dataset classes have a steady decrease in performance. Whereas classifiers trained with the IRIS dataset, with three dataset classes, have high fluctuation in performance, followed by ISIC with four classes. The least percentage of data poisoning is more effective on parametric models. The 10% poisoning has a steady and practical impact on parametric models, whereas 15% poisoning leads to impractical effects. From Fig. 4.5(a) and Fig. 4.8(a), parametric models, with minimum no. of classes, are failing at 15% poisoning. But, $\Delta L = 15\%$ is very effective for non-parametric models. Conclusively, $10\% \leq \Delta L \leq 15\%$ are the optimal poisoning rates for multiclass models, where $\Delta L > 15\%$ shows an impractical success.



(a) Poisoning random forest with IRIS dataset

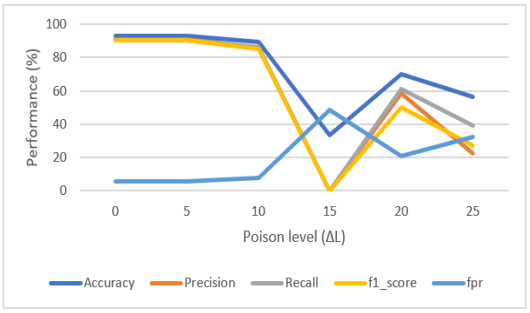


(b) Poisoning random forest with MNIST dataset

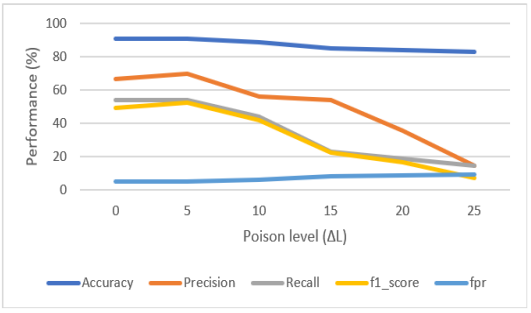


(c) Poisoning random forest with ISIC dataset

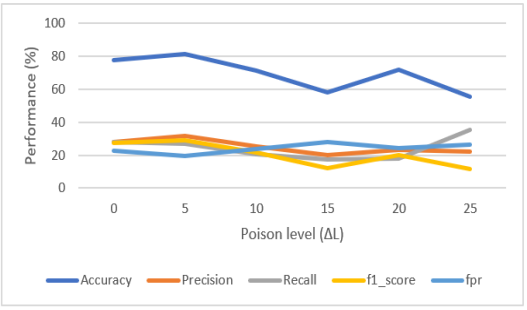
Figure 4.4: Performance analysis of random forest with consistent poisoning



(a) Poisoning gaussian naive bayes with IRIS dataset



(b) Poisoning gaussian naive bayes with MNIST dataset

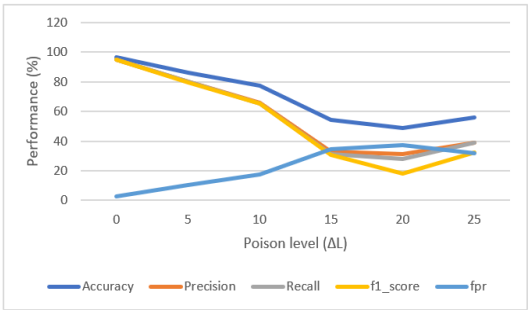


(c) Poisoning gaussian naive bayes with ISIC dataset

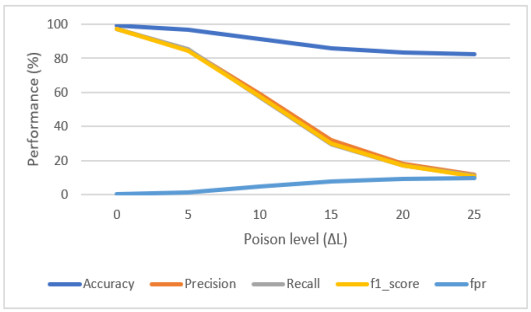
Figure 4.5: Performance analysis of gaussian naive bayes with consistent poisoning

4.6.3 Model Sensitivity to Poison

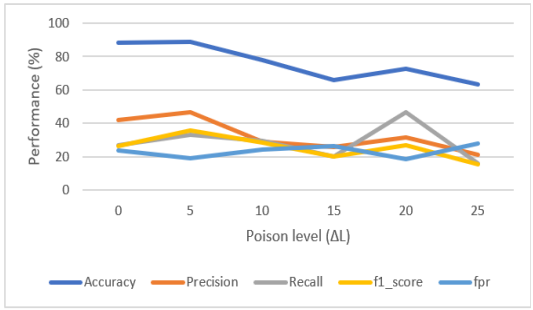
The sensitivity of the poisoned model is investigated by analysing the relationship between model variance and ASR. Table 4.11 illustrates the variance in ML models in response to the OOP attack.



(a) Poisoning k-nearest neighbors with IRIS dataset

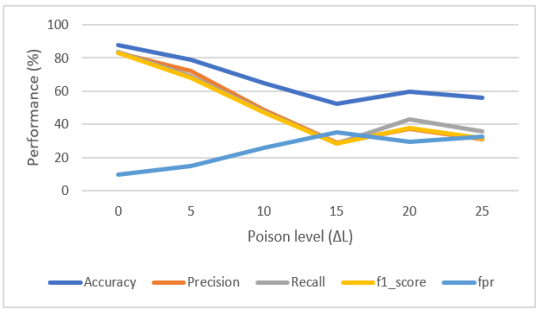


(b) Poisoning k-nearest neighbors with MNIST dataset

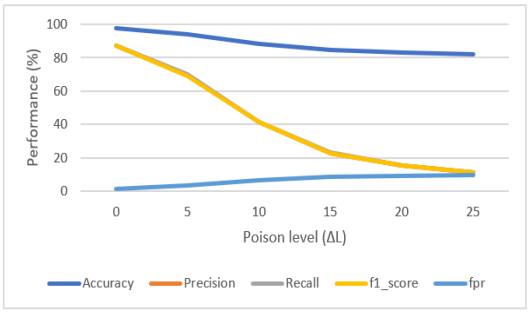


(c) Poisoning k-nearest neighbors with ISIC dataset

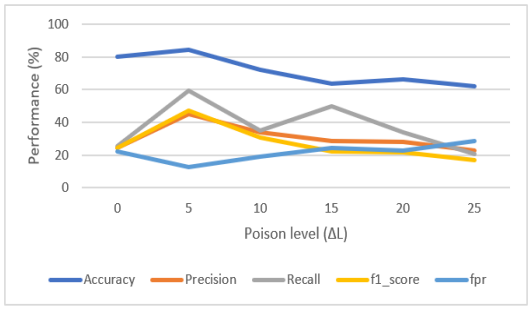
Figure 4.6: Performance analysis of k-nearest neighbors with consistent poisoning



(a) Poisoning decision tree with IRIS dataset



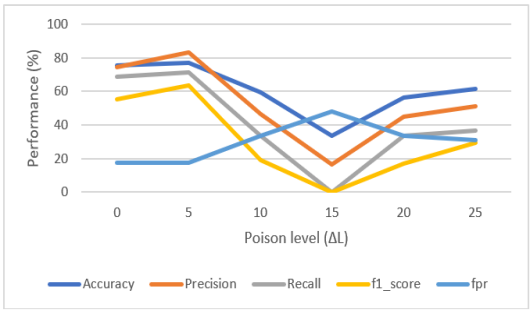
(b) Poisoning decision tree with MNIST dataset



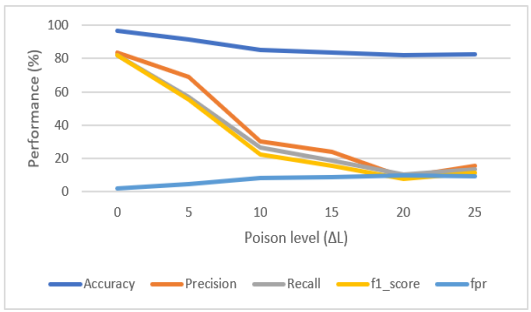
(c) Poisoning decision tree with ISIC dataset

Figure 4.7: Performance analysis of decision tree with consistent poisoning

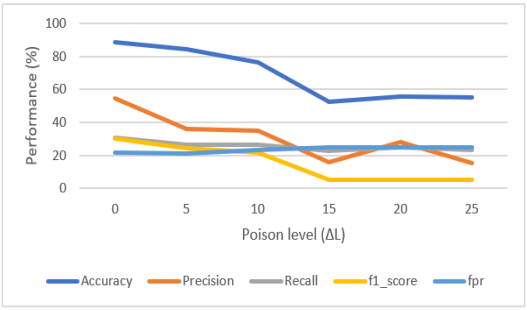
This attack significantly increased the sensitivity of all tested models, with GNB exhibiting the highest sensitivity. Its variance leads to 0.8 at $\Delta L = 10\%$, for the IRIS dataset, almost equivalent to DT, where it fails. Similarly, 0.10 variance increases for KNN at $\Delta L = 15\%$, highlighting its high



(a) Poisoning multilayer perceptron with IRIS dataset



(b) Poisoning multilayer perceptron with MNIST dataset



(c) Poisoning multilayer perceptron with ISIC dataset

Figure 4.8: Performance analysis of multilayer perceptron with consistent poisoning

sensitivity and the effectiveness of the OOP attack. RF and DT are proven to be less sensitive to this attack. Interestingly, on average, models trained with MNIST and ISIC are also less affected by the poisoning attack compared to models trained with the IRIS dataset, with high impact.

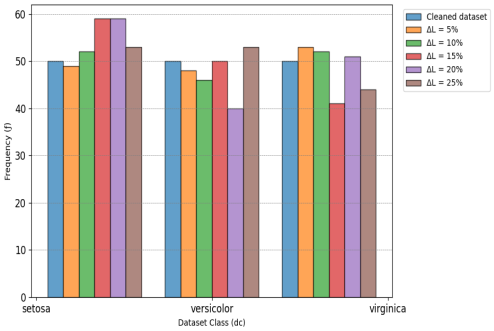
Further analysis is conducted on the dataset distribution to ascertain its impact on data poisoning and performance degradation in models. Fig. 4.9 shows the change in data distribution with the OOP attack at $0\% \leq \Delta L \leq 25\%$. The findings suggest that balanced datasets with a larger number of classes tend to mitigate the effects of poisoning on model performance, particularly in terms of model accuracy. In contrast, imbalanced and noisy datasets work as catalysts and boost the poisoning effects of this attack, leading to an impractically high decrease in performance, such as for the ISIC dataset, as shown in Fig. 4.9(c). This analysis identifies relationships between various classification characteristics and subsequent rates of data poisoning, as given in Table 4.12.

Table 4.11:
Model variance at different poisoning levels

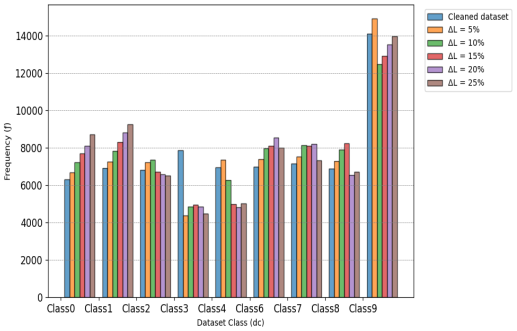
Dataset	Algorithm	Clean Dataset	$\Delta L = 10\%$	$\Delta L = 15\%$
IRIS	SVM	0.33	0.36	0.57
	RF	0.62	0.60	0.63
	GNB	0.65	0.73	0.68
	KNN	0.81	0.82	0.91
	DT	0.59	0.68	0.78
	MLP	0.65	0.69	1.45
MNIST	SVM	8.33	8.06	7.97
	RF	8.24	7.69	7.71
	GNB	11.25	12.68	12.74
	KNN	8.36	8.37	8.38
	DT	8.33	7.81	8.02
	MLP	8.31	8.38	8.31
ISIC	SVM	1.33	0.97	1.36
	RF	1.11	1.17	1.32
	GNB	1.27	1.66	1.19
	KNN	0.31	0.37	0.27
	DT	0.31	0.37	0.27
	MLP	1.52	1.48	1.59

Table 4.12:
Analysing one-to-one relation between poison and various parameters of machine learning algorithms

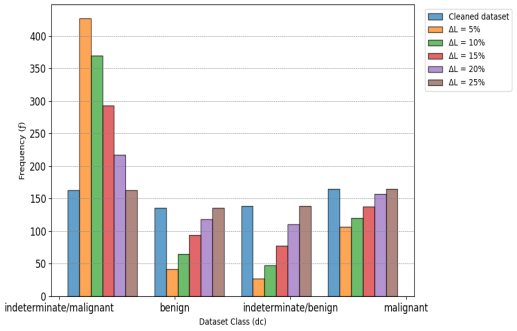
Algorithm	Algorithmic Parameters	Relation to ΔP
SVM	Margin score	Minimal impact
	Decision boundary	Minimal impact
	Features importance score	Minimal impact
DT	Features importance score	Minimal impact
	Asymmetric features space	High impact
KNN	Decision boundary	High impact
	k-neighbors	Inverse impact
GNB	Decision boundary	High impact
	Class probabilities	High impact
RF	No. of trees	Inverse impact
	Features importance score	Minimal impact
MLP	Weights	High impact



(a) Data distribution of IRIS dataset with OOP attack



(b) Data distribution of MNIST dataset with OOP attack



(c) Data distribution of ISIC dataset with OOP attack

Figure 4.9: Data distribution with OOP attack

4.7 Case Study 1: OOP Attack Against Adversarially Trained Skin Cancer Diagnostics

The ML systems have seen wide adoption in healthcare, providing proficiency in handling complex decision-making tasks and optimising their diagnostic processes efficiently. Skin cancer is a severe disease with a high frequency rate. Due to its potential seriousness and prevalence, an early diagnosis is crucial for its treatment. Traditional diagnosis requires biopsy and visual examination, which is time-consuming and subject to human error. In this context, ML tools show promising results. These models can handle various data types and ensure efficient and optimised diagnosis capability compared to manual processing.

On the other hand, data poisoning Lu et al. (2023), Wei et al. (2023) can poison the ML model's dataset and harm the victim model's diagnosis mechanism. Various solutions are provided in the literature to mitigate such data poisoning attacks, including data sanitisation (Paudice et al., 2018b), ensemble learning (Ahmed et al., 2022), and adversarial training (Tao et al., 2021). Adversarial training is one of the prominent solutions through which the ML model is trained with a combination of cleaned and poisoned data samples to allow the model to understand the patterns of the poisoned data. This mechanism makes the model resilient and enables it to surpass similarly poisoned dataset samples if injected.

However, adversarial training can be breached with novel data poisoning attacks Wen et al. (2023), Tao et al. (2022b). This study leverages the capabilities of the OOP attack and analyses its effectiveness against adversarial training. Another important consideration of this case study is that adversarial training is mostly analysed to secure DL models, as discussed in Section 2.4 in Chapter 2, whereas this case study analysed the significance of adversarial training on the traditional multiclass models in non-complex settings. The contribution of this case study 1 is to conduct a thorough analysis of the OOP attack against adversarially trained multiclass SVM, RF, and MLP models.

4.7.1 Attack Model

A grey-box approach is followed to poison the adversarially trained skin cancer application, developed in multiclass model settings. The attack model comprises four steps as given in Fig. 4.10. The details of each stage of the attack model are given as follows.

Attack Generation Generating appropriate data poisoning attacks, the poisoning level of 10% is set for the complete dataset to achieve a maximum effectiveness of the data poisoning against adversarially trained models. Following research (Paracha et al., 2024a), 10%-15% of dataset poisoning is the most effective poisoning level. Subsequently, following the threat model, the OOP attack is implemented and set the dataset=ISIC dataset, and algorithm=targeted model algorithm with different

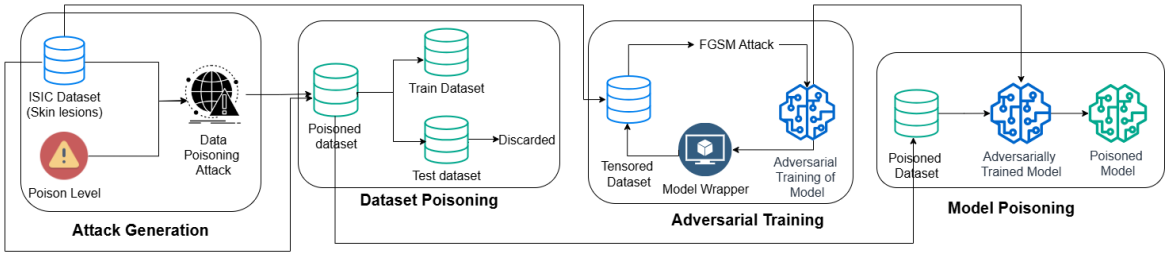


Figure 4.10: Overview of the attack model - Analysing efficacy of outlier data poisoning attack against adversarially trained models for skin cancer diagnosis

Table 4.13: Configurations of the models

S.No.	Model	Model Configurations
1	SVM	decision_function_shape='ovo', kernel='poly', degree=3, C=1
2	RF	criterian='log_loss', n_estimators=9, max_depth=4
3	MLP	penalty='elasticnet', alpha=0.0001, max_iter=1000

configurations.

Dataset Poisoning At the dataset poisoning stage, the data poisoning attack with the ISIC dataset is implemented and generated a poisoned version of the dataset, implemented with simple baseline algorithms: SVM, RF, and MLP.

Adversarial Training Using the baseline line algorithms for SVM, RF, and NN, the wrapper models were developed for each baseline model to accommodate adversarial training with the tensor-formatted dataset. The cleaned dataset D is distributed $dict(D)$ into training and testing with a 75%-25% percentage. The algorithm to perform the adversarial training of the baseline model is given in Alg. 5.

Model Poisoning Access to the victim model that is adversarially trained to be resilient against data poisoning, the model is evaluated against an outlier data poisoning attack (Paracha et al., 2025a) to assess the reliability of the adversarial training to secure baseline ML models. Furthermore, the experimentation highlighted that adversarially trained baseline ML models are highly affected by data poisoning attacks. The attack model implementation is given in Alg 6.

4.7.2 Experimentation Results and Analysis

The experimentation is conducted following two different dataset distributions. 75%-25% dataset distribution is followed to develop the resilient models with adversarial training and for performance analysis. Whereas, the adversary follows 80%-20% dataset distribution to poison the dataset. This setup allowed us to assess how data poisoning deceives the skin cancer diagnostic application even when the adversary follows a different data distribution. This case study analyses the resilience of

Algorithm 5 Adversarial Training of Baseline ML Model

$training\_epochs = 15$
$pert\_epsilon = 0.1$
$batch\_size = 4$
procedure ADVERSARIAL_TRAINING(x_train,y_train,$model$)
$N = length(y\_train)$
▷ X_train is the training dataset and y_train are the training dataset labels
$optimiser = SGD\_Optimiser(model.parameters(), lr = 0.1)$
$model \leftarrow training\_state$
for epoch in 1,2 ... training_epochs **do**
$perm = random\_permutation(N)$
$sum\_loss = 0$
for i in 1,2 ... batch_size **do**
$data\_sample = x\_train[perm[i : i + batch\_size]]$
$data\_sample\_label = y\_train[perm[i : i + batch\_size]]$
$gradient(optimiser) = 0$ ‘
$output = squeeze(model(data\_sample))$
$weight = squeeze(model.weight)$
$loss = mean(clamp(output, -1, 1))$
$loss = loss + 0.01 * (weight.t()@weight)/2.0$
$loss\_gradient = gradient(loss)$
$stepper(optimiser)$
$data\_grad = gradient(data\_sample)$
$x\_adv = FGSM\_ATTACK(data\_sample,$
$epsilon,$
$data\_grad)$
▷ FGSM attack is one of the principal data poisoning attacks
$adv\_train\_x = merge(x\_train, X\_adv)$
$adv\_train\_y = merge(y\_train, y\_adv)$
end for
end for
$train(model, adv\_train\_x, adv\_train\_y)$ ▷ performing adversarial training of the model
end procedure
return *model*

the models, their variance, and the impact of the OOP attack against traditional ML. This study assesses the limitations of adversarial training against novel data poisoning attacks.

The performance of the adversarially trained models is evaluated against their poisoned versions. Fig. 4.11 highlights the decrease in model accuracy when trained with the poisoned dataset compared to its counterpart. Data poisoning attack shows its effectiveness against adversarially trained ML models. The results highlighted that adversarial training profoundly secured neural networks against new data poisoning attacks, whereas it does not work against other baseline models. Adversarially trained SVM and RF models are vulnerable to data poisoning. Table 4.15 provides the test-time accuracy, precision, recall, f1-score, and false positive rate of cleaned and poisoned models with poisoning rate $\Delta L = 10\%$. From these results, SVM is the most affected model with the highest decrease in test time accuracy of around 50%; however, the accuracy of the neural networks model only decreased to 2%.

This analysis is extended to study the effects of data poisoning on each dataset class. The test time dataset comprises 151 instances in total, from which the biggest class is **Benign(Non-cancerous)**,

Algorithm 6 Poisoned Training Of The ML Model

Inputs: ISIC Training Dataset = D_{ISIC}
Outputs: Poisoned Model = M'_{SVM}, M'_{RF}, M'_{NN}
Initialise: Wrapper Model = W_M,
Neural Networks Module ($module = torch.nn.Module$),
Number of Dataset Classes ($N_c = 4$)
Poison Level ($\Delta L = 10\%$),
Adversarial Training ($Adv_t = Adv(Attack = FGSM)$),
Poisoning Attack = $Attack_{outlier}$
algorithms = ['SVM', 'RF', 'NN']

for alg in algorithms **do**
$M_{wrapper} = W_M(module, alg) \leftarrow size(D_{ISIC}), num\_classes)$
end for
$D_{train}, D_{test} = split\_dataset(D_{ISIC}, train\_size = 75\%, test\_size = 25\%)$
$X\_train\_tensor = inst\_to\_tensor(D_{train})$
$y\_train\_tensor = inst\_to\_tensor(labels(D_{train}))$
$M_{wrapper} \leftarrow Train\_SVM(X\_train\_tensor, y\_train\_tensor, M_{wrapper}, FGSM\_Attack)$
$D'_{ISIC} = Attack_{outlier}(D_{ISIC}, M_{wrapper}, poison = 10\%)$
$D'_{train}, D'_{test} = split\_dataset(D'_{ISIC}, train\_size = 80\%, test\_size = 20\%)$
$X\_poisoned\_tensor \leftarrow inst\_to\_tensor(D'_{train})$
$y\_poisoned\_tensor = inst\_to\_tensor(labels(D'_{train}))$
$M'_{wrapper} \leftarrow Train\_M_{wrapper}(X\_poisoned\_tensor, y\_poisoned\_tensor, M_{wrapper}, FGSM\_Attack)$
return $M'_{wrapper}$

having 115 samples and **Indeterminate/malignant** is the smallest class of 5 samples only. Table 4.14 provides the class-level test time misclassification rate with 10% training time poison.

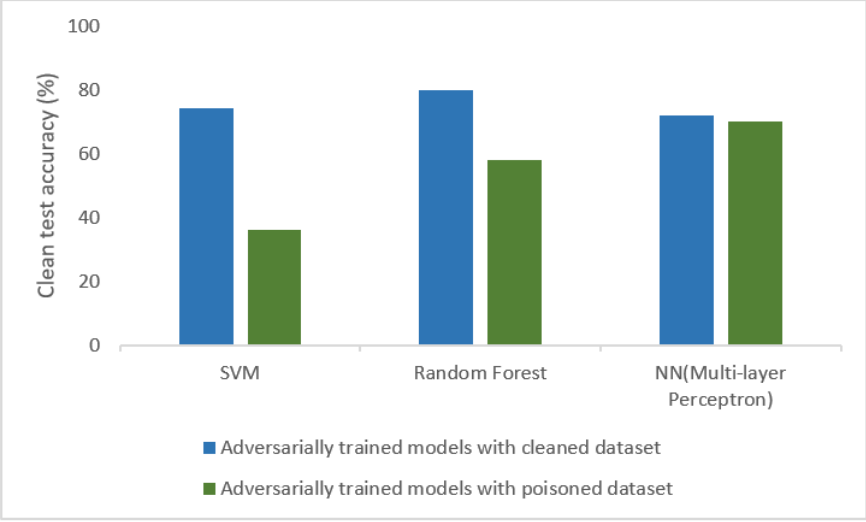


Figure 4.11: Model accuracy - Decrease in model's performance with data poisoning attack

This study investigated the relationship between model variance and the data poisoning attack at $\Delta L = 10\%$. Fig. 4.13 illustrates an increase in model variance and attack success rate with poisoned training. MLP has shown the lowest variance increase of 0.02, whereas SVM exhibits high sensitivity against a data poisoning attack with a variance of 1.15. Consequently, RF is a stable model with the

Table 4.14: Misclassification rate for each dataset class

Model	Dataset	Benign= 115	Malignant= 24	Indeterminate/ Benign=7	Indeterminate/ Malignant=5	Misclassification Rate
SVM	Cleaned ISIC	3	24	5	1	0.22
RF	Cleaned ISIC	5	20	6	1	0.21
MLP	Cleaned ISIC	2	24	6	1	0.21
SVM	Poisoned ISIC	24	25	5	1	0.36
RF	Poisoned ISIC	63	13	2	0	0.51
MLP	Poisoned ISIC	27	14	6	1	0.31

Table 4.15: Classification results of adversarially trained models

Model	Accuracy	Precision	Recall	F1-score	FPR
Clean SVM	74.5	30.35	29.62	29.88	21.06
Poisoned SVM	36	29.60	28.08	27.78	24.63
Clean RF	80	52.81	28.72	27.87	22.77
Poisoned RF	58.2	23.87	26.15	22.82	23.53
Clean MLP	72	52.55	26.27	24.48	24.25
Poisoned MLP	70.1	50.05	26.63	23.91	24.23

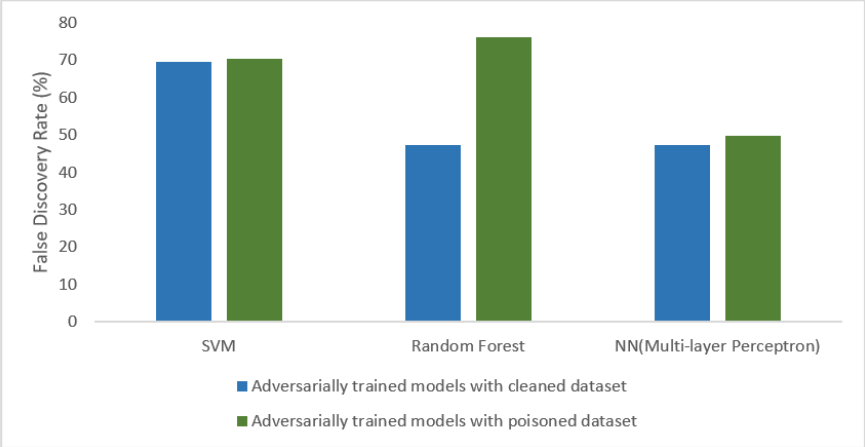


Figure 4.12: Model's false discovery rate - Increase in the false diagnosis rate of the models when trained with the poisoned dataset

least variance of 0.28 for its cleaned version, increasing to 0.60 with poisoned training. Overall, the highest impact of poisoning is visible on the SVM model.

This research examined the impact of the outlier data poisoning attack on the decisions of the individual models by taking random samples. It analysed the change in the confidence scores of randomly taken samples from the poisoned models and their impact on the application of healthcare diagnosis. Results of change in diagnosis confidence can be seen in Fig. 4.14 to Fig. 4.16. From these results, the least disruption of 0.1/0.86 prediction confidence for the Benign(Non-cancerous) sample is shown from the poisoned MLP model. Whereas, training time poisoning is shifting a Benign(Non-cancerous) sample to a Malignant(Cancerous) with a slight change in confidence that is 0.2/0.55 from Benign(Non-cancerous) to Indeterminate/Benign, in Fig. 4.14. However, a complete change in

the diagnosed class confidence of 100%, for the poisoned SVM model, as seen in Fig. 4.15. Conclusively, adversarial training is effective in making neural networks resilient against novel data poisoning attacks, whereas it is ineffective for other baseline ML models, RF, and SVM in this research.

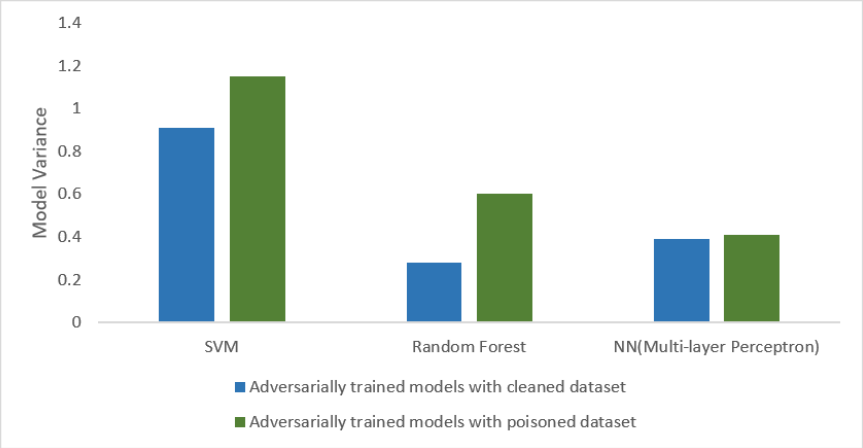


Figure 4.13: Model variance - Sensitivity of the models against data poisoning attack

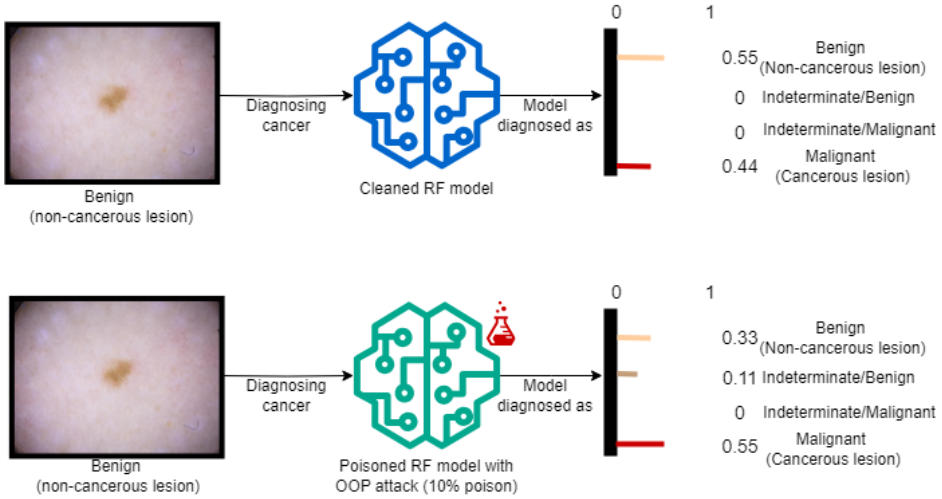


Figure 4.14: Random forest poisoned sample 1 - Analysing change in confidence score with poisoned training

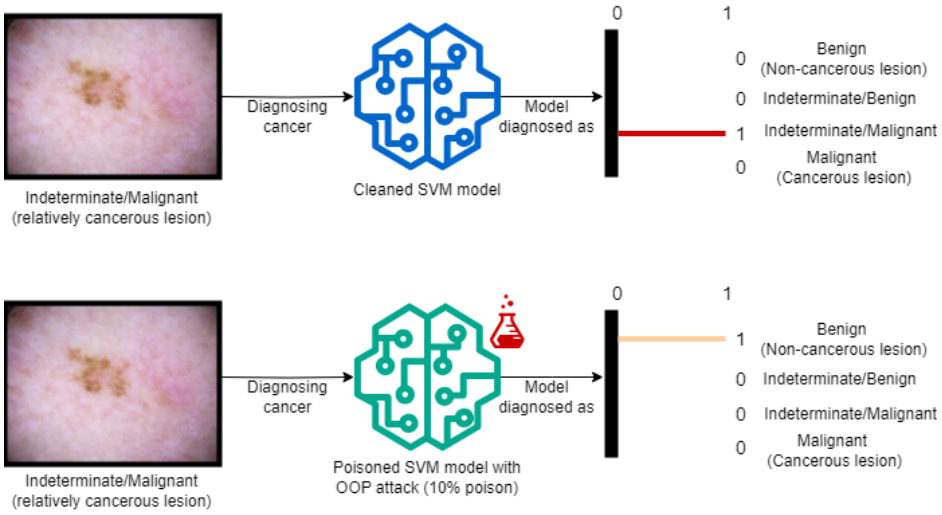


Figure 4.15: Support vector machines poisoned sample 1 - Analysing change in confidence score with poisoned training

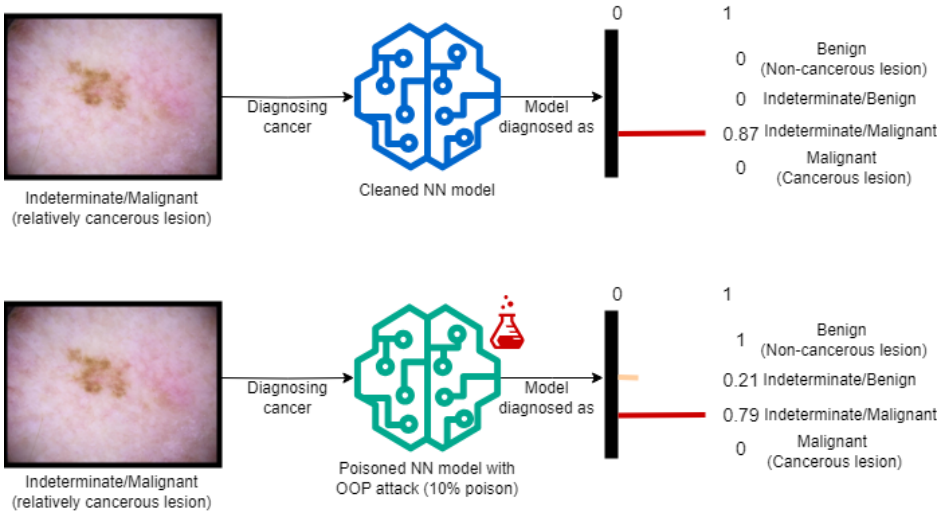


Figure 4.16: Multilayer perceptron poisoned sample 1 - Analysing change in confidence score with poisoned training

4.8 Case Study 2: OOP Attack Against Industrial 5G Private Network

The technological industrial revolution, known as "Industry 4.0", is receiving steady adoption across a wide range of industries. These industrial transformations are driven by digital transformation, enabled by advanced connectivity, artificial intelligence, and robust computing. ML is playing a vital role in developing, managing and optimising the systems and networks in Industry 4.0. Given the enabling capabilities of these technologies and the significance of their application in different sectors, the impact of cyber security threats and the consequential effect on industrial outputs can not be overemphasised.

This study explores the performance disruption and impact of data poisoning within an industrial context, with a use case of a material recycling facility (MRF). The work presented in (Baiyekusi et al., 2024), showed the value of applying an ML-based approach in spectrum sharing within industrial MRF settings. In this research paper, the authors have highlighted the significance of unlicensed spectrum in 5G networks (5G NR-U) and IEEE 802.11ax networks. By analysing the flexibility limitations of using these networks in industrial use cases, they proposed an ML-based data-driven approach to calculate the optimal performance metrics in shared unlicensed bands between these networks. Despite the usefulness of these ML models, they are found to be highly vulnerable to various adversarial attacks Sharma et al. (2019), Tu et al. (2021). Data poisoning attack Li et al. (2024), Wang et al. (2023) is one of these attacks which poison the training dataset of ML models and transmit poison to these cellular networks. However limited research is provided to study the impact of data poisoning on unlicensed shared spectrum networks in industrial scenarios.

This case study leverages the implementation of a novel multiclass data poisoning attack, based on the multi-network scenario adopted in (Baiyekusi et al., 2024), to poison the data-driven model predictions in 5G NR-U and 802.11ax shared spectrum networks. With this experimentation, the effectiveness and impact of multiclass data poisoning in an industrial scenario were analysed. Four supervised models: SVM, RF, DT and MLP were selected. The contribution of this case study is to analyse the impact of a novel poisoning attack in an industrial 5G private network and evaluate the efficacy and impact of data poisoning in 5G and 802.11ax WIFI shared-spectrum networks.

4.8.1 Attack Model

This case study describe the scenario of the MRF. The MRF consists of a network of conveyor belts through which the recycled wastes are moved through the facility to be sorted by robotics arms. The robotic arms are equipped with cameras that feed the edge server with live video of moving waste materials. Object detection is performed at the edge server using computer vision, and the robotic arm selects any material of interest to be separated and recycled. Due to stringent data rates and

delay requirements, the network performance needs to be optimal and stable, otherwise, this could significantly impact MRFs operations negatively. The robotic arms are connected to the edge server via the 5G NR-U network. The 5G NR-U network offers resilience in data rates and delay through its error correction handling scheme. For monitoring purposes, the MRF is also equipped with digital display boards which offer visualisation of operations. This displays also offer live feed from CCTV around the MRF. The network traffic to the digital display screens have lesser priority but still required to function at an acceptable level. These digital display boards are connected to the edge server through the 802.11ax network. Given, the coexisting scenario of both 5G NR-U and 802.11ax, the ML-based spectrum sharing technique proposed in (Baiyekusi et al., 2024), ensures each network maintains expected performance based on the number of nodes contending over the channel and the expected data rates.

The ML-based spectrum sharing scheme proposed in (Baiyekusi et al., 2024) is predicated based on

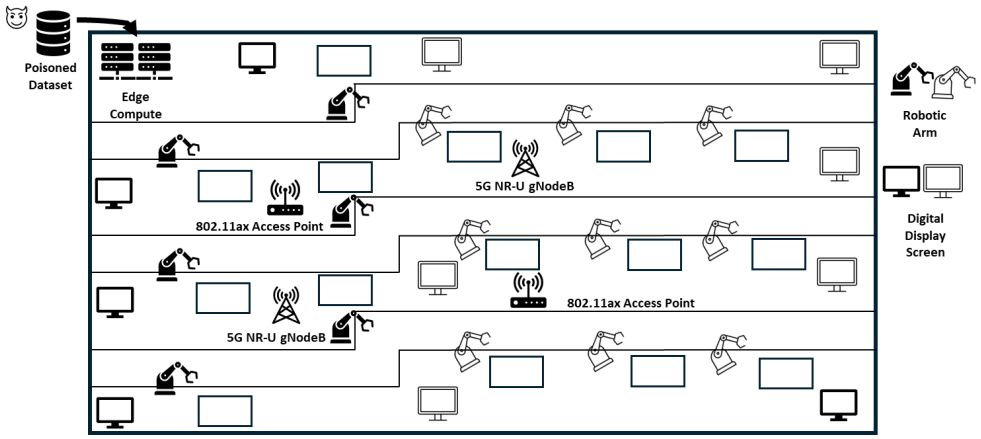


Figure 4.17: Material recycling facility scenario - A network of conveyor belts

5G NR-U and 802.11ax and operates relatively similar channel access schemes. The channel access scheme follows an arbitration process where transmission over the channel is made after channel sensing is performed and a random backoff period has been observed. The random variable for the backoff period is selected using a uniform distribution, which ideally offers fairly similar channel access to all nodes contending for transmission over the channel. The ML-based spectrum scheme proposed enables estimation of the number of nodes contending over the channel in a mixed technology scenario and adjusts the contention window for each node based on the network it is transmitting over i.e. either 5G NR-U or 802.11ax.

Experimental Dataset This case study has extended the dataset, developed in the research paper (Baiyekusi et al., 2024). This dataset is developed in MATLAB. This dataset is created with various on-networks devices in the range of 8-40 at a scale of 8. These devices are sharing the network spectrum of 5G NR-U and 802.11ax networks. Onboarding different devices on different range between

8-40, data rates and bandwidth of these devices is calculated under various transmission time and transmission delay. Transmission time and transmission delay is set to 5ms, 6ms, 8ms and 10ms. Data rates and bandwidth of these devices are measured at different contention stages for both 5G NR-U and 802.11ax networks. The contention stages to five states from 1-5 with a minimum $32 \leq$ contention rate $\leq 2^n$ and n is in the range of 6-10 at the scale of 1. The details of the dataset properties are given in Table 4.16.

Attack Method In this case, the dataset captured in real-time from the 5G network is poisoned.

Table 4.16: Properties of the shared-spectrum 5G NR-U and 802.11ax dataset

S.No.	Dataset Property	Property Description
1	No. of nodes	Number of devices on network
2	Contention stage	Wait after transmission frame
3	Data rate	Data rates based on transmission time
4	Transmission delay	Delay transmitting data

Following the threat model, only access to the dataset is provided to poison without providing any other knowledge of the target system. The details of the attack method are given as follows.

The OOP attack on the shared-spectrum 5G NR-U and 802.11ax networks dataset is implemented to structure a successful attack within an industrial 5G private network. Considering realistic attack settings, the attack in a grey-box scenario is implemented without leveraging the adversary to access the target model. Only dataset access is provided to the adversary; therefore the attack is implemented based on surrogate models to poison the dataset. The mathematical notation of developing surrogate models M_s is given in Eq 4.16.

$$M_s = \mathcal{T}(M(D;\theta), \Delta L) \tag{4.16}$$

where, D is the 5G private network dataset, θ are the model configurations and poisoning level $\Delta L = (10, 15, 20)\%$. The surrogate model M_s allows the adversary to develop the poisoned datasets D'. The mathematical notation for developing D' is given in Eq 4.17.

$$D' = \sum_{i=1}^{n \to \Delta L} f_p(f_{l'}([max\ \vec{d}(x_i)], l'_i), \Delta L) \tag{4.17}$$

where, $f_{l'}$ is the function of manipulating the label l of the targeted data point x and changing it to a new label l', f_p is the function of developing poisoned dataset D' with a dependent function $f_{l'}$ with poisoning level $\Delta L = (10, 15, 20)\%$. $f_{l'}$ is the function of updating the label of the farthest data point in the dataset. Consequently, by providing the poisoned dataset to the ML model, the

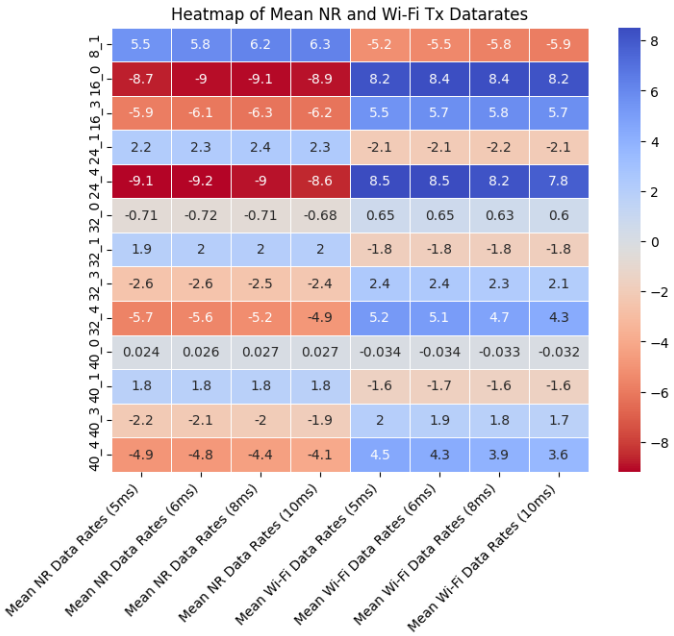


Figure 4.18: Mean data-rate poisoning performance results for multilayer perceptron showing performance across multiple nodes and contention window stages

shared-spectrum network is poisoned.

By implementing the OOP attack, the ML model has misclassified contention rates which subsequently manipulate data rates and transmission rates, leading to a disruption in the network. With this network poisoning, the impact of data poisoning attacks on 5G and 802.11ax shared-spectrum networks is evaluated at various poisoning levels. Overall, it shows a high impact on the optimised shared-spectrum network implemented with SVM followed by DT. Furthermore, Section 4.8.2 gives an in-depth analysis of the experimentation results.

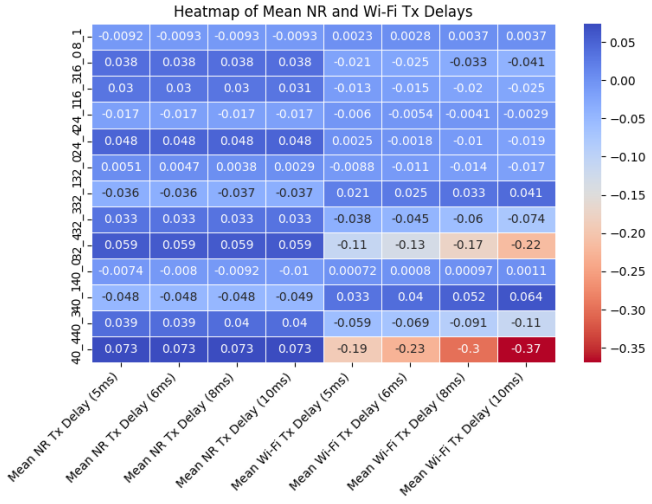


Figure 4.19: Mean delay poisoning performance results for multilayer perceptron showing performance across multiple nodes and contention window stages

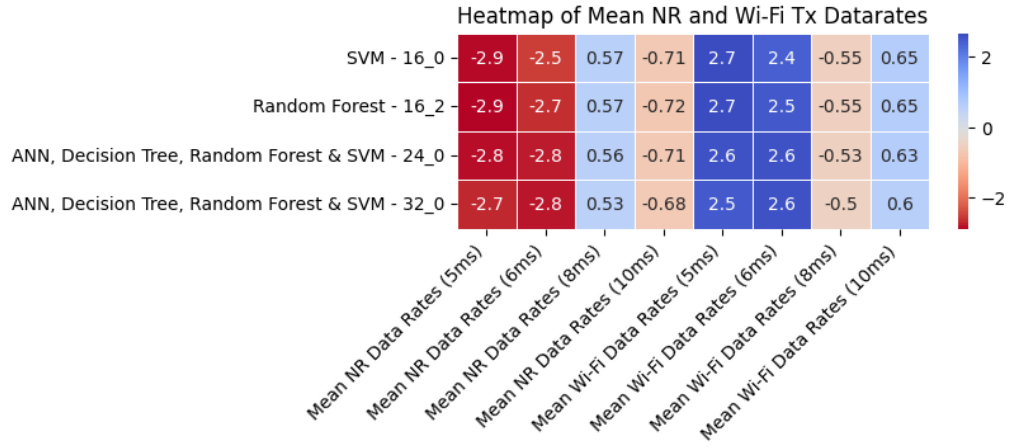


Figure 4.20: Mean data-rate poisoning performance results for other models showing performance across multiple nodes and contention window stages

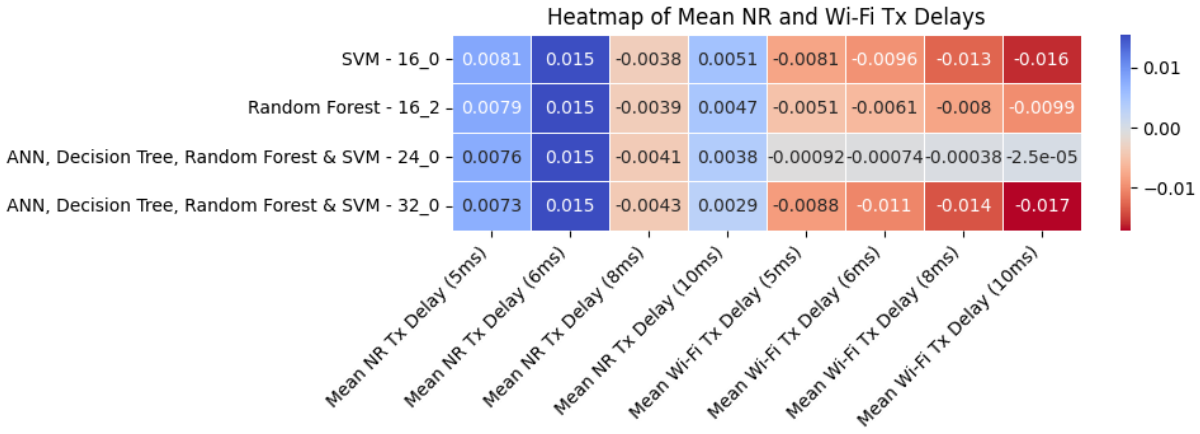


Figure 4.21: Mean delay poisoning performance results for other models showing performance across multiple nodes and contention window stages

4.8.2 Experimentation Results and Analysis

The performance of various poisoned ML models against multiclass data poisoning in a shared-spectrum private 5G network is assessed to analyse their performance against cleaned models. Fig. 4.22 highlights the decrease in model accuracy with the OOP data poisoning attack. A linear decrease is visible on the test time accuracy of models with an increase in data poisoning level ΔL where $10\% \leq \Delta L \leq 20\%$ at a scale of 5. SVM is the most affected model and MLP has minimal effects of data poisoning. Further, Fig. 4.23 highlights an abrupt change in the false discovery rate of models where RF and DT are highly disrupted at $\Delta L = 20\%$ and SVM and MLP show the highest error rates at $\Delta L = 15\%$. Overall, SVM has shown the highest poisoning effects with a decrease of 6.6% accuracy and MLP proved to be more stable against data poisoning with 2.9% accuracy decrease at $\Delta L = 20\%$.

The network performance results shown in Fig. 4.20 to Fig. 4.21 highlight the impact of the data

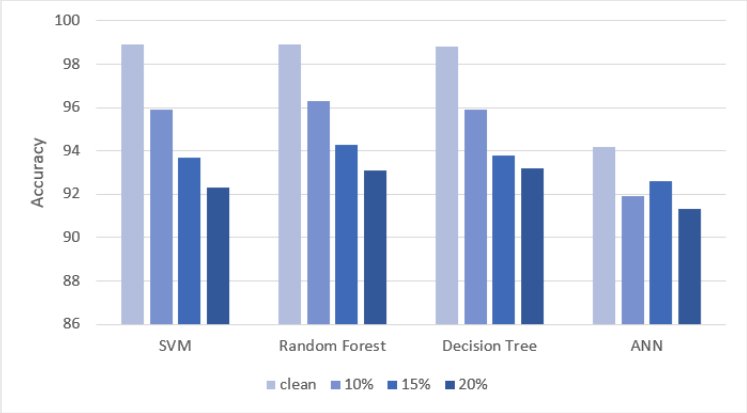


Figure 4.22: Decrease in model accuracy with data poisoning attack

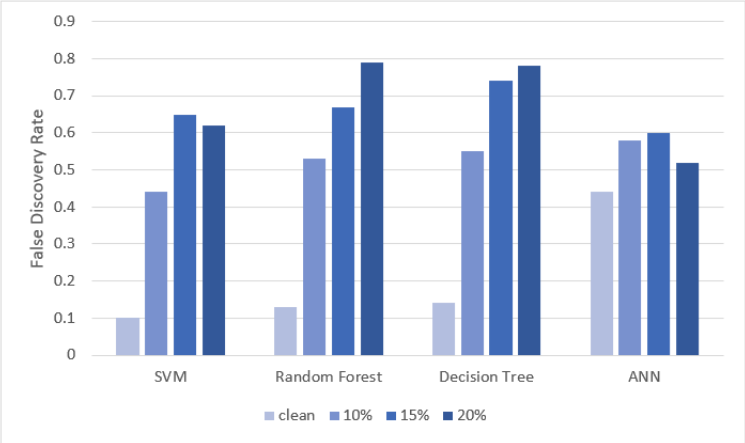


Figure 4.23: Increase in model false discovery rate with data poisoning attack

poisoning attack. The data rates and transmission delay constitute the main metrics for evaluating the impact on performance. Fig. 4.20 shows a heatmap of the impact of the poisoning attack on the MLP model on NR and WiFi data rates in light of their severity. It can be seen that 5G NR-U is more impacted. This may be due to 5G NR-U network's slightly better modulation and coding scheme (MCS) and less management data overheads. Consequently, the impact of the poisoning attack is more noticeable for 5G NR-U when node numbers and contention windows are wrongly estimated. Given the industrial scenario being evaluated, the robotic arms operate over the NR networks. The wide variations with smaller nodes, e.g. 16 and 24 nodes, indicate a significant impact on the performance of the robotic arms across the recycling centre. For instance, a drop in data rates by 8.7Mbps for each robotic arm will lead to significant degradation in the operations of the robotic arm and potentially reduce the MRF's output. From a delay perspective, the 802.11ax network shows a more negative impact of the poisoning attack. The delay reflects more on 802.11ax due to additional overheads for data transmission; hence, any wrong estimation of the number of nodes and contention window will further impact the delay profile and performance of the 802.11ax network.

4.9 Discussion and Limitations

A novel grey-box attack is formalised to attempt poisoning multiclass models, describing their efficacy and analysing the factors affecting their classification behaviour. Although several adversarial poisoning techniques are proposed in the literature, limited experimentation is provided on multiclass classifiers. Existing research papers Steinhardt et al. (2017), Chen and Koushanfar (2023), Hayase et al. (2021), Weerasinghe et al. (2021) proposed solutions that focus on discrete data set features and detect outliers to lessen poisoning effects. The attack has taken the outliers into the feature space that effectively poisoned the model. Following this, certain factors affecting individual algorithms were highlighted and effective levels of poisoning for parametric and non-parametric multiclass models were also determined. These results showed that a 10% poisoning rate shows the highest performance degradation for parametric models and 15% for non-parametric models. At these optimal poisoning, a lower level of model sensitivity is analysed, which does not allow the model to over-fit, highlighting the efficacy of the OOP attack.

Implementing the OOP attack, this chapter conducted a deep behavioural analysis of multiclass ML, identifying factors affecting the confidence of models. From these results, GNB and KNN are found to be highly affected by this poisoning attack, whereas DT and RF are less affected models. Manipulating the outliers class label, class probabilities of GNB, and proximity distance calculation of KNN are highly disrupted.

These results highlighted that the dataset size and number of classes are inversely proportional to poisoning effects. Whereas, an accelerating impact of an imbalanced dataset on model poisoning. Imbalanced classes in multiclass datasets help penetrate poison in the model effectively, to an extent. Also, a fundamental relation between dataset noise and data poisoning is found where dataset noise works as a catalyst towards poisoning, but becomes impractical with poisoning level $> 15\%$ with unrealistic performance degradation.

Following these results, two case studies were conducted to examine the impact of OOP poisoning in real-world applications. In this first case study, the efficacy of adversarial training is studied against data poisoning attacks in two traditional ML models in non-complex settings. Various literature studies Shi et al. (2018), Lu et al. (2024) experimented data poisoning against adversarially trained models mostly on DL models, including many complex Convolutional Neural Networks(CNN). This case study analysed how data poisoning, if 10% poison is injected, impacts adversarially trained SVM, RF, and MLP in their simplified forms. These results showed that SVM is highly vulnerable to data poisoning and adversarial training does not work on SVM and RF, as expected. However, is significant in making MLP resilient.

These results highlighted that after implementing adversarial training with the FGSM attack, the

integrity of the MLP model is intact and is not affected by data poisoning attacks. In contrast, the model integrity of the SVM model is breached. The test time prediction variance of the SVM model is significantly high when trained with a cleaned dataset of 0.9 and a higher variance of the model is calculated when trained with 10% poisoned dataset reaching 1.15. Whereas, RF and MLP have significantly lower variance compared to SVM. Because the attack is implemented perturbed outliers and RF is resilient to outliers, in general, this leads to a lesser impact on the RF model.

Trustworthiness and robustness of smart healthcare applications should be of the highest priority. The results highlighted a major impact of data poisoning that worsened the diagnosis mechanism of these applications. The results of this case study highlighted the need to secure the traditional ML models that provide good results in the diagnosis of diseases at an early stage. But these models are the target of adversaries who try to spoil these security-sensitive applications.

Next, the impact of OOP poisoning in 5G industrial testbed network is analysed. The results from this study show the impact of data poisoning on a private 5G NR-U and 802.11ax network. It is clear, depending on the level of poisoning, data poisoning introduces instability in the network, causing the network to perform differently from how it is designed to operate. Hence, the attack can significantly degrade the performance and consequently the industrial output, making it not fit for purpose. The results show a high impact of data poisoning in violating the integrity of advanced networks. Data poisoning against four ML models is evaluated, implemented for network optimisation. This analysis highlighted that data poisoning has minimal impact on the MLP at all three poisoning levels, whereas the highest disruption in the integrity of SVM can be seen. Fig. 4.20 shows the highest impact on SVM with a value of -2.9 NR data rates transmission with 16 nodes, and similarly, Fig. 4.21 highlights a maximum transmission delay of -0.016 and -0.017 for SVM with 16 and 32 nodes, respectively. Also, Fig. 4.22 shows a continuous decrease in performance for SVM, RF, and DT irrespective of their design, highlighting the efficacy of data poisoning on wireless networks.

This chapter is limited to the analysis of classification algorithms which can be extended to the regression algorithms. With this limitation, the factors affecting classification behaviours and their confidence in this poisoning attack were analysed. Also, comparing this attack with existing attacks from the literature helps demonstrate the efficacy of the OOP attack which is also out of the scope of this chapter.

4.10 Summary

This chapter extends the answer to RQ 2 to highlight risks, impact, and consequences of novel data poisoning attacks in real-world applications in multiclass settings. It analyses the behaviour of multiclass models developed for real-world applications, identifying individual characteristics of the algorithms against OOP data poisoning to understand how the decision process can be compromised

in multiclass classifiers. It also addresses RQ 3 by highlighting the limitations of adversarial training in securing traditional ML models against data poisoning attacks and the need for improved mitigation solutions to secure ML models from poisoning attacks.

This chapter examines the consequences and impact of data poisoning in healthcare and Industry 4.0 network applications, emphasizing the necessity for robust and generalisable security solutions that would be applicable to both ML and DL models. This examination assessed individual algorithms against the OOP attack, identifying their key vulnerabilities in their characteristics. These results and limitations of existing mitigation solutions enable us to propose an improved attack-agnostic solution in the next chapter. Such a solution will address the limitations of existing adversarial training mechanisms and is adaptable to both ML and DL models.

Chapter 5. SecureLearn: Improvised Machine Learning Defence

5.1 Introduction

Chapters 3 and 4 demonstrates the effectiveness of data poisoning attacks in binary and multiclass classification models and highlight the limitations of existing defences against these attacks. Chapter 3 determined the underlying changes in the learning dynamics with poisoned training, whereas Chapter 4 emphasised the impact of data poisoning by exploiting a training characteristic of ML. These findings highlight the need to develop an attack-agnostic solution, securing models from evolving data poisoning attacks. This chapter answers RQ 3 and proposes an attack-agnostic solution called SecureLearn and analyses the limitations of existing solutions to mitigate data poisoning attacks and highlights the effectiveness of the proposed mitigation solution as an attack-agnostic defence. By answering RQ 3 , it fulfills objectives 3 and 4. Objective 3 is achieved with the task of proposing a robust and generalised solution and objective 4 is aligned with proposing a novel mechanism called the 3D evaluation matrix. Previously, no such matrix was proposed in the literature to assess defence solutions thoroughly across various dimensions. The 3D evaluation matrix evaluates the defensive capabilities of SecureLearn from three dimensions: data poisoning attacks, data sanitisation, and adversarial training, across three attacks and two existing defences.

Existing literature highlights several techniques, such as Meng et al. (2022), Tao et al. (2021), to mitigate data poisoning attacks; however, these are largely attack-specific or algorithm-specific. For example, the research study (You et al., 2019) added a noise layer in neural networks to regularise the adversarial noise in these models. However, this approach does not apply to other models, like SVM, RF, and DT, as they do not comprise layers of nodes in their architecture. Similarly, adversarial training is one of the solutions that improves the adversarial robustness of DL models and does not applies to traditional models. The experimentation, presented in Section 4.7 in Chapter 4, highlighted the limited effectiveness of adversarial training in securing traditional models. These limitations highlighted the need for a generalised solution that is independent of the model architecture and capable of countering current and evolving threats. Therefore, this chapter proposes an attack-agnostic solution, SecureLearn, to mitigate data poisoning attacks in multiclass ML.

SecureLearn offers an enhanced data sanitisation that combines the fundamental principles of nearest neighbor voting strategy to correct data labels, followed by calculating the statistical deviations of each data point to detect and correct anomalies. Furthermore, SecureLearn introduced a new approach of feature-oriented adversarial training (FORT) influenced by a common characteristic of feature importance score of ML to identify important data points to generate adversarial examples

for training.

A 3D evaluation matrix is proposed in this chapter to thoroughly assess SecureLearn, following three orthogonal dimensions: data poisoning attack, data sanitisation and adversarial training. The experiments are conducted on four ML algorithms: RF, DT, GNB and MLP, trained with three differently structured datasets: IRIS (Fisher, 1936), MNIST(Deng, 2012), USPS (Hull, 2002). Selecting these algorithms allows this study to cover most classification models. SecureLearn is evaluated against three distinct data poisoning attacks and compared with two state-of-the-art mitigations, highlighting the better performance and generalisation of SecureLearn over others. The contributions of the chapter are given as follows.

- SecureLearn is the first defence solution that works against all types of data poisoning attacks and is adaptable for traditional classification models and neural networks. SecureLearn provides defence with minimal defender capabilities, without requiring prior knowledge of attacks and configurations of the targeted model.
- This chapter proposes a new adversarial training mechanism called FORT as a component of SecureLearn, enhancing the adversarial robustness of traditional multiclass ML, including neural networks. The results show that FORT improves the adversarial robustness of the model with a minimal trade-off between accuracy and robustness, i.e., the accuracy is decreased $< 3\%$, while enhancing the adversarial robustness.
- This chapter proposes a new 3D evaluation matrix to comprehensively evaluate SecureLearn against three data poisoning attacks and compare it with two existing defences Chan et al. (2018b), Paudice et al. (2018b). The evaluation is set up for four types of ML models trained with three distinct datasets. The results highlight that SecureLearn has outperformed other mitigations and is effective against all selected attacks for all models.

5.2 Related Work

5.2.1 Existing Multiclass Poisoning

Data poisoning attacks showed success in perturbing traditional ML in multiclass settings. The Outlier-Oriented Poisoning (OOP) attack (Paracha et al., 2025a), discussed in Chapter 4, manipulated the feature space by exploiting outliers and successfully poisoned six ML models. The research paper (Biggio et al., 2012) introduced a label flipping attack to perturb dataset labels, which can be extended to multiclass ML. Jagielski et al. (2021) introduced a subpopulation poisoning that injects a cluster of poisoned points in a dataset. Pantelakis et al. (2023) poisoned multiclass IoT networks with JSMA, FGSM, and DeepFool attacks and evaluate performance disruption.

5.2.2 Limitations of Existing Defences

Various defences are proposed in the literature to secure ML and DL models from data poisoning attacks. Such as the research (Peri et al., 2020) developed a deep k-NN to remove clean label poison by detecting falsified data points with k-neighbours. Deep k-NN defence is experimented against feature collision and convex polytope in deep neural networks. Paudice et al. (2018b) used the k-NN algorithm to mitigate label poisoning in binary SVM. Carnerero-Cano et al. (2023) computed the limitations of hyperparameters to resist data poisoning impact on DNN models. Barreno et al. (2008) has given the concept of reject on negative impact to remove affected data points, which is extended in (Chan et al., 2018b) to filter poisoned data from the given dataset.

Adversarial training is useful in improving the adversarial robustness of ML/DL models. In Ho et al. (2022a), Tao et al. (2021), Shafahi et al. (2020), adversarial training is implemented to improve the robustness of models against data poisoning attacks. However, these adversarial training methods are experimentally limited to neural networks and DL models because it is designed following gradient learning of the model at each iteration of training, which does not apply to traditional models, which do not follow gradient learning, and so makes adversarial training ineffective in securing traditional ML models.

Conclusively, some attack-agnostic solutions have been proposed in the literature; however, these are designed to secure DL models. A few solutions are proposed to secure traditional ML, mostly improving the robustness of binary models, whereas no prominent solution is proposed for multiclass ML. SecureLearn is the first attack-agnostic solution, designed to secure multiclass ML against data poisoning attacks. It is also adaptable to DL and binary models and effective against various aforementioned attacks. A brief comparison of existing solutions with SecureLearn is provided in Table 5.1, highlighting that existing solutions have either proposed data sanitisation or adversarial training, where data sanitisation solutions are experimented on binary ML models and adversarial training is experimented with only DL models.

Table 5.1: Summary of existing similar defences against data poisoning attacks proposed in various settings

Research paper	Data Sanitisation	Adv. Training	ML model	Model Settings
Chen et al. (2021)	✓	✗	GAN, CNN and LASSO	Binary and Multiclass DNN
Paudice et al. (2018b)	✓	✗	Stochastic Gradient Descent	Binary ML
Chan et al. (2018b)	✓	✗	SVM	Binary ML
Barreno et al. (2008)	✓	✗	SVM	Binary ML
Shafahi et al. (2020)	✗	✓	ResNet and InceptionV1	Multiclass DNN
Tao et al. (2021)	✗	✓	VGG-16, VGG-19, ResNet-18, ResNet-50 and DenseNet-121	Multiclass DL
SecureLearn	✓	✓	DT, RF, GNB, MLP	Multiclass ML

5.3 SecureLearn Overview

The problem of poisoning the training dataset is defined as follows: D_c represents the clean dataset, while D'_c denotes the poisoned substitute in the dataset. The combined dataset is defined as $D_o = D_c \cup D'_c$. SecureLearn relies on the general observation that the poisoned dataset tricks the model training to classify differently from the clean dataset, resulting in performance degradation. Therefore, SecureLearn identifies anomalies and misalignments in the features and labels to sanitise the dataset. Since no ground truth is provided, SecureLearn aims to sanitise D_o to correct data points and align features by identifying outliers in each dataset class and updating class labels by averaging the labels of nearest neighbouring data points. Furthermore, it quantifies the drift in features in each data point and filters those that exceed limits from the normalised dataset. Additionally, SecureLearn enhances the adversarial robustness of the model through FORT training to increase the reliability against new attacks. The complete process of SecureLearn is illustrated in Fig. 5.1. The algorithm of SecureLearn is provided in Alg. 7.

As an attack-agnostic solution, SecureLearn determines and relabels the uncertain data points in the dataset, regardless of being poisoned or outliers, whereas it safeguard mislabeling by calculating the average confidence of the neighbour data points following Eq. 5.1. Its efficacy is limited to generalised scenarios, which can be extended and integrated in specialised applications.

Algorithm 7 SecureLearn Mitigation Mechanism

Input: Training Dataset D_o, perturbation limit ε, feature importance scores: F
Initialise: b=0.001, c=0.01, nearest neighbours (k)=7
for $x_i \in D_o$ **do**
  $d = min(k, dist(x_i, x))$
  $l_i = avg(x_i, d)$
  $D_{san} \leftarrow (x_i, l_i)$
end for
for $x_i \in D_{san}$ **do**
  $Compute\ \delta_i\ following\ Eq.\ 5.5$
  if $\delta < |g|$ **then**
    $D_{san} \leftarrow (x_i, l_i)$
  end if
end for
if $M == M_{GNB}\ or\ M_{MLP}$ **then**
  $F \leftarrow \arg max\ Probability(D_{san})$
end if
if then$(M == M_{RF}\ or\ M_{DT})$:
  $F \leftarrow \sum_{i=1}^{L} f_i(1 - f_i)$
end if
for $(x_i \in D_{san})$ and $(f_i \in F)$ **do**
  $D_{adv} \leftarrow \mathbb{E}_{(x,y)\sim D_o}[\mathcal{L}(M, (x_i + (c * sign((f_i * x_i) + b)))$
end for

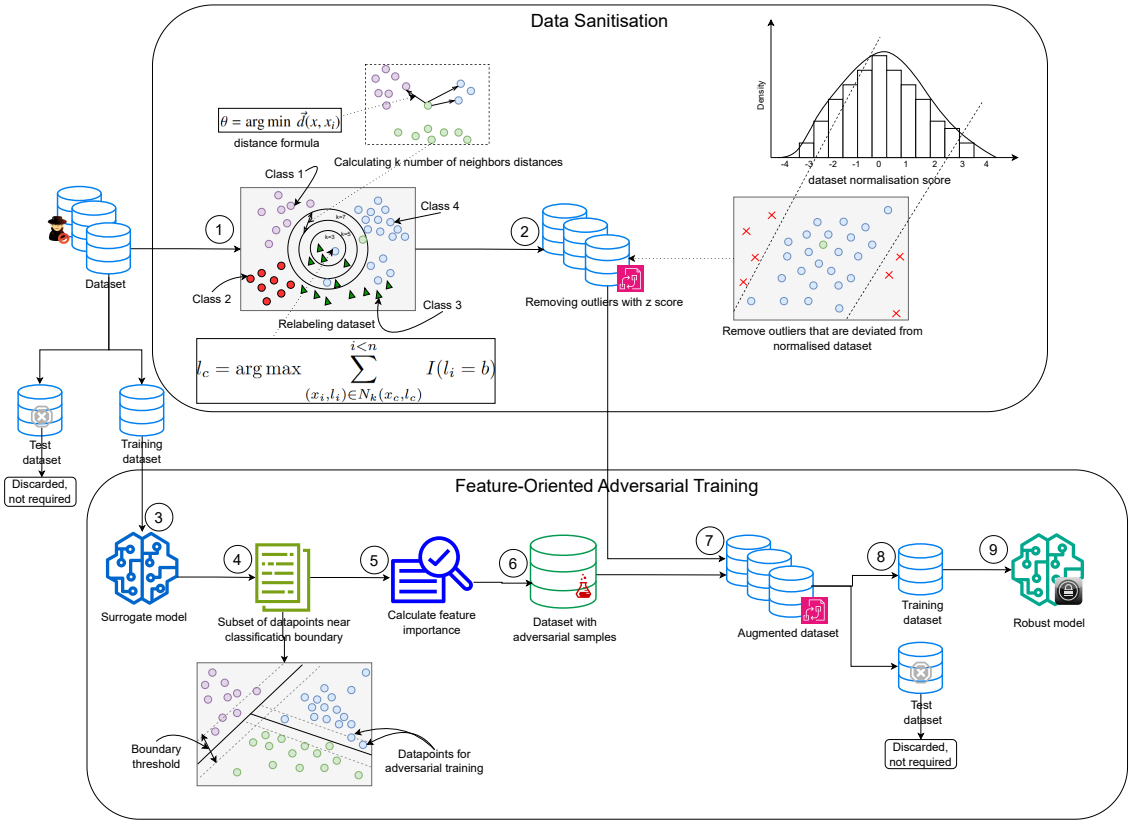


Figure 5.1: Architectural overview of SecureLearn illustrating a two-layer approach to secure the training pipeline of a machine learning model

5.3.1 Improvised Data Sanitisation

The data sanitisation module of SecureLearn comprises two parts: relabeling the data points with corrupted labels in D_o and removing anomalies to produce a sanitised dataset D_{san}. The relabeling mechanism is defined as:

$$D_{san} = \{(x, l)|x \in D_o\}$$
$$and\, l_i = \begin{cases} l_i\, if\, C(x_i, l_i) < \gamma \\ l\, if\, C(x_i, l_i) \geq \gamma \end{cases} \tag{5.1}$$

where $C(x, l)$ is the label confidence of neighbouring data points, l_i is the existing label of the data point x_i, l is the new label with the highest confidence received from the nearest data points. The confidence limit is defined as $\gamma \geq 40\%$ neighbouring votes, following an incremental majority voting approach (Abdulboriy and Shin, 2024). The optimal threshold for voting confidence to detect irrelevant data points is identified using a brute force mechanism and experimented with a threshold between 30% and 60% average of neighbouring votes. This detection mechanism identifies irrelevant data points that may be intentionally perturbed or misaligned data points, regardless of the dataset's application. The calculation of the label of each data point, given in Eq. 5.1, follows the confidence

score $C(x, l)$ of neighbouring data points, calculated with Eq. 5.2.

$$C(x, l) = \arg\max \frac{1}{k} \sum_{(x_j, l_j) \in \theta}^{j<n} I(l_j = l_c) \tag{5.2}$$

where l_c is the original class label, k is the number of nearest neighbours set to seven following the kTree method given by Zhang et al. (2017), x is the data point with label l and θ is the function of distance measure, given in Eq. 5.3.

$$\theta = \min \vec{d}(x_i, x) \tag{5.3}$$

The next step of data sanitisation is to remove outliers from the dataset. The anomalous data points are removed from the dataset, where the deviation of the given data point exceeds the limits of the normalised dataset distribution, following Eq. 5.4. The deviation δ is calculated with Eq. 5.5 where μ is the mean of the dataset and the deviation limit $|g| = 3$ (Abdi, 2007).

$$D_{san} = \{x_i \in D_o || \delta \leq |g|\} \tag{5.4}$$

$$\delta = \frac{x_i - \frac{1}{n} \sum_{i=1}^{n} x_i}{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}} \tag{5.5}$$

5.3.2 Feature-Oriented Adversarial Training

After obtaining the sanitised dataset, SecureLearn aims to improve the adversarial robustness of the model with feature-oriented adversarial training. In the literature, it is noticed that the existing adversarial training mechanism is unable to improve the resilience of traditional ML models (Paracha et al., 2024b) because existing approach follows the gradient-oriented training which is ineffective for traditional models, therefore SecureLearn introduced a new method to train models, where adversarial data D_{adv} is generated by augmenting data points with high feature importance score and lie near the decision boundary. This is done by solving Eq. 5.6, followed by generating the perturbation in Eq. 5.7.

$$D_{adv} \leftarrow \mathbb{E}_{(x,y) \sim D_o}[\mathcal{L}(M, ((x_i + \varepsilon), l) \tag{5.6}$$

where M is the training model, $\mathcal{L}$ is the training loss and ε is the perturbation given in Eq. 5.7.

$$\varepsilon = c * sign((f_i * x_i) + b) \tag{5.7}$$

where, in Eq. 5.7, f_i is the feature importance score of the model M, $c = 0.01$ is the perturbation constant. This constant is set following the average perturbation value, given in (Liu and Wen, 2021). x_i is the data point, and $b = 0.001$ is the non-zero coefficient which restricted the zero output.

Combining output of Eq. 5.1 and Eq. 5.6, the sanitised dataset D_s is given in Eq. 5.8:

$$D_o = D_{san} + D_{adv} \tag{5.8}$$

Intuitively, the model is trained to mitigate the data poisoning effects and improve the overall performance. Unlike traditional adversarial training based on gradient optimisation, FORT adds slight perturbations to the data points that are close to the decision boundaries of the model to widen these boundaries, making them robust to poisoning. This way, SecureLearn improves the security and robustness of ML models against data poisoning attacks. Next, this chapter extends the threat model, defined in Chapter 4, to evaluate SecureLearn as an effective defence.

5.4 Extended Threat Model

A threat model for this thesis is defined in Chapter 4 to study the adversarial impact of data poisoning attacks and draw attention to evolving threats in this domain. The threat model is extended with new attack vectors to assess SecureLearn's efficacy in mitigating multiclass data poisoning attacks.

5.4.1 Attack Goals

This threat model defines two attacker goals to assess the effectiveness of selected mitigation solutions. The first goal is to disrupt the model's availability and reduce its overall performance by employing the OOP attack (Paracha et al., 2025a) and label flipping attack (Shahid et al., 2022). The second goal is to harm the model's integrity by augmenting clustered poisoned data points employing the subpopulation attack to disrupt targeted class predictions (Jagielski et al., 2021).
Consider the poisoning of supervised classification models, e.g. RF or MLP, given the dataset $D_o = \{(x, l)\}_{i=1}^{n}$ with data points x and labels l of class c, the attacker can manipulate the labels l' or the features x' of the dataset or augment poisoned datapoints(x', l') into the dataset to prevent the trained victim model from attaining the intended performance.

5.4.2 Attacker Knowledge

In this threat model, the attacker possesses limited knowledge of the targeted model M and dataset D_o. Under these constraints, all selected data poisoning attacks are formulated as grey-box attacks. In this scenario, the attacker has a partial understanding of the dataset and model: the dataset and algorithm names are known, but the dataset distribution, model settings, and parameters remain unknown. Additionally, the attacker has no knowledge and access to the target system.

5.4.3 Attacker Capability

In these experiments, the attacker has leveraged the capability to poison the training datasets. Thus, the attacker can manipulate the training dataset in different ways. The attacker can modify labels or features of the dataset and introduce poisoned data points into the dataset. However, this capability is limited to injecting a maximum 20% poisoning level as the upper bound limit and a minimum 10% poisoning as the lower bound limit. These limits are defined as the most effective poisoning limits, as explained in Chapters 3 and 4 Paracha et al. (2024a), Paracha et al. (2025a).

5.4.4 Attack Strategy

In this threat model, three data poisoning attacks of varying attack vectors, i.e., OOP, SubP and RLPA attacks are considered. Following these attacks in multiclass classifiers, the effectiveness of SecureLearn is evaluated, demonstrating that it is an attack-agnostic and promising solution capable of mitigating all the aforementioned attacks.

5.4.5 Defender Capability

The capabilities of the defender, under this threat model, are leveraged to complete access to the dataset and the targeted model to assess selected mitigations. The data sanitisation (Paudice et al., 2018b) assumes full access to the dataset and algorithm used. The mitigating solution (Chan et al., 2018b) also requires full access to the model and the training data, as it compares the change in the model's performance for every data point in the training dataset. Conversely, SecureLearn is exempted from these capabilities and only requires access to the dataset, assuming that the dataset may or may ot be poisoned. This way, SecureLearn can be implemented in restricted and third-party solutions that do not provide complete access to their systems.

5.5 Experimental Datasets

This chapter utilises three publicly available datasets: IRIS, MNIST, and USPS. The IRIS and MNIST are extended from Chapter 4; however, the ISIC dataset is replaced with the USPS dataset, considering that ISIC is an unbalanced dataset. These datasets differ in structure, feature correlations, and number of classes. They have been widely used in studies of data poisoning attacks Drews et al. (2020), Wang et al. (2021c), Paracha et al. (2025a) and defences Xu et al. (2021), Jia et al. (2021), Zhang et al. (2022). The diversity of these datasets facilitates a comprehensive analysis of the adaptability and effectiveness of defence mechanisms against various poisoning attacks.

Table 5.2 presents the structural details of these datasets. The IRIS dataset contains 150 instances and three classes. The USPS dataset consists of scanned images of postal digits across 10 classes. In

contrast, the MNIST dataset comprises a large collection of digital images of handwritten digits.

The statistical correlations between the features of each dataset are given in Table 5.3. The spearman correlation highlights monotonic strength (van den Heuvel and Zhan, 2022) between dataset features and the p-value reflects statistical significance (Di Leo and Sardanelli, 2020). A low statistical significance highlights high variability in the dataset, which may strengthen the poisoning impact. Features in the MNIST dataset are highly associated and show high strength with a very low p-value of 0.0141, highlighting direct proportionality between its features. However, a low to nominal statistical significance is shown in the IRIS datasets with a p-value of 0.07, and spearman correlation of 0.12 shows a weak monotonic relationship between its features. In USPS, negative spearman correlation highlights an inverse monotonic relationship between the dataset features. In contrast, the dataset with a p-value of 0.2397 indicates a very low statistical significance of the dataset.

Table 5.2: Dataset description of all three datasets utilised in this study

Dataset	No. of classes	No. of features	No. of instances
IRIS	3	4	150
MNIST	10	784	70,000
USPS	10	256	9298

Table 5.3: Features correlation in dataset to analyse interdependence between features in individual dataset

Dataset	Spearman correlation	p-value
IRIS	0.1238	0.0791
MNIST	0.009282	0.0141
USPS	-0.008742	0.2397

5.6 Crafting Poisoned Samples

After the attacker's capabilities and goals are established, poisoned data points were constructed with selected data poisoning attacks. Considering the attacks discussed in Section 5.2, the selected attacks formulate poison from three different aspects: perturbing dataset labels, perturbing dataset features, and augmenting perturbed data samples. The details of crafting poison with each attack are given as follows.

5.6.1 Random Label Poisoning Attack

One of the most common and early proposed attacks against supervised models is the label poisoning attack Nguyen et al. (2023), Paudice et al. (2018b). Though initially proposed against binary classifiers, it is later extended to manipulate multiclass models by randomly perturbing dataset labels (Rosenfeld et al., 2020). In the random label poisoning attacks (RLPA), the computation of

perturbations is simple. Given the poisoning limits, the labels of randomly selected data points were changed, ensuring that the targeted label is updated to another class label. Let $D'_{train} = \{(x_i, l'_i)\}_{i=1}^{n}$ be the poisoned training dataset where l_i is manipulated with RLPA as:

$$l'_i = Y/\{l_i\} \tag{5.9}$$

where $l \in Y$ and features x_i of ith data point remains intact. Following the threat model of this chapter, the poison is constrained based on the poisoning limits ΔL. The training dataset is manipulated with Eq. 5.10.

$$D'_{train} = f(D_{train}, \Delta L) \tag{5.10}$$

where f is the label manipulation function given in Eq. 5.12.

5.6.2 Subpopulation Poisoning Attack

The subpopulation attack (SubP) augments a poisoned cluster into the training dataset (Jagielski et al., 2021) instead of poisoning segregated data points. The attacker selects a random class and replicates its samples. In the next step, the attacker changes the labels of the replicated data points to another class and augments these data points into the training dataset to generate D' following the perturbation function given in Eq. 5.11.

$$\forall (x, l') \in D' \tag{5.11}$$

such that l' is generated with Eq. 5.12. Then, D'_{train} is given as follows.

$$D'_{train} = D' \cup D_{train} \tag{5.12}$$

5.6.3 Outlier-Oriented Poisoning Attack

This thesis proposed the OOP attack in Chapter 4. The OOP attack perturbs the feature space by interpolating outliers of varying classes. For this perturbation, the adversarial samples are generated by selecting the most distant data points in the dataset and changing their labels to manipulate the feature space of the multiclass classifier and shift its decision boundaries.

Let $D_{train} = \{(x_i, l_i)\}_{i=1}^{n}$ be the training dataset with O outliers, such that the OOP attack calculates the distance vector of each data point, finding the most distant points. The distance is calculated given Eq. 5.13.

$$\arg\max \ \vec{d}(b_c, X_c) \tag{5.13}$$

where b_c is the decision boundary of class c. Given the poisoning levels, ΔL defines the perturbation limits of the number of poisoned data points. The function of manipulation is given in Eq. 5.15.

$$D'_{train} = f(O, \Delta L) \tag{5.14}$$

such that, f implies to change label of ith data point (x_i, l_i) with Eq. 5.15:

$$l'_i = O/\{l_i\} \tag{5.15}$$

Following these discussed attacks, the 3D evaluation matrix is proposed to thoroughly evaluate SecureLearn.

5.7 3D Evaluation Matrix

This chapter proposed an innovative and comprehensive 3D evaluation matrix to evaluate defence solutions against data poisoning attacks, particularly SecureLearn in this study. This matrix evaluates SecureLearn in three dimensions and compares it with two typical defences against three data poisoning attacks as given in Table 5.1. The 3D evaluation matrix is given in Fig. 5.2. Its dimensions are explained as follows.

5.7.1 Dimensional Space 1

The *dimensional space 1 (DS1)* lies between the dimensions of data sanitisation and data poisoning attack. Here, SecureLearn is analysed by experimenting with it against three data poisoning attacks and by comparing it with two existing similar data sanitisation defences to highlight the effectiveness of the data sanitisation of SecureLearn. The DS1 evaluates the strength of mitigations and their attack-agnostic resistance to data poisoning attacks, followed by highlighting the profound performance of SecureLearn compared to other solutions.

5.7.2 Dimensional Space 2

The *dimensional space 2 (DS2)* lies between the dimensions of data poisoning attacks and adversarial training. In DS2, the effectiveness of the proposed FORT training component of SecureLearn is evaluated against selected data poisoning attacks and analysing improvements in the adversarial robustness of the model. In this dimensional space, SecureLearn is evaluated to distinguish between benign and poisoned data points and assess the resilience of the model against corrupted data points to minimise the impact on decision boundaries during training.

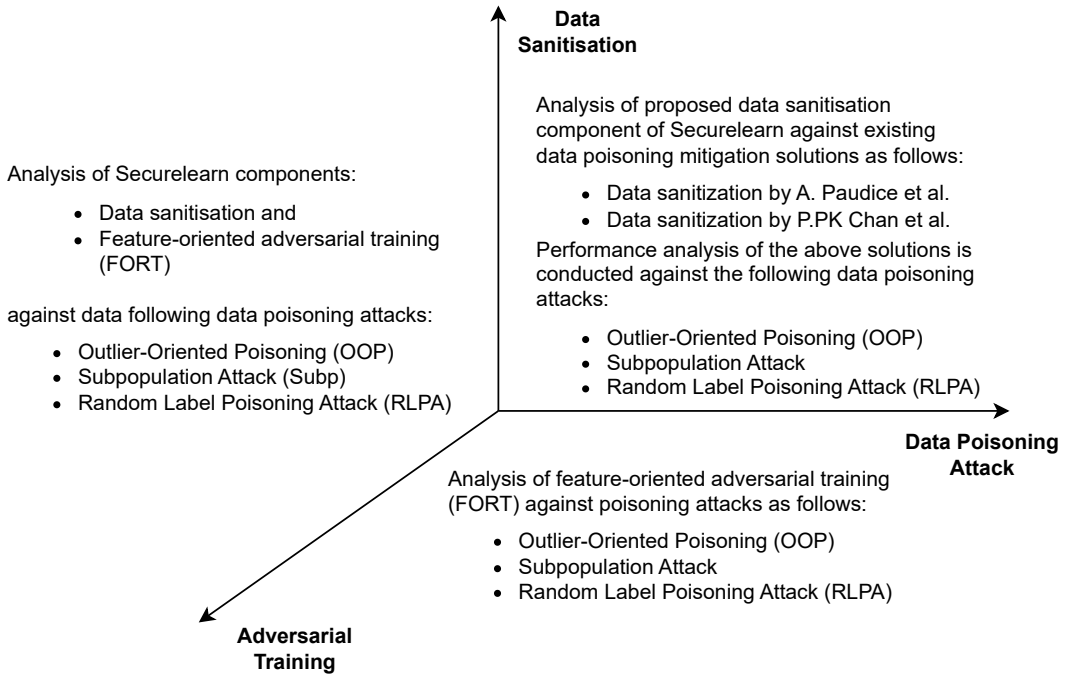


Figure 5.2: 3D evaluation matrix to evaluate SecureLearn three different aspects. The first dimension between data sanitisation and data poisoning attacks evaluates SecureLearn against selected data poisoning attacks and compares it with existing mitigations. The second dimension between data poisoning attacks and adversarial training evaluates FORT adversarial training of SecureLearn against selected data poisoning attacks. The third dimension between adversarial training and data sanitisation evaluates the overall performance of SecureLearn against selected data poisoning attacks

5.7.3 Dimensional Space 3

The *dimensional space 3 (DS3)* lies between the dimensions of adversarial training and data sanitisation. In DS3, the overall effectiveness of SecureLearn is evaluated in securing multiclass ML from reactive and proactive data poisoning attacks. The resilience and robustness of the model are assessed by analysing the false discovery rate of the model at varying poisoning levels against selected data poisoning attacks. The evaluation metrics used for the assessments in this chapter are discussed as follows.

5.8 Evaluation Metrics

To evaluate model performance in 3D evaluation matrix, the standard performance metrics: Accuracy, Recall and F1-Score are adopted. Furthermore, the detection rate (DR), correction rate (CR) and false discovery rate (FDR) are utilised for the detailed evaluation. The DR and CR prominently highlight the efficacy of SecureLearn in sanitising poisoned data points and FDR highlights the strengthened robustness of the model against poisoned training. Accuracy is the measure of correct classifications, where the poisoned data points remain disjointed in the incorrect classes and do not affect the model's

availability. Recall measures correct predictions of positive classifications over all positive answers, defining high separability. F1-score quantifies the overall defence performance, where the decision boundaries are aligned. Let the classification function be given in Eq. 5.16, the evaluation metrics can be found in Eq. 5.17, 5.18, 5.19.

$$f(C(x_t)) = \begin{cases} true & \text{if } x_t \in Class\ c \\ false & \text{otherwise} \end{cases} \tag{5.16}$$

where f is the classification function, x_t is the data point from the test dataset D_t split from D_o, and $C(.)$ is the class predictor. After sanitising dataset with SecureLearn, false positives(FP) is defined as $f_{tr}(C(x_{t_i})|l'_c)$, where l'_c is the wrong class label and false negative(FN) is defined as $f_{fs}(C(x_{tr_i})|l_c)$ where data points are not sanitised correctly. Whereas, true positive is defined as $f_{tr}(C(x_{t_i}))$ and true negative is defined as $f_{fs}(C(x_{tr_i}))$.

$$Acc = \frac{\sum_{i=0}^{n} f_{fs}(C(x_{t_i})) \wedge \sum_{i=0}^{n} f_{tr}(C(x_{t_i}))}{(x_t \in D_t)} \tag{5.17}$$

$$Rcl = \frac{\sum_{i=0}^{n} f_{tr}(C(x_{t_i}))}{\sum_{i=0}^{n}(f_{tr}(C(x_{t_i}))) \wedge \sum_{i=0}^{n}(f_{fs}(C(x_{t_i})))} \tag{5.18}$$

$$where\ f_{fs}(C(x_{t_i})) \in D_t$$

$$F1\_scr = \frac{\sum_{i=0}^{n} f_{tr}(C(x_{t_i})) * Rcl}{2 * \{(\sum_{i=0}^{n} f_{tr}(C(x_{t_i})) \wedge \sum_{i=0}^{n} f_{tr}(C(x_{t_i}))) + Rcl\}} \tag{5.19}$$

Let x' be the poisoned data point in D_o, and detection of these points with SecureLearn is given in Eq. 5.20, and setting these points in the appropriate class is shown in Eq. 5.21. After corrections, the false discovery rate of the model is analysed with Eq. 5.22.

$$Detection\ rate(DR) = \frac{\sum_{i=0}^{n} P(x'|l_c)}{\sum_{i=0}^{n} P(x|l_c) \wedge P(x'|l_c)} \tag{5.20}$$

$$Correction\ rate(CR) = \frac{\sum_{i=0}^{n} P(x' \rightarrow x|ll_c)}{\sum_{i=0}^{n} P(x|l_c) \wedge P(x'|l_c)} \tag{5.21}$$

$$False\ discovery\ rate(FDR) = \frac{\sum_{i=0}^{n} f_{tr}(C(x_{t_i})|l'_c)}{\sum_{i=0}^{n} f_{tr}(C(x_{t_i}|l'_c)) \wedge f_{tr}(C(x_{t_i}))} \tag{5.22}$$

5.9 Experimentation and Analysis

This section performed the experimental evaluation of SecureLearn using the 3D evaluation matrix. This section answers how SecureLearn is better at detecting and sanitising different types of data poisoning attacks. It first evaluates it in DS1 of the 3D evaluation matrix to analyse SecureLearn against three poisoning attacks and two existing data sanitisation techniques. The results are described in Sections 5.9.1 and 5.9.2. The evaluation of the FORT component is then conducted in DS2 of the 3D evaluation matrix. The results show that FORT effectively enhances the adversarial robustness of traditional ML models and neural networks against adversarial perturbations. The results are detailed in Section 5.9.3. Following the results of DS1 and DS2, the overall effectiveness of SecureLearn is further analysed at different poisoning levels in DS3. The relationship between increasing poisoning levels and the resilience of SecureLearn in mitigating the poisoning impact is identified, given in Section 5.9.4.

5.9.1 Detection and Correction Boundaries

The analysis begins by determining the detection and correction rates of each data poisoning attack, as presented in Table 5.4. The lower bound (LB) and upper bound (UB) of DR and CR are calculated at three defined poisoning levels using Eq. 5.20 and Eq. 5.21. Results show that SecureLearn detects at least 50% of poisoned data in training datasets, regardless of the type of attack or dataset used in these experiments. The minimum CR is approximately 30% for the RF model under the RLP attack, likely due to the unpredictable placement and impact of poisoned data points in untargeted attacks. The UB of DR and CR for SecureLearn reaches 100% against data poisoning attacks trained with IRIS dataset for most algorithms.

SecureLearn is highly effective in sanitising the IRIS dataset, followed by the USPS dataset, compared to the MNIST dataset, across all poisoning levels. These results indicate an inverse relationship between SecureLearn's performance and dataset size, with profound effectiveness observed in smaller datasets. SecureLearn is generalisable across different data poisoning strategies and dataset structures, performing independently of the number of classes in the dataset.

5.9.2 SecureLearn vs Existing Sanitisation Techniques

This section evaluates the effectiveness of the data sanitisation provided by SecureLearn. Model performance is analysed using Eq. 5.17 to Eq. 5.19, with the poisoning level set between 10% and 20%. Model accuracy after training with sanitised datasets is presented in Fig. 5.3 to Fig. 5.5. The results of these experiments demonstrate that SecureLearn consistently outperforms other solutions and maintains an accuracy of at least 90% across implemented data poisoning attacks. The recall

Table 5.4: Detection and correction boundaries of machine learning models after mitigating data poisoning attacks with SecureLearn

Algorithm	Dataset		Attack					
			OOP		Subp		RLP	
			LB	UB	LB	UB	LB	UB
RF	IRIS	DR	86.6	100	86.6	100	76.6	100
		CR	80	90.9	80	91	76.6	93.3
	MNIST	DR	56.3	65.5	56.3	66.3	52.4	66.3
		CR	33.5	49.2	33.5	49.2	29.7	47.6
	USPS	DR	87.94	89.13	56.29	65.78	50.48	62.56
		CR	44.47	49	38.42	44.54	35.22	43.24
DT	IRIS	DR	83.3	93.3	83.1	92	93.3	95.4
		CR	86.6	90.9	80	91	76.6	91
	MNIST	DR	49.6	66.7	49.8	66.7	46.4	64.1
		CR	44.69	57.88	45.1	58	44.97	55.08
	USPS	DR	44.69	57.88	44.69	57.88	44.97	55.08
		CR	15.98	36.93	15.98	37	18.1	34.51
GNB	IRIS	DR	100	100	100	100	80	100
		CR	93.3	100	93.3	100	66.6	93.3
	MNIST	DR	98.6	99.1	98.6	99.1	96	98.4
		CR	94.9	95.9	94.9	95.9	92.4	95.3
	USPS	DR	99.24	99.71	99.24	99.71	97.09	99.49
		CR	97.63	97.99	97.63	97.99	95.53	97.99
MLP	IRIS	DR	83.3	100	83.3	100	73.3	95.4
		CR	76.6	95.4	70	95.4	66.6	86.6
	MNIST	DR	56.3	65.5	56.3	66.3	52.4	66.3
		CR	59.33	49.2	33.5	49.2	29.7	47.6
	USPS	DR	71.16	85.36	70.79	84.7	64.28	82.5
		CR	59.33	78.9	59.11	79.76	51.47	76.42

and f1-score are provided in Table 5.5.

SecureLearn outperformed the mitigations proposed by Paudice et al. (2018a) and Chan et al. (2018b) in sanitising poisoned datasets. Compared to SecureLearn, the data sanitisation method proposed by Paudice et al. (2018a) achieved similar accuracy for DT with an average of 96%. SecureLearn provided an average recall of 84.22% with a 3% higher f1-score. Similarly, the average accuracy for GNB provided by Paudice et al. (2018a) is 94%, equivalent to SecureLearn; however, its recall and f1-score are 3.69% and 3.63% lower, respectively. Furthermore, the sanitised accuracy provided by Paudice et al. (2018a) dropped to 79% for the RLP attack and to 82% for the OOP attack when the model is trained with the MNIST dataset.

The data sanitisation proposed by Chan et al. (2018b) is highly unstable, particularly for MLP models. The accuracy of each model consistently decreases with increasing poisoning levels. For example, the accuracy of MLP substantially decreases after 10% poisoning, reached approximately 52% when trained on the IRIS and MNIST datasets, and 80% when trained on the USPS dataset. This instability arises because the method removes anomalous data points, which potentially decreases model accuracy. However, removing such data points also reduces the dataset size, which leads to underfitting, particularly in neural networks.

Table 5.5: Impact of data poisoning on recall and f1-score of secured machine learning models

Metric	Alg	Dataset	Defence	Attack								
				OOP			Subp			RLP		
				$\Delta L = 10\%$	$\Delta L = 15\%$	$\Delta L = 20\%$	$\Delta L = 10\%$	$\Delta L = 15\%$	$\Delta L = 20\%$	$\Delta L = 10\%$	$\Delta L = 15\%$	$\Delta L = 20\%$
Recall	RF	IRIS	A. Paudice et al.	**97.33**	92.85	**91.66**	91.88	88.09	91.66	87.17	92.85	80.55
			M. Barreno et al.	92.09	78.57	75.04	**97.43**	99.99	**96.07**	84.61	84.12	69.75
			SecureLearn	93.73	**99.99**	86.53	94.87	**99.99**	91.88	**94.87**	**96.96**	**92.09**
		MNIST	A. Paudice et al.	88.22	85.20	82.19	88.13	85.44	82.38	88.57	85.20	81.96
			M. Barreno et al.	**92.09**	78.57	75.04	**97.43**	**99.99**	**96.07**	84.61	84.12	69.75
			SecureLearn	91.31	**86.63**	**84.19**	91.34	86.61	84.38	**90.76**	**86.30**	**83.40**
		USPS	A. Paudice et al.	91.48	89.08	81.51	90.57	88.95	81.38	91.06	87.65	80.07
			M. Barreno et al.	86.84	81.14	80.50	83.26	80.35	80.40	82.85	75.96	75.86
			SecureLearn	**95.18**	**91.02**	**90.16**	**95.36**	**90.51**	**90.56**	**95.33**	**90.22**	**89.16**
	DT	IRIS	A. Paudice et al.	93.74	94.88	82.05	91.66	**97.91**	74.64	85.79	92.85	79.48
			M. Barreno et al.	86.66	81.81	77.77	**99.99**	93.93	**85.18**	84.70	84.84	69.62
			SecureLearn	**97.77**	**97.91**	**88.88**	95.55	94.21	84.12	**95.55**	**94.21**	**83.33**
		MNIST	A. Paudice et al.	**86.93**	81.84	78.09	86.90	81.38	78.21	**86.71**	81.93	**78.28**
			M. Barreno et al.	86.66	81.81	77.77	**99.99**	**93.93**	**85.18**	84.56	**84.84**	69.62
			SecureLearn	85.45	**85.13**	**78.20**	85.45	84.38	78.40	85.45	84.56	77.44
		USPS	A. Paudice et al.	85.67	80.39	80.63	86.14	79.60	80.41	86.27	80.12	79.61
			M. Barreno et al.	81.65	74.01	80.29	81.34	79.41	80.07	73.85	62.92	79.25
			SecureLearn	**87.42**	**81.51**	**81.00**	**87.37**	**81.58**	**81.50**	**87.40**	**81.55**	**79.82**
	GNB	IRIS	A. Paudice et al.	91.11	**94.11**	77.77	88.88	94.11	71.96	86.11	94.11	86.11
			M. Barreno et al.	85.18	84.40	85.30	92.59	**94.65**	94.74	90.74	86.96	94.74
			SecureLearn	**95.39**	92.59	**98.03**	**95.39**	94.44	**98.03**	**95.39**	**94.44**	**98.03**
		MNIST	A. Paudice et al.	57.12	60.34	58.98	58.39	57.78	52.72	56.49	59.48	50.50
			M. Barreno et al.	**85.18**	**84.40**	**85.30**	**92.59**	**94.65**	**94.74**	**90.74**	**86.96**	**94.74**
			SecureLearn	57.71	57.12	57.15	57.93	58.38	57.65	58.48	57.33	57.16
		USPS	A. Paudice et al.	75.39	73.11	77.28	74.01	**77.54**	75.01	74.01	77.67	76.64
			M. Barreno et al.	76.70	75.94	76.19	76.85	71.12	75.82	75.73	75.94	75.83
			SecureLearn	**76.97**	**78.16**	**77.50**	**77.34**	76.80	**77.23**	**76.57**	**77.93**	**77.26**
	MLP	IRIS	A. Paudice et al.	96.29	97.77	**99.99**	96.27	91.11	**97.22**	96.3	90.47	97.22
			M. Barreno et al.	31.11	28.61	36.01	36.30	18.72	28.51	33.92	16.34	28.51
			SecureLearn	**99.90**	**98.01**	99.90	**99.99**	**97.98**	96.96	**99.99**	**99.90**	**99.99**
		MNIST	A. Paudice et al.	96.29	**97.77**	**99.99**	96.15	91.11	97.22	96.29	90.47	97.22
			M. Barreno et al.	31.11	28.61	36.01	36.30	18.72	28.51	33.92	16.34	28.51
			SecureLearn	**97.93**	97.45	97.05	**98.08**	**97.82**	**97.37**	**97.32**	**97.60**	**97.25**
		USPS	A. Paudice et al.	96.29	82.92	83.52	96.30	81.05	79.69	96.29	81.04	82.33
			M. Barreno et al.	85.56	78.9	83.52	86.10	51.47	79.69	86.04	82.33	79.20
			SecureLearn	**98.42**	**97.76**	**98.40**	**97.69**	**98.19**	**98.05**	**98.36**	**97.87**	**97.06**
F1-Score	RF	IRIS	A. Paudice et al.	**97.33**	91.81	**91.65**	91.93	86.49	91.72	86.06	91.65	80.37
			M. Barreno et al.	91.98	75.94	72.38	**97.33**	99.99	**95.13**	83.59	84.12	68.05
			SecureLearn	93.73	**99.99**	86.58	93.88	**99.99**	91.94	**93.88**	**97.40**	**91.98**
		MNIST	A. Paudice et al.	86.05	82.90	78.60	85.94	83.13	78.88	86.27	82.91	78.40
			M. Barreno et al.	**91.98**	75.94	72.38	**97.33**	**99.99**	**95.13**	83.59	**84.12**	68.05
			SecureLearn	90.90	**84.46**	**81.54**	90.91	84.39	81.78	**90.31**	84.11	**80.65**
		USPS	A. Paudice et al.	91.36	88.60	80.65	90.45	88.53	80.49	91.00	87.18	78.82
			M. Barreno et al.	86.47	79.05	79.47	83.26	78.42	79.23	82.85	74.38	74.84
			SecureLearn	**95.17**	**90.85**	**88.94**	**95.36**	**90.44**	**89.34**	**95.26**	**90.09**	**87.96**
	DT	IRIS	A. Paudice et al.	93.52	94.88	78.80	91.31	**97.47**	73.68	89.98	85.85	75.42
			M. Barreno et al.	88.15	81.56	70.85	**99.99**	93.88	83.81	84.56	84.84	61.16
			SecureLearn	**97.77**	**97.16**	**89.16**	94.66	94.21	**83.82**	**95.53**	**94.21**	**82.50**
		MNIST	A. Paudice et al.	86.38	79.11	75.56	86.27	78.61	75.50	**86.08**	79.33	**75.75**
			M. Barreno et al.	**88.15**	81.56	70.85	**99.99**	**93.88**	**83.81**	84.56	**84.81**	61.16
			SecureLearn	84.70	**84.58**	**75.80**	84.70	83.70	75.85	84.52	83.89	74.49
		USPS	A. Paudice et al.	83.12	77.46	77.40	83.48	76.58	77.03	83.79	77.18	76.06
			M. Barreno et al.	79.78	70.65	76.71	79.24	76.10	76.42	71.73	60.37	75.67
			SecureLearn	**85.09**	**78.66**	**81.00**	**84.97**	**78.82**	**81.50**	**84.77**	**81.55**	**78.70**
	GNB	IRIS	A. Paudice et al.	90.89	**92.77**	76.31	87.77	92.77	69.88	84.56	92.77	86.46
			M. Barreno et al.	82.32	83.76	84.74	91.87	**94.75**	94.74	91.41	86.58	94.74
			SecureLearn	**95.39**	91.87	**97.23**	**95.39**	94.44	**97.23**	**95.39**	**94.44**	**97.23**
		MNIST	A. Paudice et al.	53.35	57.86	56.42	54.81	55.09	49.86	52.64	56.94	49.19
			M. Barreno et al.	**82.32**	**83.76**	**84.74**	**91.87**	**94.75**	**94.74**	**91.41**	**86.58**	**94.74**
			SecureLearn	53.92	53.68	53.49	54.19	54.95	54.09	54.67	53.90	54.38
		USPS	A. Paudice et al.	75.14	73.56	77.36	73.73	**77.45**	75.72	73.98	**77.90**	76.97
			M. Barreno et al.	76.54	**78.37**	76.61	74.27	71.42	71.45	72.91	75.19	66.96
			SecureLearn	**76.79**	77.77	**77.47**	**77.20**	76.62	**77.33**	**76.55**	77.59	**77.14**
	MLP	IRIS	A. Paudice et al.	97.18	97.77	97.70	97.18	91.11	90.70	97.18	90.47	90.52
			M. Barreno et al.	31.61	29.75	30.74	36.53	18.19	26.96	32.93	15.25	26.96
			SecureLearn	**99.90**	**99.87**	**99.90**	**99.99**	**97.06**	**97.07**	**99.99**	**99.90**	**99.99**
		MNIST	A. Paudice et al.	97.18	**97.70**	**99.99**	97.18	90.70	**97.54**	97.18	90.52	96.96
			M. Barreno et al.	31.61	29.75	30.74	36.53	18.19	26.96	32.93	15.25	26.96
			SecureLearn	**97.96**	97.46	97.06	**98.08**	**97.84**	97.39	**97.34**	**97.61**	**97.26**
		USPS	A. Paudice et al.	86.00	81.71	82.48	85.35	80.83	80.98	87.39	80.75	80.77
			M. Barreno et al.	**99.99**	78.9	83.52	14.96	51.47	79.69	13.88	82.33	79.20
			SecureLearn	98.42	**97.77**	**98.41**	**97.74**	**98.22**	**98.08**	**98.40**	**97.95**	**97.08**

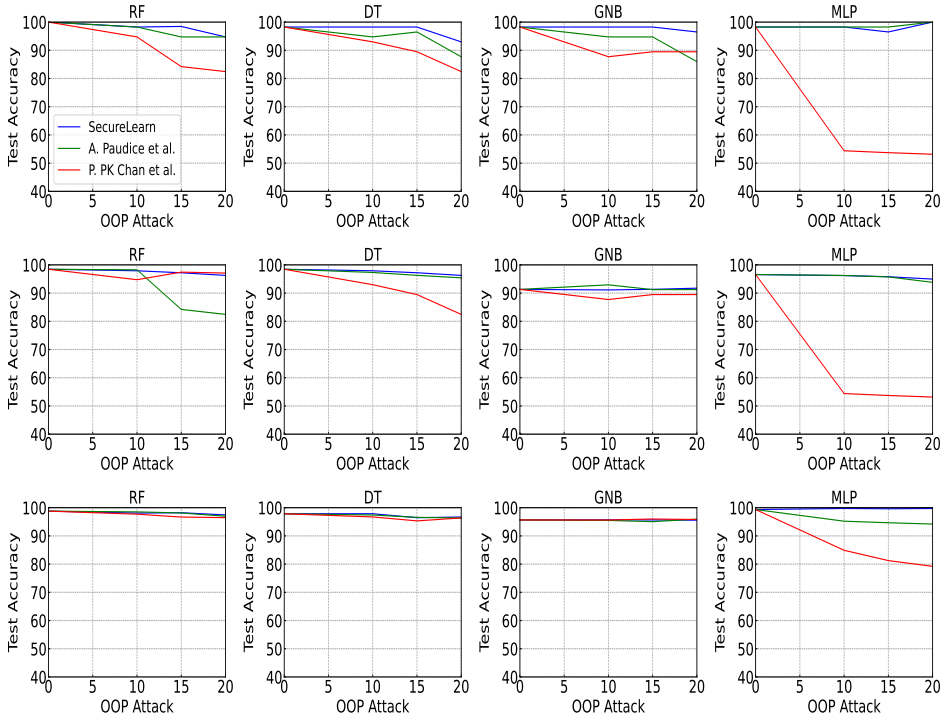


Figure 5.3: Impact of outlier-oriented poisoning attack on accuracy of models at various poisoning levels. The first row illustrates all models trained with the IRIS dataset, the models in the second row are trained with the MNIST dataset, and in the third row, the models are trained with the USPS dataset

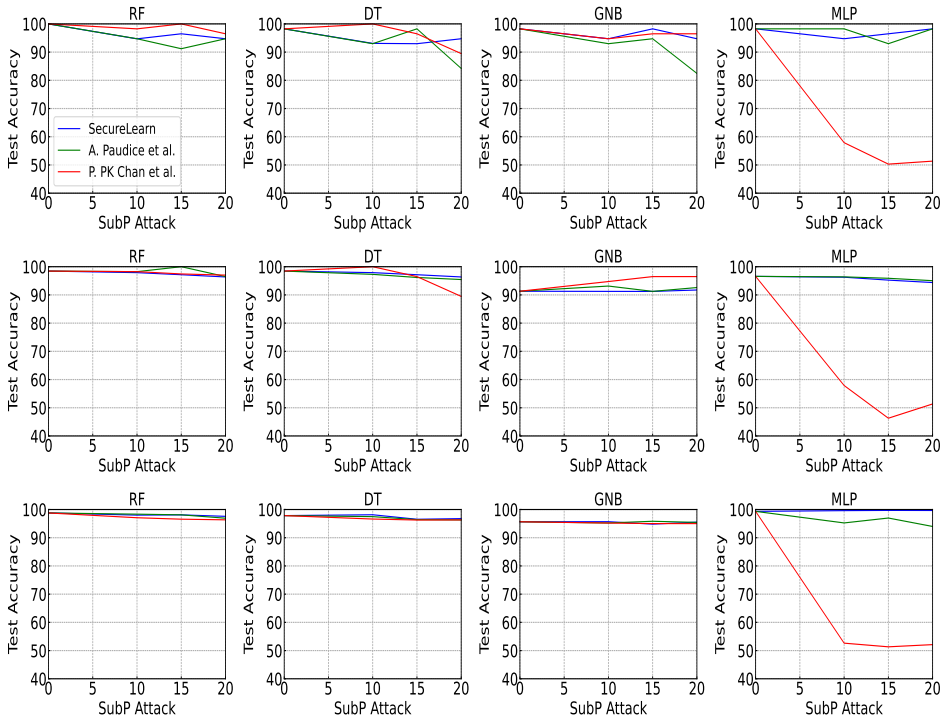


Figure 5.4: Impact of subpopulation attack on the accuracy of models at various poisoning levels. The first row illustrates models trained with the IRIS dataset, the models in the second row are trained with the MNIST dataset, and the models in the third row are trained with the USPS dataset

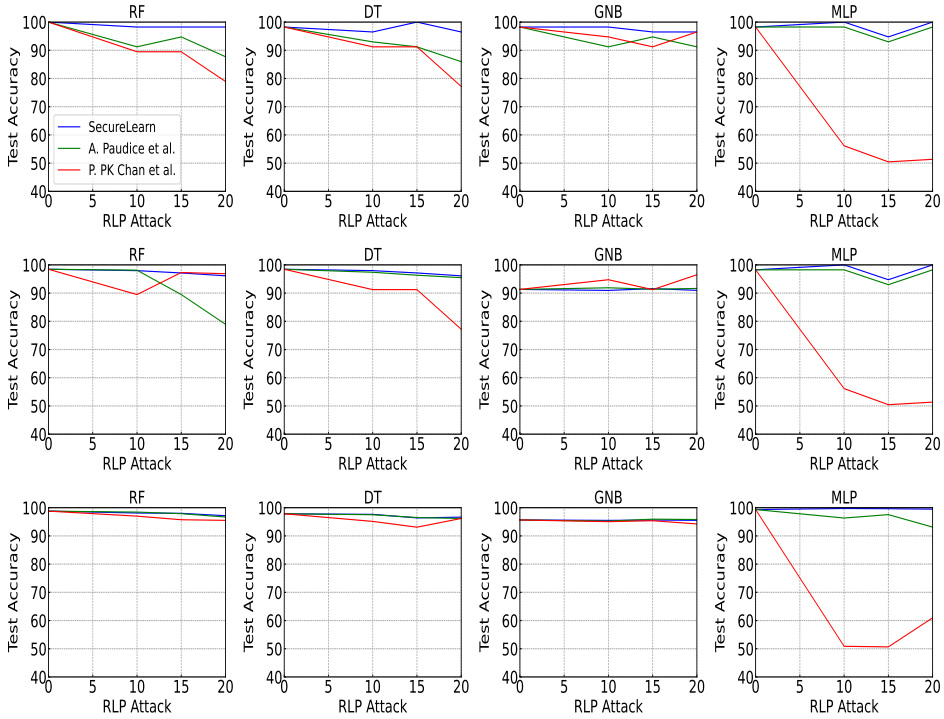


Figure 5.5: Impact of random label poisoning attack on accuracy of models at various poisoning levels. The first row illustrates models trained with the IRIS dataset, the models in the second row are trained with the MNIST dataset, and the models in the third row are trained with the USPS dataset

5.9.3 Analysing Feature-Oriented Adversarial Training

This section evaluated the FORT training of SecureLearn in enhancing adversarial robustness against data poisoning attacks. The adversarial robustness of the model is analysed by assessing FDR following Eq. 5.22. The results are provided in Tables 5.6 to 5.9. These results highlighted that FORT highly improved the adversarial robustness of multiclass models against all implemented data poisoning attacks.

These improvements are attributed to FORT's design, which leverages feature importance scores to guide adversarial training of ML. The adversarial samples for the training are developed by slightly perturbing data points close to decision boundaries and with high feature importance scores. Generalising over these perturbations enables the model to resist changes in its decision mechanisms with poisoned datasets.

The results given in the Table 5.6 highlighted that FORT reduces the FDR of the RF model to 0.06 when the model is trained on the poisoned IRIS dataset with $\Delta L = 10\%$. Similarly, for the same dataset, FDR=0.02 at $\Delta L = 15\%$ and FDR=0.05 at $\Delta L = 20\%$ across all attacks. Similar stability is visible for all adversarially trained models with FORT, as shown in Tables 5.7 and 5.9, highlighting the effectiveness of FORT.

Table 5.6: Effectiveness of feature-oriented adversarial training on false discovery rate of random forest after poisoning

Attack	Dataset	FDR					
		$\Delta L = 10\%$	FORT	$\Delta L = 15\%$	FORT	$\Delta L = 20\%$	FORT
OOP	IRIS	0.05	0.06	0.1	0.0001	0.19	0.13
	MNIST	0.02	0.01	0.16	0.14	0.21	0.16
	USPS	0.09	0.04	0.15	0.08	0.2	0.09
SubP	IRIS	0.08	0.06	0.1	0.0001	0.21	0.07
	MNIST	0.02	0.01	0.16	0.14	0.2	0.16
	USPS	0.1	0.04	0.16	0.08	0.2	0.09
RLP	IRIS	0.08	0.06	0.09	0.01	0.27	0.07
	MNIST	0.02	0.01	0.21	0.14	0.27	0.17
	USPS	0.12	0.04	0.21	0.08	0.26	0.09

Table 5.7: Effectiveness of feature-oriented adversarial training on false discovery rate of decision tree after poisoning

Attack	Dataset	FDR					
		$\Delta L = 10\%$	FORT	$\Delta L = 15\%$	FORT	$\Delta L = 20\%$	FORT
OOP	IRIS	0.03	0.02	0.1	0.03	0.19	0.07
	MNIST	0.15	0.14	0.19	0.14	0.26	0.23
	USPS	0.15	0.11	0.21	0.19	0.27	0.2
SubP	IRIS	0.07	0.05	0.15	0.05	0.13	0.15
	MNIST	0.14	0.14	0.19	0.15	0.26	0.23
	USPS	0.14	0.12	0.2	0.19	0.26	0.19
RLP	IRIS	0.15	0.03	0.12	0.05	0.23	0.11
	MNIST	0.19	0.14	0.25	0.15	0.33	0.24
	USPS	0.19	0.12	0.26	0.19	0.34	0.22

5.9.4 Increasing Poisoning Rate

SecureLearn maintains effectiveness across all evaluated attacks, independent of increasing poisoning levels. In this section, the analysis is extended to understand the relationship between the impact of increasing poisoning levels and the effectiveness of SecureLearn. Following the adversarial capabilities of the threat model, the data poisoning levels are set to $10\% < \Delta L < 20\%$. SecureLearn achieves a minimum sanitised accuracy of 90% for all models developed with four selected algorithms, highlighting no significant trade-off between model accuracy and adversarial robustness. The results are shown in Fig. 5.3 to Fig. 5.5. Data poisoning, however, impacts recall and f1-score differently for each model. The results are given in Table 5.5. For RF models, SecureLearn stabilises these models with a minimum recall of 84.19% and f1-score of 81.54% at 20% OOP poisoning. For DT models, the minimum recall is 78.20% and the f1-score is 75.80%. However, it is observed that SecureLearn does not sufficiently stabilise the GNB model trained with the MNIST dataset, as recall remains approximately 57% and the f1-score 56% across poisoning levels. In contrast, SecureLearn is highly effective in securing MLP models, achieving a minimum recall and f1-score of 97%, which demonstrates its potential to enhance the security of DL models. Overall, these results indicated that SecureLearn effectively mitigates the impact of data poisoning across datasets, even as poisoning levels increase.

Table 5.8: Effectiveness of feature-oriented adversarial training on false discovery rate of gaussian naive bayes after poisoning

Attack	Dataset	FDR					
		$\Delta L = 10\%$	FORT	$\Delta L = 15\%$	FORT	$\Delta L = 20\%$	FORT
OOP	IRIS	0.06	0.04	0.13	0.08	0.1	0.03
	MNIST	0.3	0.29	0.31	0.29	0.31	0.29
	USPS	0.2	0.2	0.2	0.19	0.22	0.2
SubP	IRIS	0.08	0.04	0.05	0.05	0.13	0.03
	MNIST	0.29	0.29	0.32	0.28	0.3	0.28
	USPS	0.2	0.19	0.2	0.2	0.23	0.19
RLP	IRIS	0.06	0.04	0.11	0.05	0.12	0.03
	MNIST	0.3	0.3	0.33	0.28	0.34	0.28
	USPS	0.21	0.19	0.22	0.19	0.24	0.2

Table 5.9: Effectiveness of feature-oriented adversarial training on false discovery rate of multilayer perceptron after poisoning

Attack	Dataset	FDR					
		$\Delta L = 10\%$	FORT	$\Delta L = 15\%$	FORT	$\Delta L = 20\%$	FORT
OOP	IRIS	0.07	0.02	0.04	0.02	0.15	0.05
	MNIST	0.06	0.01	0.06	0.02	0.08	0.02
	USPS	0.1	0.01	0.14	0.02	0.18	0.01
SubP	IRIS	0.03	0.03	0.05	0.02	0.2	0.07
	MNIST	0.06	0.01	0.08	0.02	0.08	0.02
	USPS	0.1	0.02	0.13	0.01	0.16	0.01
RLP	IRIS	0.03	0.0001	0.07	0.04	0.37	0.05
	MNIST	0.07	0.02	0.09	0.02	0.1	0.02
	USPS	0.1	0.01	0.13	0.01	0.16	0.02

5.10 Discussion and Limitations

This chapter proposed SecureLearn as a two-layer defence to mitigate data poisoning attacks and improve the resilience of traditional multiclass models, including neural networks. SecureLearn proposes an improvised data sanitisation along with a generic formulation of adversarial training, considering a common characteristic of the feature importance score. SecureLearn is analysed and compared with two existing solutions and three data poisoning attacks at three poisoning levels $10\% < \Delta L < 20\%$. The results showed that SecureLearn outperformed others in improving both the security and adversarial robustness of ML against various data poisoning attacks.

SecureLearn effectively enhanced the resilience of multiclass ML across RF, DT, GNB and MLP, confirming its generalisability beyond algorithm-specific defences. For all evaluated models, SecureLearn consistently maintained a minimum accuracy of 90%, recall and f1-score to at least 75%, and reduced the FDR to at least 0.06 against three distinct poisoning attacks. In the context of neural networks, SecureLearn achieved a minimum of 97% recall and f1-score against all selected data poisoning attacks. Furthermore, the adversarial robustness of models is improved with an average accuracy trade-off of only 3%.

Although various solutions Ho et al. (2022a), Tao et al. (2021) are provided in the literature, none

have proposed a two-layer approach to enhance the resilience and adversarial robustness of ML. Also, existing adversarial training mechanisms, for example, (Ho et al., 2022a) are limited to a gradient-oriented approach, which enhances the adversarial robustness of neural networks and advanced ML and DL models but is ineffective for traditional models, such as RF, DT and GNB models. The FORT follows a baseline property of feature importance, this method applies to all types of ML models, including traditional models and neural networks. The feature importance score informs the decision criteria of the model and helps generalise the model. By adding a small fraction of perturbation into the features with high importance, the model is taught to distinguish benign and poisoned data points. In this way, the resilience of the ML model is improved.

The experimentation of SecureLearn is conducted under assumptions to strengthen the efficacy and resilience of ML applications against data poisoning attacks; however, it may require additional constraints to be adapted for sensitive data and applications. For example, it might be limited in distinguishing between exceptional non-intrusive traffic packets and poisoned packets.

5.11 Summary

This chapter presented SecureLearn, a new attack-agnostic method to defend traditional multiclass models from data poisoning attacks. It is a two-layer solution that secures the training pipeline of the models by first effectively sanitising the dataset to remove poison and then improving adversarial robustness with the FORT adversarial training. SecureLearn secures the model from existing attacks and strengthens resilience against evolving attacks, hence fulfilling the aim of this thesis. This chapter answered RQ 3 by analysing the effectiveness of SecureLearn, highlighting its strengths and weaknesses and comparing it with existing solutions. To answer thoroughly, a new 3D evaluation matrix is proposed that has evaluated the proposed solution from three orthogonal dimensions. The experiments are conducted with three differently structured datasets to analyse the generalisability of SecureLearn. SecureLearn defends against black-box and grey-box poisoning attacks without requiring prior knowledge of the model and any additional dataset. SecureLearn provides robustness to the model in a two-layer approach, with first an improvised data sanitisation and second with a new FORT adversarial training. This way, making it an attack-agnostic and generic method that applies to all types of classification models.

Chapter 6. Conclusion and Future Work

The thesis proposed an attack-agnostic solution to defend multiclass ML against data poisoning attacks. Following this aim, the thesis has answered three research questions from two aspects of adversarial and defence standpoints. These answers provided a thorough analysis of ML security from new perspectives and covered traditional multiclass ML, which are overlooked in the literature. Firstly, this thesis explored various security vulnerabilities and highlighted limitations of existing solutions, stating that most solutions are attack-specific. Furthermore, it practically explored security vulnerabilities in traditional ML, motivated to assess the change in their learning dynamics with poisoned training. Conclusively, this assessment contributed to identifying vulnerabilities in ML models and their characteristics with poisoned training. These identifications helped fulfill the aim of this research by highlighting the relationships between ML characteristics and data poisoning.
Secondly, this thesis answered what the impact and consequences of multiclass data poisoning attacks would be under limited adversarial capabilities and knowledge to resolve a complex challenge of analysing the practicality and risks of data poisoning attacks. The answer highlighted that data poisoning attacks are successful and have severe consequences, even when implemented under limited adversarial capabilities. The new attack vectors may significantly degrade their performance, leading to a 50% decrease in accuracy and have successfully exploited adversarially robust ML models, trained with adversarial training. This analysis practically answered the limitations of existing solutions to defend against data poisoning attacks.
Following the above answers of ML security from an adversarial perspective, this thesis proposed a strengthened defence called SecureLearn, inspired by the outcomes of the above analyses. SecureLearn is a security-by-design and attack-agnostic solution that shows capabilities to mitigate various data poisoning attacks and is adaptable to many ML models. It mitigates reactive poisoning impact by cleaning datasets with its data sanitisation mechanism and provides proactive resilience to ML by improving adversarial robustness through FORT training.
Conclusively, it comprehensively assessed traditional ML against data poisoning attacks through an adversarial lens, followed by proposing an enhanced attack-agnostic solution which mitigates data poisoning attacks, irrespective of new attack vectors and is supported in all types of ML classification models.

6.1 Evaluate Research Objectives

Following the development of SecureLearn, an attack-agnostic solution, this thesis fulfilled the defined research objectives. The evaluation of objectives is given as follows:

- The first objective of the thesis is explore data poisoning attacks and their impact under limited

adversarial capabilities and knowledge. This objective is achieved in Chapter 4 by proposing the OOP attack and implementing it under grey-box scenarios. The attack has been practically successful and highlighted a high impact on three ML applications. Moreover, two case studies are conducted in Chapter 4, further assessing its impact against two real-life applications and highlighting the adverse impact of data poisoning in breaching adversarial training, a significant resilience mechanism against data poisoning attacks.

- The second objective of the thesis is achieved in Chapter 5 by proposing SecureLearn, an attack-agnostic defence, to mitigate data poisoning attacks in multiclass ML. SecureLearn is evaluated against four traditional multiclass models, trained with three datasets, highlighting its efficacy in enhancing the resilience of all selected models against three data poisoning attacks.

- The third objective of the thesis is achieved by proposing a 3D evaluation matrix to assess SecureLearn thoroughly against data poisoning attacks and compare it with existing mitigations in various settings. Chapter 5 provided a thorough assessment of SecureLearn in 3D evaluation matrix and its effectiveness against three data poisoning attacks against three distinct datasets in four multiclass models. The evaluation also compared it with two existing mitigations, highlighting that SecureLearn outperformed others as an attack-agnostic defence in all provided scenarios.

6.2 Contribution to Defence

Having revisited the adversarial aspect of analysing ML against data poisoning attacks, this thesis proposed deep behavioural analysis, revealing vulnerable characteristics of individual ML models and their relations with data poisoning attacks. Following deep behavioural analysis, this thesis designed the OOP attack to assess the resilience of multiclass models with constrained adversarial capabilities. The examination of OOP attack also strengthened the implications of behavioural analysis by analysing the underlying behaviours of poisoned multiclass models. The contributions to defence of these analyses are given as follows.

- Alongside the efficient development of classification systems with ML, this method allows practitioners to understand algorithm-level vulnerabilities and weaknesses of ML. The practitioners utilise these findings to define their training methodology, pre-training assessments of the datasets, and develop test cases for the security assessment of the trained model

- This approach enables defenders to improve the resilience of the ML by highlighting relationships between data poisoning and individual model characteristics. It also helps design stealthy data poisoning attacks by exploiting the characteristics of models that are sensitive to poison.

- The security analysis of multiclass ML under limited adversarial capabilities and knowledge helps practitioners understand the practicality and real-world impact of data poisoning attacks against traditional multiclass models.
- This chapter also puts forward the understanding of exploiting a common ML characteristic, called outliers, and its effectiveness in manipulating the learning fundamentals of models.
- This chapter emphasized the limitations of adversarial training in securing traditional classification models, highlighting the need to develop adaptive adversarial training mechanisms to enhance the resilience of traditional ML algorithms, not only for DL models and neural networks.

Following the adversarial aspect, this thesis strengthened the security and resilience of multiclass ML against existing and evolving data poisoning attacks by proposing an improved defence, SecureLearn. The contributions to defense made by SecureLearn are given as follows.

- SecureLearn is a step towards an enhanced mitigation that is an attack-agnostic solution and secures multiclass models against data poisoning, irrespective of particular attacks. Practitioners can implement it during the model training when they are unsure about the reliability of the dataset or to strengthen the adversarial robustness of the model against potential attacks. However, SecureLearn is a generalised defence and may require additional safeguards to assure particular scenarios of the applications.
- SecureLearn has provided a new approach of adaptable FORT adversarial training in traditional ML and neural networks with a trade-off of less than 3% in accuracy and robustness. This solution helps practitioners improve the adversarial robustness of traditional ML.
- SecureLearn is a standalone solution that does not require additional data for FORT training. It generates adversarial training data on its own and hence reduces the overhead of Practitioners.

6.3 Limitations

Each chapter has discussed the limitations of the individual contributions. Following those, this section presents the broader perspective of the limitations of the thesis, which could be addressed in future work. This thesis investigated the adversarial impact and consequences of data poisoning attacks; nonetheless, the scope of the work exclusively focused on supervised classification models, which can be extended to investigate the impact of poisoning in regression, unsupervised and DL models. Next, this thesis examined poisoned training of labeled datasets for the classification models, which can be extended to study poisoning impact in unlabeled datasets. Understanding the data poisoning impact in other ML types will provide a more comprehensive knowledge of poisoned training,

which ultimately helps strengthen defences against data poisoning attacks.

Furthermore, this thesis developed the analysis schemes under restricted threat models to study poisoning impact under limited adversarial capabilities. However, studying these poisoning implications under various threat models of extended adversarial capabilities, enhancing adversarial goals to inject backdoors, or inferring models beyond performance degradation enables practitioners to understand poisoning threats more comprehensively.

Finally, this thesis assessed the adversarial attacks and existing defences in real-world applications; however, the proposed defence can also be evaluated against real-world and security-sensitive applications to highlight its strength in securing these applications. Another limitation of presenting SecureLearn is that it has been evaluated in a grey-box adversarial setting. The threat model can be enhanced by analysing SecureLearn with white-box adversaries, which involve providing complete knowledge of the targeted system to adversaries to develop more stealthy and intrinsic attacks.

6.4 Future Work

Following the contributions of the thesis, this section discusses the future directions that can be taken as next steps to this research in the context of secure and trustworthy ML.

Chapter 3 presented a deep behavioural analysis of classification models against data poisoning, which can be extended to explore the behaviour of regression models and unsupervised ML to deepen the understanding of how data poisoning changes the learning fundamentals of other ML types. It should also be explored in DL, as DL models are largely embedded in digital systems nowadays. This future direction provides a thorough understanding of poisoning criteria to strengthen threat models, testing use cases against data poisoning attacks. Another potential direction is to conduct the behavioral analysis of poisoned models in real-time systems to understand poisoned behaviour in complex use cases. The behavioural analysis can also be extended to analyse model hallucination and jailbreak attacks by identifying changing characteristics with poisoned training.

Extending the implementation of the OOP attack, a potential future direction is to utilise it to exploit other mitigations, such as data sanitisation and differential privacy. This direction highlights the limitations of these mitigation solutions, providing an opportunity to improve these and strengthen the security of ML. The OOP attack is designed for multiclass models, which can be extended to poison multilabel models and enable practitioners to develop strong resilience against these attacks. Exploiting outliers in the OOP attack opens directions to develop new attack vectors by exploiting other characteristics of ML algorithms, such as exploiting the majority voting mechanism in KNN or altering feature importance scores in RF and DT. This direction will provide detailed knowledge of ML vulnerabilities and derive potential directions towards secure ML.

The above future directions highlight prospective work from an adversarial perspective, the other

perspective to discuss future directions from secure ML. This thesis primarily examined adversarial training, data sanitisation, and SecureLearn, their efficacy and limitations in classification models, which can be analysed to secure unsupervised models and DL. This direction will also improve SecureLearn in other types of ML. Though this thesis thoroughly evaluated SecureLearn under the 3D evaluation matrix, which can be further extended to other attacks, including inference and backdoor attacks. Furthermore, SecureLearn next extended to secure real-time applications as a potential avenue to explore, and would help improve SecureLearn itself.

Another very important direction in this regard is the development of security processes and policies for the secure development and deployment of ML models. These policies define the security principles in various domains and applications. Following the knowledge and understanding provided in this thesis, practitioners can develop security policies to use these models and define checklists for the secure development of ML.

References

Abdi, H. (2007). Z-scores. *Encyclopedia of measurement and statistics*, pages 1055–1058.

Abdulboriy, A. and Shin, J. S. (2024). An incremental majority voting approach for intrusion detection system based on machine learning. *IEEE Access*, 12:18972–18986.

Adarsh, P. and Jeyakumari, D. (2013). Multiclass svm-based automated diagnosis of diabetic retinopathy. In *2013 International Conference on Communication and Signal Processing*, pages 206–210. IEEE.

Aghakhani, H., Meng, D., Wang, Y.-X., Kruegel, C., and Vigna, G. (2021). Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In *2021 IEEE European symposium on security and privacy (EuroS&P)*, pages 159–178. IEEE.

Ahmed, U., Lin, J. C.-W., and Srivastava, G. (2022). Mitigating adversarial evasion attacks by deep active learning for medical image classification. *Multimedia Tools and Applications*, 81(29):41899–41910.

Ain, Q. U., Javed, A., and Irtaza, A. (2025). Deepevader: An evasion tool for exposing the vulnerability of deepfake detectors using transferable facial distraction blackbox attack. *Engineering Applications of Artificial Intelligence*, 145:110276.

Aïvodji, U., Gambs, S., and Ther, T. (2019). GAMIN: an adversarial approach to black-box model inversion. *CoRR*, abs/1909.11835.

Al-Haijaa, Q. A. and Ishtaiwia, A. (2021). Machine learning based model to identify firewall decisions to improve cyber-defense. *Int. J. Adv. Sci. Eng. Inf. Technol*, 11(4):1688–1695.

Alarab, I. and Prakoonwit, S. (2023). Uncertainty estimation based adversarial attack in multi-class classification. *Multimedia Tools and Applications*, 82(1):1519–1536.

Anthi, E., Williams, L., Javed, A., and Burnap, P. (2021). Hardening machine learning denial of service (dos) defences against adversarial attacks in iot smart home networks. *Computers & Security*, 108:102352.

Apruzzese, G., Andreolini, M., Colajanni, M., and Marchetti, M. (2020a). Hardening random forest cyber detectors against adversarial attacks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(4):427–439.

Apruzzese, G., Andreolini, M., Marchetti, M., Venturi, A., and Colajanni, M. (2020b). Deep reinforcement adversarial learning against botnet evasion attacks. *IEEE Transactions on Network and Service Management*, 17(4):1975–1987.

Aryal, K., Gupta, M., and Abdelsalam, M. (2022). Analysis of label-flip poisoning attack on machine learning based malware detector. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 4236–4245. IEEE.

Awajan, A. (2023). A novel deep learning-based intrusion detection system for iot networks. *Computers*, 12(2):34.

Ayub, M. A., Johnson, W. A., Talbert, D. A., and Siraj, A. (2020). Model evasion attack on intrusion detection systems using adversarial machine learning. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6.

Baiyekusi, O., Mahmoud, H., Mi, D., Arshad, J., Adeyemi-Ejeye, F., and Lee, H. (2024). An ml-based spectrum sharing technique for time-sensitive applications in industrial scenarios. In *2024 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1607–1612.

Baker, T., Li, T., Jia, J., Zhang, B., Tan, C., and Zomaya, A. Y. (2024). Poison-tolerant collaborative filtering against poisoning attacks on recommender systems. *IEEE Transactions on Dependable and Secure Computing*, 21(5):4589–4599.

Balevi, E. and Gitlin, R. D. (2017). Unsupervised machine learning in 5g networks for low latency communications. In *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, pages 1–2. IEEE.

Bao, H., Wei, K., Wu, Y., Qian, J., and Deng, R. H. (2025). Distributional black-box model inversion attack with multi-agent reinforcement learning. *IEEE Transactions on Information Forensics and Security*.

Baracaldo, N., Chen, B., Ludwig, H., and Safavi, J. A. (2017). Mitigating poisoning attacks on machine learning models: A data provenance based approach. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 103–110.

Barreno, M., Bartlett, P. L., Chi, F. J., Joseph, A. D., Nelson, B., Rubinstein, B. I., Saini, U., and Tygar, J. D. (2008). Open problems in the security of learning. In *Proceedings of the 1st ACM workshop on Workshop on AISec*, pages 19–26.

Basu, S., Izmailov, R., and Mesterharm, C. (2019). Membership model inversion attacks for deep networks. *CoRR*, abs/1910.04257.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 387–402. Springer.

Biggio, B., Nelson, B., and Laskov, P. (2011). Support vector machines under adversarial label noise. In *Asian conference on machine learning*, pages 97–112. PMLR.

Biggio, B., Nelson, B., and Laskov, P. (2012). Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1467–1474.

Bostani, H. and Moonsamy, V. (2021). Evadedroid: A practical evasion attack on machine learning for black-box android malware detection. *CoRR*, abs/2110.03301.

Brown, A., Gupta, M., and Abdelsalam, M. (2024). Automated machine learning for deep learning based malware detection. *Computers & Security*, 137:103582.

Cao, X. and Gong, N. Z. (2017). Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287.

Carlini, N. and Terzis, A. (2021). Poisoning and backdooring contrastive learning. *arXiv preprint arXiv:2106.09667*.

Carnerero-Cano, J., Munoz-Gonzalez, L., Spencer, P., and Lupu, E. C. (2023). Hyperparameter learning under data poisoning: Analysis of the influence of regularization via multiobjective bilevel optimization. *IEEE Transactions on Neural Networks and Learning Systems*.

Centurion, D. I., Chubarian, K., Fan, B., Sgherzi, F., Rashakrishnan, T., Sidiropoulos, A., and Straight, A. (2023). Geometric algorithms for k-nn poisoning. In *Proceedings of the 35th Canadian Conference on Computational Geometry (CCCG 2023)*. Canadian Conference on Computational Geometry.

Chalé, M. and Bastian, N. D. (2022). Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems. *Expert Systems with Applications*, 207:117936.

Chan, P. P., He, Z.-M., Li, H., and Hsu, C.-C. (2018a). Data sanitization against adversarial label contamination based on data complexity. *International Journal of Machine Learning and Cybernetics*, 9:1039–1052.

Chan, P. P., He, Z.-M., Li, H., and Hsu, C.-C. (2018b). Data sanitization against adversarial label contamination based on data complexity. *International Journal of Machine Learning and Cybernetics*, 9(6):1039–1052.

Chen, H. and Koushanfar, F. (2023). Tutorial: Toward robust deep learning against poisoning attacks. *ACM Transactions on Embedded Computing Systems*, 22(3):1–15.

Chen, J., Wang, W. H., and Shi, X. (2020). Differential privacy protection against membership inference attack on machine learning for genomic data. *bioRxiv*.

Chen, J., Zhang, X., Zhang, R., Wang, C., and Liu, L. (2021). De-pois: An attack-agnostic defense against data poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 16:3412–3425.

Chen, K., Lou, X., Xu, G., Li, J., and Zhang, T. (2022). Clean-image backdoor: Attacking multi-label models with poisoned labels only. In *The eleventh international conference on learning representations*.

Chen, Z. and Pattabiraman, K. (2023). Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction.

Chen, Z., Wu, J., Fu, A., Su, M., and Deng, R. H. (2023). Mp-clf: An effective model-preserving collaborative deep learning framework for mitigating data leakage under the gan. *Knowledge-Based Systems*, 270:110527.

Chenthan, H. N. (2023). Network intrusion dataset (cic-ids-2017). *Kaggle*.

Chillara, A. K., Saxena, P., Maiti, R. R., Gupta, M., Kondapalli, R., Zhang, Z., and Kesavan, K. (2024). Deceiving supervised machine learning models via adversarial data poisoning attacks: a case study with usb keyboards. *International Journal of Information Security*, 23(3):2043–2061.

Cinà, A. E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B. A., Oprea, A., Biggio, B., Pelillo, M., and Roli, F. (2023). Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, 55(13s):1–39.

Cohen, S. (2025). The evolution of machine learning: Past, present, and future. In *Artificial Intelligence in Pathology*, pages 3–14. Elsevier.

Das, A., Tariq, A., Batalini, F., Dhara, B., and Banerjee, I. (2024). Exposing vulnerabilities in clinical llms through data poisoning attacks: Case study in breast cancer. *medRxiv*.

David, M. W. (2018). Unsw_nb15. *Kaggle*.

Del Moral, P., Nowaczyk, S., and Pashami, S. (2022). Why is multiclass classification hard? *IEEE Access*, 10:80448–80462.

Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., and Roli, F. (2018). On the intriguing connections of regularization, input gradients and transferability of evasion and poisoning attacks. *CoRR*, abs/1809.02861.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, pages 141–142.

Di Leo, G. and Sardanelli, F. (2020). Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach. *European radiology experimental*, 4:18.

Dixit, M., Moholkar, A., Limaye, S., and Limaye, D. (2018). Naive bayes and svm based nids. In *2018 3rd International conference on inventive computation technologies (ICICT)*, pages 527–532. IEEE.

Drews, S., Albarghouthi, A., and D'Antoni, L. (2020). Proving data-poisoning robustness in decision trees. In *Proceedings of the 41st ACM SIGPLAN conference on programming language design and implementation*, pages 1083–1097.

Dunn, C., Moustafa, N., and Turnbull, B. (2020). Robustness evaluations of sustainable machine learning models against data poisoning attacks in the internet of things. *Sustainability*, 12(16):6434.

Ebrahimi, M., Zhang, N., Hu, J., Raza, M. T., and Chen, H. (2020). Binary black-box evasion attacks against deep learning-based static malware detectors with adversarial byte-level language model. *arXiv preprint arXiv:2012.07994*.

Farinu, U. (2025). Fairness, accountability, and transparency in ai: Ethical challenges in data-driven decision-making. *Available at SSRN 5128174*.

Fisher, R. A. (1936). Iris.

Geiping, J., Fowl, L., Somepalli, G., Goldblum, M., Moeller, M., and Goldstein, T. (2021). What doesn't kill you makes you robust (er): How to adversarially train against data poisoning. *arXiv preprint arXiv:2102.13624*.

Geiping, J., Fowl, L. H., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. (2020). Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations*.

Ghosh, A., Jagielski, M., and Wilson, C. (2022). Subverting fair image search with generative adversarial perturbations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 637–650.

Gibert, D., Planes, J., Le, Q., and Zizzo, G. (2023). Query-free evasion attacks against machine learning-based malware detectors with generative adversarial networks.

Gil-Fournier, A. and Parikka, J. (2021). Ground truth to fake geographies: machine vision and learning in visual practices. *AI & society*, 36:1253–1262.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Madry, A., Li, B., and Goldstein, T. (2020). Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *CoRR*, abs/2012.10544.

Guého, D., Singla, P., Melton, R. G., and Schwab, D. (2020). A comparison of parametric and non-parametric machine learning approaches for the uncertain lambert problem. In *AIAA Scitech 2020 Forum*, page 1911.

Gupta, P., Yadav, K., Gupta, B. B., Alazab, M., and Gadekallu, T. R. (2023a). A novel data poisoning attack in federated learning based on inverted loss function. *Computers & Security*, 130:103270.

Gupta, S., Kumar, P., and Tekchandani, R. K. (2023b). Facial emotion recognition based real-time learner engagement detection system in online learning context using deep learning models. *Multimedia Tools and Applications*, 82(8):11365–11394.

Hayase, J., Kong, W., Somani, R., and Oh, S. (2021). Spectre: Defending against backdoor attacks using robust statistics. In *International Conference on Machine Learning*, pages 4129–4139. PMLR.

He, Y., Li, B., Liu, L., Ba, Z., Dong, W., Li, Y., Qin, Z., Ren, K., and Chen, C. (2025). Towards label-only membership inference attack against pre-trained large language models. In *USENIX Security*.

He, Z., Zhang, T., and Lee, R. B. (2019). Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, page 148–162.

Ho, J., Lee, B.-G., and Kang, D.-K. (2022a). Attack-less adversarial training for a robust adversarial defense. *Applied Intelligence*, 52(4):4364–4381.

Ho, S., Reddy, A., Venkatesan, S., Izmailov, R., Chadha, R., and Oprea, A. (2022b). Data sanitization approach to mitigate clean-label attacks against malware detection systems. In *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, pages 993–998.

Hossain, K. M. and Oates, T. (2024). Advancing security in ai systems: A novel approach to detecting backdoors in deep neural networks. In *ICC 2024-IEEE International Conference on Communications*, pages 740–745. IEEE.

Hu, C. and Hu, Y.-H. F. (2020). Data poisoning on deep learning models. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 628–632. IEEE.

Huang, W. R., Geiping, J., Fowl, L., Taylor, G., and Goldstein, T. (2020). Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems*, 33:12080–12091.

Hull, J. J. (2002). A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, pages 550–554.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., and Li, B. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)*, pages 19–35. IEEE.

Jagielski, M., Severi, G., Pousette Harger, N., and Oprea, A. (2021). Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3104–3122.

Jebreel, N. M., Domingo-Ferrer, J., Sánchez, D., and Blanco-Justicia, A. (2024). Lfighter: Defending against the label-flipping attack in federated learning. *Neural Networks*, 170:111–126.

Jhong, S.-Y., Tseng, P.-Y., Siriphockpirom, N., Hsia, C.-H., Huang, M.-S., Hua, K.-L., and Chen, Y.-Y. (2020). An automated biometric identification system using cnn-based palm vein recognition. In *2020 international conference on advanced robotics and intelligent systems (ARIS)*, pages 1–6. IEEE.

Jia, J., Cao, X., and Gong, N. Z. (2021). Intrinsic certified robustness of bagging against data poisoning attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7961–7969.

Jia, J., Salem, A., Backes, M., Zhang, Y., and Gong, N. Z. (2019). Memguard: Defending against black-box membership inference attacks via adversarial examples. *CoRR*, abs/1909.10594.

Jin, J., McMurtry, E., Rubinstein, B. I. P., and Ohrimenko, O. (2022). Are we there yet? timing and floating-point attacks on differential privacy systems. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 473–488.

Kamath, C. N., Bukhari, S. S., and Dengel, A. (2018). Comparative study between traditional machine learning and deep learning approaches for text classification. In *Proceedings of the ACM Symposium on Document Engineering 2018*, pages 1–11.

Khosravy, M., Nakamura, K., Hirose, Y., Nitta, N., and Babaguchi, N. (2021). Model inversion attack: Analysis under gray-box scenario on deep learning based face recognition system. *KSII Transactions on Internet and Information Systems*, 15:1100–1119.

Khosravy, M., Nakamura, K., Hirose, Y., Nitta, N., and Babaguchi, N. (2022). Model inversion attack by integration of deep generative models: Privacy-sensitive face generation from a face recognition system. *IEEE Transactions on Information Forensics and Security*, pages 357–372.

Koball, C., Wang, Y., Vaidyan, V., and Hastings, J. (2025). Assessing evasion attacks on tree-based machine learning models: Supervised vs. unsupervised approaches. In *2025 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6. IEEE.

Koh, P. W., Steinhardt, J., and Liang, P. (2021). Stronger data poisoning attacks break data sanitization defenses.

Koh, P. W., Steinhardt, J., and Liang, P. (2022). Stronger data poisoning attacks break data sanitization defenses. *Machine Learning*, pages 1–47.

Kuppa, A. and Le-Khac, N.-A. (2020). Black box attacks on explainable artificial intelligence (xai) methods in cyber security. In *2020 International Joint Conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Laishram, R. and Phoha, V. V. (2016). Curie: A method for protecting svm classifier from poisoning attack. *arXiv preprint arXiv:1606.01584*.

Lee, J. H., Kim, B. H., and Kim, M. Y. (2021). Machine learning-based automatic optical inspection system with multimodal optical image fusion network. *International Journal of Control, Automation and Systems*, 19(10):3503–3510.

Li, H., Bai, L., Ye, Q., Hu, H., Xiao, Y., Zheng, H., and Xu, J. (2025). A sample-level evaluation and generative framework for model inversion attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18287–18295.

Li, K., Zheng, J., Yuan, X., Ni, W., Akan, O. B., and Poor, H. V. (2024). Data-agnostic model poisoning against federated learning: A graph autoencoder approach. *IEEE Transactions on Information Forensics and Security*.

Lin, J., Njilla, L. L., and Xiong, K. (2022). Secure machine learning against adversarial samples at test time. *EURASIP Journal on Information Security*, 2022(1):1.

Liu, G., Huang, X., and Yi, X. (2022). Adversarial label poisoning attack on graph neural networks via label propagation. In *European Conference on Computer Vision*, pages 227–243. Springer.

Liu, Q. and Wen, W. (2021). Model compression hardens deep neural networks: A new perspective to prevent adversarial attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 34:3–14.

Liu, T. Y., Yang, Y., and Mirzasoleiman, B. (2023). Friendly noise against adversarial noise: A powerful defense against data poisoning attacks.

Lu, Y., Kamath, G., and Yu, Y. (2023). Exploring the limits of model-targeted indiscriminate data poisoning attacks. In *International Conference on Machine Learning*, pages 22856–22879. PMLR.

Lu, Y., Yang, M. Y., Kamath, G., and Yu, Y. (2024). Indiscriminate data poisoning attacks on pre-trained feature extractors. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 327–343. IEEE.

Ma, K., Xu, Q., Zeng, J., Cao, X., and Huang, Q. (2021a). Poisoning attack against estimating from pairwise comparisons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6393–6408.

Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., Bailey, J., and Lu, F. (2021b). Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332.

Malik, F. (2022). Ctu13-csv-dataset. *Kaggle*.

Mayerhofer, R. and Mayer, R. (2022). Poisoning attacks against feature-based image classification. In *Proceedings of the Twelfth ACM conference on data and application security and privacy*, pages 358–360.

McCarthy, A., Ghadafi, E., Andriotis, P., and Legg, P. (2023). Defending against adversarial machine learning attacks using hierarchical learning: A case study on network traffic attack classification. *Journal of Information Security and Applications*, 72:103398.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35.

Melacci, S., Ciravegna, G., Sotgiu, A., Demontis, A., Biggio, B., Gori, M., and Roli, F. (2021). Domain knowledge alleviates adversarial attacks in multi-label classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9944–9959.

Meng, M. H., Bai, G., Teo, S. G., Hou, Z., Xiao, Y., Lin, Y., and Dong, J. S. (2022). Adversarial robustness of deep neural networks: A survey from a formal verification perspective. *IEEE Transactions on Dependable and Secure Computing*.

Mengara, O. (2024). A backdoor approach with inverted labels using dirty label-flipping attacks. *IEEE Access*.

Merzouk, M. A., Cuppens, F., Boulahia-Cuppens, N., and Yaich, R. (2022). Investigating the practicality of adversarial evasion attacks on network intrusion detection. *Annals of Telecommunications*, 77(11):763–775.

Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., and Roli, F. (2017). Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 27–38.

Murugan, A., Nair, S. A. H., Preethi, A. A. P., and Kumar, K. S. (2021). Diagnosis of skin cancer using machine learning techniques. *Microprocessors and Microsystems*, 81:103727.

Nazary, F., Deldjoo, Y., Di Noia, T., and Di Sciascio, E. (2025). Stealthy llm-driven data poisoning attacks against embedding-based retrieval-augmented recommender systems. In *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, pages 98–102.

Nguyen, B.-N., Chandrasegaran, K., Abdollahzadeh, M., and Cheung, N.-M. M. (2023). Label-only model inversion attacks via knowledge transfer. *Advances in Neural Information Processing Systems*, 36:68895–68907.

Nguyen, H. D. and Han, K. (2023). Safe reinforcement learning-based driving policy design for autonomous vehicles on highways. *International Journal of Control, Automation and Systems*, 21(12):4098–4110.

Pan, K., Gong, M., and Gao, Y. (2023). Privacy-enhanced generative adversarial network with adaptive noise allocation. *Knowledge-Based Systems*, 272:110576.

Pantelakis, V., Bountakas, P., Farao, A., and Xenakis, C. (2023). Adversarial machine learning attacks on multiclass classification of iot network traffic. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pages 1–8.

Paracha, A. and Arshad, J. (2024). A bibliometric study toward quantitative research assessment of security of machine learning. *Information Discovery and Delivery*.

Paracha, A., Arshad, J., Farah, M. B., and Ismail, K. (2024a). Deep behavioral analysis of machine learning algorithms against data poisoning. *International Journal of Information Security*, 24(1):29.

Paracha, A., Arshad, J., Farah, M. B., and Ismail, K. (2024b). Exploring data poisoning attacks against adversarially trained skin cancer diagnostics. In *2024 IEEE/ACM 17th International Conference on Utility and Cloud Computing (UCC)*, pages 220–225. IEEE.

Paracha, A., Arshad, J., Farah, M. B., and Ismail, K. (2024c). Machine learning security and privacy: a review of threats and countermeasures. *EURASIP Journal on Information Security*, 2024(1):10.

Paracha, A., Arshad, J., Farah, M. B., and Ismail, K. (2025a). Outlier-oriented poisoning attack: a grey-box approach to disturb decision boundaries by perturbing outliers in multiclass learning. *International Journal of Information Security*, 24(2):85.

Paracha, A., Arshad, J., Farah, M. B., and Ismail, K. (2025b). Securelearn-an attack-agnostic defense for multiclass machine learning against data poisoning attacks. *IEEE Transactions on Information Forensics and Security (Under review)*.

Paracha, A., Baiyekusi, O., Arshad, J., Mi, D., Lu, C., Zhang, Y., Wang, F., Chen, L., and Zhang, J. (2025c). Exploring multiclass data poisoning within an industrial 5g private network. In *2025 IEEE 101st Vehicular Technology Conference (VTC2025-Spring)*, pages 1–5. IEEE.

Patterson, W., Fernandez, I., Neupane, S., Parmar, M., Mittal, S., and Rahimi, S. (2022). A white-box adversarial attack against a digital twin. *arXiv preprint arXiv:2210.14018*.

Paudice, A., Muñoz-González, L., Gyorgy, A., and Lupu, E. C. (2018a). Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041*.

Paudice, A., Muñoz-González, L., and Lupu, E. C. (2018b). Label sanitization against label flipping poisoning attacks. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*, pages 5–15. Springer.

Pawlicki, M., Choraś, M., and Kozik, R. (2020). Defending network intrusion detection systems against adversarial evasion attacks. *Future Generation Computer Systems*, 110:148–154.

Peri, N., Gupta, N., Huang, W. R., Fowl, L., Zhu, C., Feizi, S., Goldstein, T., and Dickerson, J. P. (2020). Deep k-nn defense against clean-label data poisoning attacks. In *European Conference on Computer Vision*, pages 55–70. Springer.

Phan, H., Thai, M. T., Hu, H., Jin, R., Sun, T., and Dou, D. (2020). Scalable differential privacy with certified robustness in adversarial learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 7683–7694. PMLR.

Plant, R., Giuffrida, M. V., Pitropakis, N., and Gkatzia, D. (2024). Evaluating language model vulnerability to poisoning attacks in low-resource settings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Qiao, Q. and Beling, P. A. (2016). Decision analytics and machine learning in economic and financial systems. *Environment Systems and Decisions*, 36(2):109–113.

Qin, T., Gao, X., Zhao, J., Ye, K., and Xu, C.-Z. (2023). Learning the unlearnable: Adversarial augmentations suppress unlearnable example attacks. *arXiv preprint arXiv:2303.15127*.

Qiu, H., Zheng, Q., Zhang, T., Qiu, M., Memmi, G., and Lu, J. (2020). Toward secure and efficient deep learning inference in dependable iot systems. *IEEE Internet of Things Journal*, 8(5):3180–3188.

Rafiq, H., Aslam, N., Ahmed, U., and Lin, J. C.-W. (2023). Mitigating malicious adversaries evasion attacks in industrial internet of things. *IEEE Transactions on Industrial Informatics*, 19(1):960–968.

Rahman, M. H., Islam, T., Rana, M. M., Tasnim, R., Mona, T. R., and Sakib, M. M. (2023). Machine learning approach on multiclass classification of internet firewall log files.

Rosenbloom, P. S., Laird, J. E., McDermott, J., Newell, A., and Orciuch, E. (1985). R1-soar: An experiment in knowledge-intensive programming in a problem-solving architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):561–569.

Rosenfeld, E., Winston, E., Ravikumar, P., and Kolter, Z. (2020). Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR.

Russo, A. and Proutiere, A. (2021). Poisoning attacks against data-driven control methods. In *2021 American Control Conference (ACC)*, pages 3234–3241. IEEE.

Sagar, S. and Keke, C. (2021). Confidential machine learning on untrusted platforms: a survey. *Cybersecurity*, 4(1):30.

Saha, A., Subramanya, A., and Pirsiavash, H. (2020). Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965.

Salama, W. M., Aly, M. H., and Abouelseoud, Y. (2023). Deep learning-based spam image filtering. *Alexandria Engineering Journal*, 68:461–468.

Sanchez Vicarte, J. R., Schreiber, B., Paccagnella, R., and Fletcher, C. W. (2020). Game of threads: Enabling asynchronous poisoning attacks. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 35–52.

Sasmono, R. A., Agung, M. I. A., Nazaruddin, Y. Y., Oktavianus, J. A., and Tjahjono, G. (2021). Object matching in autonomous vehicle perception system using orb feature matching with svm

classifier. In *2021 International Conference on Instrumentation, Control, and Automation (ICA)*, pages 150–155. IEEE.

Seraj, S. (2022). Botdroid: Android botnet detection. *Kaggle*.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. (2018). Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31.

Shafahi, A., Najibi, M., Xu, Z., Dickerson, J., Davis, L. S., and Goldstein, T. (2020). Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5636–5643.

Shafee, A. and Awaad, T. A. (2021). Privacy attacks against deep learning models and their countermeasures. *Journal of Systems Architecture*, 114:101940.

Shahid, A. R., Imteaj, A., Wu, P. Y., Igoche, D. A., and Alam, T. (2022). Label flipping data poisoning attack against wearable human activity recognition system. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 908–914. IEEE.

Sharma, P., Austin, D., and Liu, H. (2019). Attacks on machine learning: Adversarial examples in connected and autonomous vehicles. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7. IEEE.

Shi, C. and Wang, Y. (2021). Non-parametric machine learning methods for interpolation of spatially varying non-stationary and non-gaussian geotechnical properties. *Geoscience Frontiers*, 12:339–350.

Shi, Y., Davaslioglu, K., and Sagduyu, Y. E. (2020). Over-the-air membership inference attacks as privacy threats for deep learning-based wireless signal classifiers. In *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, pages 61–66.

Shi, Y., Erpek, T., Sagduyu, Y. E., and Li, J. H. (2018). Spectrum data poisoning with adversarial deep learning. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 407–412. IEEE.

Shi, Y. and Sagduyu, Y. E. (2017). Evasion and causative attacks with adversarial deep learning. In *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, pages 243–248.

Sinanović, H. and Mrdovic, S. (2017). Analysis of mirai malicious software. In *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–5. IEEE.

Song, C., Ristenpart, T., and Shmatikov, V. (2017). Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pages 587–601.

Steinhardt, J., Koh, P. W. W., and Liang, P. S. (2017). Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30.

Strobel, M. and Shokri, R. (2022). Data privacy and trustworthy machine learning. *IEEE Security & Privacy*, 20(5):44–49.

Su, J., Vargas, D. V., and Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841.

Suya, F., Mahloujifar, S., Suri, A., Evans, D., and Tian, Y. (2021). Model-targeted poisoning attacks with provable convergence. In *International Conference on Machine Learning*, pages 10000–10010. PMLR.

Taherdoost, H. (2022). Classification of machine learning algorithms. In *Advances in Data Computing, Communication and Security: Proceedings of I3CS2021*, pages 417–422. Springer.

Tao, G., Liu, Y., Shen, G., Xu, Q., An, S., Zhang, Z., and Zhang, X. (2022a). Model orthogonalization: Class distance hardening in neural networks for better security. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1372–1389.

Tao, J. and Shokri, R. (2025). Range membership inference attacks. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 346–361. IEEE.

Tao, L., Feng, L., Wei, H., Yi, J., Huang, S.-J., and Chen, S. (2022b). Can adversarial training be manipulated by non-robust features? *Advances in Neural Information Processing Systems*, 35:26504–26518.

Tao, L., Feng, L., Yi, J., Huang, S.-J., and Chen, S. (2021). Better safe than sorry: Preventing delusive adversaries with adversarial training. *Advances in Neural Information Processing Systems*, 34:16209–16225.

Tian, J., Shen, C., Wang, B., Xia, X., Zhang, M., Lin, C., and Li, Q. (2024). Lesson: Multi-label adversarial false data injection attack for deep learning locational detection. *IEEE Transactions on Dependable and Secure Computing*.

Titcombe, T., Hall, A. J., Papadopoulos, P., and Romanini, D. (2021). Practical defences against model inversion attacks for split neural networks.

Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., and Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 1–11.

Tu, J., Wang, T., Wang, J., Manivasagam, S., Ren, M., and Urtasun, R. (2021). Adversarial attacks on multi-agent communication. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7768–7777.

Usynin, D., Rueckert, D., and Kaissis, G. (2023). Beyond gradients: Exploiting adversarial priors in model inversion attacks. *ACM Trans. Priv. Secur.*, 26(3).

van den Heuvel, E. and Zhan, Z. (2022). Myths about linear and monotonic associations: Pearson'sr, spearman's ρ, and kendall's τ. *The American Statistician*, 76:44–52.

Venkatesan, S., Sikka, H., Izmailov, R., Chadha, R., Oprea, A., and de Lucia, M. J. (2021). Poisoning attacks and data sanitization mitigations for machine learning models in network intrusion detection systems. In *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, pages 874–879.

Vidnerová, P. and Neruda, R. (2016). Evolutionary generation of adversarial examples for deep and shallow machine learning models. In *Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016*, pages 1–7.

Vivek, B., Mopuri, K. R., and Babu, R. V. (2018). Gray-box adversarial training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 203–218.

Wang, H., Cao, Z., and Hong, B. (2020). A network intrusion detection system based on convolutional neural network. *Journal of Intelligent & Fuzzy Systems*, 38(6):7623–7637.

Wang, H., Wang, S., Jin, Z., Wang, Y., Chen, C., and Tistarelli, M. (2021a). Similarity-based gray-box adversarial attack against deep face recognition. In *2021 16th IEEE international conference on automatic face and gesture recognition (FG 2021)*, pages 1–8. IEEE.

Wang, P., Fan, E., and Wang, P. (2021b). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern recognition letters*, 141:61–67.

Wang, Y., Mianjy, P., and Arora, R. (2021c). Robust learning for data poisoning attacks. In *International Conference on Machine Learning*, pages 10859–10869. PMLR.

Wang, Z., Huang, S., Huang, Y., and Cui, H. (2023). Energy-latency attacks to on-device neural networks via sponge poisoning. In *Proceedings of the 2023 Secure and Trustworthy Deep Learning Systems Workshop*, pages 1–11.

Weerasinghe, S., Alpcan, T., Erfani, S. M., and Leckie, C. (2021). Defending support vector machines against data poisoning attacks. *IEEE Transactions on Information Forensics and Security*, 16:2566–2578.

Wei, W., Chow, K.-H., Wu, Y., and Liu, L. (2023). Demystifying data poisoning attacks in distributed learning as a service. *IEEE Transactions on Services Computing*.

Wei, X., Guo, Y., and Yu, J. (2022). Adversarial sticker: A stealthy attack method in the physical world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2711–2725.

Wen, R., Zhao, Z., Liu, Z., Backes, M., Wang, T., and Zhang, Y. (2023). Is adversarial training really a silver bullet for mitigating data poisoning?

Xu, Q., Yang, Z., Zhao, Y., Cao, X., and Huang, Q. (2022). Rethinking label flipping attack: From sample masking to sample thresholding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7668–7685.

Xu, X., Wang, Q., Li, H., Borisov, N., Gunter, C. A., and Li, B. (2021). Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 103–120. IEEE.

Xue, J., Zheng, M., Sheng, Y., Yang, L., Lou, Q., and Jiang, L. (2023). Trojfair: Trojan fairness attacks. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, pages 47–56.

Yan, H., Li, X., Zhang, W., Wang, R., Li, H., Zhao, X., Li, F., and Lin, X. (2023). Automatic evasion of machine learning-based network intrusion detection systems. *IEEE Transactions on Dependable and Secure Computing*, pages 1–16.

Yang, Z., Wang, L., Yang, D., Wan, J., Zhao, Z., Chang, E.-C., Zhang, F., and Ren, K. (2023). Purifier: Defending data inference attacks via transforming confidence scores. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10871–10879.

Yerlikaya, F. A. and Şerif Bahtiyar (2022). Data poisoning attacks against machine learning algorithms. *Expert Systems with Applications*, 208:118101.

You, Z., Ye, J., Li, K., Xu, Z., and Wang, P. (2019). Adversarial noise layer: Regularize neural network by adding noise. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 909–913.

Yu, M. and Sun, S. (2022). Natural black-box adversarial examples against deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8936–8944.

Yu, T., Tao, P., and Ru, Z. (2025). Toward trustworthy ai: Explainable and robust machine learning for medical applications. *Available at SSRN 5328192*.

Zeng, X., Liu, C., Wang, Y.-S., Qiu, W., Xie, L., Tai, Y.-W., Tang, C.-K., and Yuille, A. L. (2019). Adversarial attacks beyond the image space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4302–4311.

Zhang, C., Tang, Z., and Li, K. (2023). Clean-label poisoning attack with perturbation causing dominant features. *Information Sciences*, 644:118899.

Zhang, M., Ren, Z., Wang, Z., Ren, P., Chen, Z., Hu, P., and Zhang, Y. (2021). Membership inference attacks against recommender systems. *CoRR*, abs/2109.08045.

Zhang, Q., Ma, J., Xiao, Y., Lou, J., and Xiong, L. (2020). Broadening differential privacy for deep learning against model inversion attacks. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1061–1070.

Zhang, S., Li, X., Zong, M., Zhu, X., and Wang, R. (2017). Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, 29:1774–1785.

Zhang, Y., Albarghouthi, A., and D'Antoni, L. (2022). Bagflip: A certified defense against data poisoning. *Advances in Neural Information Processing Systems*, 35:31474–31483.

Zhao, B. and Lao, Y. (2022a). Clpa: Clean-label poisoning availability attacks using generative adversarial nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9162–9170.

Zhao, B. and Lao, Y. (2022b). Towards class-oriented poisoning attacks against neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3741–3750.

Zhong, H., Liao, C., Squicciarini, A. C., Zhu, S., and Miller, D. (2020). Backdoor embedding in convolutional neural network models via invisible perturbation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 97–108.

Zhou, L., Pan, S., Wang, J., and Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237:350–361.

Zhu, C., Huang, W. R., Li, H., Taylor, G., Studer, C., and Goldstein, T. (2019a). Transferable clean-label poisoning attacks on deep neural nets. In *International conference on machine learning*, pages 7614–7623. PMLR.

Zhu, C., Huang, W. R., Shafahi, A., Li, H., Taylor, G., Studer, C., and Goldstein, T. (2019b). Transferable clean-label poisoning attacks on deep neural nets.

Zhu, C., Wang, H., Zhuang, Y., Li, J., and Cao, Y. (2022). A sparsity-limitation-based high-dimensional distribution searching algorithm for adversarial attack. *Journal of Sensors*, 2022(1):4031440.

Zhu, T., Ye, D., Zhou, S., Liu, B., and Zhou, W. (2023a). Label-only model inversion attacks: Attack with the least information. *IEEE Transactions on Information Forensics and Security*, 18:991–1005.

Zhu, Y., Miao, C., Xue, H., Li, Z., Yu, Y., Xu, W., Su, L., and Qiao, C. (2023b). Tilemask: A passive-reflection-based attack against mmwave radar object detection in autonomous driving. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1317–1331.

Zhu, Y., Wen, H., Wu, J., and Zhao, R. (2023c). Online data poisoning attack against edge ai paradigm for iot-enabled smart city. *Math. Biosci. Eng*, 20:17726–17746.

Zhu, Z., Wu, C., Fan, R., Lian, D., and Chen, E. (2023d). Membership inference attacks against sequential recommender systems. In *Proceedings of the ACM Web Conference 2023*, page 1208–1219. Association for Computing Machinery.

Zou, Y., Zhang, Z., Backes, M., and Zhang, Y. (2020). Privacy analysis of deep learning in the wild: Membership inference attacks against transfer learning. *CoRR*, abs/2009.04872.