ADVERSARIAL TRAINING FOR MACHINE LEARNING CLASSIFIERS AGAINST MULTIPLE ADVERSARIES

Hakeem Alade Quadri

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Victoria University, Australia.

Institute for Sustainable Industries and Liveable Cities

August 2025

Abstract

Convolutional Neural Networks (CNNs), particularly low-latency models like MobileNet, are widely applied in areas such as image classification, speech recognition, and language processing. Despite their efficiency and accuracy, these models remain vulnerable to adversarial attacks, small, structured perturbations to input data that can lead to misclassification without affecting human perception. Traditional adversarial training techniques, which aim to enhance model robustness, typically treat all data points equally. This uniform treatment does not account for the varying susceptibility of individual samples to adversarial perturbation.

To address this limitation, we propose a Weighted Adversarial Reinforced Stackelberg Learning (WARS) framework, which formulates the training process as a Stackelberg game between a defender (the CNN model) and an adversary. In this setup, we assign greater training weight to data points more likely to be exploited by adversaries. The strategy allows the model to adapt its training focus based on the risk level associated with each input. To further enhance robustness, we integrate a reinforcement learning (RL) agent to fine-tune hyperparameters dynamically throughout training, reducing reliance on manual configuration and improving convergence efficiency.

Experimental results on the CIFAR-10 dataset show that the WARS model achieves a robustness of 66.18% after a single epoch of training, compared to 64.72% obtained through standard adversarial training. This indicates that the WARS approach can offer measurable improvements in resilience with minimal computational overhead.

Beyond single-adversary settings, we extend our model to account for multiple attackers using a Bayesian Stackelberg game framework. This models the interaction between the classifier and a population of adversaries with different strategies, simulating more

realistic deployment conditions. The defender computes an optimal mixed strategy that considers the distribution of possible attacks. The resulting nested Bayesian Stackelberg formulation provides a scalable foundation for training models robust to varied and unpredictable threats.

Finally, we investigate quantum machine learning as an alternative defence strategy. By employing quantum support vector machines (QSVM) with ZZ feature maps, we project adversarial inputs into high-dimensional quantum spaces, allowing for enhanced separability between perturbed and unperturbed data. On adversarial perturbed MNIST and CIFAR-10 datasets, the QSVM achieved 70.6% classification accuracy, outperforming a classical SVM with an RBF kernel, which scored 51%. This demonstrates the potential of quantum kernels in defending against adversarial threats, particularly in complex, non-linear domains.

This thesis addresses three key challenges in adversarial machine learning: (1) the inability of traditional adversarial training to adapt to sample-specific vulnerabilities, (2) the inefficiency of static hyperparameter tuning in dynamic adversarial settings, and (3) the limitations of classical models in handling complex, non-linear adversarial perturbations. To overcome these challenges, we propose a Weighted Adversarial Reinforced Stackelberg Learning (WARS) framework that combines sample-weighted adversarial training with reinforcement-based hyperparameter optimization. We extend this to a Bayesian Stackelberg game to model interactions with multiple attackers and improve scalability in real-world threat environments. Finally, we explore quantum-enhanced classification using Quantum Support Vector Machines (QSVMs), demonstrating superior resilience to adversarial perturbations through high-dimensional feature mapping. Collectively, this work presents an integrated defence strategy that enhances the robustness of modern machine learning models against evolving adversarial threats.

Doctor of Philosophy Declaration

I, Hakeem Alade Quadri, certify that the thesis titled *Game Theory Adversarial Training for Machine Learning Classifiers* is entirely my own original work, except where due reference is made. The thesis does not exceed 80,000 words in length, not including tables, figures, appendices, references, or footnotes. It has not been submitted, in whole or in part, for the award of any other degree or qualification at any academic institution.

This research was conducted in accordance with the principles outlined in the Australian Code for the Responsible Conduct of Research and adheres to Victoria University's Higher Degree by Research policies and procedures.

This thesis has been edited for clarity of expression, punctuation and grammar using Generative AI tools. The use complies with VU guidelines on the use of editors in HDR theses and overall, VU policy on use of AI in research. This research was also conducted in accordance with the principles outlined in the Australian Code for the Responsible Conduct of Research and adheres to Victoria University's Higher Degree by Research policies and procedures.

Signature Date 15/07/2025

Acknowledgements

As I bring this thesis to completion, I am filled with a deep sense of gratitude for the individuals who have supported me throughout this long, challenging, and transformative journey. Completing a PhD is never a solitary pursuit; it is a path marked by the contributions, patience, and encouragement of others, and I am incredibly fortunate to have had so many remarkable people walk alongside me.

First and foremost, I express my sincere appreciation to my principal supervisor, Professor Hua Wang. Your unwavering commitment to my academic growth and your invaluable guidance throughout this research journey has been critical to its success. Your expertise in machine learning and data science, coupled with your generous mentorship, has inspired me deeply. From our very first meeting, you provided clarity in direction, rigor in research, and an expectation for excellence that challenged and motivated me. Thank you for believing in the value of this work and for steering me in the right direction at every stage of my development. Your dedication to my progress went far beyond your formal role as a supervisor, and I am truly grateful for the time, feedback, and encouragement you have consistently offered.

To Professor Sardar M. N. Islam, thank you for being a pillar of intellectual and personal support throughout this journey. Your thoughtful insights, timely feedback, and constant encouragement helped me navigate the more complex phases of this research. You helped sharpen my critical thinking and grounded my theoretical contributions with meaningful context. Working with you taught me the value of academic integrity and the importance of maintaining high standards in scholarly research. Your collaborative spirit and belief in my potential have left a lasting impression on me.

I also extend my heartfelt thanks to Dr Yonfeng (Felix) Ge and Dr Bruce Gu, whose practical advice and perspective helped me stay focused and adaptive. Your guidance, particularly during the early stages of the research, was essential to shaping my understanding of applied machine learning techniques in real-world contexts. Thank you for making time to review my work, offer recommendations, and contribute meaningfully to the evolution of this thesis.

Beyond my academic mentors, I owe a profound debt of gratitude to my wife, Halima Quadri, whose love and steadfast support made this accomplishment possible. Halima, you have been my source of strength during the most difficult periods of this journey. Your words of encouragement, your quiet patience during late nights and long weekends, and your unwavering belief in my vision helped me persist even when the path forward seemed uncertain. Thank you for carrying our family on your shoulders when I was buried in research, and for reminding me of what truly matters. You saw my potential even when I doubted it, and I dedicate this achievement to you as much as it is mine.

To my beloved children, Abdullah and Umar, you may be too young to understand the details of this work, but your presence in my life gave me a purpose far greater than any academic milestone. The sound of your laughter, your innocent questions, and the joy you bring into our home lifted me when I was tired and gave me something to look forward to at the end of each long day. I hope one day this thesis shows you the importance of perseverance, hard work, and the pursuit of knowledge. You have both taught me patience and humility, and I look forward to the day when I can share with you the story of how your little footsteps gave strength to mine.

I also wish to thank my extended family and friends, near and far, who have offered their words of encouragement, understanding, and support over the years. Whether it was a phone call, a kind message, or a shared meal during a break in writing, your presence helped me stay grounded and reminded me of the rich community behind me. Special thanks go to those who allowed me space when I needed to focus, and who celebrated even the smallest victories along the way.

To the academic and administrative staff at Victoria University, thank you for providing a vibrant and supportive research environment. The facilities, resources, and services offered through the university made this work possible. The workshops, seminars, and opportunities for scholarly exchange enriched my perspective and helped me become a better researcher. I also appreciate the work of the graduate research office for their administrative support, and for ensuring the smooth progression of my candidacy over the years.

I would like to acknowledge the wider research community in quantum machine learning, adversarial training, and security-focused AI, whose publications, discussions, and breakthroughs served as the foundation and inspiration for this work. Reading the work of pioneers in this field pushed me to ask deeper questions and refine the focus of my own research. I am especially grateful for the open-source tools and datasets that enabled experimentation and reproducibility.

To my peers and colleagues in the research program, your collaboration, discussions, and camaraderie were instrumental in making this journey less isolating. I cherish the brainstorming sessions, the critical feedback on drafts, and the mutual support that made this PhD experience a shared endeavor rather than a solitary pursuit.

Finally, I am grateful to God, whose mercy and guidance sustained me through both trials and triumphs. This journey demanded perseverance, resilience, and faith, qualities that I was only able to hold on to with divine help. All praise is due to Him who makes all things possible.

In conclusion, though my name stands on the cover of this thesis, it is the product of many hearts, minds, and hands. Each contribution, big or small, has left a lasting imprint on the work and on me personally. To all who supported me through this chapter of life, please accept my deepest and most heartfelt thanks.

Table of Contents

Dedication.

To my wife for her immense support through the program.

List of Publications

This thesis includes work that has been published or accepted for publication, which is the sole work of the author. The inclusion of these works is done with the permission of the respective publishers and adheres to academic requirements for reproduction.

Certain data, ideas, arguments, and figures contained within this thesis have been previously disseminated. These materials are integral to the development and validation of the research presented herein.

The following publications represent original contributions made by the author during the doctoral research

1. **Quadri, Hakeem**;Gu, Bruce;Wang, Hua;Islam, Sardar: Advancing MobileNet Security: Weighted Adversarial Learning in Convolutional Neural Networks; 2024 11th International Conference on Machine Intelligence Theory and Applications (MiTA), Machine Intelligence Theory and Applications (MiTA), 2024 11th International Conference on,20240714
2. **Quadri, Hakeem**;Gu, Bruce;Wang, Hua;Islam, Sardar: Mixed Bayesian Stackelberg Strategies for Robust Adversarial Classifiers; EAI Endorsed Transactions on Scalable Information Systems, 2025, Vol. 12 Issue 1, p1-10,
3. **Hakeem.Quadri**, hua.wang, Sardar.Islam, bo.li: Quantum SVM Algorithms for Efficient Defense Against Gradient-Based Adversarial Attacks; NaNA 2025 International Conference on Networking and Network.

List of Figures

List of Tables

Glossary

AI Artificial Intelligence

DOBSS Decomposed Optimal Bayesian Stackelberg Solver

FGSM Fast Gradient Sign Method

RFGSM Regulated Fast Gradient Sign Method

PGD Projected Gradient Descent

CNN Convolution Neural Network

SVM Support Vector Machine

WAR Weighted Adversarial Reinforcement Learning

QAML Quantum Adversarial Machine Learning

QML Quantum Machine Learning

CIFAR Canadian Institute for Advanced Research

MNIST Modified National Institute of Standards and Technology database

AdvSVM Adversarial Support Vector Machine

QSVM Quantum Support Vector Machine

ML Machine Learning

SARSA Statr-Action-Reward-State-Action

DNN Deep Neural Network

JND Just Needed

RL Reinforcement Learning

IFGSM Iterated Fast Gradient Sign Method

FLAT

FAT Friendly Adversarial Training

COMP Concentration of Measure Phenomenon

QVC Quantum Variational Classifiers

NISQ Noisy Intermediate-Scale Quantum

DQN Deep Quantum Network

PPO Proximal Policy Optimization

RBF Radial Basis Function

PCA Principal Component Analysis

CQ Classical Quantum

WAS Weighted Adversarial Stackelberg

QUGAN Quantum Generative Adversarial network

GAN Generative Adversarial Network

EWL Eister-Wilkens-Lewenstein

Chapter 1

Introduction

1.1 Background

With the rapid advancement of deep learning and artificial intelligence, ensuring the robustness of machine learning classifiers against adversarial attacks has become a critical research priority. The susceptibility of machine learning (ML) models to adversarial perturbations has drawn increasing attention from both the machine learning and cybersecurity communities. As ML algorithms are integrated into real-world applications, ranging from autonomous systems to medical diagnostics and financial decision-making, there is growing concern about the security risks posed by these vulnerabilities [1] [2] [3]

Recent conventional ML models are built on the assumption that both training and testing data are independently and identically distributed (i.i.d.). This assumption, while convenient for theoretical analysis, fails to account for adversarial manipulation. Attackers exploit this gap by generating inputs that are close to legitimate samples in feature space but are purposefully designed to cause misclassification [4] [5] [6] [7] [8] [9] [10]. These inputs, often called adversarial examples, introduce subtle perturbations to images or other input data. Though imperceptible to the human eye, these changes can cause the model to confidently predict an incorrect class (Goodfellow et al., 2014). As a result, adversarial examples degrade the reliability of ML models and expose them to potential exploitation [11] [12] [13] [14] [15] [16].

Addressing these vulnerabilities requires a shift in the way models are trained and evaluated. Traditional ML algorithms are not inherently designed to account for adversarial interference. In adversarial learning, the training process is adapted to account for the presence of an attacker who perturbs the input distribution [17] [18] [19] [20].These

perturbations can shift the model's latent space, resulting in learned representations that are biased toward adversarial distributions. If not corrected, the model becomes overfitted to either benign data or to a narrow range of adversarial patterns, reducing its generalization capability. [21] [22] [23] [24] [25] [26] [27] [28] [29].

To build classifiers that are truly robust, it is essential to incorporate diverse adversarial threat models into the training framework. This includes accounting for variations in the attacker's knowledge, capabilities, and goals. For example, white-box attackers may have full access to the model architecture and gradients, while black-box attackers operate with limited information. Similarly, attacks may differ in their perturbation norms (e.g., ℓ_0, ℓ_2, $\ell\infty$) and their target objectives (targeted vs. untargeted misclassification) [30] [31] [25].

Robust adversarial learning must therefore be formulated as a strategic interaction between the learner and a range of potential adversaries. This requires not only training on adversarial samples but also designing learning algorithms that can generalize across unseen attack types. Incorporating prior knowledge about likely attack strategies, perturbation bounds, or distributions of adversaries can improve the learner's ability to resist attacks. Furthermore, integrating adversarial training with techniques such as regularization, uncertainty modeling, or game-theoretic optimization can further enhance model robustness [32] [33].

Hence, developing secure and resilient machine learning systems involves understanding the behaviour and distributional impact of adversarial attacks, as well as designing models that adapt to diverse and evolving threat scenarios. As ML continues to expand into critical and high-stakes domains, adversarial robustness is not just a technical challenge, but a foundational requirement for trustworthy AI systems [29].

Common adversarial defence techniques include adversarial training, where neural networks are trained on perturbed input samples to improve their resilience to attacks. The effectiveness of this method relies heavily on the ability to generate strong and diverse adversarial examples. Without high-quality adversarial samples, the network may overfit to specific perturbations and fail to generalize under real-world adversarial conditions.

Therefore, an effective adversarial training pipeline requires a reliable and strategic method for crafting adversarial data [34] [35] [36].

Furthermore, the interaction between a machine learning classifier and an adversary can be modelled as a two-player game, where each player's move prompts a counter-response from the other. In such strategic settings, no single strategy is universally optimal against all possible adversary behaviours. Instead, each agent must observe or anticipate the other's actions and respond with a strategy that maximizes their own payoff or minimizes potential losses. The large number of possible strategy combinations quickly becomes complex, especially as the models and attacks increase in sophistication. Game theory offers a mathematical framework for navigating this complexity, providing tools to identify equilibrium strategies that balance the objectives of competing agents [26] [29] [37] [35].

In adversarial learning, this framework helps formalize the interaction between a classifier (the defender) and an attacker. Each party aims to either optimize model performance or degrade it, depending on their role. The attacker may seek to find the smallest perturbation that causes misclassification, while the defender aims to preserve accuracy under such transformations. Classical game theory assumes players behave rationally and make decisions based on their knowledge of the opponent's incentives and constraints. In adversarial learning, this translates to selecting model parameters or defence strategies that either anticipate or adapt to the attacker's moves [34] [35] [36].

Adversarial training, while effective in increasing model robustness, often leads to reduced accuracy on clean (unperturbed) samples. For example, Zhang et al. (2019) developed an advanced adversarial training algorithm that improved robustness but resulted in a CNN with 89% accuracy on the CIFAR-10 dataset, lower than the 96% accuracy achieved through standard training on the same dataset [38].Since adversarial attacks do not occur continuously in deployment, it may not be optimal to commit to a pure defence strategy that prioritizes robustness at the expense of baseline performance. Instead, when the defender possesses prior knowledge about the likelihood or nature of attacks, it becomes beneficial to adopt a mixed strategy, distributing responses across multiple models or defences according to a probability distribution [39] [40].

In object classification tasks, defenders often have access to a set of pretrained convolutional neural networks (CNNs), each optimized to minimize classification loss under standard conditions. Meanwhile, attackers may choose among multiple strategies for optimizing perturbation vectors that transform the input during an attack. A rational defender, armed with probabilistic knowledge of the attacker's behaviour, can leverage this information to mix strategies and maximize expected payoff. This avoids the trade-off between robustness and accuracy associated with committing to a single defensive model [41] [42].

Many existing game-theoretic approaches in adversarial learning focus on the adversary-classifier interaction but often assume idealized or simplified conditions. These models frequently overlook uncertainties and the dynamic nature of real-world deployment environments. Game theory, particularly in its Bayesian or Stackelberg forms, allows for more realistic modeling by incorporating partial information, asymmetric knowledge, and sequential decision-making. Optimization in these frameworks is typically formulated as a non-linear, non-convex quadratic programming problem with constraints that include classification accuracy, misclassification error, and regularization terms. These elements are weighted by probability distributions over the strategy space, enabling the formulation of defences that adapt over time and across varying threat levels [37].

Therefore, applying game theory to adversarial machine learning provides a robust and flexible foundation for modeling the strategic interplay between attackers and defenders. It supports the design of learning algorithms that balance accuracy and robustness, adapt to varying adversarial conditions, and optimize outcomes in uncertain, multi-agent environments.

Adversarial samples are designed to exploit weaknesses in a machine learning classifier by selecting inputs that are likely to cause misclassification. These attacks often use knowledge of the defender's strategy to increase their effectiveness. To build robust classifiers capable of withstanding such threats, it is essential to incorporate insights about potential adversarial behaviours into the training phase. A game-theoretic model provides a structured mathematical framework for representing this interaction. Through iterative

adversarial training, the model minimizes loss on adversarial samples, rather than relying solely on the natural data distribution. This iterative process mimics the actions of intelligent and adaptive adversaries, offering a more realistic and practical defense mechanism for deployed systems.

Moreover, adversarial examples have demonstrated transferability across different machine learning models. An input that causes one classifier to misclassify can often mislead another, even if the models differ in architecture or training data [43] [44] [45]. This phenomenon underscores the importance of general robustness in machine learning systems. Many existing game-theory-based approaches model only single-round interactions between the learner and the adversary [46]. In contrast, our proposed framework incorporates historical interactions, allowing the defender to learn from past adversarial behaviour and make more informed decisions about selecting appropriate classifiers or defences.

Consequently, we conduct a literature review to investigate the relationship between adversarial attacks and the robustness of machine learning algorithms within game-theoretic contexts. Our review explores how adversarial perturbations lead to nonlinear and complex data representations, disrupting the classifier's ability to generalize. We examine current methods used to improve robustness, assess the impact of adversarial samples on learning systems, and highlight key challenges in defending against such attacks.

Furthermore, we explore the integration of game theory, reinforcement learning, and quantum computing into adversarial training frameworks [47] [22] [34]. These emerging approaches offer promising directions for improving model resilience, particularly in high-stakes or real-time applications. Reinforcement learning enables adaptive learning strategies in dynamic environments, while quantum machine learning introduces powerful tools for modelling complex, high-dimensional input spaces [48].

In this work, we propose a game-theoretic framework,the Decomposed Optimal Bayesian Stackelberg Solver (DOBSS),to model the interaction between a single defender and multiple adversaries. This framework enables the defender, who lacks certainty about the adversary's type, to use prior knowledge over adversary distributions to compute a high-

rewarding mixed strategy. Unlike traditional adversarial training methods that primarily defend against a fixed adversarial strategy, our approach integrates multiple adversarial behaviors, attack strengths, strategy variations, and prior information into the learning process.

In conclusion, our focus is on developing robust machine learning classifiers through adversarial learning enhanced with quantum kernels, reinforcement learning, and game-theoretic optimization. By combining these methodologies, we aim to advance the development of resilient AI systems capable of maintaining accuracy and security in adversarial settings.

1.2 Thesis Contribution

This research introduces a set of novel methodologies that significantly advance the robustness of machine learning models against adversarial attacks, particularly within the context of image classification using Convolutional Neural Networks (CNNs) and Quantum Support Vector Machines (QSVMs). The core contribution lies in the development and implementation of game-theoretic frameworks, specifically leader follower game models, to design adversarial training strategies tailored for modern neural network architecture. The research also expands into the integration of reinforcement learning and quantum computing paradigms to enhance adversarial defense mechanisms.

The first major contribution is the design of an adversarial training methodology based on the Weighted Adversarial Stackelberg game. In this formulation, the training process of a MobileNet CNN model is conceptualized as a leader-follower dynamic, where the model (defender) anticipates and responds to actions of adversarial agents (attackers). This game-theoretic model introduces asymmetric weighting schemes that emphasize adversarial data points during testing, thereby guiding the model to focus more on potential vulnerabilities in the input space. By solving for the Stackelberg equilibrium, we derive a pure strategy model that optimizes the learning parameters of MobileNet. This

equilibrium effectively reduces misclassification errors and enhances the generalization ability of the model, enabling it to perform robustly in adversarial settings.

To further augment the effectiveness of the Stackelberg formulation, the study incorporates the SARSA (State-Action-Reward-State-Action) reinforcement learning algorithm into the MobileNet architecture. SARSA functions as an adaptive tuning mechanism that adjusts the model's decision-making in response to dynamically changing adversarial inputs. The integration of SARSA not only strengthens MobileNet's resilience but also enhances its ability to learn defensive policies in an online, iterative manner. Empirical evaluations demonstrate that the SARSA-augmented Stackelberg-trained MobileNet consistently outperforms conventional adversarial training methods in terms of accuracy and robustness across multiple benchmark datasets.

In addition to focusing on a pure strategy model, the research presents a Bayesian Stackelberg game framework designed to handle scenarios involving multiple intelligent adversaries. In this model, the defender lacks precise knowledge about the types of adversaries it may face but possesses prior distributions representing this uncertainty. The defender, acting as a learner, computes an optimal mixed strategy by solving the expected payoff matrices that encapsulate both defender and adversary strategies. The Bayesian Stackelberg equilibrium derived from this interaction provides a probabilistic defense strategy that can flexibly adapt to varying adversarial behaviors.

This Bayesian game formulation extends beyond MobileNet and is applied to other CNN architectures, showing generalized improvements in adversarial robustness. The mixed strategy approach proves particularly effective when adversaries employ varied and sophisticated perturbation techniques. Our empirical results validate that the Bayesian Stackelberg-trained models yield significantly lower classification errors under high-perturbation attacks compared to traditionally trained models.

Complementing the game-theoretic methods, this thesis explores the application of Quantum Support Vector Machines (QSVMs) in adversarial defense. A key contribution is the comparative evaluation between QSVMs and adversarially trained classical SVMs

(advSVMs) under increasing levels of adversarial perturbations using the MNIST and CIFAR-10 datasets. The study demonstrates that QSVMs maintain superior classification accuracy even as the strength of the adversarial attacks intensifies. This robustness is attributed to the ability of QSVMs to operate in high-dimensional Hilbert spaces, where adversarial perturbations become less effective at distorting decision boundaries.

Further analysis models the adversarial interaction between a defender deploying either QSVM or advSVM, and an attacker aiming to exploit model vulnerabilities. In this two-strategy game, the defender evaluates potential payoffs and selects between quantum and classical models. The results consistently indicate that the QSVM strategy yields higher payoffs, affirming the advantage of quantum-enhanced machine learning models in adversarial environments.

By integrating these methodologies, this thesis establishes a multi-faceted defense framework combining game theory, reinforcement learning, and quantum computing. Each component contributes to a holistic approach for developing resilient classifiers capable of withstanding a wide spectrum of adversarial threats. The contributions span theoretical modeling, algorithm design, and empirical validation, laying the groundwork for future research in adversarially robust machine learning.

In summary, the key contributions of this thesis are as follows:

1. Development of a Weighted Adversarial Stackelberg training methodology for MobileNet CNNs.

2. Integration of SARSA reinforcement learning for adaptive adversarial defense.

3. Formulation and application of Bayesian Stackelberg games to derive optimal mixed defense strategies against multiple intelligent adversaries.

4. Empirical validation of enhanced robustness across different CNN architectures.

5. Comparative analysis demonstrating the superiority of QSVMs over classical advSVMs under adversarial conditions.

6. Game-theoretic modeling of QSVM and advSVM deployment strategies, highlighting the practical benefits of quantum models.

These contributions collectively underscore the importance of strategic, adaptive, and quantum-empowered learning systems in countering the ever-evolving landscape of adversarial machine learning threats.

1.3 Significance of Research

There are multiple ways a machine learning classifier can be attacked, and no general defense guarantees robustness against all attack types. For example, a spam detector system: such systems are frequently targeted by malware designed to evade detection. A machine learning-based spam detector and the malware it targets have opposing goals. Malware developers modify malicious content to appear benign, exploiting the detector's assumptions, such as the expectation that training and test data are independently and identically distributed. This mismatch causes the detector to fail. If an attacker can manipulate the training data, the classifier may begin to misclassify inputs.

This interaction between attacker and defender can be modeled as a two-player game, where each party seeks to optimize its own outcome. In security scenarios, it is unrealistic for any player to maintain a dominant strategy that guarantees consistent success. Instead, a range of strategies must be considered. Game theory provides a mathematical framework for identifying equilibrium strategies when players interact strategically.

This thesis proposes a set of adversarial learning algorithms designed to mitigate vulnerabilities in machine learning classifiers when faced with strategic adversaries. These algorithms can be applied to a range of applications, including computational systems, data analytics, mobile networks, recommender systems, image classification, and medical or biological image processing data analysis.

ML algorithms have become foundational in tasks such as image classification, speech processing, and malicious code detection. Despite their strong predictive capabilities, these models are highly susceptible to adversarial perturbations, small, carefully constructed modifications to input data that can lead to incorrect classifications with high

confidence. These perturbations are often imperceptible to the human eye, allowing them to pass unnoticed during normal usage. When integrated into systems deployed in safety-critical or privacy-sensitive environments, the consequences of such vulnerabilities can be severe.

Adversarial examples exploit the underlying structure of DNNs by subtly altering features in ways that shift model predictions across decision boundaries. Although some attacks are designed with specific classifiers in mind, many adversarial perturbations generalize across different models. This transferability makes them a broader threat, raising concerns about the reliability of machine learning systems in real-world applications. For example, in autonomous vehicles, a misclassification of traffic signs,such as interpreting a stop sign as a speed limit sign,could result in dangerous behavior on the road. In biometric systems, adversarial tampering with facial images may allow unauthorized access or block legitimate users, thereby compromising system integrity and user privacy.

The development of adversarial examples is not only a method for attack but also a valuable tool for evaluating model robustness. By studying how and why models fail under such conditions, researchers can better understand the limitations of existing architectures and develop more effective defensive strategies. This includes exploring how DNNs make decisions, how they represent data internally, and what vulnerabilities exist at different layers of abstraction.

The rise of big data and artificial intelligence has accelerated the deployment of DNNs in autonomous systems, especially in transportation. Among the various modules in an autonomous driving system, traffic sign recognition is particularly vulnerable. Even small patches or stickers placed on physical road signs can manipulate the visual input in a way that misleads the recognition system. Studies have shown that printed adversarial patches,designed digitally and then transferred to the physical world,can fool camera-based classifiers in real-time driving scenarios. This introduces a new class of risks, where digital manipulation can result in physical-world failures, highlighting the urgent need for resilient models capable of detecting or resisting adversarial inputs.

In summary, adversarial attacks expose fundamental weaknesses in how deep neural networks process information. Addressing these challenges requires not only stronger defense mechanisms but also a deeper understanding of model behavior under adversarial conditions. This understanding is essential for building trustworthy AI systems, especially as they continue to be integrated into critical domains such as healthcare, finance, security, and autonomous transportation.

Generating adversarial samples plays a central role in assessing and improving the robustness of deep neural network (DNN) classifiers. These samples are deliberately modified inputs designed to expose weaknesses in machine learning models. By incorporating adversarial examples into training processes, models can be conditioned to recognize and resist inputs that would otherwise lead to misclassification. This process has become a fundamental step in building classifiers that maintain accuracy under adversarial conditions.

To develop effective defenses, it is necessary to understand how adversarial examples are generated. The core idea is to identify how an input sample can be perturbed in a way that causes the model to misclassify it, while the changes remain imperceptible to human observers. The perturbation must be large enough to push the input across the decision boundary of the model, yet small enough to maintain its apparent integrity. If the perturbation is excessive, it becomes detectable and loses its effectiveness. If it is too subtle, the model may classify it correctly and the intended effect is not achieved.

The key challenge lies in computing the optimal level of perturbation. Biggio et al. addressed this by applying gradient descent techniques to the cost function of the classifier, integrating the model's behavior with the input distribution to locate sensitive regions. Szegedy et al. introduced a method using the sign of the gradient vector to generate perturbations, which reliably fooled a variety of classifiers. Their work also demonstrated that retraining models using adversarial inputs could act as a regularizer, thereby improving general robustness. These early methods laid the foundation for many of the adversarial training techniques used today.

While much of the research on adversarial attacks has focused on digital environments, real-world implications are becoming increasingly evident. Kurakin et al. showed that adversarial patches, once printed and photographed, could still deceive trained classifiers like Inception v3. Evtimov and colleagues applied altered patterns to road signs, successfully misleading vision systems used in autonomous vehicles. These results confirmed that adversarial attacks are not confined to simulated settings and can be transferred into physical scenarios with tangible consequences.

The emergence of physical-world adversarial attacks highlights critical concerns about the reliability and safety of AI-driven systems. As attack techniques become more adaptable, transferable, and harder to detect, defending against them becomes more complex. Understanding the mechanisms and behavior of these attacks is essential for designing classifiers that can withstand a broader range of threats. This is particularly relevant in domains where the margin for error is minimal, such as autonomous driving and biometric authentication.

Moreover, advancements in AI and big data have accelerated the adoption of convolutional neural networks (CNNs) in driverless technologies. Among various modules in such systems, traffic sign recognition is notably exposed to adversarial interference. Even small, well-placed patches on a road sign can cause a CNN-based system to misidentify critical information, such as mistaking a stop sign for a speed limit sign. These vulnerabilities translate into real safety risks and emphasize the need for models that can generalize under both normal and adversarial input conditions.

Building robust CNN models requires an understanding of the adversary's behavior and constraints. However, a persistent challenge is the uncertainty surrounding the type and capability of attackers the system may face at deployment. Even when a model is trained on diverse, high-quality data, it may still be vulnerable to unknown adversarial strategies. In such cases, it becomes important to formulate defense strategies that account for this uncertainty. Game-theoretic approaches, such as Bayesian modeling or Stackelberg games, offer potential frameworks for choosing optimal strategies under uncertain adversarial conditions.

In conclusion, improving CNN resilience requires a combination of techniques that go beyond conventional training. It involves adversarial data generation, targeted retraining, physical-world testing, and uncertainty modeling. These elements are essential to develop classifiers that are not only accurate under standard conditions but also capable of withstanding adversarial manipulation in real-world systems with critical safety and security implications.

Justification For Perturbation (PGD) Attack

The PGD (Projected Gradient Descent) attack method is widely recognized as one of the strongest first-order adversarial attack algorithms due to its ability to repeatedly refine perturbations through multiple gradient steps while ensuring the adversarial sample remains within a bounded ε-ball. Although its per-sample processing time is higher than simpler methods like Fast Gradient Sign Method (FGSM), the increased accuracy and success rate of the attack justify its use, particularly in security-critical evaluations. Unlike single-step methods, PGD explores the local loss surface more effectively, making it less prone to gradient masking effects.

When implemented on the Mini ImageNet dataset, PGD demonstrates an attack success rate comparable to or exceeding that of I-FGSM. For example, PGD-JND achieved an attack success rate of 98.2% versus 97.5% for I-FGSM-JND. While the average time to generate adversarial samples using PGD is approximately 1.25 seconds, higher than I-FGSM's 0.7 seconds, it consistently produces adversarial examples that are closer to the decision boundary and more robust under model defences or transformations.

Evaluations on CIFAR and MNIST datasets show that across all distance metrics (ℓ_0, ℓ_1, ℓ_∞), PGD finds adversarial samples with high success rates and better alignment with perturbation constraints. Unlike some attack methods that may fail to converge in high-dimensional spaces, PGD maintains strong attack performance across models and input dimensions. For instance, under ℓ_∞ constraints, PGD reliably generates imperceptible yet highly effective perturbations on ImageNet-class images, often requiring only minor

pixel-level changes. It has been shown that PGD retains a 100% attack success rate under controlled ε-balls while also remaining resistant to common defence techniques.

PGD is particularly effective against robust models due to its iterative refinement process and is often used as a benchmark attack in adversarial training. It is also less sensitive to initialization, as each iteration projects the perturbed input back into the valid ε-ball, ensuring stability and repeatability of the attack process. As learning tasks and model complexity increase, PGD remains effective and scalable, unlike some second-order or optimization-heavy methods that degrade in performance or efficiency. For example, PGD consistently finds stronger ℓ_0 and ℓ_2 adversarial examples with lower perceptual distortion, especially in scenarios where previous methods underperform due to gradient obfuscation.

Overall, PGD stands out as a rigorous and adaptable method for generating adversarial images. It balances attack strength and perceptual quality effectively and remains a standard for evaluating model robustness. When combined with perceptual tuning approaches such as JND (Just Noticeable Difference), PGD further improves visual fidelity while retaining high attack effectiveness, making it a reliable choice for adversarial robustness studies and training.

Goodfellow et al. introduced the fast gradient method to generate adversarial perturbations by exploiting the direction of the cost function gradient. This approach emphasized the efficiency and relevance of the gradient's direction in crafting effective perturbations. Subsequent work proposed targeting classes with the lowest prediction confidence or perturbing features most influential to the model's output based on forward gradients. These methods aim to fine-tune the adversarial input generation process to exploit vulnerabilities in the classifier more effectively. In both cases, the attacker is modelled as the leader who samples strategies stochastically, while the classifier (follower) responds by searching for a strategy that leads to equilibrium based on the available knowledge.

1.4 Thesis Structure

This report is structured into six chapters. Chapter 1 provides the introduction and outlines the motivation and objectives of the study. Chapter 2 presents a literature review,

examining existing research in adversarial learning and defence mechanisms. Chapter 3 offers a detailed survey of the robustness of deep neural networks, covering various attack types, defence strategies, and safety concerns. Chapter 4 introduces the concept of robust machine learning in the presence of multiple adversaries. Chapter 5 explores adversarial learning using a Bayesian Stackelberg game framework to model interactions between a learner and multiple intelligent attackers. Chapter 6 discusses the application of quantum machine learning, specifically quantum SVM algorithms, as a defence against gradient-based adversarial attacks. Finally, the report concludes with a summary of findings and suggestions for future research directions.

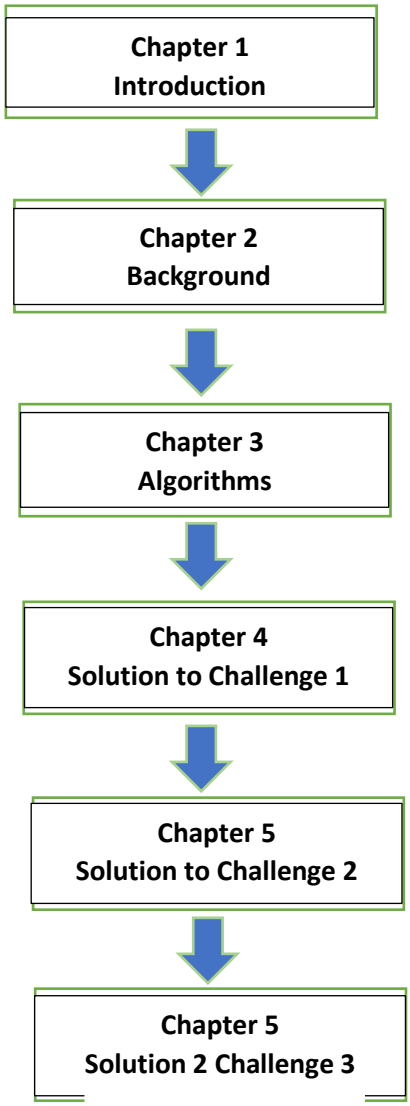

Fig 1.1 Thesis Structure

Chapter 2

Background and Literature Review

This section covers existing adversarial learning algorithms, attack strategies, and defence mechanisms aimed at improving classifier robustness. It also summarizes current developments in quantum algorithms and quantum machine learning within the context of cybersecurity.

2.1 Game theory framework

Previous research demonstrates that conventional neural-network training methods often fail to ensure robustness in convolutional neural networks (CNNs). Azulay *et al.* (2018) showed that simple regularization and data-augmentation techniques only partially mitigate misclassification and do not address underlying model fragility. Likewise, Mathew *et al.* (2021) mathematically proved that vanilla neural-network training remains inherently unstable or inaccurate for classification tasks [54–57, 51]. These works collectively suggest that accuracy improvements achieved through standard optimization are not synonymous with resilience. Ali Shafahi *et al.* (2020) further demonstrated empirically that the dimensionality and visual complexity of inputs strongly affect a classifier's real-world vulnerability, reinforcing that structural properties of the data not merely training heuristics determine robustness. Consequently, adversarial learning becomes a necessary paradigm for developing CNNs resistant to practical attack methods [58]. Building on this insight, our research employs adversarial training within a Stackelberg-game formulation to seek a mixed-strategy equilibrium that jointly optimizes accuracy and robustness for fixed-dimension CNNs [51, 59–61].

Game-theoretic modeling has been widely used to formalize the interaction between a classifier and an adversary. Gauthier *et al.* (2021) applied the minimax theorem to derive equilibrium conditions for non-convex, non-concave games, revealing that deterministic pure Nash equilibria often fail to capture the stochastic nature of adversarial learning [37, 47]. Ya-Ping *et al.* introduced a two-player algorithmic framework capable of efficiently computing mixed Nash equilibria [62], while Avishek *et al.* (2021) formulated a max-min adversarial game to clarify the fundamental trade-off between accuracy and robustness in neural networks. However, several of these models assume idealized rational behavior or convergence that is rarely attainable in practice. Chivukula *A.S. et al.* (2021) extended this work using a Stackelberg formulation with variational adversaries but did not prove equilibrium existence [2]. Tanner Flez and Chi Jin (2020) further examined local optimality in sequential games [63], yet their analyses remain largely theoretical. These studies collectively highlight that most adversarial-learning games treat the classifier–attacker dynamic as non-cooperative but leave open how equilibrium can be achieved under realistic constraints. By contrast, Jie Ren et al. (2021) explored cooperative-game defenses [24], offering conceptual breadth but less relevance to hostile environments. Our work focuses instead on non-cooperative Stackelberg interactions, where an adaptive attacker continuously perturbs inputs to increase misclassification, requiring the defender to anticipate and counter probabilistic strategies.

A prerequisite for designing resilient learners is a precise understanding of how adversarial data are generated [64]. The essence of adversarial example creation lies in crafting perturbations that remain imperceptible to humans yet push the input across the classifier's decision boundary [65]. If the perturbation is excessive, the sample becomes visibly distorted and trivial to detect; if it is too small, the model's prediction remains unchanged [66, 5, 7]. This delicate balance underpins both attack realism and defense evaluation.

Seminal works established the foundation for modern adversarial training. Szegedy *et al.* (2014) first demonstrated that adding a small perturbation vector aligned with the sign of the loss gradient reliably induces misclassification across diverse models [67]. Their findings introduced the principle that training with worst-case adversarial perturbations acts as a regularizer, improving generalization under attack. Goodfellow *et al.* (2014) formalized

this idea through the Fast Gradient Sign Method (FGSM), emphasizing the critical role of gradient direction in generating effective perturbations [51]. Madry et al. (2018) advanced this approach via Projected Gradient Descent (PGD), recasting adversarial robustness as a min-max optimization problem that better approximates constrained, iterative attack scenarios [3, 5, 66]. Despite these advances, existing methods remain computationally intensive and often trade off clean accuracy for robustness gaps that motivate our Stackelberg-based framework for achieving a balanced, theoretically grounded equilibrium between the two.

To develop optimal defensive strategies, adversarial attack models must be explicitly defined, since no single learning approach can universally defend against all threat types. Existing neural-network defenses often overfit to specific attacks, leaving models vulnerable to unseen perturbation schemes. Several studies highlight a persistent trade-off between accuracy and robustness, where improving one tends to degrade the other [68, 38]. Moreover, the diversity of adversarial settings defined by norms such as L_0, L_1, L_2,and L_∞ complicates the generalization of defenses, as varying perturbation magnitudes yield different sensitivities and adversarial accuracies [69–71]. These limitations underscore the urgent need for algorithms that generalize across multiple attack classes without sacrificing baseline accuracy or stability.

Huang et al. (2011) conceptualized machine learning as a dynamic interaction between a learner and an adversary, where the learner aims to classify inputs correctly while the adversary perturbs data to induce misclassification [72]. Their model positioned adversarial learning as an inherent cybersecurity threat in domains such as email filtering, autonomous vehicle vision, and medical imaging. Building on this, Kantarcioglu et al. (2011) used a Stackelberg equilibrium with a simulated annealing algorithm to identify optimal feature subsets for classification [73]. Similarly, Liu et al. [74] relaxed the common assumption of mutual knowledge between players' payoff functions, proving that knowing only the adversary's payoff can still yield optimal strategies. Both studies treated the adversary as the leader and the classifier as the follower, reflecting asymmetric knowledge and initiative in real-world attacks.

Madry et al. (2018) extended this framework through min-max robust optimization, formally defining adversarial training as a constrained optimization problem [75, 76]. This theoretical foundation remains one of the most influential contributions to adversarial defense, although it assumes ideal convergence and significant computational resources. Subsequent works introduced refinements to improve robustness and efficiency. Huimin Zeng et al. (2021), for example, incorporated a learnable weighted minimax function into the loss objective, improving balanced performance under both uniform and non-uniform attacks [77]. Other studies explored non-zero-sum Stackelberg formulations, computing optimal learner actions against worst-case linear attacks [63]. Hardt et al. (2016) modelled adversarial interaction as a sequential game in which the attacker perturbs inputs based on shared logits, demonstrating that classifier parameter secrecy can be crucial for sustained generalization [78]. Dritsoula et al. (2017) further applied a non-zero-sum framework where defenders distinguished malicious from benign interactions [47, 79, 80]. Their findings emphasized that equilibrium existence depends heavily on loss design, regularization, and penalty structure. Bruckner et al. subsequently derived necessary conditions for the existence of unique Nash equilibria in such adversarial classifications [81, 5, 9, 82–88], establishing theoretical underpinnings but offering limited practical scalability.

Game-theoretic frameworks remain appealing because they are algorithm-agnostic applicable to models ranging from Support Vector Machines (SVMs) to deep neural networks provided that payoff functions for the learner and adversary are well-defined. However, achieving practical equilibrium requires careful modeling of the attack space. Zhou et al. (2016) formulated an adversarial SVM problem as a single-leader, multi-follower Stackelberg game, countering multiple attacker types using the DOBSS (Decomposed Optimal Bayesian Stackelberg Solver) method [89]. Chengzi et al. (2019) applied a Bayesian Stackelberg game to infrastructure defense, again using DOBSS to manage heterogeneous adversaries and enhance protection in real time [90]. Despite these advances, most prior research continues to treat adversarial learning as a simultaneous game with convex probability strategy spaces, limiting the applicability of equilibrium results in non-convex deep-learning landscapes.

Our study addresses this limitation by formulating adversarial learning as a Stackelberg game under DOBSS, proving that a Stackelberg equilibrium can exist for CNNs of fixed structure and that this equilibrium yields the highest adversarial accuracy and optimal defense strategy among comparable models [91, 92]. While adversarial training has traditionally been posed as an optimization problem, it can equivalently be interpreted as a hierarchical Stackelberg interaction in which the classifier acts as the leader. In our Bayesian Stackelberg formulation, the learner selects an optimal mixed strategy under uncertainty, while followers (adversaries) vary in type and intent. The classifier does not know the specific adversary it faces but can infer its distribution or type through prior observations.

Comparable work, such as that of [93], modeled a single-leader, single-follower Bayesian Stackelberg game to represent interactions between a security agency and an unknown-type criminal. Their model solved the equilibrium through linear programming, where the defender first commits to a strategy and the adversary responds optimally after observation. Extending this reasoning, our approach constructs a generalizable defense mechanism capable of adapting to multiple attack distributions by probabilistically selecting optimal strategies that maximize robustness without explicit prior knowledge of the attacker's exact type.

2.2 Adversarial Interaction

2.2.1 Non-Cooperative Game

Adversarial learning can be modeled as a 2 player non cooperative game. A non-cooperative game can be defined as an interaction between 2 or more players over a utility to be shared by the players. The normal form representation of the game is expressed as (N, A, U), where N is set players in the game, $A = \{A_i\}$ where A_i is the set of actions for each player i and $U = \{U_i\}$, for $U(a_i, a_{-1})$ is a valued outcome receivable by each player $i \in \{1, \dots N\}$ when it chooses a strategy $a \in A_i$ and other players jointly select strategy $a_{-i} \in A_{-i,}$ the utility gained for each player determines its preference over the different possible outcomes of the game that result from the joint actions of other players. Player

$i's$ action set given as $S_i = \{p(A_i): p(A_i) \geq 0\ \forall i, \sum_{A_i} p(A_i) = 1\}$ shows the probability distributions for it actions. A technique used to evaluate best response by players in a game is the Nash equilibrium. A strategy profile $S = (s_1 \ldots\ldots, s_N)$ is a Nash equilibrium if a strategy s_i played by player i is the best response for all possible strategies S_{-1} that is $s_i \epsilon\ BR(s_{-1}), \forall i.$ Thus, , the best response R_i of player i to other players playing s_{-i} is $R_i(s_{-1}) = argmax\pi_i(s_i, s_{-1})$ given that π_i is the payoff to player i when other players play s_{-1}Actions since adversarial learning is a 2-player game between a learner and adversary NE solution techniques [49] can be used to determine the both players strategy.

Bruckner et al. studied prediction games where both players acted simultaneously, each committing to a strategy without prior knowledge of the other's move. In such settings, an optimal strategy cannot be explicitly defined in advance. However, assuming both players are rational and seek to maximize their own utility, it is expected that each would adopt a strategy corresponding to a Nash Equilibrium. The authors modelled this interaction as a static two-player game, where both players know their respective action spaces and cost functions, and aim to maximize utility while minimizing cost. Bruckner et al. also provided sufficient conditions for the existence of a Nash Equilibrium in these settings.

2.2.2 Zero sum and Sequential games

Adversarial learning problems can be modelled as a zero-sum game between a learner and an adversary where one player's gain is another player's loss which can typically be expressed as a minimax strategy. In the this case the learner chooses the best strategy assuming the adversary is responding to maximize the learners loss$min_{\omega^*} max_{\delta^*} L(f, x, \delta)$, where $\omega^*\ and\ \delta^*$ are the optimal learning parameters of the classifier and the optimal distortioned that can generated by the adversary tand applied x go maximize the learners loss function L. In a sequential game where both players are trying to minimize their cost, the players follow a sequential action. A player (the leader) commits to a strategy first, while the other player (the follower) observes the first player's actions and plays to maximize their loss. For instance, the leader commits to a classifier f by learning the

parameterω the the follower, who is the adversary after observing ω chosses δ to minimize his own cost but maximizes the loss L of the leader as shown by the following

$$min_{\omega^*} max_{\delta^*}\ L_l(f, x, \delta)\ \text{ s.t } \delta^* \epsilon\ argmin L_f(\omega, x, \delta),$$

given that $L_l\ and\ L_f$ are the loss functions of the learner and adversary respectively

An interaction between players in which the total payoff is a constant or sums up to zero can be modelled as a zero-sum game. In such interaction, one player's loss is the other players gain, adversarial learning problems can be modeled as zero-sum games where the adversary's objective is to maximize the loss of machine learning classifier and following a minimax strategy. Suppose a set of data $\{(x_i, y_i)\epsilon(X, Y)\}\ _{i=1}^{n}$, where $y_i \epsilon \{-1,1\}$ is the target, $X \subseteq \mathbb{R}^d$, d is dimensional feature space and n is the total number of examples. The adversary aims to move the malicious data In any direction by addition some perturbation vector δ_i to $x_i|_{y_i=1}$. `The adversary needs to balance between the risk of exposure and potential profit from the attack, a common strategy its Suppose Madry et al (2018) investigated the adversarial robustness of neural networks with the use of robust optimization [49]. The authors defined security attacks using a min-max formulation in a theoretical framework, which captured the importance of using adversarial training and existing methods for attacking neural networks.

In sequential games, players are typically aware of each other's moves. The leader makes the first move, and the follower selects a response based on the observed strategy. This type of interaction is modelled as a sequential game and is referred to as a Stackelberg game (SE). Kantarcıoğlu et al. solved an SE using a genetic algorithm to identify an optimal set of attributes. Liu et al. addressed a similar problem but assumed only the follower's payoff function was known. In both cases, the attacker is modelled as the leader who samples strategies stochastically, while the classifier (follower) responds by searching for a strategy that leads to equilibrium based on the available knowledge [50] [51] [52] [53] [54] [55].

2.2.3 Stackelberg Game

A Stackelberg game models a setting where one player, the leader, commits to a strategy first, and other players, the followers, observe this choice before selecting their own strategies. The leader must commit to the chosen strategy and cannot switch during gameplay. The strategy can be either pure or mixed. A Stackelberg game can be described as a tuple (N, A, H, Z, χ, ρ, σ, μ), where *N* is the set of players, *A* is the set of possible actions, *H* is the set of nonterminal nodes, χ defines the available actions at each node, ρ indicates the active player at a node, *Z* is the set of terminal nodes, σ is the successor function determining the next node based on the current action, and μ represents the utility functions for the players. To define a Stackelberg equilibrium, a best response function maps a leader's strategy to the follower's optimal response. This function $f: S_1 \rightarrow S_2$ satisfies the condition $u_2(s_1, f(s_1)) \geq u_2(s_1, s_2)$ for all $s_2 \in S_2$. The expected utility from a mixed strategy is the weighted sum of utilities from pure strategies. A strategy profile $(s_1, f(s_1))$ is a Stackelberg equilibrium if the follower's strategy is a best response to the leader's strategy under the utility function.

2.2.3.1 Bayesian Stackelberg Game

Adversarial learning problem can be defined as an input space $X \in \mathbb{R}^d$ where d is the number of attributes in the vector space. For a learning model classifier † with an input $x \epsilon X$ and a corresponding output give as $y \epsilon \{+1, -1\}$, there is an adversary able to corrupt the model at test time by an amount δ such that a malicious instance x will be misclassified as benign given by $f(x) \neq f(x + \delta)$. Thus adversarial machine learning focuses to obtain a robust algorithm such that the probability of the algorithm misclassifying even under attack is as small as possible $P\big(f(x) \neq f(x + \delta)\big) < \varepsilon \; for \; \varepsilon > 0$.

If we have input samples $x_{i|i=1,\dots n} \epsilon \chi$ and want to estimate target label $y_i \epsilon \Upsilon$ where $\Upsilon = \{+1, -1\}$ to be classified by a learner function $g: \chi \rightarrow \mathbb{R}$ with a feature vector $\phi(x \epsilon X) \in \mathbb{R}^d$ The predicted value $\hat{y} = g(w, x_i)_{|w \epsilon \mathbb{R}^N}$ is obtained by optimizing a loss function L. The learner's loss function with a regularization is given as $L=\sum_{i=1}^{n} \ell(\hat{y}_i, y_i) + \lambda||\omega||^2$ where λ is a regularization parameter that penalizes weights ω of the classifier. A cost

vector c is included in the loss function to reflect the weights of individual input data, and the learner now optimizes the equation:

$$\underset{\omega}{\operatorname{argmin}} L = \underset{\omega}{\operatorname{argmin}} \sum_{i=1}^{n} c_i . \ell(\hat{y}_i, y_i) + \lambda ||\omega||^2.$$

The loss function can be extended to an adversarial learning problem. If an adversary wishes to influence the learner by modifying the input data, then the learner's classification task to obtain $\hat{y}$ on the transformed data becomes $\hat{y} = \omega^T . \phi\big(f_t(x_i, \omega)\big)$ where f_t is the function used by the adversary to transform the data:

$$f_t(x_i, \omega) = x_i + \delta_x(x_i, \omega),$$

δ_x is the displacement vector that determines the level of perturbation of original input x_i, hence the adversarial learning can be defined as $\underset{\omega}{\operatorname{argmin}} \underset{\delta_x}{\operatorname{argmax}} L(\omega, x, \delta_x)$.

The machine learning classifier and adversarial follower play a Stackelberg game. The learner chooses the best model g from a set of fitted models $\mathcal{G}$: $\underset{g \epsilon \mathcal{G}}{\operatorname{argmin}} \ell(g)$, that is the optimal prediction function based on the data transformation by the adversary. The learner plays the optimal mixed strategy preassigned to the models $g \epsilon \mathcal{G}$, allowing the learner to randomize over available strategies with a probability distribution.

2.2.3.2 Single Leader Follower Stackelberg Game

A set of learning models $\mathcal{G}$ can be modelled as a non-zero sum and sequential two player game between the learner and the adversary, where the learner commits first to a move observable by the adversary follower who now plays an optimal strategy to minimize the learners' payoffs while maximizing his. The leader's loss is the misclassification error given as

$$L_\ell^g = \sum_{i=1}^{n} c_i . \ell_\ell(\hat{y}_i, y_i) + \lambda_\ell ||\omega||^2,$$

While the follower's loss comprising of the cost exposure to the learner due to misclassification and also the cost of data transformation

$$L_f = \sum_{i=1}^{n} c_{f,i} . \ell_\ell(\hat{y}_i, y_i) + \lambda_f \sum_{i=1}^{n} \big|\big|\phi(x_i) - \phi\big(f_t(x_i, \omega)\big)\big|\big|^2.$$

Where λ_l, λ_f and c_ℓ, c_f are the weights of the penalty terms and cost of data transformation for the leader and follower. The adversarial learning model can be found by solving the optimization problem: $\min_{\omega^*} \max_{\delta_x^*} L_\ell(\omega, x, \delta_x)$ such that $\delta_x^* \epsilon \operatorname{argmin}_{\delta_x} L_f(\omega, x, \delta_x)$.

2.3 Multiple Agent Interaction

Interaction between multiple agents aiming to optimize their strategies often depends on equilibrium analysis, especially when a leader-follower structure is involved. This can be effectively modelled using a Stackelberg game framework. In this model, the leader commits to a strategy first, anticipating the best possible reaction from the follower. A key defines strategy is to derive a Stackelberg equilibrium that maximizes the defender's expected payoff, considering that the attacker may or may not choose to launch an attack. The defender's approach depends on what triggers strategy selection and the nature of the player's reasoning process. This includes whether players act deterministically or stochastically, and whether they rely on prior knowledge, observed behaviour, or inferred beliefs about the opponent's actions. The sophistication of each agent reflected in their strategic assumptions, learning capabilities, and payoff estimation affects the dynamics and outcome of the game. an SE using a genetic algorithm to identify an optimal set of attributes. Liu et al. addressed a similar problem but assumed only the follower's payoff function was known. In both cases, the attacker is modelled as the leader who samples strategies stochastically, while the classifier (follower) responds by searching for a strategy that leads to equilibrium based on the available knowledge.

The learner commits to a set of learning models $g \epsilon \mathcal{G}$ and $\mathcal{G} = \{g_s, g_{f_1} \dots .\}$. The learning function g_s is the Stackelberg equilibrium solution $g_s(\omega_s, x) = \omega_s^T . \phi(x)$, given that ω_s is the Stackelberg solution for the leader, the other functions $g_f$is depends on the followers solution of the Stackelberg equilibrium (SE), which is obtained from the optimal data transformation $\delta_x(\omega_s)$. $g_f$is not a robust solution to the adversary's response since the adversary can easily defeat the model g_f by transforming the data, however if the adversary transforms x using δ_x then g_f will perform better than the Stackelberg solution

learning function g_s in terms of classification error. A set of g_f can be trained by varying the impact of data transformation on the follower's loss function. When the penalty term λ_f is large, the adversary's best strategy is to reduce the disparity between the original data and the transformed data or simply do not transform data at all, conversely when λ_f is relatively small, the adversary has more leeway to perform data transformation, however the adversary cannot transform data arbitrarily because of the increase in the cost of misclassification that follows. Therefore, a spectrum of adversary types can be specified from that range from least aggressive to most aggressive.

Reinforcement learning (RL) extends the idea that an agent tends to adopt strategies that have yielded better payoffs in past interactions. In traditional RL settings, neither the defender nor the adversary maintains a formal model of the other's behaviour. Instead, decisions are guided by observed rewards or penalties in response to selected actions. Adversarial reinforcement learning builds on this by integrating an adversarial training environment into the learning loop. This allows an agent to improve its response strategies in the presence of varying and potentially disruptive inputs.

Studies introduced a model where the training environment evolves in tandem with the learning process. The environment adjusts its reward responses based on the classifier's performance over perturbed data, simulating adversarial conditions. This feedback structure enhances the classifier's capacity to generalize and adapt. The reinforcement learning model does not require complete information about the adversary but instead learns effective policies through interaction and reward shaping [56].

ML algorithms were investigated for their behaviour under significant perturbations and implemented a reward-based adversarial training system for reinforcement learning agents. They reformulated the adversarial risk function to find a balance between model robustness and accuracy. Their training approach led to policies that withstand stronger attacks and performed more reliably than conventional methods. The adversarial setting forced the learning agent to refine its strategy over time, improving its effectiveness in environments where adversarial actions are expected. to identify an optimal set of attributes. Liu et al. addressed a similar problem but assumed only the follower's payoff

function was known. In both cases, the attacker is modelled as the leader who samples strategies stochastically, while the classifier (follower) responds by searching for a strategy that leads to equilibrium based on the available knowledge [57].

Reinforcement learning algorithm was integrated with the training environment of a classifier such that the model generates new adversarial sample by randomly extracting data from training set while generating rewards based on the accuracy of the classifier's predictions. This presentation of adversarial reinforcement learning resulted in increased final performance for intrusion detection [58].

The defender minimizes the loss on perturbed datasets, observes the payoff and fine-tunes the learning environment based on a positive or negative reward, in this case the accuracy on perturbed dataset. The primary objective of the reinforcement of learning algorithms is to increase the total sum of rewards during adversarial training. The reward ensures that the accuracy of the present state is higher than the previous state. The Q-value estimate gives the total discounted reward when the defender selects strategy a in state s. this serves as an updating procedure by which the adversarial training process starts with arbitrary training hyperparameters with initial values of $Q(s, a)$ and updates the Q-values using the function.

2.4 Adversarial Algorithms for robust machine learning

2.4.1 Adversarial Data Generation

To improve the resilience of machine learning models against adversarial attacks, it is important to understand how adversarial samples are generated. The goal of adversarial data generation is to identify mechanisms by which an adversary can create perturbed inputs. The adversary aims to modify a valid input sample in a way that is imperceptible to humans but causes the machine learning model to misclassify the input. This is typically achieved by introducing just enough perturbation to move the input across the decision boundary of the classifier. If the perturbation is too large, the sample becomes visibly distorted. If it is too small, it remains within the classifier's correct classification zone and has no impact.

The key objective is to determine the optimal level of perturbation that causes misclassification without altering the perceptual integrity of the input. Biggio et al. used a gradient descent method that incorporates the gradient of the model's cost function with the data distribution to compute this optimal perturbation. Szegedy et al. proposed adding a small vector to an input based on the sign of the gradient of the model's cost function, which was found to reliably mislead various classifiers. They showed that training the model with adversarial examples instead of the original input acts as a regularization method and improves robustness.

Goodfellow et al. introduced the fast gradient method to generate adversarial perturbations by exploiting the direction of the cost function gradient [59]. This approach emphasized the efficiency and relevance of the gradient's direction in crafting effective perturbations. Subsequent work proposed targeting classes with the lowest prediction confidence or perturbing features most influential to the model's output based on forward gradients. These methods aim to fine-tune the adversarial input generation process to exploit vulnerabilities in the classifier more effectively. In both cases, the attacker is modelled as the leader who samples strategies stochastically, while the classifier (follower) responds by searching for a strategy that leads to equilibrium based on the available knowledge.

2.4.2 Machine Learning Attack Models

There are numerous ways a machine learning based system can be attacked, a broad classification attack type can be such that the adversary has a freedom to attack the feature set of a particular data, conversely an adversary can also be restrained by the range of perturbation than can impose on the data. Generally, an adversary will be reluctant to move data too far away from its original position since greater distortions usually incur more cost and loss of malicious utility. In the free-range attack the adversary is familiar with the range in feature set of the data. If $x_{.j}^{max}$ and $x_{.j}^{min}$ be the upper and lower bound values of the j^{th} feature of a data point x_i , a free-range attack is defined such that the attacked data appear legitimate in the given domain [60].

$$C_f\left(x_{.j}^{min} - x_{ij}\right) \leq \delta_{ij} \leq C_f\left(x_{.j}^{max} - x_{ij}\right), \forall j \in [1, d],$$

Given $C_f \epsilon [0,1]$ controls the impact of the attacks.

Conversely an attack method that penalizes excessive data corruption can be modelled. if x_i is a malicious data point from data X and x_i^t is target benign point the adversary would like to distort x_i to. Choosing benign x_i^t is unidirectional because it must be in a good benign class, therefore choosing this point will involve a high level of knowledge on the path of the attacker. Practically, however, the attacker may not be able to modify x_i directly to x_i as desired because the malicious point x_i may lose much of its malicious features. Hence for each feature j in the dimensional feature space, and assuming the adversary perturbs x_{ij} byδ_{ij}

$$\left|\delta_{ij}\right| \leq \left|x_{ij}^t - x_{ij}\right|, \forall j \epsilon d,$$

and δ_{ij} is further bounded as shown as:

$$0 \leq (x_{ij}^t - x_{ij})\delta_{ij} \leq \varphi (x_{ij}^t - x_{ij})^2$$

$$\varphi = \left(1 - C_\delta \frac{\left|x_{ij}^t - x_{ij}\right|}{\left|x_{ij}\right| + \left\lceil x_{ij}^t \right\rceil}\right).$$

where $C_\delta \epsilon [0,1]$ is a constant that regulates the loss of malicious utility because of the distortion on x_{ij}. The model regulates how much the attacker can force x_{ij} towards x_{ij}^t based on how far they are apart. The parameter φ is the ratio of $\left|x_{ij}^t - x_{ij}\right|$ that is the maximum value which δ_{ij} can be. When C_δ is held constant the closer $x_{ij}$is to the target x_{ij}^t the larger the range for which x_{ij} can move towards the target data point x_{ij}^t .$\left(x_{ij}^t - x_{ij}\right)\delta_{ij} \geq 0$ ensures δ_{ij} is in the same direction as the as the target x_{ij}^t. C_δ determines how much malicious utility the attacker is willing to risk for crossing the decision boundary, a larger C_δ means that they will be a smaller loss of malicious utility with the δ_{ij} value selected, while a smaller C_δ means they will be more loss of malicious utility and a more aggressive attack [60].

2.4.3 Adversarial Attack Methods for Deriving Perturbation

The aim of training a dataset with a classifier image is to correctly label all input images to target label set. A classifier model can correctly classify a sample x to a it corresponding label y expressed as

$$\arg\max P(y_i|\, x) = y_{true}.$$

Given that $y \epsilon Y = \{y_1, y_2, \ldots, y_k\}$ is an output label class with k unique classes. $P(y_i|x)$ shows the confidence value of model in predicting a sample x to y_i . Hence the adversarial attack aims to generate adversarial sample such as small perturbation δ added to x will lead the classifier model to predict another label other that the correct label y_{true}

$$\arg\max P(y_i|\, x') \neq y_{true}\,, x = x + \delta.$$

2.4.3.1 Fast Gradient Sign Method (FGSM)

The method generates adversarial samples by adding perturbations in the direction of the loss function that is the positive direction of the slope gradient, a normal input image x, FGSM calculates a similar adversarial example x' to fool the classifier. x' is derived by optimizing the loss function, defined as the cost of classifying x'as a label lx with minimum possible perturbation

$$x' = x + \varepsilon.sign(\nabla_x\, Loss(x, l_x)).$$

2.4.3.2 IFGSM

This is an extension of FGSM but computes perturbations in iterations rather than in a single shot, achieving samples of better image quality than FGSM. The FGSM algorithm is simply applied multiple times with miniature perturbations rather than a single large one. After the completion of each iteration the pixels are cropped such that the perturbation remains as close as possible to the input image x.

$$x_i = clip_{x,\epsilon}(x_{(i-1)} + \epsilon.sign\left(\nabla\, x_{(i-1)} Loss\left(x_{(i-1)}, y\right)\right)).$$

Where $Loss(x, l_x)$ shows the cost function given x as an input image, l_x as the corresponding true output label and ε the parameter that determines the magnitude of perturbation for x.

2.4.3.3 Deepfool

The Deepfool algorithm is a non-targeted obtains adversarial samples by evaluating the minimum perturbations near the classifier model decision boundary and then minimally modifies the input image to reach the bound to result in a classification error. Deepfool has a more consistent success rate than its FGSM counter since the magnitude of perturbation is generally small, and the samples are more difficult to detect [61] [25] [20] [62].

2.4.3.4 Carlini and Wagner (I-FGSM)

The algorithm is calculated using the L norm methods. The algorithm achieves higher success rates with minimal perturbations compared to other methods. I-FGSM algorithm with the L_2 norm has the best performance which is a form of optimization attack that solves the generates adversarial samples by optimizing the expression

$$||x - x'||_2 + \lambda \max(-k, Z(x')_k - \max\{Z(x')_{k'} : k \neq' k\}).$$

Given that k can be adjusted to control the confidence at which misclassification occurs that is the confidence gap between the real and sample category [18].

2.4.4 Adversarial Defense Methods for Individual and Ensemble Neural Networks

NeuralNets are unique for their efficiency in mobile and edge devices primarily due to their depthwise separable convolutions, which reduce computation in the first few layers. However, studies have revealed that MobileNets are prone to adversarial attacks that can significantly impair their performance in image classification tasks. Even slight perturbations on images can cause substantial declines in classification accuracy. To counter these vulnerabilities, adversarial training methods have been proposed, aiming to bolster the resilience of deep neural networks against such attacks. Adversarial training methods were

initially proposed to enhance the resilience of deep neural networks against adversarial attacks. Over time, this approach has proven to be highly adaptable, finding applications in various domains of machine learning. The core idea revolves around the generation of adversarial examples during the training process, which forces the model to adjust and refine its decision boundaries. Prominent methodologies utilised include the Fast Gradient Sign Method , Projected Gradient Descent (PGD) and adversarial training employing generative models.

Research indicates that models trained with single-step adversarial training methods may overfit, reducing their effectiveness against adversaries. However, integrating dropout scheduling into single-step adversarial training can result in more robust models. A hyperparameter introduced to control overfitting enables these models to defend not only against single-step but also multi-step attacks. For instance, Feature-Level Adversarial Training (FLAT) is designed to ensure consistent predictions for both original and adversarial example pairs, and utilizing variational word masks further guides the model to focus on datapoints that enhances accuracy and robustness against adversarial attacks.

Numerous studies have also modelled adversarial training as a simultaneous game between a classifier and an adversary. In such games, the adversary perturbs data using point-wise perturbations to transform the training data, with the goal of increasing misclassification errors for the classifier while avoiding detection. The problem is formulated as a worst-case min-max game, where both the classifier and the adversary aim to minimize the adversarial loss. Strong perturbation attacks are achieved through Projected Gradient Descent (PGD) to train robust learning models in a single-step min-max interaction. Additionally, results from PGD-based attacks can be emulated using Fast Gradient Sign Method (FGSM) by reducing the curvature along the perturbed direction projected by FGSM. This is accomplished by regularizing the curvature of the attack and restraining the projection to align with those generated by PGD attacks [63] [64] [65]. An introduced hyperparameter controls the curvature along the attack direction and regularizes the model. A game theory framework proposed by Ambar et al. explores attacks and defenses, leading to equilibrium in a simultaneous game setting.

In the context of adversarial attacks on reinforcement learning algorithms, these attacks are presented as generated noises that result in the misclassification of the learning algorithm. Rajeswaran et al. investigate an ensemble of models for robust reinforcement learning, combining deep neural networks with reinforcement learning to create a robust agent. The interaction between the adversary and the reinforcement learning agent is akin to a min-max game theory formulation. Adversarial training in reinforcement learning enhances robustness against attacks that mislead the reinforcement learning agent into believing it is in a worst-performing trajectory state, leading to sub-optimal actions. While adversarial training based on mini-max formulation is often overly pessimistic and may not generalize well over test distributions, a more practical approach involves sequential interactions between classifiers and adversaries. In this scenario, the defender initially selects a model while knowing the existence of an optimal adversary. The adversary then chooses a strategy while considering the defender's choice. This hierarchical nature of Stackelberg games provides the defender with a first-mover advantage, constraining the adversary's choices to optimize their own payoff. For example, a game can be modelled as an optimization problem between a data generator and a learner within a Stackelberg game framework. Gao et al. demonstrated the existence of Stackelberg equilibrium that converges to an optimal robust classifier in interactions between Deep Neural Networks (DNNs) and adversaries. Adversaries not only focus on perturbing data but can also manipulate the dataset distribution to maximize classification errors during test time. Traditional adversarial defense mechanisms train models on uniform training data distribution, which may not generalize well to unseen adversarial data distributions at test time. The Adversarial Risk Importance method is effective in generalizing well under both uniform and non-uniform attacks. Furthermore, Distributionally Robust Optimization (DRO) has been combined with adversarial training to produce more robust models. The goal of adversarial training is to reduce classification loss during test time, which necessitates a hierarchical interaction occurring sequentially between classifiers and adversaries [66] [67] [68] [69] [70] [71] [72] [73].

Combining both Stackelberg game and weighted adversarial learning methods provides an effective defense mechanism that generalizes well across test distributions for a

defender. While several works have independently explored game theory frameworks, reinforcement learning and distribution-based robust optimization, this paper introduces a novel approach by combining both Stackelberg games and reinforced weighted adversarial training [74] [75] [76] [77] [78]. The objective is to obtain a classifier that effectively generalizes to both perturbation and targeted attacks particularly those deployed against mobile and edge devices using a deep neural networks defence mechanisms.

2.4.4.1 Friendly Adversarial training (FAT)

Given a dataset $S = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ and $B_\epsilon[x]$ is a closed ball of radius $\epsilon > 0$ at x in $\mathcal{X}$. The objective function of standard adversarial training (Madry et al,2018) is.

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left\{ \max_{\tilde{x} \in B_\epsilon[x_i]} \ell(f(\tilde{x}), y_i) \right\}.$$

FAT defines a margin $\rho > 0$ such that an adversarial data is classified with a certain amount of confidence. $\tilde{x}_i$ is generated using the inner minimization problem rather than the usual maximization solution as shown.

$$\tilde{x}_i = arg \min_{\tilde{x} \in B_\epsilon[x_i]} \ell(f(\tilde{x}), y_i)$$

$$s.t\ \ell(f(\tilde{x}), y_i) - \min_{y \in \mathcal{Y}} \ell(f(\tilde{x}), y) \geq \rho.$$

The constraints ensure $y_i \neq arg \min \ell(f(\tilde{x}), y_i)$ or $\tilde{x}_i$ is misclassified, and also ensures that for the adversary $\tilde{x}_i$ the wrong prediction is better than the desired prediction y_i by at least p in terms of loss value. From all x satisfying the constraint, the one minimizing $\ell(f(\tilde{x}), y_i)$ is selected. Hence, the adversarial loss is minimized given that some confident adversarial data has been obtained. The adversarial data $\tilde{x}_i$ is referred to as a 'friend' among the adversaries, hence the term friendly adversarial data. Consequently, an upper bound was also derived for the adversarial risk. Given any classifier f any loss function ℓ that upper bounds the classifer and any probability distribution $\mathcal{D}$, we have the adversarial risk defined as

$$\mathcal{R}_{rob}(f) \leq \underbrace{\mathbb{E}_{(X,Y)\sim\mathcal{D}}\ell(f(X),Y)}_{For\ standard\ test\ acurracy} + \underbrace{\mathbb{E}_{(X,Y)\sim\mathcal{D},X'\in B_\epsilon[X,\epsilon]}\ell^*(f(X),Y)}_{For\ robust\ test\ acurracy}$$

$$\ell^* = \begin{cases} \min \ell(f(X'),Y)+\rho, & if\ f(X')\neq Y, \\ \max \ell(f(X'),Y), & if\ f(X')=Y\,. \end{cases}$$

ρ is the small margin that the friendly adversarial data would be classified with some amount of confidence. Equation above shows that the upper bound is tighter than those of standard adversarial training such as TRADES (Zhang et al,2019b), where the loss is maximized regardless of the model prediction $\ell^* = \max \ell(f(X'), Y)$. On the other hand, the FAT bound takes the model prediction into account in that when the model makes a correct prediction on the adversarial data the loss is still maximized but when the model makes a wrong prediction on adversarial data X' the inner loss minimized through a small constant ρ.

2.4.4.2 Weighted Minimax Risk Models

Standard methods of adversarial training solve a minimax problem between a classifier minimizing the loss over an update on input perturbations and seeking a convergence to equilibrium [49]. The inner loop generates the strongest perturbation δ_i within the radius ϵ of each input example x_i, and the model minimizes the expectation of adversarial loss function $(f(x_i + \delta_i), y_i)$ according to the equation

$$\min_\theta \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}}\left[\max_{\delta_i:\|\delta_i\|<\epsilon} \ell(f_\theta(x_i+\delta_i), y_i)\right].$$

Hence the model parameters adjust to the generated perturbations added to the training data to combat potential adversaries at test time. A potential problem to the method is that all generated adversarial examples despite their distances to the decision boundary an varying risk of being misclassified are treated equally when empirically estimating the adversarial loss. Furthermore, due to the adversarial nature of attacks, the adversarial samples at test time x'_{test} may not have the same distribution as the adversarial examples generated at training time. Thus, it is likely that the distribution of adversarial risk is not the same as the i.i.d of clean data points.

To tackle the problem the weighted minimax risk approach defines a margin such that for a data point (x_i, y_i), the margin is the difference between the classifiers confidence in predicting the correct model y_i and the maximal probability of an incorrect label t. The Adaptive margin aware minimax risk uses an exponential family parameterized by the margin of the adversarial examples in training

$$\min_{\theta} \sum_{i=1}^{m} \max_{\delta_i : \|\delta_i\| < \epsilon} e^{-\alpha\, margin(f_\theta(x_i+\delta_i), y_i)}\, \ell(f_\theta(x_i + \delta_i), y_i).$$

Where $\alpha > 0$ is a hyperparameter of the exponential weight kernel. The impact of this is that there is a positive correlation between the weight kernel and individual loss, that is a larger individual loss will induce a larger weight and vice versa. Also, the distribution of the attack $\mathcal{D}'$ deployed may deviate from the empirical distribution $\mathcal{D}$ represented by the training examples. The true distribution is often intractable; however, it is assumed that the divergence between the empirical distribution and the attack distribution is bounded by a threshold divergence (Namkoong and Duchi 2016). Hence, a risk estimator for each data point to express the distribution of the adversarial examples and learn it via training is beneficial. A distributionally robust optimization will only require evaluation an importance weight at each minibatch of training data $(x_i, y_i)_{i=1}^{N}$. and can improve distributional robustness against adversarial perturbations $(x_i', y_i)_{i=1}^{N}$. An importance weight $s(f_\theta, x_i', y_i)$ which is a ratio of the adversarial examples distribution and clean data points is evaluated at training as

$$s(f_\theta, x_i', y_i) = \frac{\mathcal{D}'(x_i', y_i)}{\mathcal{D}(x_i, y_i)}.$$

Therefore, the re-weighting strategy – adaptive margin-aware risk trains the object function considering a full batch gradient decent as follows

$$\tilde{\mathcal{L}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} s(f_\theta, x_i', y_i)\,\ell(f_\theta(x_i + \delta_i), y_i)$$

$$\approx \mathbb{E}_{(x,y)\sim\mathcal{D}}[s(f_\theta, x', y)\,\ell(f_\theta(x'), y)]$$

$$\approx \mathbb{E}_{(x',y)\sim\mathcal{D}'}[\ell(f_\theta(x'), y)].$$

Since the weighting factor is learnable, the objective of this method can be thought of as learning the adversarial example distribution conditioned on a model θ through learning of the importance weight using the loss $\tilde{\mathcal{L}}(\theta)$.

2.4.4.3 Single Step Adversarial Training with Dropout Scheduling Method

Models trained using sing-step adversarial training method prevent the generation of single-step adversaries due to overfitting of the model during training. A single-step adversarial training with dropout scheduling learns a more robust model. Typical setting of adding drop out layer with fixed probability does not help the single-step trained model in gaining robustness. Considering the empirical training objective formulated as a minimax optimization problem

$$\min_{\theta} \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}} \left[\max_{\delta_i:\|\delta_i\|<\epsilon} \ell(f_\theta(x_i+\delta_i), y_i) \right]$$

$$R_\epsilon = \frac{loss_{adv}}{loss_{clean}}.$$

R_ϵ can be interpreted as that if $R_\epsilon > 1$ which is the same as $loss_{adv}$ being greater than $loss_{clean}$ then there is an adversarial perturbation. Conversely, if $R_\epsilon < 1$ meaning $loss_{adv}$ is less than $loss_{clean}$ the generated perturbation is not an adversarial perturbation, and the attack method fails to generate adversarial perturbations for the model. The single-step training method introduces a dropout layer after each non-linear layer of the model to be trained. The dropout layers are initialized with a high dropout probability P_d. Then during the training $P_d$is linearly decayed for all dropout layer and controlled by the hyper-parameter r_d. The hyperparameter r_d is expresses in terms of maximum training iterations meaning that the dropout probability reaches zero when the current training iteration reaches half of the maximum training iterations. This training method learn to prevent the generation of adversaries due to over-fitting during training and the resultant model achieves robustness not only in single-step attacks but also against multiple step attacks.

2.4.4.4 Regularized FGSM (FGSMR) for Adversarial Training

Tianjin H. et al (2020) increased the similarity between vanilla FGSM and Projected Gradient Descent (PGD) attack by reducing the curvature along the perturbed direction projected by FGSM. This was achieved by regularizing the curvature of the FGSM and restraining the projection to make the perturbed direction close to those generated by PGD-inf attacks. Restraining the gradient direction along the FGSM, which is the second direction derivative, gives a perturbed direction that can be expressed as

$$\nabla_{xg}^2 L_\theta(x) = \lim_{h \to 0} \frac{\nabla_x L_\theta(x+hg) - \nabla_x L_\theta(x)}{h},$$

also given a curvature regularization term R_θ then the adversarial training optimization objective is to minimize the expression: $\min_\theta L(x + \epsilon g) + \lambda R_\theta$. The hyperparameter λ is penalizing factor for controlling the curvature along the FGSM direction. Robust models trained by adv.FGSMR had higher perturbed data accuracy than adv.PGD for PGD-infinity and FGSM attacks, also adv.FGSMR models achieved state of the art accuracy on clean MNIST datasets. For further comparison, the times spent on training 50 epochs with adv.FGSMR for ResNet-18/34 models was considerable lower that adv.PGD since the later takes k (usually k is set to 20) iterations of forward and backward process to find an optimum perturb vector in the l_∞ ball while adv.FGSM takes only 1 iteration for the forward and backward process to find a perturbed vector and 2 times forward and backward process for the curvatures regularization.

2.4.4.5 Feature - Level Adversarial Training (FLAT)

FLAT seeks to increase model resilience by ensuring that a model consistently predicts original or adversarial examples pairs. To do this, FLAT uses variational word masks to choose the appropriate words from an original/adversarial example pair so that the model can predict them. Variational word masks act as a bottleneck in the training process, educating the model to base predictions on key words to guarantee the accuracy of the model's predictions. Throughout training, they learn the global word significance. Also, FLAT normalizes the global relevance of the terms that were changed in an original example and

their replacements in the adversarial counterpart so that the model would perceive the related phrases as having the same importance.

Using original/adversarial example pairs, FLAT aims to build a robust model with consistent prediction behaviors.

$$\min_{\theta,\phi} \mathcal{L}_{pred} + \gamma \mathcal{L}_{imp}$$

$$\mathcal{L}_{pred} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f_\theta(g_\theta(x)), y)] + \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L\left(f_\theta\left(g_\theta(x')\right), y\right)\right]$$

$$\mathcal{L}_{imp} = \mathbb{E}_{(x,y)\sim\mathcal{D}\cup\mathcal{D}'}\left[\sum_{i,x_i\neq x_i'} \left|\phi_{x_i} - \phi_{x_i'}\right|\right].$$

where cross entropy loss is indicated by $\mathcal{L}(.,.)$ is a teachable vector with the same dimension as the specified text vocabulary, where $\theta_{x_i} \in (0,1)$ denotes the word x_i overall importance. $\gamma \in \mathbb{R}_+$ is a coefficient. By bringing $\theta_{x_i}$and $\theta_{x_2'}$close together, $\mathcal{L}_{imp}$ normalizes the global importance scores of the replaced words $\{x_i\}$ and their replacements $\{x_i'\}$ in an original/adversarial example pair $(x_i;\ x_i')$. The related word pair $(x_i;\ x_i')$ would be selected by $g_\phi(\cdot)$ based on similar relevance score. Based on the crucial phrases $g_\theta(x)$ and $g_\theta(x')$, respectively,$\mathcal{L}_{pred}$ encourages the model to produce the same and accurate predictions on the original and adversarial examples. By maximizing the objective, the model develops a consistent pattern of behavior when predicting comparable texts, increasing its robustness to adversarial assaults (Hanjie et al, 2022).

2.4.4.6 Ensemble models

Suppose $\mathcal{K}$ base networks make up the ensemble model $\mathcal{F}$ represented by the notation $F(x;\ \theta_k)$ for k = 1, 2, …, K. A strategy for modelling $\mathcal{F}$ is obtaining the average over each predictor i.e., $\hat{y}_\mathcal{F} = \frac{1}{K}\sum_{k=1}^{K} F(x;\ \theta_k)$. In simultaneous training and for each training iteration all classifiers are trained on the same mini batch of data. Traditionally, the objective function is only the sum of the individual CE losses plus the ensemble cross-entropy (ECE) loss $\mathcal{L}_{ECE} = \sum_{k=1}^{K} \mathcal{L}_{CE}(\hat{y}_k, y)$, where $\hat{y}_k = F(x;\ \theta_k)$ contains the predictive score of the k-th network and y is a one-hot encoding of the true label for x.

The method is based on diversified learning on the feature level for simultaneous training and involves two regularization schemes. First, the Priority Diversified Dropouts (PDD) that aims to encourage each member to learn diversified feature representations of the input and the Dispersed Ensemble Gradients is amended to the ECE loss as a penalty term for gradient descents of classifiers in similar directions of the learning space. The two parts work to enhance each other. Consequently, members in the ensemble can have more dispersed gradients when learning more diversified features, and vice versa. To enforce diverse learning of deep feature representations among the ensemble networks, the technique additionally involves creating an adjustable dropout in simultaneous training. Each base network may be thought of as choosing this as a feature. Given that the dropout creates sparsity in feature representation by ignoring certain high-level information, resulting in different activation patterns between networks. The ensemble range of activation strength is divided into m intervals and the number of neurons k base networks that fall in the intervals $G_{i=1,2,..,m}$ is counted. The interval with the largest counts are considered as having the priority for activation their neurons. Hence, let the k-th network have $N_m^{(k)}$neurons in the m-th interval G_m, then the total number of neurons in the k-th network is $C_k = \sum_m^M N_m^{(k)}$. Given that k-th network has an activation priority within the interval G-th $t_1 \neq t_2 \neq t_k$ then the keep rate for the k-th network with activation length in the interval G_m is given as

$$p_m^{(k)} = \begin{cases} \alpha, & m = t_k \\ \beta\left(1 - N_m^{(k)}/C_k\right), & m \neq t_k \end{cases}.$$

Where α and β are coefficient parameters that range between [0,1]. Lastly, since the goal is to make adversarial examples on one network less transferable to the other network the Dispersed Ensemble gradient (DEG) is used to obtain a gradient regularization. The conventional CE losses $\mathcal{L}_{CE}(\hat{y}_k, y)$ are calculated as usual as well as their corresponding $g_k = \partial\mathcal{L}_{CE}(\hat{y}_k, y)/\partial x$ for k=1, 2,…, K and the penalty term for the dispersed gradient is given

$$\mathcal{L}_g = \sum_{1\leq i<j\leq K} \frac{(g_i, g_j)}{\|g_i\|\|g_j\|}.$$

Where $\mathcal{L}_g$ is effectively the sum of cosine values for the pairwise input gradients. The gradient dispersion is affected by including the regularization term to the CE loss $\mathcal{L}_{ECE}$:

$$\mathcal{L}_{DST} = \mathcal{L}_{ECE} + \lambda \mathcal{L}_g.$$

2.4.4.7 Collaboratively Promoting and Demoting Adversarial Robustness

This ensemble method considers a model f with the aim of making f robust over a consistent prediction over a ball $\mathcal{B}(x, \epsilon) \coloneqq \{x' : \|x' - x\| \leq \epsilon\}$ around an adversarial data example x'_a in the dataset $\mathcal{D}$ and a distortion boundary ϵ.

$$\mathcal{B}_{secure}(x, y, f, \epsilon) \coloneqq \{x' \in \mathcal{B}(x, \epsilon) : argmax_i f_i(x') = y\},$$

$$\mathcal{B}_{insecure}(x, y, f, \epsilon) \coloneqq \{x' \in \mathcal{B}(x, \epsilon) : argmax_i f_i(x') \neq y\},$$

based on the definition of secure and insecure sets an adversary example can be expressed in four subsets $S_{11} = \mathcal{B}_{secure}(x, y, f^1, \epsilon) \cap \mathcal{B}_{secure}(x, y, f^2, \epsilon)$, where the x_a is predicted correctly by both models. The subset S_{10}, S_{01} are the intersection of a secure set of f^1 model and an insecure set of f^2. Lastly, the S_{00} is equivalent to $S_{11} = \mathcal{B}_{secure}(x, y, f^1, \epsilon) \cap \mathcal{B}_{secure}(x, y, f^2, \epsilon)$,e) i.e. both models wrongly predicts the true label y. Hence, the insecure region of the ensemble should be related to the union $S_{10} \cup S_{01} \cup S_{00}$. The method encourages adversarial samples x'_a inside S_{00} to transfer to S_{10}, S_{01} while the model is trained, and those of S_{10}, S_{01} to move to the subset S_{11}. The transfer flow is implemented via promoting and demoting adversarial robustness to leverage the information of an adversarial example from improving the robustness of the model. To promote the adversarial robustness of a given adversarial example x_a w.r.t the model , empirical adversarial training (Madry et al., 2018) by minimizing the cross-entropy loss $l(f(x'_a), y)$ and x'_a is transformed to the secure set $\mathcal{B}_{secure}(x, y, f, \epsilon)$. On the other hand, to demote x'_a w.r.t the model by max $H(f(x'_a))$ where H is the entropy.

The collaboration strategy that allows an ensemble of multiple individual members. Thus, given an ensemble of N members $f^{en}(.) = \frac{1}{N}\sum_{n=1}^{N} f^n(.)$ parameterized by θ_n, the loss function for a model $f^n, n \in [1, N]$:

$$\mathcal{L}^n(x, y, \theta_n) = \mathcal{C}(f^n(x), y) + \mathcal{C}(f^n(x_a^n), y) + \frac{1}{N-1}\sum_{I \neq n}\left(\lambda_{pm} f_y^n(x_a^i)\mathcal{C}(f^n(x_a^i), y)\lambda_{dm}(1 - f_y^n(x_a^i))H\left(f^n(x_a^i)\right)\right).$$

$\mathcal{B}_{secure}(x, y, f, \epsilon)$ is the set of elements in $\mathcal{B}(x, \epsilon)$ for which the classifier f makes the correct prediction y. Also, the insecure $\mathcal{B}_{secure}(x, y, f, \epsilon)$ is the set of elements in t $\mathcal{B}(x, \epsilon)$ for which f wrongly predicts the true label y.

For a given loss we can see the strength of an adversary that can be derived. That can guide the selection of a loss to settle for, that will in turn be robust.

2.4.5 Data Privacy and Security in Adversarial Learning

Evasion attacks target the decision boundary of a trained classifier by introducing small perturbations to input samples at test time. Biggio et al. highlighted the risks posed by such attacks during both training and deployment of machine learning models. In their work, they evaluated classifiers across different adversarial scenarios in malware detection applications. To improve robustness, they proposed a classifier that uses gradient descent optimization applied to the discriminant function. In this context, the adversary seeks to minimize the classifier's discriminant score, generating data samples likely to cross the model's decision boundary and lead to incorrect classifications.

To counter this, the classifier can incorporate prior knowledge specific to the domain and the adversarial context. This includes understanding the adversary's strategy, attack probabilities, classification priorities, and payoff functions. By modelling these elements, a more informed and strategic defence can be developed, allowing the classifier to anticipate and resist targeted evasion attempts more effectively.

Poisoning attacks manipulate the training data by introducing carefully crafted adversarial samples that shift the learned decision boundary. This undermines model accuracy and generalization. A key challenge lies in the assumption that training data may not fully represent the true data distribution. Attackers exploit this by injecting data points that increase misclassifications. Poisoning can be particularly damaging to models like Support Vector

Machines (SVMs), which are sensitive to outliers and rely on representative training samples.

To analyse and defend against poisoning, adversarial perturbations are introduced into the training set to assess model robustness. Training SVMs using kernels such as polynomial or radial basis function (RBF) kernels helps capture complex relationships in the data. In this context, gradient descent is used to identify adversarial samples that degrade model performance. The optimization process seeks samples that are likely to shift the margin or decision boundary of the classifier.

By simulating these attacks, researchers can better understand how model accuracy deteriorates and develop countermeasures such as data sanitization, robust optimization, or reweighting of training points. These insights contribute to the development of classifiers that are better equipped to detect, adapt to, and recover from adversarial data poisoning while preserving overall performance.

2.5 Quantum Adversarial Machine Learning

Quantum Machine Learning (QML) is a rapidly evolving field at the intersection of quantum computing and classical machine learning. It explores how quantum algorithms can be used to enhance traditional machine learning tasks like classification, clustering, and regression. With the rapid progress in quantum hardware and software in recent years, quantum versions of many standard machine learning algorithms have been proposed. These innovations have sparked excitement over QML's potential to be among the first areas to demonstrate quantum advantage, where quantum systems outperform classical ones in meaningful ways.

QML leverages uniquely quantum phenomena such as superposition, entanglement, and quantum interference to process data in richer, high dimensional Hilbert spaces. This capability offers new opportunities not just for improved speed, but also for novel forms of data representation and analysis. For instance, variational quantum circuits, which are trainable quantum models analogous to neural networks, have become a central framework in many QML applications. These circuits typically operate by encoding classical

data into quantum states using feature maps, then manipulating those states with parameterized gates, and finally measuring them to yield a prediction.

Early QML research focused heavily on algorithmic speedups, a compelling new direction is the quest for robustness, particularly in adversarial contexts. The field of Quantum Adversarial Machine Learning (QAML) has emerged to examine whether quantum models can defend against adversarial attacks, where small but maliciously designed input perturbations cause misclassifications. Although classical models like convolutional neural networks are known to be vulnerable to such attacks, QAML explores the potential for quantum models to offer improved defences.

The geometry of quantum Hilbert spaces introduces both risks and opportunities in this domain. Early studies have highlighted a counterintuitive effect known as the concentration of measure phenomenon. This property implies that in high dimensional Hilbert spaces, the majority of quantum states cluster close to a median value. This clustering of the state space can make variational quantum classifiers especially vulnerable to adversarial attacks, independent of the classifier's design. In other words, adversarial perturbations can push input states outside the model's generalization boundary due to the narrow concentration of valid state space.

Moreover, existing theoretical frameworks for generalization bounds in QML, while promising for evaluating training performance, fall short in adversarial contexts because adversarial examples are intentionally crafted and not drawn from the natural data distribution. This makes defending against them fundamentally different from traditional overfitting scenarios.

Despite this vulnerability, the unique properties of quantum information processing may also provide novel mechanisms for adversarial defence. For example, quantum noise, entanglement, and probabilistic measurement outcomes could be harnessed to create models that are harder to exploit. Several recent studies have demonstrated that quantum enhanced classifiers can outperform classical ones in adversarial settings when trained with appropriately designed defences.

QML is not only a platform for achieving computational speedups but also a promising research area for tackling the growing challenge of adversarial robustness. As both attack strategies and defensive techniques evolve within QAML, the interplay between quantum geometry and model learning continues to offer important theoretical and practical insights into the future of secure and reliable machine learning.

2.5.1 Perturbation Attacks on Quantum ML algorithms

Quantum Machine Learning (QML) integrates quantum computing with machine learning methods. While offering computational benefits, quantum classifiers exhibit susceptibility to adversarial attacks. These vulnerabilities result from structural properties of quantum systems, especially the geometry of high-dimensional Hilbert spaces, and the way data is encoded. This section outlines the foundational reasons for these vulnerabilities and presents empirical and hardware-based evidence that supports their existence.

A key reason for vulnerability in quantum classifiers arises from the concentration of measure phenomenon (COMP). In high-dimensional Hilbert spaces, most data points are near decision boundaries of quantum classifiers. As a result, small changes to inputs can shift them across boundaries, causing misclassifications. Mathematically, the average distance between a random point and its nearest adversarial variant scales as $O(2^{-n})$, where n is the number of qubits. This means quantum classifiers using even modest qubit counts can be impacted by adversarial examples.

Another issue is the existence of universal adversarial examples, which are inputs designed to affect multiple classifiers simultaneously. The required perturbation for such examples scales as $O(\log(k)\,.\,2^{-n})$, where k is the number of models. These examples compromise ensemble methods and increase risk in multi-model systems.

The form of input encoding plays a role in determining adversarial susceptibility. In practice, quantum classifiers do not operate on all of Hilbert space, but on subspaces formed by encoded classical data. For example, in phase encoding, where $x \to \bigotimes_{i=1}^{n} (\cos x_i|0\rangle + \sin x_i|1\rangle)$, perturbations scale as $O(1/\sqrt{n})$. While this scale is less severe than in full-space models, it still demonstrates a measurable risk.

The choice of encoding affects classifier behaviour. Encoding strategies should be evaluated not only for computational or expressive efficiency but also for robustness. Empirical testing is necessary to validate these risks on realistic data distributions and quantum encodings.

A study by Lu et al. tested quantum variational classifiers (QVCs) against adversarial inputs using both classical and quantum datasets. Attacks adapted from classical methods such as FGSM, PGD, and Carlini-Wagner were applied. These methods successfully degraded performance on QVCs trained on clean data, even when images remained visually similar. The models failed to retain accuracy when exposed to crafted perturbations, indicating real risks even on simple tasks like MNIST binary classification.

The role of the quantum states being classified is central to vulnerability. This factor, combined with the encoding method, directly affects whether classifiers are stable under perturbation. Studies show that the data subset used in classification may or may not follow COMP behaviour, further complicating generalizations about robustness.

Although simulations have identified vulnerability patterns, validation requires tests on actual quantum systems. In a recent experiment, Ren et al. trained and attacked QVCs on classical and quantum data using real quantum hardware. The classifiers were successfully deceived using adversarial samples that closely resembled original inputs. The results confirm that adversarial issues are not limited to theory or simulation but exist in current noisy quantum devices.

These experiments highlight that QML is not inherently protected against manipulation. The structure of Hilbert space, design of encoding circuits, and limitations of NISQ hardware contribute to the system's weaknesses. Models that generalize well on clean data can still be exposed when faced with crafted adversarial samples.

In conclusion, quantum classifiers are vulnerable to attack due to both theoretical principles and practical constraints. Hilbert space geometry, encoding strategy, and model design all influence robustness. Theoretical models predict weaknesses, and empirical work confirms their presence on simulated and actual hardware. Defending quantum models must be a priority for QML research, alongside improving efficiency or achieving

quantum advantage. Efforts should focus on developing quantum-aware defence approaches, including data encoding techniques, optimization modifications, and circuit-based mitigation. The study of adversarial robustness in QML remains essential for building dependable and secure quantum learning systems.

2.5.2 FGSM and PGD Justification

The Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) were selected as representative adversarial attacks because they define the lower and upper bounds of attack strength. FGSM performs a single linear perturbation step, providing efficiency and interpretability for baseline evaluation. PGD extends FGSM into iterative updates with projection onto an ε-ball, producing stronger, more adaptive perturbations that test model resilience under worst-case conditions.

Alternative attacks such as the Momentum Iterative Method (MIM), Carlini–Wagner (CW), and DeepFool can achieve high attack success rates but incur heavier computational cost and less analytical transparency. FGSM and PGD thus offer a balanced benchmark suite for evaluating robustness while maintaining reproducibility. Future work can incorporate MIM to analyse momentum-based perturbation accumulation and adaptive adversaries.

2.5.2 Defending Quantum Classifiers

Defending quantum classifiers against adversarial attacks has emerged as a critical research focus in quantum machine learning. A leading strategy replicates classical adversarial training. A quantum classifier is trained using attack-aware loss functions. This technique adapts adversarial inputs during training, promoting resilience to bounded-norm perturbations. Their information-theoretic analysis shows that the generalization gap due to adversarial training decreases with the square root of the sample size and

vanishes in high-dimensional input spaces [27]. This provides theoretical support for adversarial training, especially for models employing rotation-based data encoding.

2.5.2.1 Adversarial Training in Quantum Classifiers

Adversarial training is a prominent defence strategy adapted from classical machine learning to the quantum setting. The method involves training quantum classifiers on adversarial examples to enhance their robustness. Georgiou et al. [27] proposed adversarial training methods tailored for quantum variational classifiers (QVCs), showing that when adversarial perturbations are incorporated during training, the resulting models demonstrate improved stability against input manipulations. Their analysis provides generalization bounds for adversarial trained quantum models, especially under rotation-based encoding. While effective, adversarial training demands significant computational overhead and can be challenging on near-term quantum hardware (NISQ) due to gate fidelity and circuit depth constraints.

2.5.2.2 Quantum Noise as a Defensive Mechanism

Quantum noise, traditionally seen as a barrier to reliable computation, has recently been explored as a potential defence mechanism. Du et al. [3] showed that incorporating depolarization noise into quantum circuits can mitigate adversarial effects by acting similarly to differential privacy mechanisms in classical systems. This approach reduces the sensitivity of the classifier to small input perturbations. Further, Huang et al. [9] introduced a robustness certification method using randomized noise injections during the training and inference phases. Their work demonstrated that added noise could bound the classifier's response to adversarial perturbations while preserving prediction fidelity. These findings suggest that quantum noise, when controlled and well-characterized, can be strategically employed to improve robustness.

2.5.2.3 Randomized Encoding and Benchmarking

Randomization in quantum encoding schemes has also proven effective. Huang et al. [16] proposed a randomized data encoding strategy using random quantum rotations, which obfuscate gradient directions necessary for constructing adversarial examples. This makes

gradient-based attack methods ineffective, improving classifier resilience. Additionally, West et al. [8] conducted benchmarking of multiple quantum classifiers under various classical attack methods, such as Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), on both quantum and classical datasets. Their findings indicate that quantum classifiers exhibit unique robustness patterns not always mirrored in classical counterparts, although they remain vulnerable to certain carefully crafted perturbations.

2.5.3 Challenge and Opportunities

2.5.3.1 Adversarial Attacks

Quantum classifiers face vulnerability due to the structure of the Hilbert space into which classical inputs are embedded. The concentration of measure phenomenon (COMP) causes most quantum states to cluster near decision boundaries, making small perturbations highly effective. The expected distance to the nearest adversarial example decreases exponentially with the number of qubits, scaling as $O(2^{-n}$ [1].

As in classical machine learning, transferability also appears in quantum settings. Adversarial examples crafted for one model often deceive others, as many models tend to learn the same non-robust features from training data [2]. In classical systems, adversarial training helps models identify more robust features, leading to more semantically meaningful perturbations [3].

Quantum adversarial machine learning (QAML) introduces additional complexity. Studies show that quantum-generated perturbations may target robust features by default, even without adversarial training [4]. These structured perturbations can deceive classical models, enabling quantum-to-classical transferability. However, the full mechanism behind this is still unclear and remains a research gap.

Interestingly, the feature-targeting nature of adversarial attacks may also offer advantages to QML. Quantum models may learn data features that are classically inaccessible, simply due to their distinct encoding and transformation processes [5]. As a result, adversarial examples generated by attacking a classical model may fail to transfer to a quantum model. Even if quantum-discovered features do not improve clean-data accuracy, their uniqueness may confer additional robustness. Initial studies confirm that QAML networks exhibit such behaviour [4].

Conversely, the extent to which quantum-generated adversarial examples transfer across quantum models or from quantum to classical systems is still an open question. If QML systems prove

resistant to attacks from classical adversaries, those without access to quantum computers, this could provide an early strategic advantage to QML adopters in critical applications.

2.5.3.2 Data Encoding

Data encoding plays a central role in quantum machine learning (QML) and quantum adversarial machine learning (QAML). It affects the classifier's ability to represent data, generalize patterns, and resist adversarial attacks. In current quantum systems, encoding classical data into quantum states remains a major limitation due to hardware constraints.

Two common encoding methods are amplitude encoding and phase encoding. Amplitude encoding uses the amplitudes of quantum states to represent data, requiring fewer qubits but needing deep quantum circuits. Phase encoding maps input values to qubit rotations. It uses simpler circuits but demands more qubits, which limits its application in current devices.

An alternative is interleaved encoding, which combines encoding layers with trainable quantum gates. This setup improves model expressiveness without significantly increasing circuit depth.

Encoding also affects the robustness of quantum models. Studies have linked specific encoding schemes to model behaviour under noise. Although standard noise and adversarial noise differ, encoding influences how well models tolerate both. Some strategies confine data to structured regions of the Hilbert space, improving resistance to adversarial perturbations compared to methods like amplitude encoding that explore wider spaces.

The encoding method also shapes how data distributes in Hilbert space. Due to the concentration of measure effect, data can cluster near decision boundaries, making models more prone to adversarial perturbations. Choosing an encoding that spreads data more evenly may help reduce this vulnerability.

Encoding influences how adversarial examples transfer between models. Quantum models that capture features not accessible to classical models may resist classical attacks. Conversely, some encodings may expose shared weaknesses.

Current hardware limitations often require compact encodings like amplitude encoding. But as quantum systems improve, more encoding options will become viable. This shift will make encoding a key design factor in QAML, affecting performance and security.

In short, encoding is a structural component that shapes resource use, model performance, and robustness. As hardware advances, careful encoding selection will be critical for building effective and secure quantum models.

2.5.3.3 Quantum Noise

Quantum noise is a fundamental feature of current quantum computing systems, particularly in the Noisy Intermediate-Scale Quantum (NISQ) era. It introduces significant challenges in quantum machine learning (QML) and quantum adversarial machine learning (QAML) by affecting model performance through decoherence, gate errors, and readout inaccuracies. However, quantum noise may also disrupt adversarial attacks, providing a potential defence mechanism.

In QAML, noise introduces variability during model training and evaluation. This variability interferes with the gradient-based optimization processes typically used to generate adversarial examples, reducing their effectiveness. Unlike classical attacks, which rely on deterministic behaviour, quantum models affected by noise are less predictable, making attacks harder to craft and apply consistently.

Quantum noise can also serve as a form of regularization. In variational quantum circuits, repeated evaluations under noisy conditions can prevent overfitting to narrow regions of the data space, including adversarial zones. Some approaches deliberately introduce noise during training to make models more robust. This is similar to classical techniques like dropout or noise injection.

Despite these potential advantages, too much noise leads to poor model fidelity and unstable outputs. A balance must be found between using noise to improve robustness and maintaining accuracy. Strategies such as noise-aware training and error mitigation techniques are required to make use of noise without degrading model quality.

Recent studies have demonstrated these effects in practical settings. Ren et al. conducted experiments on quantum hardware, showing that while adversarial attacks could still affect quantum variational classifiers, the presence of hardware-induced noise weakened their impact and reduced consistency across trials. This illustrates the role of noise in shaping model robustness.

However, hardware limitations remain. Current devices have a limited number of qubits and relatively high noise levels. While small-scale QML and QAML demonstrations exist, more complex and reliable implementations will need improvements in quantum hardware. Short-term progress may come from quantum error mitigation techniques, but long-term scalability requires fault-tolerant systems.

Fault tolerance aims to suppress errors below a critical threshold using error correction codes. One method is the surface code, which encodes logical qubits across many physical qubits to detect and correct errors. Surface codes are among the most studied schemes for building reliable quantum operations. Some early demonstrations of surface code applications show that quantum hardware is approaching the capability needed for small-scale, error-corrected computation.

Quantum noise presents both a problem and a possible solution in QAML. While it limits the depth and reliability of current models, it can also resist adversarial strategies by disrupting their optimization process. Achieving effective and robust QAML will require careful use of noise, supported by error mitigation, and future adoption of fault-tolerant architectures.

Chapter 3

Quantum and Classical Machine Learning Algorithms and Datasets

This chapter introduces the quantum and classical machine learning algorithms, along with the datasets used in this thesis. It covers quantum models and traditional approaches that form the basis for later chapters. These methods support the analysis of how quantum computing can be applied to machine learning tasks. The chapter outlines the core concepts, implementations, and relevance of each algorithm in relation to the datasets. We begin with quantum machine learning algorithms, followed by classical machine learning and deep learning methods, and end with a description of the datasets used.

3.1 Classical Learning Algorithms

In the context of adversarial training, this section outlines the core machine learning algorithms used throughout the thesis. These algorithms serve as the baseline models for evaluating the impact of adversarial examples and the effectiveness of defence mechanisms. The selection includes widely used classifiers such as support vector machines, decision trees, and neural networks, chosen for their relevance in adversarial robustness research.

Each algorithm is introduced with a focus on its vulnerability to adversarial attacks, computational efficiency, and role within the adversarial training framework. This foundation sets the stage for later chapters, where these models are trained and evaluated under adversarial settings to assess their resilience and performance.

3.1.1 Reinforcement Learning

Reinforcement Learning (RL) is a learning paradigm in which an agent interacts with an environment to learn optimal behaviours by maximizing cumulative rewards over time. It differs from supervised learning in that it does not rely on labelled input/output pairs but instead learns from trial and error. The RL framework is often modelled as a Markov

Decision Process (MDP), defined by a tuple (S, A, P, R, γ) where S represents states, A actions, P the transition probability, R the reward function, and γ the discount factor [57] [56] [79] [80].

$$G_t = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\right].$$

The agent's goal is to learn a policy $\pi(a|s)$, which defines the probability of taking action a in state s, in order to maximize the expected return:

Common RL methods include value-based approaches like Q-learning and policy-based methods like policy gradients. In Q-learning, the agent updates an action-value function $Q(s, a)$ that estimates the expected return of taking action a in state s and acting optimally thereafter. Deep Q-Networks (DQN) extend this to high-dimensional spaces using deep neural networks.

In adversarial training contexts, RL plays a dual role. First, RL agents can be targets of adversarial attacks, where small perturbations in state observations lead to suboptimal decisions. This has been demonstrated in environments like Atari games, where imperceptible noise misleads agents into making poor actions [79, 56, 57] . Second, RL can be used as a tool for adversarial defence, where agents are trained under adversarial conditions to become more robust. This typically involves adversarial perturbed states being included during training, following a min-max formulation where the agent learns a policy that performs well even under worst-case inputs.

Adversarial training in RL can also improve safety and reliability in real-world applications such as autonomous vehicles, robotics, and cybersecurity. For example, robust RL has been applied to train agents that maintain performance even when attackers manipulate sensor data or inject malicious behaviour into the environment.

Recent advances such as Proximal Policy Optimization (PPO) and robust adversarial RL frameworks have made it easier to incorporate adversarial resilience into RL models [80]. While challenges remain in sample efficiency and convergence stability, RL continues to be a promising approach for building adaptable and secure AI systems.

3.1.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are widely used for image classification and recognition. They learn spatial features through convolutional layers, activation functions, and pooling operations. The core operation in a CNN is the convolution:

$$Z_{i,j}^{l} = \sum_{m} \sum_{n} x_{i+m,j+n}^{l-1} \cdot w_{m,n}^{(l)} + b^{(l)} .$$

Here, $x^{(l-1)}$ is the input feature map, $w^{(l)}$ is the kernel, $b^{(l)}$ is the bias, and $z^{(l)}$ is the output of layer l. After convolution, a non-linear function and pooling may be applied.

Several CNN architectures have become standard in the field. VGGNet [81] employs deep stacks of small 3×3 convolutions. ResNet [82] introduces residual connections that address the degradation problem in deeper networks, using the identity mapping formula:

$$y = \mathcal{F}(x, \{W_i\}) + X,$$

where $\mathcal{F}$ is a residual function.

MobileNet focuses on computational efficiency, using depthwise separable convolutions to reduce the number of parameters and operations [83]. These architectures are commonly trained on large datasets such as ImageNet [84], which serves as a benchmark for evaluating image recognition models.

Although CNNs perform well in various tasks, they are vulnerable to adversarial examples. These are inputs that have been intentionally perturbed in subtle ways that are often imperceptible to humans but can cause a model to make incorrect predictions. The Fast Gradient Sign Method (FGSM) is a common technique for generating such inputs and is defined by the following equation:

$$x_{zdv} = x + \varepsilon . sign(\nabla_x J(\theta, x, y)).$$

Where x is the original input, ε is a small scalar controlling the perturbation size, J is the model's loss function, and $\nabla_x J$ is the gradient of the loss with respect to the input.

Projected Gradient Descent [49] extends this method by applying iterative updates while keeping the perturbed input within a specified norm ball around the original input.

To improve robustness against these attacks, adversarial training modifies the learning process by including adversarial examples during training. The training objective becomes a minimax optimization problem [85] [84] [49]. The formulation forces the model to learn parameters that perform well even under worst-case perturbations. Variants of ResNet and MobileNet have been successfully trained using this method, resulting in improved robustness, although often at the cost of increased computational requirements and reduced accuracy on clean data.

In addition to adversarial training, other strategies such as architectural modifications, input transformations, and ensemble models have been explored to reduce CNN sensitivity to adversarial inputs. These efforts aim to enhance model reliability in environments where adversarial manipulation may be present[185][189][199].

CNNs remain essential in visual recognition, but their susceptibility to adversarial attacks highlights the need for robust design and training techniques. Ongoing research continues to address these challenges to improve the security and generalizability of CNN-based models[200].

3.1.3 Support Vector Machine

Support Vector Machines (SVMs) are supervised learning models used for classification and regression. SVMs aim to find a decision boundary that maximizes the margin between different classes in the feature space. Given a set of labelled training examples (x_i, y_i) where $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$, the optimal decision boundary is obtained by solving the following optimization problem:

$$min\frac{1}{2}\|w\|^2 \text{ subject to } y_i(W^T x_i + b) \geq 1 \ .$$

Where w defines the orientation of the hyperplane, and b is the bias term. For non-linearly separable data, kernel functions such as radial basis function (RBF) or polynomial kernels are used to project data into higher-dimensional spaces.

SVMs are known for their generalization capability in high-dimensional spaces, but they are not immune to adversarial attacks. Like neural networks, SVMs can be deceived by small perturbations crafted to move an input across the decision boundary. These attacks exploit the geometry of the margin and the model's reliance on boundary decisions.

Recent studies have shown that even simple linear SVMs can be manipulated through carefully generated adversarial examples. Zhang et al. (2020) demonstrated that adversarial perturbations can be constructed using gradient-based methods, and these attacks transfer across different kernel types and datasets. The vulnerability persists even when SVMs are trained with regularization or input preprocessing [86].

To address this, several defence strategies have been proposed. One approach is robust optimization, which modifies the SVM objective to account for worst-case perturbations within an allowable region around each input. This results in a formulation similar to:

$$\min_{w,b} \max_{\|d_i\| \le \epsilon} \frac{1}{2}\|w\|^2 + C \sum_i \max\left(0, 1 - y_i(w^T(x_i + \delta_i) + b)\right)$$

. This formulation trains the SVM to classify correctly even under input perturbations of bounded size ϵ\epsilonϵ. Biggio et al. proposed gradient masking and feature squeezing as preprocessing techniques to make adversarial manipulation more difficult. However, these methods may reduce model accuracy on clean data or be bypassed by adaptive attacks.

A more recent direction involves integrating adversarial training with SVMs by including adversarial examples during training to improve robustness. Liang et al. showed that this approach improves classification margins under perturbations while maintaining performance on original data [87]. Other efforts focus on using SVMs as part of ensemble defences, combining their stable decision boundaries with deep learning classifiers to mitigate attack transferability.

Although SVMs are less commonly used in large-scale image classification, they remain relevant in structured data tasks and as components of hybrid systems. Their decision

margins and convex optimization framework provide a basis for analysing and improving robustness to adversarial inputs [86] [35] [88].

3.2 Principal Component Analysis

Principal Component Analysis (PCA) is a linear dimensionality reduction technique used to project data onto a lower-dimensional subspace that captures the directions of maximum variance. Given a dataset $X \in \mathbb{R}^{n \times d}$ where n is the number of samples and d the number of features, PCA computes a set of orthogonal basis vectors (principal components) by solving the eigenvalue decomposition of the covariance matrix:

$$\sum = \frac{1}{n} X^T X \; and \; \sum v_i = \lambda_i v_i \; .$$

The eigenvectors v_i represent the directions of maximum variance, and the corresponding eigenvalues λ_i indicate their importance. The data can then be projected onto the top k components:

$$X_{PCA} = XV_k \; ,$$

where $V_k \epsilon \mathbb{R}^{d \times k}$ contains the top k eigenvectors.

In adversarial settings, PCA has been explored both as a defensive preprocessing step and as an analysis tool for understanding model vulnerability. By reducing the input dimensionality, PCA can remove noise or irrelevant features that adversarial perturbations may exploit. Xu et al. (2018) showed that applying PCA before feeding data into classifiers such as SVMs or neural networks can reduce the effectiveness of certain adversarial attacks, particularly those that rely on high-dimensional noise.

However, PCA is not inherently robust to adversarial manipulation. Researchers have found that adversarial examples can still be crafted in the reduced feature space or that perturbations can be projected to align with principal components to maintain their impact after transformation. Jagielski et al. demonstrated that attackers can adaptively generate perturbations that survive dimensionality reduction, especially when the defence mechanism is known [89] [90] [91] [71].

Robust PCA variants have been proposed to improve resistance to outliers and adversarial points. These methods attempt to decompose the data into a low-rank structure and a sparse error matrix, isolating adversarial noise. However, these approaches typically introduce additional computational cost and are sensitive to hyperparameters.

In practical applications, PCA is often used in combination with other defences such as adversarial training or feature denoising. While it can reduce attack surface by eliminating redundant input dimensions, PCA alone is insufficient for ensuring robustness. It is better suited as a complementary component in a layered defence strategy [179-181].

3.3 Quantum-Classical Hybrid Models

Quantum-classical hybrid models describe machine learning systems that combine classical data with quantum computational components. These models belong to the CQ category in the quantum machine learning classification proposed by Schuld and Petruccione, which considers the nature of the data (classical or quantum) and the platform used for learning (classical or quantum). In CQ models, classical input data is processed by quantum circuits as part of the learning process, while optimization is generally performed using classical algorithms [92] [93].

This thesis focuses exclusively on CQ quantum machine learning: learning models that operate on classical data and use quantum computing for model evaluation. Within this category, there are multiple architectural possibilities. Some approaches aim to use quantum computing for both model evaluation and optimization [65]. Although these techniques show theoretical promise in accelerating classical, they often rely on hardware capabilities that are not yet available on current quantum devices. Therefore, this thesis does not explore those methods [93] [47] [94].

Table 3. 1 Quantum Machine Learning grouped with respect to nature of Model and data used

	Classical Algorithm	Quantum Algorithm
Classical Data	CC	CQ
Quantum Data	QC	QQ

Instead, this work focuses on CQ models that are compatible with noisy intermediate-scale quantum (NISQ) devices. These models use quantum circuits to evaluate input data, while training is performed using classical optimization routines. Given an input $x \in \mathbb{R}^n$ the data is encoded into a quantum state $|\psi(x)\rangle$. A parameterized quantum circuit $U(\theta)$ is then applied, and an observable $\hat{O}$ is measured to compute the output:

$$f(x;\theta) = \langle\psi(x)\big|U^{\dagger}(\theta)\hat{O}U\big|\psi(x)\rangle.$$

The goal is to find parameters θ that minimize a classical loss function, typically using gradient-based or heuristic optimization methods.

CQ quantum machine learning includes several model types which includes:

3.3.1 Quantum Support Vector Machines (QSVMs)

These models adapt classical SVMs by using quantum kernels that map classical inputs into a space of quantum states. The model operates entirely on classical data but relies on a quantum computer to compute inner products in the quantum feature space.

3.3.2 Quantum Neural Networks (QNNs)

These models are built entirely from quantum circuits and are inspired by the structure of classical neural networks. They use layers of parameterized quantum gates to represent learnable transformations.

3.3.3 Hybrid Networks

These models combine quantum components with classical architectures, such as connecting a quantum neural network to a classical dense layer. Hybrid networks aim to balance the expressive power of quantum circuits with the stability and maturity of classical models.

These models are evaluated in this thesis with a particular focus on their behaviour under adversarial attack. Classical adversarial methods are adapted to test the robustness of quantum classifiers and hybrid networks, examining whether quantum-based architectures provide advantages in adversarial settings [93].

The QQ category, where both the data and processing are quantum, represents a promising future direction. In QQ models, quantum states may be input directly into quantum circuits without intermediate measurement [95] [96] [97]. However, such models require hardware and quantum memory capabilities that exceed the scope of this thesis and current technology. As such, they are excluded from the present analysis.

3.4 Feature maps

In machine learning, a feature map refers to a transformation that projects input data from its original space into a new feature space, often of higher dimension. This transformation

enables models to capture patterns or relationships in the data that may not be linearly separable in the original space. In quantum machine learning, feature maps serve a similar purpose but operate through the preparation of quantum states that encode classical input data.

A quantum feature map is typically defined by a parameterized quantum circuit that depends on the input data. This circuit acts as a data embedding mechanism, it takes a classical input vector $x \in \mathbb{R}^n$ and prepares a corresponding quantum state $|\phi(x)\rangle$. The structure and parameters of the circuit determine how the data is represented in Hilbert space.

Mathematically, a feature map in quantum machine learning is expressed as a unitary transformation $U(x)$ such that:

$$|\phi(x)\rangle = U(x)|0\rangle^{\otimes n}.$$

Where $|0\rangle^{\otimes n}$ is the initial state of the quantum system, and $U(x)$ encodes the data x into a quantum state using parameterized gates. The expressivity and usefulness of a quantum model heavily depend on the design of this feature map, as it defines the geometry of the feature space in which quantum learning models operate.

3.4.1 Angle Encoding

Angle encoding is one of the most used quantum feature maps due to its simplicity and compatibility with near-term quantum hardware. It encodes classical data into quantum states by interpreting each input value as a rotation angle applied to a qubit.

In an n qubit system, angle encoding can embed up to n real-valued input features. The encoding is implemented by applying a rotation gate to each qubit, where the rotation angle corresponds to the input value. Typically, the rotation gate $R_y(.)$ or $R_z(.)$ is used. For example, if x= $(x_1, x_2, \dots, x_n)$ is the input vector, then each qubit q_i undergoes a transformation:

$$|\psi(x)\rangle = \bigotimes_{i=1}^{n} R_y(x_i)|0\rangle^{\otimes n}.$$

This operation results in a product state where the quantum state of each qubit depends directly on the corresponding input feature.

However, the choice of rotation gate matters. If the circuit uses only R_y gates applied to the computational basis state, $|0\rangle$ the resulting state lies entirely in the $y-z$ plane of the Bloch sphere. If instead, R_z gates are used on $|0\rangle$, no transformation occurs, since $R_z(\theta)|0\rangle = |0\rangle$, Therefore, to make the encoding effective when using R_z gates, a layer of Hadamard gates is commonly applied before the rotations. The Hadamard gate transforms $|0\rangle$ into a superposition state, enabling the subsequent rotations to affect the state meaningfully.

Normalization of input data plays a crucial role in angle encoding. Input features should be rescaled to lie within a specific interval, such as $[0, \pi]$ or $[-\pi, \pi]$ depending on the desired spread of encoded quantum states. For example, if data is normalized within $[0, \pi]$ then values near 0 and π are mapped to similar quantum states due to the periodicity of the trigonometric functions involved in rotation gates. In fact, since rotation gates operate modulo 2π, 0 and 2π represent the same angle, leading to ambiguity if the normalization range wraps around this boundary.

This trade-off affects how the feature space is explored. A wider normalization interval allows for broader distribution of states in the Hilbert space, which can improve class separability. However, it may also cause overlap between inputs at opposite ends of the data range. The choice of normalization range must balance the need for spread in the feature space with the need to preserve distinctions between extreme values.

In this thesis, angle encoding is used as the primary feature map in the quantum support vector machine (QSVM). Its structure is hardware-efficient, easy to implement, and allows for clear interpretation of how classical inputs are embedded into quantum states. Moreover, its sensitivity to input values makes it an important component when studying adversarial perturbations, as small changes in input can induce noticeable shifts in the encoded quantum state, potentially affecting classification decisions.

3.4.2 ZZ Feature Maps

The ZZ feature map is a quantum feature encoding method that introduces nonlinearity and entanglement into the data embedding process, making it suitable for quantum models that aim to exploit quantum advantage in classification tasks. Unlike simpler methods such as angle encoding, the ZZ feature map uses both single-qubit rotations and entangling gates to map classical input data into quantum Hilbert space in a way that captures pairwise interactions between features.

Given an input vector $x \in \mathbb{R}^n$ the ZZ feature map operates on an n qubit system, where each input feature x_i is used to parameterize a rotation gate on qubit i. The circuit typically applies a Hadamard gate to each qubit, followed by a rotation around the Z-axis $R_z(x_i)$, and then introduces entanglement through controlled-ZZ interactions between qubit pairs (i, j), parameterized by the product $x_i.x_j$. The unitary transformation for the ZZ feature map is given by:

$$U_{zz}(x) = \left[\prod_{(i,j)\in E} \exp\left(i\gamma x_i x_j Z_i Z_j\right)\right].\left[\bigotimes_{k=1}^{n} R_z(x_k)H_k\right],$$

where:

- $H_k$is the Hadamard gate on qubit k,
- $R_z(x_k)$is the Z-rotation gate parameterized by input x_k,
- $\exp\left(i\gamma x_i x_j Z_i Z_j\right)$ is the ZZ interaction between qubits iii and jjj,
- γ is a tunable hyperparameter that controls the entanglement strength,
- E is the set of qubit pairs over which the ZZ interaction is applied.

The key property of the ZZ feature map is that it captures second-order feature correlations by encoding the product terms $x_i x_j$ directly into quantum phase relationships.

This makes it particularly powerful in cases where feature interactions are relevant to the decision boundary, such as in non-linear classification problems.

The resulting quantum state $|\phi(x)\rangle = U_{zz}(x)|0\rangle^{\otimes n}$ is used in quantum kernel methods, including the Quantum Support Vector Machine implemented in this thesis. The model relies on computing inner products between these states to form the quantum kernel matrix:

$$K(x, x') = |\langle\phi(x)|\phi(x')\rangle|^2.$$

The structure of the ZZ feature map allows for highly expressive embeddings while remaining implementable on current noisy intermediate-scale quantum devices. In this work, the ZZ feature map is used in conjunction with classical optimization techniques to train a QSVM for classification tasks. The entanglement introduced by the feature map is expected to improve class separation in the quantum feature space.

In adversarial settings, the sensitivity of the ZZ map to feature correlations makes it an important point of analysis. Perturbations in the input not only affect the individual rotations but also alter the interaction terms $x_i x_j$ potentially resulting in large shifts in the encoded quantum state. This thesis evaluates how such perturbations affect the decision boundaries of the QSVM trained using the ZZ feature map, and whether the encoding provides any resilience or unique vulnerabilities under adversarial attack.

3.4.3 Amplitude Encoding

Amplitude encoding is a quantum data encoding strategy where classical input data is embedded into the amplitudes of a quantum state. Unlike methods that encode each feature into individual gate parameters or circuit structures, amplitude encoding represents the entire input vector globally within the quantum state. This approach leverages the superposition principle of quantum mechanics to encode N data points using only $\log_2 N$ qubits, providing an exponentially compact representation.

Given a classical input vector $x = (x_0, x_1, \dots, x_{N-1}) \in \mathbb{R}^N$, it is first normalized such $\|x\|^2 = \sum_{i=0}^{N-1} |x_i|^2 = 1$. The amplitude encoding maps this vector to a quantum state $|\psi(x)\rangle$ over $\log_2 N$ qubits:

$$|\psi(x)\rangle = \sum_{i=0}^{N-1} |i\rangle .$$

Each component x_i becomes the amplitude of the computational basis state $|i\rangle$. The encoded state contains the full structure of the classical input and can support inner product evaluations and quantum linear algebra subroutines.

This encoding is particularly relevant for quantum algorithms that rely on linear algebra operations, such as the Harrow-Hassidim-Lloyd (HHL) algorithm for solving linear systems, quantum principal component analysis, and quantum kernel methods. It enables quantum models to operate on data in high-dimensional spaces without requiring a proportional number of physical qubits.

However, state preparation is the main bottleneck of amplitude encoding. Constructing arbitrary quantum states with specific amplitudes typically requires circuits with depth and complexity that scale poorly with input size. Efficient loading of classical data into amplitude-encoded quantum states often assumes access to quantum random access memory (QRAM), which is currently not available on most NISQ devices. For this reason, amplitude encoding is often studied theoretically or in simulations but rarely implemented in real quantum hardware workflows today.

From a machine learning perspective, amplitude encoding offers a dense and information-rich representation, but its sensitivity to perturbations can be significant. Adversarial changes in input values affect global state amplitudes and can result in non-local effects in downstream quantum processing. This raises open questions regarding the robustness of amplitude-encoded quantum models under adversarial attack.

Due to these constraints, this thesis does not utilize amplitude encoding in the experimental implementation. Instead, we focus on the ZZ feature map, which offers more hardware-compatible encoding while still enabling non-linear classification through

entanglement. Nonetheless, amplitude encoding remains a foundational concept in quantum data representation and may become practical with the advancement of quantum memory and efficient state preparation methods.

3.5 Datasets

To evaluate the performance of the CNN and quantum support vector machine (QSVM) model and its robustness under adversarial conditions, this thesis employs two well-established benchmark datasets: MNIST and CIFAR-10. These datasets are widely used in machine learning research due to their standardized structure and relevance for image classification tasks.

The MNIST dataset consists of grayscale images of handwritten digits (0–9), offering a relatively simple classification task that is suitable for initial model validation. CIFAR-10, by contrast, contains coloured images across ten distinct classes, introducing higher visual complexity and a more challenging classification environment.

These datasets enable systematic comparison between classical and quantum models and provide a basis for evaluating adversarial vulnerability and defence mechanisms under consistent experimental conditions. The following subsections describe each dataset in more detail.

Although this research focused on benchmark datasets such as CIFAR-10 and MNIST for controlled experimentation, the proposed models can be extended to large-scale, real-world datasets including ImageNet and medical-imaging repositories. Such datasets provide higher-resolution and domain-specific complexity that would better assess generalization under diverse threat conditions.

Scaling to these datasets requires efficient feature-compression and quantum-encoding schemes to manage high-dimensional inputs under current hardware limits. A hybrid classical–quantum pipeline using classical convolutional feature extraction followed by quantum kernel mapping can enable feasible large-scale evaluation of adversarial robustness.

3.5.1 CIFAR-10 Dataset

The CIFAR-10 (Canadian Institute for Advanced Research) dataset is a benchmark dataset widely used in computer vision and machine learning for evaluating image classification models. It consists of 60,000 color images uniformly divided into 10 classes, with 6,000 images per class. Each image in the dataset has a resolution of 32 × 32 pixels and is represented in RGB format, meaning each image contains three color channels.

The ten categories included in CIFAR-10 are: *airplane, automobile, bird, cat, deer, dog, frog, horse, ship,* and *truck*. The dataset is split into 50,000 training images and 10,000 test images, providing sufficient diversity and volume to train and evaluate models effectively. Because the images span low-resolution representations of various natural and man-made objects, CIFAR-10 is considered more complex than simpler datasets like MNIST. This complexity makes it particularly useful for testing generalization, model robustness, and resistance to adversarial perturbations.

In the context of this thesis, CIFAR-10 is used to evaluate the performance of quantum support vector machines (QSVMs) implemented with the ZZ feature map. However, quantum machine learning models often have constraints in terms of input size and qubit availability, particularly on noisy intermediate-scale quantum (NISQ) hardware. Therefore, preprocessing is necessary to adapt CIFAR-10 to formats that are compatible with quantum circuits.

3.5.1.1 Preprocessing and Dimensionality Reduction

Each CIFAR-10 image originally contains 32 × 32 × 3 = 3,072 features when flattened into a one-dimensional vector. Directly encoding this high-dimensional data into a quantum state is not currently practical due to hardware limitations. As such, dimensionality reduction techniques, such as Principal Component Analysis (PCA), are applied to compress the data into a lower-dimensional space. This allows the transformed features to be encoded into a manageable number of qubits, typically 4 to 8, depending on the available resources.

In this thesis, PCA is used to reduce the feature vector to a size compatible with the quantum circuit used in the QSVM. The number of principal components is selected to retain most of the variance in the original data while allowing efficient quantum processing.

Additionally, for initial experiments and binary classification evaluation, a subset of CIFAR-10 is used, typically involving two distinct classes such as *cat vs. dog* or *automobile vs. truck*. This binary setting simplifies the model structure and makes kernel-based classification more tractable, especially under adversarial training or attack scenarios.

3.5.1.2 Relevance to Adversarial Machine Learning

CIFAR-10 is known to be vulnerable to a range of adversarial attacks, including Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and others. These attacks can modify pixel values slightly to cause misclassification, even though the image remains visually like a human observer. As such, CIFAR-10 serves as a useful benchmark for evaluating the robustness of QSVMs under adversarial conditions.

By incorporating CIFAR-10 into this study, the thesis demonstrates how quantum-enhanced models behave under realistic image-based classification tasks and investigates whether quantum embeddings like the ZZ feature map introduce any meaningful robustness or sensitivity to adversarial perturbations.

3.5.2 MNIST Dataset

The MNIST (Modified National Institute of Standards and Technology) dataset is one of the most used benchmarks in image classification and machine learning research. It consists of 70,000 grayscale images of handwritten digits ranging from 0 to 9, spread across 10 classes. Each image has a size of 28 × 28 pixels, resulting in 784 features when flattened into a one-dimensional vector. The dataset is divided into 60,000 training samples and 10,000 test samples, offering a well-balanced distribution for model development and validation.

MNIST's simplicity, structured labelling, and relatively low-dimensional input space make it an ideal choice for early experimentation, algorithm prototyping, and benchmarking classification models. Although modern deep learning models have achieved near-

perfect performance on MNIST, it remains highly valuable in contexts such as model interpretability, low-resource learning, and adversarial robustness areas that are particularly relevant in the evaluation of quantum machine learning approaches.

In this thesis, MNIST is used as one of the primary datasets to assess the behaviour and effectiveness of the Quantum Support Vector Machine (QSVM), implemented using the ZZ feature map. Its structured yet diverse content provides a reliable test case for evaluating model performance and susceptibility to adversarial perturbations.

3.5.2.1 Preprocessing and Adaptation for Quantum Models

Quantum models, particularly those targeting NISQ (Noisy Intermediate-Scale Quantum) hardware, are limited by the number of qubits and circuit depth. As such, the original 784-dimensional feature vectors of MNIST images cannot be directly encoded into a quantum circuit. To address this, dimensionality reduction is performed as a preprocessing step. In this thesis, Principal Component Analysis (PCA) is used to reduce the dimensionality of the images while retaining most of the data variance.

The reduced features are then scaled and encoded into quantum states using the ZZ feature map. The number of PCA components is chosen based on trade-offs between information retention and quantum resource constraints. This allows the data to be embedded into a quantum circuit of manageable size, enabling meaningful experimentation and evaluation on current quantum simulation platforms or restricted real devices.

In addition, to simplify the classification task and reduce computational requirements, binary classification settings are used in some experiments. For example, distinguishing between the digits *3 vs. 8* or *4 vs. 9* allows focused evaluation of quantum kernel methods in a controlled scenario. This is particularly useful in adversarial robustness tests, where specific class boundaries are examined under perturbation.

3.5.2.2 Relevance to Adversarial Studies

MNIST is widely used in adversarial machine learning literature due to its clean structure and clear visual features. Small perturbations, imperceptible to the human eye, can often lead to significant misclassification, especially in linear or shallow models. In this thesis,

adversarial attacks such as FGSM and PGD are applied to MNIST samples to test the robustness of the QSVM model.

Using MNIST provides a clear baseline to assess whether quantum feature mappings, such as the ZZ feature map, enhance robustness to adversarial inputs or introduce new forms of vulnerability. By comparing results on MNIST and CIFAR-10, the thesis evaluates how quantum models perform across datasets of varying complexity.

3.6 Performance Metrics

Evaluating the performance of classification models, including quantum and classical machine learning systems, requires systematic use of performance metrics. These metrics not only quantify how well the model performs overall but also reveal specific strengths or weaknesses, such as whether the model is better at identifying positive or negative classes, or how it balances precision and recall. In this section, we explore the key performance metrics employed in this thesis to assess model behaviour under both normal and adversarial conditions.

3.6.1 Confusion Matrix

The confusion matrix is a fundamental tool in evaluating classification tasks. It is a square matrix that summarizes the number of correct and incorrect predictions made by the model, categorized by actual and predicted labels. For binary classification, the confusion matrix has four key components:

- True Positive (TP): Instances where the model correctly predicted the positive class.
- True Negative (TN): Instances where the model correctly predicted the negative class.
- False Positive (FP): Negative instances that the model incorrectly classified as positive.
- False Negative (FN): Positive instances that the model incorrectly classified as negative.

This matrix forms the basis for several derived metrics, which are discussed below.

Accuracy

Accuracy measures the overall proportion of correctly classified instances among all predictions. It is a commonly used metric but may be misleading in imbalanced datasets, where one class dominates the other.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}.$$

Accuracy is helpful in balanced scenarios but should not be the sole metric in evaluating model effectiveness, especially under adversarial perturbations where one class may be disproportionately misclassified.

Precision

Precision quantifies the proportion of true positive predictions among all instances predicted as positive. It measures how much the model can be trusted when it labels an instance as positive.

$$Precision = \frac{TP}{TP + FP}.$$

High precision indicates low false positive rates, which is crucial in contexts where false alarms are costly, such as fraud detection or intrusion detection.

Recall (Sensitivity)

Recall, also known as sensitivity, measures the proportion of actual positives that were correctly identified by the model.

$$Recall = \frac{TP}{TP + FN}.$$

Recall is important in scenarios where missing a positive instance is more harmful than mistakenly labelling a negative instance, such as in medical diagnosis or threat detection.

F1-Score

The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances both concerns, particularly when there is a trade-off between minimizing false positives and false negatives.

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} .$$

The F1-score is particularly useful when dealing with imbalanced datasets or when the cost of false negatives and false positives are comparable.

Chapter 4

Adversarial Training: Reinforced Weighted Adversarial Learning for Convolutional Neural Networks (CNN)

This section addresses challenge 1 and has been published in 2024 11th International Conference on Machine Intelligence Theory and Applications (MiTA), Machine Intelligence Theory and Applications (MiTA).

4.0 Introduction

Given training set of n pairs $(x_i, y_i)_{i=1}^{N} \in \mathcal{X} \times \mathcal{Y}$ drawn independently and identically (iid) from a distribution $\mathcal{D}$. Here x_i represents the CIFAR-10 data examples and y_i denotes the corresponding labels. Our primary goal is to develop a robust MobileNet classifier model parameterized by θ that effectively maps the input space to the output space, denoted as $f_\theta: \mathcal{X} \to \mathcal{Y}$ while minimizing a loss function on adversarial data x'. In this context, we introduce the L_∞ norm metric $d(x, x')$ on $\mathcal{X}$ and a boundary ball $B_\epsilon(x) = \{x': d(x, x_i) \leq \epsilon\}$ around x, an adversary's goal is to perturb the data examples x_i to x_i' within a defined budget $\epsilon > 0$ with the aim of maximizing the adversarial during the training process.

4.1 Methodology

4.1.1 Stackelberg Game formulation

Consider a sequential 2-player non-zero sum Stackelberg game $\mathcal{G}= (S_L, S_F, u)$ where S_l and S_f are strategy spaces for the classifier leader and adversary follower of game and $u: S_L \times S_F \to R$ is the payoff function. The leader has a set of strategies $s_l \in S_L$ and the followers set of strategies is given by $s_f \in S_F$. For a Stackelberg equilibrium there exist a

rational best response mapping function $f: S_L \rightarrow S_F$ such that $u_2\left(s_{l,} f(s_l)\right) \geq u_2(s_{l,} s_f)\ \forall s_l\ \in S_L\ ,\ s_f \in S_F$.

The leader makes the first move by selecting a strategy. $s_l \in S_L$ to minimize the u_1, knowing the existence of a follower. After knowing s_l, the follower picks $s_{f2} \in S_F$ to maximize their own payoff u_2 where $s_{f2} = f(s_l)$. Hence, the Stackelberg equilibrium strategies (s_l^*, s_f^*) pair for leader and follower is $s_l^* \in argmin_{s_l \in S_L} u_1(\ s_l, s_{f2})$ and $s_f^* \in argmax_{s_l \in S_L} u_2(\ s_l^*, s_f)$ respectively such that $u_2(s_l^*, f(s_l^*)) \leq u_2(s_l, s_{f2})$. This gives the leader an advantage that imposes a solution favorable for himself while optimizing against the follower's anticipated strategy s_{f2}.

Proposition 3.1 A Stackelberg equilibrium strategy exists with the defender as the leader and adversary the follower if S_L and S_F are compact sets and U_L and U_F are continuous on $S_L \times S_F$.

Proof. Since the rational adversarial response strategy $\left(s_{l,} f(s_l)\right)$ is a subset of the compact set $S_L \times S_F$ we only need to show that set of adversarial responses is closed. If (s_l^0, s_f^0) is the closure of Ω_f and (s_l^n, s_f^n) are sequence of points converging to (s_l^0, s_f^0) in Ω_f. We show that Ω_f is closed and (s_l^0, s_f^0), a point on the boundary, is contained in Ω_f. If $(s_l^0, s_f^0) \notin\ \Omega_f$ then $\exists (s_l^0, s_f^*) \in \Omega_f$ such that $U_f(s_l^0, s_f^*) > U_f(s_l^0, s_f^0)$. Let $U_f(s_l^0, s_f^*) - U_f(s_l^0, s_f^0) = \beta$. since U_F is continuous on $S_L \times S_F$ and $(s_l^{n0}, s_f^{n*}) \rightarrow (s_l^0, s_f^*)$ then $\exists\ \delta_1 > 0$ such that $\left|U_f(s_l^{n0}, s_f^{n*}) - U_f(s_l^0, s_f^*)\right| < \frac{\beta}{3}$. Similarly, as $(s_l^{n0}, s_f^{n*}) \rightarrow (s_l^0, s_f^*)\ \exists\ \delta_2 > 0$ such that $\left|U_f(s_l^{n0}, s_f^{n0}) - U_f(s_l^0, s_f^0)\right| < \frac{\beta}{3}$ and $\left|U_f(s_l^{n0}, s_f^{n0}) - U_f(s_l^0, s_f^0)\right| < \frac{\beta}{3}$, $\forall (s_l, s_f) \in S_L \times S_F$. Therefore, we have

$$\left|U_f(s_l^{n0}, s_f^{n0}) - U_f(s_l^0, s_f^0)\right| < \frac{\beta}{3} = U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^0, s_f^0) + \frac{\beta}{3}$$

$$U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^0, s_f^*) - \beta + \frac{\beta}{3} = U_f(s_l^{n0}, s_f^{n0}) < U_f(s_l^0, s_f^*) - \frac{2\beta}{3}$$

$$= U_f\left(s_l^{n0}, s_f^{n0}\right) < U_f\left(s_l^{n0}, s_f^*\right) - \frac{\beta}{3}$$

$$= U_f\left(s_l^{n0}, s_f^{n0}\right) < U_f\left(s_l^{n0}, s_f^*\right).$$

This contradicts the fact that $U_f\left(s_l^{n0}, s_f^{n0}\right)$ is a sequence in U_F, therefore $\left(s_l^{n0}, s_f^{n0}\right) \in U_F$ and U_F is closed.

4.1.2 Adversarial Training as a Stackelberg game

Traditional methods of adversarial training aim to solve a minimax problem between a classifier and attacker by minimizing the loss on the input perturbation. The solution converges to an equilibrium such that for a given dataset $S = \{(x_i, y_i)\}_{i=1}^n$, the model f_θ minimizes the expectation of adversarial loss function as shown

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} \left\{ \max_{x_i' \in B_\epsilon[x_i]} l(f_\theta(x_i'), y_i) \right\}$$

. The model adjusts its parameters θ to the adversarial perturbations by treating all generated adversarial samples x' equally when estimating the adversarial loss at test time. The classifier strategy $s_l \in S_L$ is a parameter θ that gives minimum training loss on a training set $(x_i, y_i)_{i=1}^N$. The strategy minimizes the payoff empirical risk on the dataset, as shown below:

$$s_l = \min_\theta \mathcal{L}(\theta) = \min_\theta \frac{1}{n} \sum_{i=1}^{n} \left(l(f_\theta(x_i), y_i) \right)$$

. The payoff function $u_F: S_L \times S_F \rightarrow R$ of the follower is the adversarial loss derived during attack at test time. After observing the classifier f_θ the adversary chooses a strategy $s_f \in S_f$ that maximally perturbs the original data. To achieve the attack, the optimal strategy $s_f = \{x': x + \delta\}$ is the best response to θ and maximizes the loss $\mathcal{L}$ in equation below. The maximum perturbation δ is derived using projected gradient descent (PGD) algorithm in the l_∞ norm ball. The payoff function $\mathcal{L}$ of the adversary selecting s_f is given as

$$u_F = \mathcal{L}'(\theta) = \sum_{i=1}^{n} \max_{x_i' \in B_\epsilon(x,\delta)} \big(l(f_\theta(x_i'), y_i)\big)$$

$$s.t\ B_\epsilon(x,\delta) = \{\delta : d\big(x, x'\big) \leq \epsilon\}\ .$$

The adversary selects a best response s_f that guarantees a high payoff. The solution to [eq] obtains a perturbation δ which also maximizes $\mathcal{L}'(\theta)$.

In other words, the adversary searches for a strategy s_f obtained using (PGD) that maximizes the adversary's payoff while observing the classifier's strategy s_l.

On the other hand, the best response for the leader is calculated by considering the adversary's strategy $s_f = \{x' : x + \delta\}$ as a function of the classifier's payoff $\mathcal{L}(\theta)$. The leaders Stackelberg strategy s_l^* is consequently denoted as

$$s_l^* = \min_\theta \mathcal{L}'(x_i') = \min_\theta \max_{x_i' \in B_\epsilon(x,\delta)} \big(l(f_\theta(x_i'), y_i)\big).$$

4.1.3 Defining the Weighting Parameter c_i

Learning the model parameters requires estimating the loss imposed by potential adversaries. The losses which differ from natural data are derived from adversarial samples generated by adversarial perturbations added to the original samples. The derivative of the summation of individual losses from x_i' in a training batch updates the parameter of the model. To maximize the loss in the inner loop, strong x_i', that is adversarial samples that guarantee high losses, are more represented, weaker x_i' are less represented and x_i' that do not misclassify y_i at all are least represented in the adversarial distribution. The loss in fact guides the model into ultimately learning the parameter of the model to accurately predict the on the adversarial samples. Afterall, the essence of an adversarial attack is to generate the maximum possible loss, and adversarial samples do not contribute equally to the overall loss of the distribution $\mathcal{D}'$.

$$\mathcal{L}' = \mathbb{E}_{(X,Y)\sim D, X\prime \in B_\epsilon[X,\epsilon]}\big(l(f(X'), Y)\big).$$

A priority attacker selects a strategy $s_f \in S_f$ that not only perturbs the data but also ensures a maximum adversarial payoff loss $\mathcal{L}'$. Not all adversarial samples result in incorrect predictions with PGD attack; therefore, a priority attacker modifies the data distribution $\mathcal{D}$ such that the effective adversarial samples that confidently mislead the model f_θ into generating outputs different from y are more represented.

For the model to be aware of the underlying distribution of strong adversary samples and generalize effectively over benign adversarial data, we introduce the weighting mechanism that prioritizes adversary data x_i' during training. Stronger adversarial examples, those that result in misclassifications i.e., $f(x_i) = z$ such that $z \neq y_i$ label y_i with a higher margin are assigned greater weight, while the weaker adversarial examples are given lower weight. The strength of an adversarial sample is determined by its classification margin which is the difference between the probabilities of the wrongly predicted label and the correct label. A larger difference indicates a stronger adversarial sample and vice versa. We define the weight $c_i > 0$ as a function of the classification margin m of the adversarial sample hence we have:

$$m(x, y, f) = \max_{z \neq y} P(f(x) = z) - P(f(x) = y).$$

4.1.4 Weighted Adversarial Reinforced Training

Adversarial training involves the exploration of hyperparameters to achieve an optimized model. Once a hyperparameter configuration is established, it remains unchanged until the completion of the entire training epoch, resulting in the acquisition of a robust model. We propose an alternative approach, wherein instead of adhering to a single hyperparameter throughout all epochs, we dynamically adjust the hyperparameter during training. This adaptation process aims to yield a better-optimized model for the defender by end of the training. In pursuit of hyperparameter optimization, the defender employs the SARSA (State-Action-Reward-State-Action) algorithm. Specifically, the objective is to learn the hyperparameter denoted as φ with the intention of enhancing the accuracy of the selected strategy within a single training epoch. Indeed, the retraining is at the cost of additional

overall epochs until an optimal accuracy is reached. A Q-value function $Q(s_l^*, \varphi)$ is estimated using a Stackelberg equilibrium strategy-state s_l^* and an action φ from a previous φ'. The defender takes and action φ and observes the next strategy state $s_l^{*\prime}$ and reward r. The reward r ensures that the accuracy of the current state is higher than the previous one, the Q-value estimate uses the following update rule:

$$Q(s_l^*, \varphi) = Q(s_l^*, \varphi) + \propto \left(r + \gamma Q\left(s_l^{*\prime}, \varphi'\right) - Q(s_l^*, \varphi)\right).$$

where $\propto$ and γ is the learning rate and discount factors of the reinforcement learning process.

4.2 Experiment

For reproducibility, all experiments were implemented in PyTorch 2.0 and executed on an GPU. Each adversarial example was generated with ε ∈ {0.01, 0.03, 0.07} and PGD iteration k ∈ {1, 3, 5, 7}. Training employed the Adam optimizer (learning rate = 0.001, batch size = 64, epochs = 30). These parameters were kept constant across architectures (MobileNet, ResNet-56, VGG13BN, ShuffleNetv2) to ensure comparability. Code and configurations were maintained in version-controlled repositories for transparency and reproducibility.

We conducted experiments using the Weighted Adversary Stackelberg (WAS) Training model and fine-tuned its performance with a Reinforcement Learning (RL) algorithm on a pretrained MobileNet, resulting in the Weighted Adversarial Reinforced Stackelberg (WARS) model. In our experiment, we employed an adversarial attacker to perturb the CIFAR-10 dataset using the PGD attack. We varied the attack's strength by adjusting the parameter k. The perturbed dataset was used to assess the accuracy and robustness of the WAS MobileNet.

We evaluated the adversarial robustness of our WARS model on the CIFAR-10 dataset, benchmarking it against traditional adversarial training methods under PGD attacks. We applied the WARS algorithm to enhance 3 additional pre-trained models: ResNet-56, shufflenetv2, and vgg13_bn , using different values of k, such as 7 and 20 to evaluate the

effectiveness of our algorithm. The results demonstrated that our method consistently achieved higher test accuracy compared to traditional adversarial training methods. We used the concept of Natural accuracy A_n representing the accuracy of the pre-trained model f_θ on the natural CIFAR-10 dataset. After subjecting the model to PGD attacks with varying k, denoted as k-steps, the corresponding accuracy A'_n of the pre-trained model on the perturbed dataset x' consistently fell below A_n, for all the values of k. After training, the resulting WAS model becomes more robust than the initial pre-trained f_θ showing accuracy A_R consistently greater than A'_n but still less than A_n. The WARS model fine-tunes the hyper-parameter φ of the WAS to achieve an accuracy A_R^* equals to or greater than A_R, such that $A'_n < A_R \leq A_R^*$.

The hyper-parameter φ was initially set to 0.7 in the WAS model but improved by the WARS training process for enhanced robustness. As shown in Fig4.1, Fig.2, Fig4.3 and Fig4.4 we observe that in addition to the improved test accuracy, the training loss reduced significantly in a single training epoch, a contrast to traditional adversarial training, which does not exhibit the same behaviour. It's worth noting that the WARS training resulted in a wider range of loss values compared to AT training, and we attribute this to the distribution-aware weight assigned to potential adversarial data points during training, increasing the overall training loss of the model.

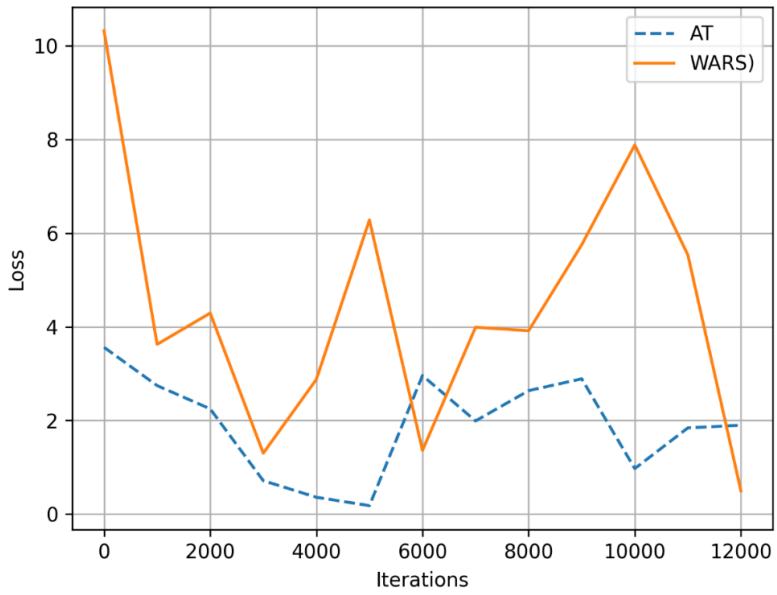

Figure 4. 1 Epoch training loss for Adversarial Trained and WARS trained mobilenetv2

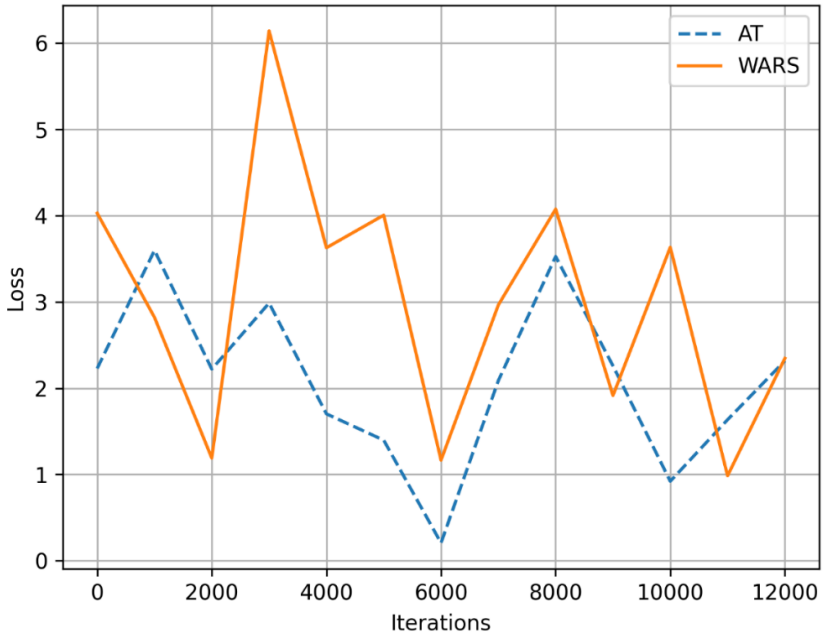

Figure 4. 2 Epoch training loss for Adversarial Trained and WARS trained shufflenetv2

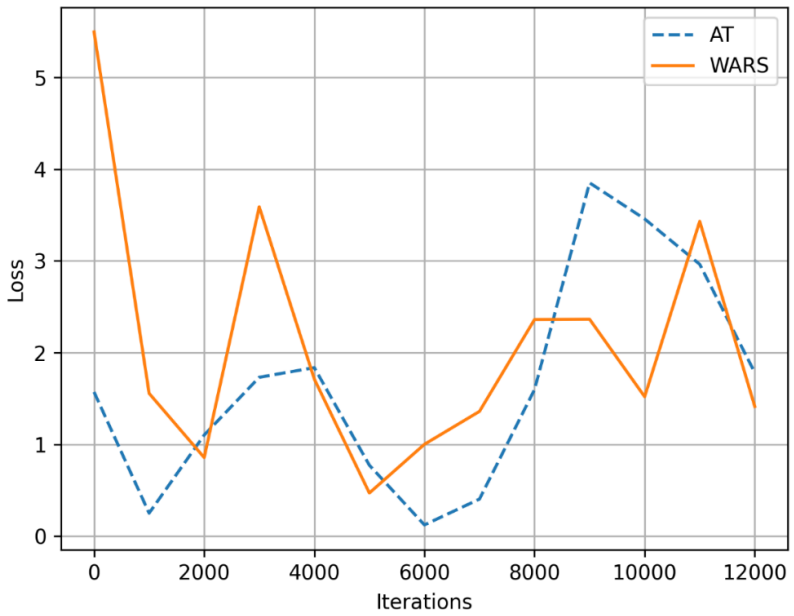

Figure 4. 3 Epoch training loss for Adversarial Trained and WARS trained RestNet56

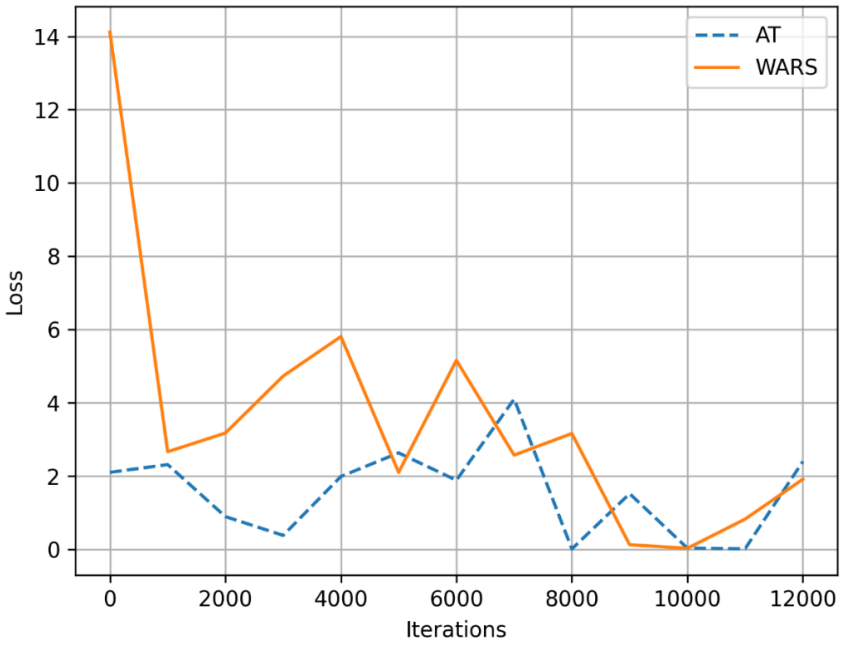

Figure 4. 4 Epoch training loss for Adversarial Trained and WARS trained vggh

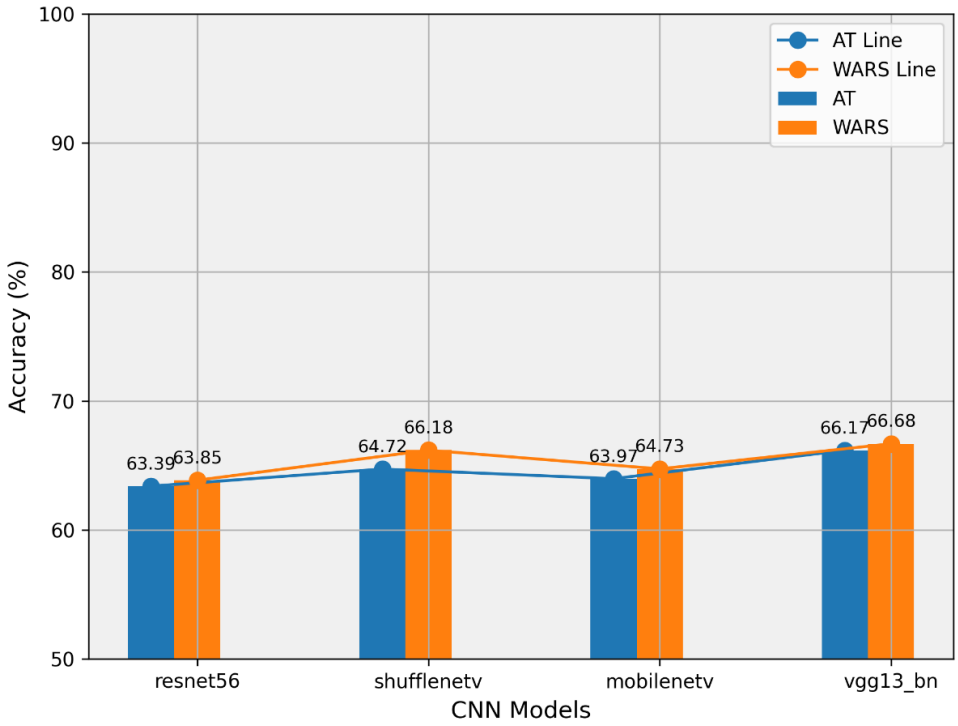

Figure 4. 5 Accuracy for the different Adversarial Trained and WARS trained CNN in a single Epoch

We illustrate how an attacker, observing the pre-trained model f_θ, employs PGD to perturb and launch an attack against the target model. The extent of perturbation depends on the selected value of k, subsequently reducing the accuracy of the pre-trained models. In our Stackelberg game illustration, the defender selects an equilibrium strategy by observing the attack and choosing a WAS model parameter (through retraining on the perturbed dataset) to minimize losses on the perturbed dataset.

.

Table 4. 1 WARS training for various PGD steps for a ResNet-56 Model on CIFAR-10 dataset

Models	$A_n\%$	k	$A'_n\%$	$A_R\%$	φ	$A^*_R\%$
vgg13_bn	94.24	20	14.17	78.22	0.8	78.22
3-7		7	17.78	78.22	0.7	78.22
mobilenetv2_x1_4	93.88	20	7.21	74.91	0.8	79.1
3-7		7	10.66	78.11	0.9	80.01
shufflenetv2_x2_0	93.63	20	12.24	78.14	0.8	79.83
3-7		7	16.89	76.26	0.8	79.11
ResNet-56	94.46	20	6.81	79.83	0.8	81.23
3-7		7	10.23	79.33	1	80.45

Weighted Adversarial Stackelberg Training leads to improved accuracy compared to the original pre-trained model. Further enhanced learning accuracy is achieved after retraining with a hyper-parameter φ . For a moderate preset hyper-parameter φ = 7 an overall increase in accuracy is observed across all models. The WARS model further improves the training hyper-parameter during retraining.

As seen in Table 4.1, a PGD attack with k=20 results in stronger attack dataset, significantly reducing the accuracy of all models. Attack steps with k=7 used by the attacker also lead to decreased accuracy in the models. Larger values consistently decrease the overall accuracy of all models. For k values consistently reduce the, the impact of the attack is more pronounced in ResNet-56, with accuracy dropping to 10.23% from the initial natural accuracy of 94.46%. The higher the K, the greater the image distortion, and even when the distortion is imperceptible, the attack still significantly reduces the model's accuracy. After retraining, using distribution-aware Stackelberg training, the accuracy

improves to 60.67% , and the WAS model fine-tunes it further to an accuracy of 65.67% with a WARS φ of 1.0. The training involved 8 epochs for the WARS training, with additional epochs based on when the model reaches optimal accuracy. For the ResNet-56 model, default epoch training and adversarial accuracy reached 78.11%, and the WARS trained model optimized the accuracy to 80.01% with a φ of 0.8.

From Table II, epoch accuracy for WAS training gradually improves after each epoch from an initially low A'_n of the original model. The original pre-trained model exhibits reduced accuracy after the attack, with MobileNet showing an A_R of 10.66%, dropping to 78.11% at the final epoch after achieving 78.68% accuracy. However, the WARS model fine-tunes the model back to an optimized accuracy of 80.01%. The ResNet-56 model's accuracy is optimized to 80.45% after reaching an φ 1.0, up from a previous WAS accuracy of 79.33%, while the vgg13_bn accuracy for both WAS and WARS training remained 78.22% at the default φ of 0.7.

Table 4. 2 Epoch accuracy of the WAS training for k=7 on various CNN models using CIFAR-10 dataset.

Models	E=2	E=3	E=4	E=5	E=6	$A_R^*\%$
vgg13_bn	79.46	78.29	78.68	77.32	78.42	78.22
mobilenetv2	78.56	77.73	77.96	77.57	78.68	79.12
Shufflenetv2	77.06	76.68	78.39	78.89	77.99	79.83
ResNet-56	79.52	80.18	79.99	79.92	80.19	81.23

Table 4. 3 Epoch accuracy of the WAS training for k=20 on various CNN models using CIFAR-10 dataset.

Models	E=1	E=2	E=3	E=4	E=5	$A_R^*\%$
vgg13_bn	77.33	77.53	77.75	77.79	78.22	78.22

Models	E=1	E=2	E=3	E=4	E=5	$A_R^*\%$
mobilenetv2	76.73	77.23	77.98	79.35	75.92	80.01
Shufflenetv2	76.19	76.12	77.94	78.75	77.93	79.11
ResNet-56	78.86	76.58	80.2	78.52	77.73	80.45

4.3 Discussion

In this research, we have developed a novel adversarial training approach for MobileNet CNNs, conceptualizing it as a dynamic interaction within a WAS game framework. By strategically emphasizing adversarial data points during training, our methodology has substantially improved the model's accuracy. This is achieved by prioritizing adversarial inputs that are more likely to cause misclassifications, thereby training the MobileNet model to develop a bias that enhances its resilience during adversarial attacks.

When comparing our WAS model to traditional AT methods, we observe a notable superiority in terms of robustness under adversarial conditions. Although the WAS model initially shows a broader range of training losses compared to AT models, it demonstrates a more rapid decrease in training loss within a single epoch, particularly when applied to dataset like CIFAR-10, tailored for MobileNet's architecture.

Moreover, our research introduces the WARS training methodology. This refined approach further strengthens the MobileNet model's resilience against adversarial attacks. Our empirical findings, as detailed in the accompanying tables, show consistent enhancements in the performance of MobileNet across various levels ofφ increments in the training process. This iterative and strategic reinforcement leads to a discernible improvement in accuracy with each successive training epoch, underscoring the efficacy of the WARS approach in crafting a more robust MobileNet CNN.

The experimental findings demonstrate that the Weighted Adversarial Stackelberg (WAS) and Weighted Adversarial Reinforced Stackelberg (WARS) frameworks significantly enhance the robustness of convolutional neural networks against adversarial perturbations. The progressive improvement from WAS to WARS highlights the

effectiveness of integrating reinforcement learning (RL) into the Stackelberg game formulation, where the defender adaptively fine-tunes the hyperparameter $\phi$to optimize both model accuracy and stability.

The observed relationship between the attack strength parameter $k$and model performance confirms the theoretical expectation that higher attack intensities induce greater perturbations in the input space, thereby amplifying misclassification rates. As $k$increases from 7 to 20, the perturbations generated by PGD become increasingly aggressive, leading to substantial declines in natural accuracy across all pre-trained models. This degradation is particularly evident in deeper architectures such as ResNet-56, where accuracy dropped from 94.46% to 10.23%, suggesting that complex gradient landscapes in deep networks are more exploitable by iterative adversarial attacks. Nevertheless, retraining with distribution-aware Stackelberg optimization improved performance substantially, elevating accuracy to 60.67%, and further to 65.67% under WARS fine-tuning with $\phi = 1.0$.

The key insight lies in how reinforcement learning complements adversarial Stackelberg optimization. By dynamically updating ϕ, the defender (learner) progressively identifies optimal policy adjustments in response to the adversary's strategy, effectively learning equilibrium behavior through experience. Unlike static hyperparameter tuning in conventional adversarial training, WARS introduces an adaptive feedback mechanism that adjusts weights in real time, leading to faster convergence and improved model resilience. This adaptive process is evidenced by the sharp reduction in training loss within a single epoch, as shown in Figures 4.1–4.4, contrasting with the slower convergence patterns typical of standard adversarial training.

The broader training loss range observed in WARS, compared to adversarial training (AT), is indicative of distribution-aware weighting across adversarial data samples. Rather than uniformly treating all perturbations, the WARS algorithm assigns higher weights to regions in the data manifold more likely to be exploited by the adversary. This results in a controlled increase in training loss but yields better generalization to unseen

adversarial distributions, validating the trade-off between training variance and robustness.

Performance metrics across models further reinforce this conclusion. The consistency of post-training improvements across MobileNet, ShuffleNetV2, VGG13BN, and ResNet-56 suggests that the WARS optimization process generalizes well across architectures with varying depth and parameter density. For instance, ResNet-56's adversarial accuracy increased from 78.11% to 80.01%, and VGG13BN maintained stable robustness at 78.22%, confirming that WARS enhances or at least preserves robustness even for architectures sensitive to adversarial gradients. These improvements, though incremental, signify stability under high-dimensional perturbation stress, an essential property for deploying CNNs in adversarially exposed environments.

The results also highlight a meaningful theoretical implication: Stackelberg equilibrium is empirically attainable in adversarial learning contexts when reinforcement feedback is embedded into the optimization process. The defender's reinforcement-guided parameter tuning mirrors equilibrium adaptation, where the leader (classifier) iteratively adjusts strategies based on the observed payoff dynamics of the adversary. This dynamic optimization aligns with the principles of hierarchical game theory, demonstrating that adversarial learning can transition from purely reactive defense to proactive, equilibrium-seeking behavior.

Moreover, the superior convergence speed observed in WARS training compared to standard adversarial methods implies reduced computational cost per effective epoch. This finding supports the hypothesis that reinforcement-guided Stackelberg optimization not only enhances robustness but also improves training efficiency, which is crucial for scaling adversarial defense to large models or resource-constrained environments.

In summary, the WARS model achieves a robust–accuracy balance unattainable by traditional adversarial training or fixed Stackelberg formulations. The integration of reinforcement learning provides adaptability, allowing the system to infer optimal defense weights and policies dynamically as adversarial pressure evolves. These findings validate the theoretical framework proposed in this research: that weighted and reinforced Stackelberg

learning yields superior adversarial resilience, improved convergence behavior, and generalizable robustness across diverse CNN architectures under varying attack intensities.

4.4 Summary

In this paper, we have designed a novel adversarial training methodology, conceptualized as a Weighted Adversarial Stackelberg game, specifically tailored for training a robust MobileNet CNN. Our research demonstrates the effectiveness of the Stackelberg equilibrium model in enhancing MobileNet's resilience against adversarial attacks. We further augment this model's robustness by incorporating a SARSA algorithm, which acts as a defensive mechanism, fine-tuning the MobileNet architecture to counteract such attacks more effectively, we also showed the effectiveness of our methods on other CNN models.

Our approach in the Stackelberg game formulation centres on assigning asymmetric weights that focus more on adversarial data points during testing. This strategy significantly reduces misclassification errors in MobileNet. We derive a pure strategy model with optimized learning parameters by solving the Stackelberg game. This outcome empowers the MobileNet model to generalize more effectively and exhibit increased robustness to targeted and perturbation attacks.

Chapter 5

Adversarial Learning with Multiple adversaries Using Bayesian Stackelberg Game

This section addresses challenge 2 and has been published in EAI Endorsed Transactions on Scalable Information Systems, 2025.

5.0 Introduction

This section focuses on the application of the Decomposed Optimal Bayesian Stackelberg Solver (DOBSS) as a framework for enabling a machine learning-based defender to compute optimal mixed strategies in the presence of multiple adversaries. The goal is to determine how the defender can maximize classification performance when faced with strategic attacks. In adversarial settings, attackers often manipulate inputs or influence training data to reduce model reliability. To address this, the defender must not only detect and respond to attacks but also plan strategies that remain effective under uncertainty. By incorporating a probabilistic game-theoretic model, the research seeks to formalize the interaction between a single learner and several potential adversaries, each with multiple possible strategies. The DOBSS framework allows the defender to consider uncertainties in adversary behavior and select a distribution over its strategies that provides the best expected outcome.

A central focus is improving the robustness of the defender in machine learning environments where multiple adversaries operate with varying and possibly unknown objectives.

These adversaries may exploit different vulnerabilities, such as input perturbation, poisoning, or model inversion. The defender, therefore, must be able to generalize across threat scenarios and adopt a policy that performs well against a distribution of potential attacks. This section investigates how a mixed strategy approach, where the defender selects among multiple classification models or parameter settings probabilistically, can reduce error rates and mitigate risk under diverse and uncertain threat landscapes.

The section answers several guiding questions. The first explores whether learning and deploying a mixed strategy can improve the defender's ability to maintain classification accuracy and robustness across a range of adversarial attacks. It also asks what specific types of prior knowledge about the adversaries, such as attack frequency, model targeting, or payoff structures, are needed to construct an optimal strategy. A key consideration is the computational feasibility of solving the Bayesian Stackelberg game formulation, particularly when the number of adversaries or their strategy spaces grow. The study evaluates how the scalability of the DOBSS algorithm is affected by these factors and whether approximations or reductions in strategy space are necessary to make the solver practical in real-world settings. By examining these questions, the research contributes to the understanding of how machine learning defenses can be designed using principled strategic reasoning in adversarial environments.

5.1 Stackelberg game formulation

Let Adversarial machine learning problem is defined as an input space $X \in \mathbb{R}^d$ where d is the number of attributes in the vector space. For a learning model classifier † with an input $x \epsilon X$ and a corresponding output given as $y \epsilon \{+1, -1\}$, there is an adversary able to corrupt the model at test time by an amount δ such that a malicious instance x will be misclassified as benign given by $f(x) \neq f(x+\delta)$. Thus adversarial machine learning focuses to obtain a robust algorithm such that the probability of the algorithm misclassifying even under attack is as small as possible $P\big(f(x) \neq f(x+\delta)\big) < \varepsilon \; for \; \varepsilon > 0$. If we have input samples$x_{i|i=1,\dots n} \epsilon \chi$ and want to estimate target label $y_i \epsilon \Upsilon$ where $\Upsilon = \{+1, -1\}$ to be classified by a learner function $g: \chi \rightarrow \mathbb{R}$ with a feature vector $\phi(x \epsilon X) \in \mathbb{R}^d$ The predicted

value $\hat{y} = g(w, x_i)_{|w\epsilon\mathbb{R}^N}$ is obtained by optimizing a loss function L. The learner's loss function with regularization is given as $\sum_{i=1}^{n} \ell(\hat{y_i}, y_i) + \lambda||\omega||^2$ where λ is a regularization parameter that penalizes weights ω of the classifier. A cost vector c is included in the loss function to reflect the weights of individual input data, and the learner now optimizes the equation:

$$\underset{\omega}{\operatorname{argmin}} L = \underset{\omega}{\operatorname{argmin}} \sum_{i=1}^{n} c_i . \ell(\hat{y_i}, y_i) + \lambda||\omega||^2 \ .$$

The loss function can be extended to an adversarial learning problem. If an adversary wishes to influence the learner by modifying the input data, then the learner's classification task to obtain $\hat{y}$ on the transformed data becomes $\hat{y} = \omega^T . \phi\big(f_t(x_i, \omega)\big)$ where f_t is the function used by the adversary to transform the data:

$$f_t(x_i, \omega) = x_i + \delta_x(x_i, \omega).$$

δ_x is the displacement vector that determines the level of perturbation of original input x_i, hence the adversarial learning can be defined as $\underset{\omega}{\operatorname{argmin}} \underset{\delta_x}{\operatorname{argmax}} L(\omega, x, \delta_x)$.

5.2 Adversarial Attacks

The aim of training a classifier algorithm on a dataset is to correctly label all input images to target label set. A classifier model can correctly classify a sample x to its corresponding label y expressed as $\arg\max P(y_i|\, x) = y_{true}$. Given that $y \epsilon Y = \{y_1, y_2, \ldots, y_k\}$ is an output label class with k unique classes. $P(y_i|x)$ shows the confidence value of model in predicting a sample x to y_i . Hence the adversarial attack aims to generate adversarial sample such as small perturbation δ added to x will lead the classifier model to predict another label other than the correct label y_{true} $\arg\max P(y_i|\, x') \neq y_{true}$, $x = x + \delta$.

5.2.1 Fast Gradient Sign Method

The method generates adversarial samples by adding perturbations in the direction of the loss function that is the positive direction of the slope gradient, a normal input image x, FGSM calculates a similar adversarial example x' to fool the classifier. x' is derived by

optimizing the loss function, defined as the cost of classifying x'as a label lx with minimum possible perturbation

$$x' = x + \varepsilon.sign(\nabla_x Loss(x, l_x)).$$

5.2.2 IFGSM

This is an extension of FGSM but computes perturbations in iterations rather than in a single shot, achieving samples of better image quality than FGSM. The FGSM algorithm is simply applied multiple times with miniature perturbations rather than a single large one. After the completion of each iteration the pixels are cropped such that the perturbation remains as close as possible to the input image x.

$$x_i = clip_{x,\epsilon}(x_{(i-1)} + \epsilon.sign\left(\nabla x_{(i-1)} Loss\left(x_{(i-1)}, y\right)\right)).$$

Where $Loss(x, l_x)$ shows the cost function given x as an input image, l_x as the corresponding true output label and ε the parameter that determines the magnitude of perturbation for x.

5.2.3 Analyzing Existing Works using Regularized FGSM (FGSMR) for Adversarial Training

Tianjin H. et al (2020) increased the similarity between vanilla FGSM and Projected Gradient Descent (PGD) attack by reducing the curvature along the perturbed direction projected by FGSM [91] [98] [99] [88]. This was achieved by regularizing the curvature of the FGSM and restraining the projection to make the perturbed direction close to those generated by PGD-inf attacks. Restraining the gradient direction along the FGSM, which is the second direction derivative, gives a perturbed direction that can be expressed as

$$\nabla^2_{xg} L_\theta(x) = \lim_{h \to 0} \frac{\nabla_x L_\theta(x + hg) - \nabla_x L_\theta(x)}{h},$$

also given a curvature regularization term R_θ then the adversarial training optimization objective is to minimize the expression:$\min_\theta L(x + \epsilon g) + \lambda R_\theta$. The hyperparameter λ is penalizing factor for controlling the curvature along the FGSM direction. Robust models trained by adv.FGSMR had higher perturbed data accuracy than adv.PGD for PGD-infinity and FGSM attacks, also adv.FGSMR models achieved state of the art accuracy on clean MNIST datasets. For further comparison, the times spent on training 50 epochs with adv.FGSMR for ResNet-18/34 models was considerable lower that adv.PGD since the later takes k (usually k is set to 20) iterations of forward and backward process to find an optimum perturb vector in the l_∞ ball while adv.FGSM takes only 1 iteration for the forward and backward process to find a perturbed vector and 2 times forward and backward process for the curvatures regularization.

Table 5. 1 Qualitative Analysis of FGSMR and PGD Attack Methods

	Attack Method Accuracy		
Trained Models	Clean	FGSM	PGD-inf
Vanilla train	0.98	0.361	0.27
adv.PGD(e:0.1)	0.993	0.897	0.974
adv.PGD(e:0.2)	0.992	0.966	0.982
adv:FGSMR(e:0.1)	0.994	0.961	0.979
adv:FGSMR(e:0.2)	0.992	0.968	0.982

Robust models trained on *adv*.FGSMR with e=0.1, 0.2 achieved some improvement in accuracy of 0.994 and 0.992 on clean MNIST Dataset while *adv*.PGD (e=0.1) achieved 0.993. The accuracy decreased slightly for both training methods under a PGD-inf attack but the accuracy on the *adv*.FGSMR trained models was still higher with and accuracy of 0.983 when epsilon is set at 0.2. For a less aggressive FGSM attack the accuracy of the *adv*.PGD dropped to 0.897 while the *adv*.FGSMR dropped only slightly to 0.961 for the same epsilon =0.1.

5.2.3 FGSM and PGD Attack Strategies

An attacker's goal is to both increase the misclassification error of the learner and remain undetected. The learner's objective is to derive optimum accuracy at every stage of the game even when faced with an uncertain adversary. In our work, the strategy of the adversary regardless of the type is to transform the data using FGSM or PGD as a method to obtain the perturbation δ for transforming the input vector [100] [101] [102] [103]. A stealthy attacker who is more concerned about being undetected than increasing the error of the defender classifier will more likely implement a FGSM to solve the inner maximization by only projecting by a small ε in the direction of the gradient of the input image. Conversely an aggressive attacker will implement the PGD to optimize the δ in the l_∞ ball, the attacker wants to maximize δ to guarantee a misclassification by the learner.

$$\max_{||\delta||\leq\epsilon} \ell(h_\theta(x+\delta), y).$$

In a Stackelberg game the follower who is the attacker observes the model parameters of the learner and selects transformation strategy based on their type. The leader observes the follower's strategy and chooses a model that minimizes the error based on the transformation

$$L = \min_\theta \sum_{(x,y)\in T} \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta), y)\,.$$

The leader solves the equation to minimize the loss to obtain set of models $\mathcal{G} = \{g_S, g_{f_1} \ldots.\}$ strategies.

Single Leader Single Follower Game Illustration

A convolutional neural network (CNN) with the parameters shown on Fig 3.5 was trained using the MNIST dataset to obtain the 3 different models based on the methods of the data transformation by the untargeted attacker (follower), we obtain the payoff table for the Defender as shown in Table 3.2

```
Net(
  (conv1): Conv2d(1, 10, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(10, 20, kernel_size=(5, 5), stride=(1, 1))
  (conv2_drop): Dropout2d(p=0.5, inplace=False)
  (fc1): Linear(in_features=320, out_features=50, bias=True)
  (fc2): Linear(in_features=50, out_features=10, bias=True)
```

Figure 5. 1 Parameters of CNN model in Pytorch

Table 5. 2 Accuracy of a CNN model trained on MNIST data transformed using FGSM and PGD for an untargeted Attack

Defender \ Attacker	No Attack	FGSM	PGD
Vanilla Train	.98	0.09	0.01
*adv.*FGSM(epsi-lon=0.3)	.97	0.89	0.07
adv.PGD(epsilon=0.3)	0.96	0.88	0.86

Table 5.2 illustrates a zero-sum game between the defender and the adversary. The payoff of the defender is the accuracy obtained choosing a model g_l that was trained over a transformation w_f of the adversary strategy W. From the attacker's standpoint w_{f1} is a dominated strategy and wouldn't be played by even by a stealthy attacker during the game and hence can be eliminated from the payoff matrix. The adversary therefore chooses between w_{f2} and w_{f3} , which also varies depending on the type of adversary. As shown in the figure 3.5 a stealthy attacker benefits from choosing w_{f2} since after the attack has been implemented the handwritten digits for epsilon=0.05,0.1 almost do appear like the original, maintaining the cover of the attacker. Strategy w_{f2} on the other hand even at epsilon=0 looks visibly perturbed and gives the attacker away. However, an attacker that uses w_{f2} is more concerned about questioning the accuracy or trustworthiness of the classifier by implementing an unrestrained form of attack. The payoff of the unrestrained adversary for the zero-sum game is given as

$$error_{learner} = 1 - Accuracy_{Learner} ,$$

the payoff of the stealthy adversary is the same as the aggressive adversary but discounted with the distance between the original image and the perturbed image. The discount penalizes images that are too disparate from the original in terms of the level of perturbation and measured as the average distance from the original. Given that x' is the generated adversarial sample by the attacker, and $\mathcal{D}$ is the size of the test dataset the discount factor is defined as follows

$$d = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}}\frac{||x' - x||}{||x||_2}$$

$$Payoff_{adversary} = error_{learner} - \lambda d.$$

Figure 5. 2 FGSM attack on MNIST showing the intensity (less aggressive) on varying epsilon

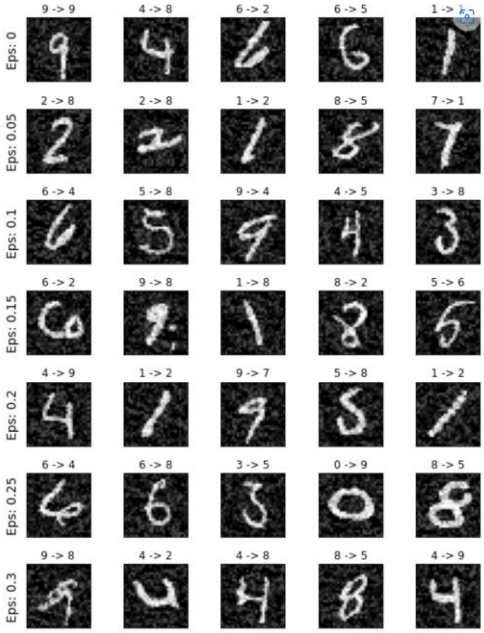

Figure 5. 3 PGD attack on MNIST showing the attack intensity (more aggressive) on varying epsilon

5.3 System Modelling and Analysis

Adversarial attack. Given a classifier f_θ: $\mathcal{X} \to \mathcal{Y}$ and a dataset $(x_i, y_i)_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$, the adversary finds a perturbation d that changes x from its original class to adversarial data,

yet the changes on the adversarial data x' is imperceptible to the human eye, this action is called an adversarial attack. To ensure the attack is undetectable the attacker constrains the perturbation within a defined budget $\epsilon > 0$ in a boundary ball around x such that $B_\epsilon(x) = \{x' : d(x, x_i) \leq \epsilon\}$. While the classifier is pretrained on x by reducing the empirical loss function the adversary aims to increase the classifier's loss on the adversarial data x.

5.3.1 Game Theory Perspective.

In this game the defender is the row player and the attacker the column player, q denotes the defender strategies consisting of a vector of pure strategies in this case a pre-trained model and an adversarial trained model. The value of q_i is the proportion of time where the defender uses the strategy i in their set q. Similarly, p denotes the vector of possible strategies deployed by the attacker. Q and P represent the sets of both the attacker and defender's pure strategies. The payoff matrices D and R are defined such that D_{ij} represents the accuracy of the classifier and R_{ij} is the misclassfication rate of the classifier when the defender chooses a classifier q_i and the attacker deploys an attack j . Given an attacker, the defender maximises their payoff by selecting the optimal classifier to attack p_j as the following:

$$\max \sum_{q \in Q} \sum_{p \in P} D_{ij}\, p_i q_j$$

$$s.t \sum_{q \in Q} q_i = 1 \ .$$

The objective function maximizes the expected payoff given q, while the constraints ensure a mixed strategy j for the defender. The attacker maximizes his payoff function given the the policy q of the defender by selecting a pure strategy p_j in response. The attacker solves the following objective function.

$$\max \sum_{p \in P} \sum_{q \in Q} R_{ij}\, q_i p_j$$

$$s.t \sum_{p \in P} p_j = 1 \ .$$

5.3.2 Stackelberg 2 Player Game

Similar to adversarial training, the defender solves its objective function to minimize the empirical loss for a classifier $q \in Q$ which is either pretrained on natural data x or re-trained on adversarial data x' depending on the strategy $p \in P$ deployed by the attacker. The solution for the set of strategies $q \in Q$ converge to an equilibrum that minimizes the expectation of adversarial loss on the dataset such that for a given dataset minimizes the expectation of adversarial loss function. Q denotes the set of possible strategies by the defender as shown

$$Q \quad = \begin{cases} \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left(l(f_\theta(x_i), y_i) \right) \\ \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \left\{ \max_{x_i' \in B_\epsilon[x_i]} l(f_\theta(x_i'), y_i) \right\} \end{cases} .$$

The classifier $q \in Q$ selected by the defender updates its learning parameters θ to the minimising the adversarial loss across all data points to improve accuracy. The attacker aiming to increase the loss or misclassification rate of the selected classifier, perturbs the natural data $(x_i, y_i)_{i=1}^{N}$. To achieve the attack, the attacker finds an optimal pure strategy $p \in P$, $p_j = \{x' : x + \delta\}$ which is the best response to θ that maximizes the loss. The maximum perturbation δ is derived using projected gradient descent (PGD) algorithm.

The adversary selects a best response pure strategy q_j that guarantees a high payoff after observing the defender's selection.

In a Stackelberg game the defender seeks a mixed strategy of q that maximizes his payoff, given that the attacker selects an optimal response $p(q)$, hence the defender solves the following optimization

$$\max_{q} \sum_{q \in Q} \sum_{p \in P} D_{ij} \, p(q) q_i$$

$$s.t \sum_{q \in Q} q_i = 1$$

$$q_i \in [0 \ldots 1]$$

$$p_j \in \{0,1\}.$$

5.3.3 Payoff for the Defender and Attacker

A Bayesian Stackelberg game models the interaction between a defender and multiple adversaries, where the defender only knows the prior probabilities p of the different types of attackers $t \in T$. We assume that an attacker t_n has two strategies of attack, a selective strategy p_1 that focuses only on the impact of adversarial data x' on the classifier selected by the defender , and the other, a universal strategy p_2 focused on the overall accuracy of the attack on both natural x and adversarial data x'. The prior probability that an attacker of type will appear is p^t , while probability the other attacker appearing is $1 - p^t$.

With PGD attack we can model a range of attack types using k to vary the strength of attack. A small k value yields a small perturbation corresponding to a weak attack, while a large k value yields a large perturbation leading to a stronger attack. The payoff of p_1 is the classification error caused by the perturbed data D' on the classifier $q \in Q$. Hence for a classifier q_i with accuracy A on dataset D' the payoff R of attacker t_n using the selective strategy p_1 is given as

$$R_1 = 1 - A.$$

The universal adversary strategy p_2 also attacks a classifier using varying values of k PGD attack. However, p_2 derives from both the classification error of selected classifier $q \in Q$ by the defender on adversary data D' and natural data D. The intuition for this is that the more the classifier is fitted to the adversarial data D', the less accurately it predicts on the natural data D. For instance, a classifier that is retrained on D' will be less accurate on D since the distribution of both datasets varies due to the perturbations added to D'. Hence, along with the classification error on D', strategy p_2 is also concerned with the

classification error of the pretrained classifier on D. The payoff R of strategy p_2 for a q_i classifier selected by the defender, given that the accuracy of q_i on a dataset D is $A_{q_i}(D)$ is given below

$$R_2 = A_{q_i}(D, D') = 2 - \left(A_{q_i}(D) + A_{q_i}(D')\right),$$

an adversary $t \in T$ changes the value of k in projected gradient descent attack to vary the intensity of an attack. A small value of k value of yields a small perturbation δ, and vice versa. Therefore, a spectrum of adversary types can be specified that ranges from least aggressive to most aggressive. Using the payoff matrices of the classifier and the adversaries, a single defender with T possible multiple follower types can be modeled using decomposed Multiple integral Linear programming (Paruchuri 2008) to obtain an optimal strategy for the leader.

5.3.4 Stackelberg Solution for Multiple Adversaries

When multiple types of attackers are considered in adversarial training, the adversary chooses an optimal pure strategy after observing the defender's strategy, the formulation can be solved by a Bayesian Stackelberg Equilibrium. The defender's strategy Q which is a vector probability distribution of defender's pure strategies q, where the value q_i is the proportion of times where strategy i is used. Q^t denotes the vector of strategies of the attacker $t \in T$ type, and the corresponding payoff for the attacker and defender is giving as D_{ij}^t and R_{ij}^t respectively. M is some large constant and r^t is the upper bound that corresponds to the highest payoff obtainable by the attacker.

$$\max_{q,p,r} \sum_{q \in Q} \sum_{t \in T} \sum_{j \in J} p^t \, D_{ij} p_i j_j^t$$

$$s.t \sum_{q \in Q} p_i = 1$$

$$\sum_{q \in Q} p_j^t = 1$$

$$0 \leq \left(r^t - \sum_{p \in P} A_{ij}^t \, q_1 \right) \leq (1 - p_i^t) M$$

$$q_i \in [0 \ldots 1]$$

$$p_j \in \{0,1\}$$

$$r^t \in \mathbb{R} \, .$$

The prior probability of the occurrence of an attacker type t is denoted by p^t. p_i denotes the probability that the defender selects a mixed strategy i. p_j^t represents the probability that the attacker with type t adopts a pure strategy. Constraints 1 and 4 enforce a feasible mixed strategy for the defender, while constraints 2 and 5 enforce a feasible pure strategy for the attacker. Constraint 3 enforces the feasibility of the attacker's problem to ensure an optimal pure strategy with a maximum payoff of $a = \sum_{q \in Q} R_{ij} \, p_i$ when $p^t = 1$. The quadratic programming problem can be linearized by combining the $p_i q_j^t$ such that $z_{ij}^t = p_i q_j^t$, and obtaining the following equations.

$$\max_{q,p,r} \sum_{q \in Q} \sum_{t \in T} \sum_{j \in J} p^t \, D_{ij} z_{ij}^t$$

$$s.t \sum_{q \in Q} \sum_{p \in P} z_{ij}^t = 1$$

$$\sum_{p \in P} z_{ij}^t \leq 1$$

$$\sum_{q \in Q} p_i = 1$$

$$0 \leq \left(r^t - \sum_{p \in P} A_{ij}^t \left(\sum_{p \in P} z_{ij}^t \right) \right) \leq (1 - p_i^t) M$$

$$\sum_{p \in P} z_{ij}^{t} = \sum_{p \in P} z_{ij}^{1}$$

$$z_{ij}^{t} \in [0 \ldots 1]$$

$$p_j \in \{0,1\}$$

$$r^t \in \mathbb{R}\,.$$

5.4 Experiment

For reproducibility, all experiments were implemented in PyTorch 2.0 and executed on an GPU. Each adversarial example was generated with ε ∈ {0.01, 0.03, 0.07} and PGD iteration k ∈ {1, 3, 5, 7}. Training employed the Adam optimizer (learning rate = 0.001, batch size = 64, epochs = 30). These parameters were kept constant across architectures (MobileNet, ResNet-56, VGG13BN, ShuffleNetv2) to ensure comparability. Code and configurations were maintained in version-controlled repositories for transparency and reproducibility.

In this experiment, we use the CIFAR-10 dataset as the test data to be perturbed by the adversary and evaluate the impact of adversarial attacks on four different CNN classifiers: MobileNet, ResNet, VGG13BN, and ShuffleNet. The original CIFAR-10 dataset is evaluated on each of the pre-trained models to obtain the initial accuracy A of the models. The perturbations added to the natural dataset are derived using the Projected Gradient Descent (PGD) algorithm, with varying k values to adjust the strength of the attack. A higher value of k corresponds to a higher attack strength, and vice versa. The attack algorithm takes in the natural dataset and returns adversarial datasets generated with respect to the corresponding pre-trained model and bounded by epsilon ϵ. The pre-trained models are then evaluated with the generated adversarial dataset to observe the accuracy A_k of the models after the PGD attack, which is lower than the initial accuracy A, as shown in Table 1. Using adversarial training, the pre-trained models are retrained to obtain models robust to perturbed adversarial data. The accuracy results show a significant improvement

from the pre-trained models. The accuracy A'_k of the retrained models is also shown in Table 1. The accuracy of the model decreases with the strength of the PGD attack, which can be varied by changing the value of k. Increasing the value of k in the PGD algorithm produces more perturbed CIFAR-10 datasets, leading to more misclassifications of the pre-trained models. For the pre-trained ResNet-53 model, the accuracy reduced from 94.24% to 10.24% with a PGD k value ranging from 1 to 7 ($k = \{1,3,5,7\}$). Similarly, ShuffleNetv2, MobileNetv2, and VGG13BN also show reduced accuracy as k increases, as depicted in Figure 2.

To observe the impact of adversarial data on the robust retrained model, the retrained model is evaluated on the natural dataset. We find that the accuracy A'_k of the retrained model on the natural dataset is significantly lower than the accuracy of the pre-trained model on the natural dataset.

Table 5. 3 Mixed Bayesian Stackelberg Accuracy A^* for Multiple Adversary Types k=(1,3)

Models	A	k	A_k%	A'_k%	A^*_{min}%	A^*_{max}%
vgg13_bn	94.24	1	53.41	17.93	38.05	43.81
		3	14.38	16.97	–	
mobilenetv2_x1_4	93.88	1	52.31	10.54	43.29	46.39
		3	21.72	10.78	–	
shufflenetv2_x2_0	93.63	1	53.76	20.80	39.20	43.08
		3	15.14	20.09	–	
ResNet-56	94.46	1	54.06	32.00	45.84	48.69
		3	22.93	32.32	–	

Table 5. 4 Mixed Bayesian Stackelberg Accuracy A for Multiple Adversary Types k=(5,7)*

Models	A	k	$A_k\%$	$A'_k\%$	$A^*_{min}\%$	$A^*_{max}\%$
vgg13_bn	94.24	5	25.17	15.66	35.96	37.59
		7	10.24	17.58	–	
mobilenetv2_x1_4	93.88	5	32.37	10.67	40.99	42.23
		7	16.94	10.72	–	
shufflenetv2_x2_0	93.63	5	25.96	20.77	37.06	38.56
		7	10.69	20.25	–	
ResNet-56	94.46	5	33.5	34.53	44.66	46.14
		7	17.85	33.65	–	

We performed experiments on four pre-trained classifiers: MobileNet, ResNet56, VGG13BN, and ShuffleNet. Using the PGD attack, we modeled two pairs of attacks: a mild adversarial perturbation and a strong perturbation attack, corresponding to a weak attacker g and a stronger attacker G by varying the k value in the PGD algorithm. The pairs of attacks represent the adversary type; a lower value of k denotes a weak adversary g, while a higher value of k denotes a stronger adversary G. In an attack scenario, adversary type t_1, which is the weak adversary g, will have a lower k value compared to attack t_2, which is the stronger adversary G. In addition to these, each adversary has two strategies to choose from to maximize their payoff. The payoff for each strategy is derived from Equations (6) and (7) to confront a defender that chooses between deploying a pre-trained or retrained model.

As an illustration, a defender deploys a pre-trained model with an accuracy of 94.24% on the CIFAR-10 dataset. After an adversary uses PGD with $k = 1$ to perturb the dataset, the pre-trained model's accuracy drops to 53.41%. However, by using adversarial training to retrain the pre-trained model on the perturbed dataset, the accuracy improves from the previous 53% to 63%. On evaluating the retrained model on the original CIFAR-10 dataset, we observe that even though the retrained model has improved accuracy on the

adversarial data, its accuracy on the original data dropped to 17.93%. The accuracy of the retrained model facing an adversary t_2 with $k = 5$ is even lower. The adversarial training accuracy is 46.92%, while the retrained accuracy on CIFAR-10 is 16.97%.

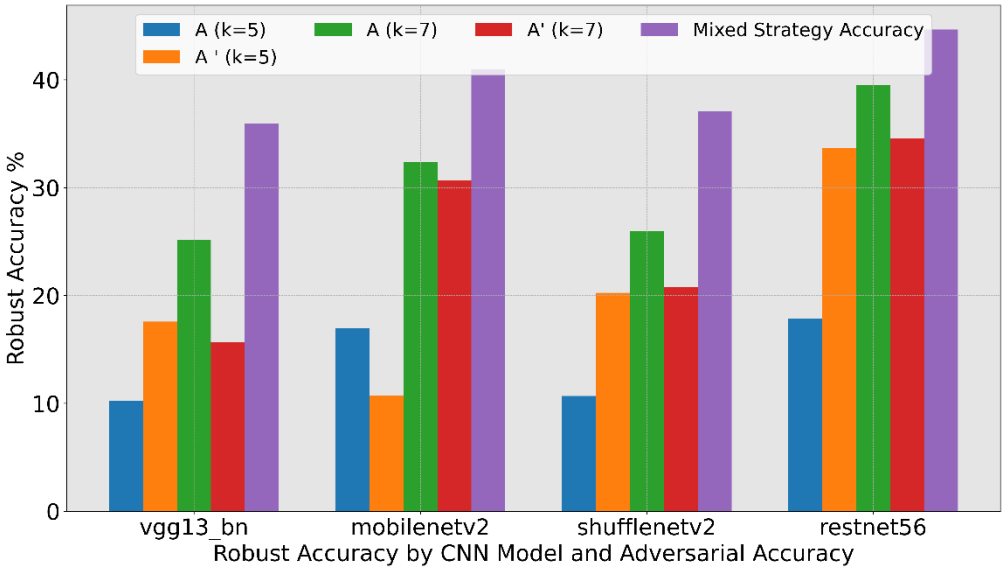

Figure 5. 4 Robust Accuracy for CNN Models Considering Adversary Types k=(1,3)

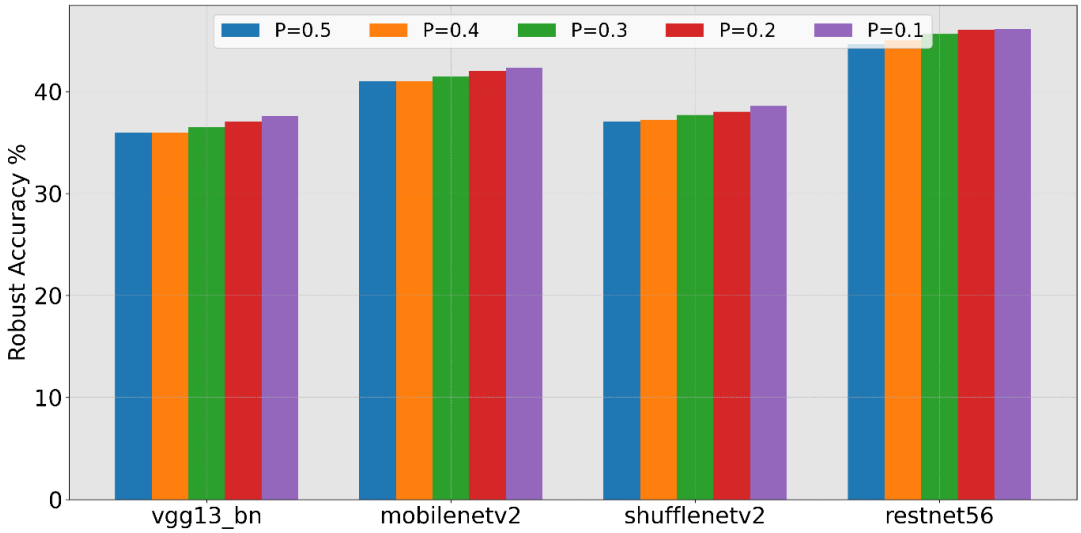

Figure 5. 5 Robust Accuracy for CNN Models Considering Adversary Types k=(5,7)

To obtain a model that performs well on both natural and adversarial datasets, a mixed Bayesian Stackelberg algorithm is employed. The problem is modeled with two types of adversaries using two different strategies: a global strategy and a direct strategy. The payoffs for both adversary strategies are given by Equations (6) and (7). The optimal mixed strategy of the defender is obtained by solving the mixed integer quadratic equation (9) and the corresponding accuracy payoff. The goal is to develop a randomized classifier selection strategy such that the adversary cannot deploy a perturbed dataset to undermine the accuracy of the selected classifier. The relationship between the defender and the adversary is framed as a Bayesian Stackelberg game consisting of t adversary types, $1, \dots, t$. The defender's set of pure strategies includes two CNN models: a pre-trained model and a retrained model. The defender can choose a mixed strategy such that the adversary is uncertain about which CNN model is being deployed, although the adversary may be aware of the mixed strategy the defender is implementing. For instance, the adversary can observe how often each CNN model is deployed over time and then select an attack strategy that guarantees maximum impact. The adversary will receive a lower payoff if it uses a direct attack targeted at a pre-trained model while the defender deploys a retrained model. Conversely, the adversary will achieve a higher payoff if it uses the global attack while the defender chooses a pre-trained model.

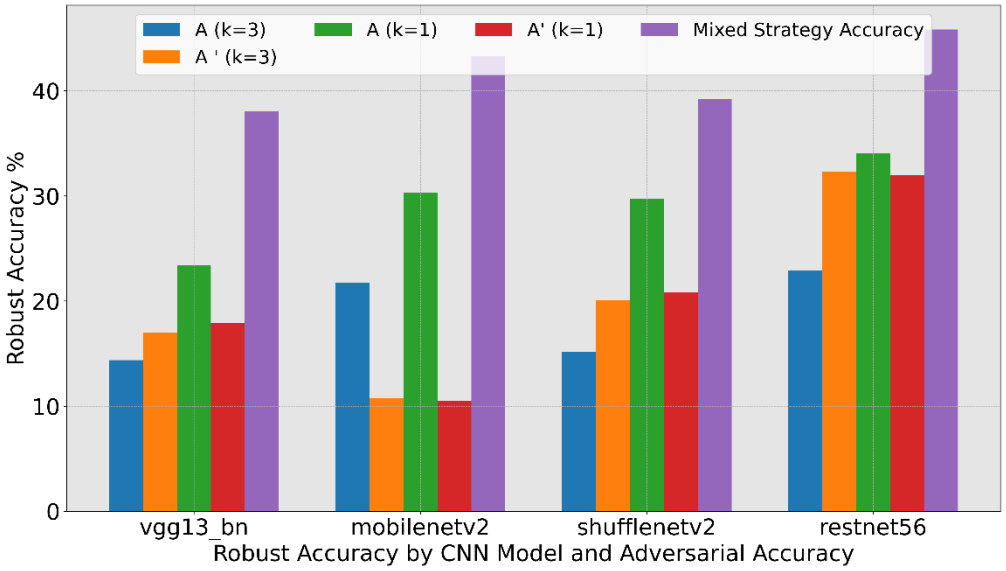

Figure 5. 6 Accuracy of CNN Models based on the Prior Probability of Adversary Type

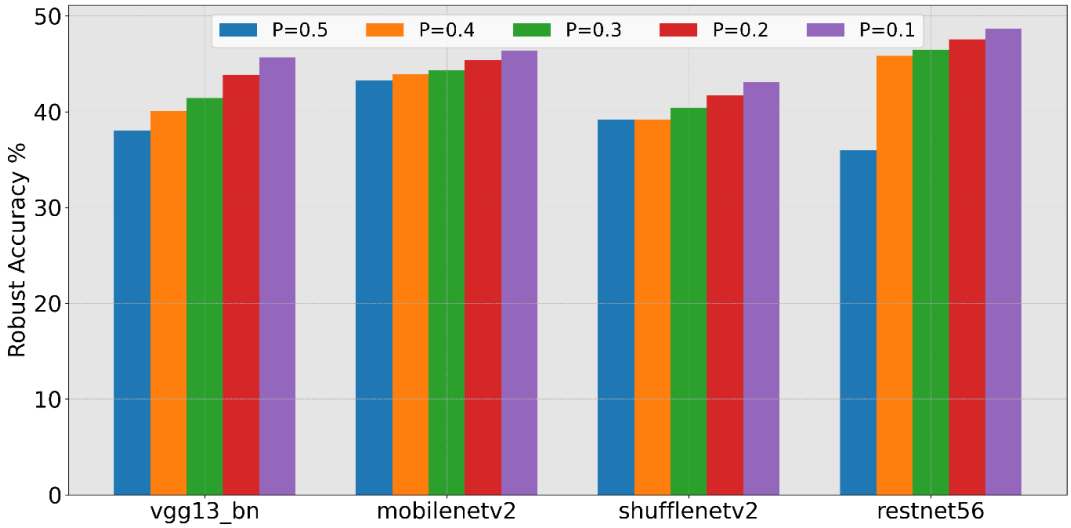

Figure 5. 7 Accuracy of CNN Models based on the Prior Probability of Adversary Type k=(5,7)

To reconcile the effect of the significant reduction in accuracy, the Bayesian Stackelberg algorithm finds a mixed strategy, as shown in Fig. 1, for the defender. This strategy ensures that the accuracy after retraining the model is consistently better than the accuracy of the pre-trained model when attacked by the strongest adversary, and also better than the accuracy of the retrained model on the original CIFAR-10 dataset. The pre-trained VGG13BN model experienced the highest impact from adversarial attacks, with a notable reduction in accuracy after perturbation for both $k = 3$ and $k = 7$. Figure above shows that the pre-trained accuracy A_k and the retrained accuracy A'_k after the attack are 25.17% and 15.66% for $k = 3$, respectively, and even lower, at 10.24% and 17.58% for $k = 7$, as shown in Fig. 2. However, the mixed strategy for the defender, which combines both pre-trained and retrained models, achieves an accuracy of 35.96% as shown in Fig. 1. Similar results are observed for MobileNetV2, ShuffleNetV2, and ResNet-56.

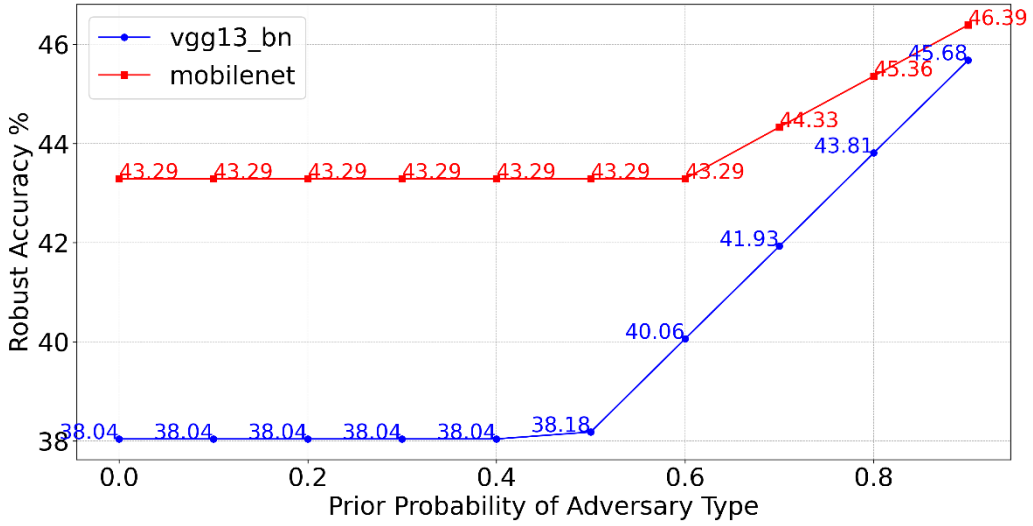

Figure 5. 8 Accuracy of CNN Models based on the Prior Probability of Adversary Type k=(1,3)

Before committing to a mixed strategy, the defender considers the prior probability P of encountering type of adversary. With varying probabilities P that a strong adversary G may not appear, the defender only begins to see a notable increase in accuracy when there is at least 60% certainty that they will confront a weaker adversary g, as shown in Fig. 7. This indicates that, with the knowledge that the models are more susceptible to a strong attack, the mixed strategy accuracy is conservative and only improves when there is a higher likelihood that a strong attack will not occur. As shown in Fig. 5 and Fig. 6, the knowledge of the prior probability of an adversary type perturbing the dataset also affects the accuracy achieved by the mixed strategy implemented by the defender. Intuitively, a higher prior probability of a weak adversary g perturbing the dataset, as opposed to a stronger adversary G, results in higher accuracy from the mixed strategy. Conversely, if there is a higher probability that the adversary is stronger, the resulting accuracy from selecting the mixed Bayesian Stackelberg strategy will be lower.

5.5 Finding Summary

The experimental results underscore the inherent trade-off between adversarial robustness and natural-data performance. The observed accuracy degradation of all pre-trained models under increasing PGD iteration $k$values demonstrates that adversarial perturbations amplify the model's sensitivity to small input variations, exposing weaknesses in gradient-based decision boundaries. The sharper accuracy decline in models such as VGG13BN and ResNet-56 suggests that deeper or more parameter-rich architectures may possess larger attack surfaces due to the greater number of gradient pathways that adversaries can exploit. This finding aligns with existing theoretical analyses indicating that model complexity often correlates with increased vulnerability to perturbations.

Adversarial retraining significantly mitigates this vulnerability by reorienting model gradients toward smoother local minima, thereby improving the model's capacity to resist high-frequency perturbations. However, this improvement comes at a measurable cost: retrained models exhibit reduced generalization on clean data, as observed in the drop of A'_kon the natural CIFAR-10 dataset. This accuracy reduction reflects a phenomenon known as robust overfitting, where models trained on adversarially augmented datasets adapt excessively to synthetic perturbations while underperforming on unaltered examples. Consequently, while retraining enhances defensive resilience, it introduces a tension between robust accuracy and standard accuracy, echoing the theoretical robustness–accuracy frontier discussed in prior works.

The implementation of the mixed Bayesian Stackelberg strategy effectively balances this trade-off by probabilistically selecting between pre-trained and retrained models according to the inferred adversary type. The resulting equilibrium demonstrates that a randomized defence mechanism can outperform either pure strategy alone. When the defender employs this mixed policy, the adversary faces uncertainty about the classifier type, leading to a reduced expected payoff for any deterministic attack. Empirically, this is reflected in the improved mixed-strategy accuracy (e.g., 35.96% for VGG13BN under $k = 7$), which exceeds both the standalone retrained and pre-trained model accuracies under the same attack conditions.

From a game-theoretic perspective, these findings validate the existence of a practical Stackelberg equilibrium in the adversarial learning framework. The Bayesian formulation further highlights the role of prior belief distributions in shaping defensive outcomes. Figures 5–7 reveal that accuracy under the mixed strategy increases when the defender's prior probability $P(g)$of encountering a weaker adversary exceeds 0.6. This indicates that the mixed strategy is conservative optimizing for average-case rather than worst-case scenarios consistent with rational behaviour under incomplete information. When the likelihood of facing a strong adversary rises, the mixed strategy adjusts defensively but at the expense of accuracy, illustrating a real-world manifestation of the accuracy–robustness trade-off predicted by equilibrium theory.

The results also suggest that adversarial uncertainty introduces a stabilizing effect: by avoiding commitment to a single defensive posture, the defender reduces vulnerability to exploitative attacks targeting predictable model behaviour. This adaptive equilibrium principle underpins the proposed framework's novelty, demonstrating that rational defence randomization guided by Bayesian inference can yield superior resilience without extensive retraining. The convergence of experimental outcomes across MobileNetV2, ShuffleNetV2, ResNet-56, and VGG13BN further indicates that the mixed Stackelberg formulation generalizes effectively across diverse architectures, reinforcing its potential as a scalable and architecture-agnostic defence strategy.

In summary, these results confirm that while adversarial training enhances robustness, its benefits are most effectively realized within a game-theoretic defense framework that accounts for uncertainty, model heterogeneity, and adversary adaptation. The mixed Bayesian Stackelberg approach provides a principled mechanism to navigate the robustness–accuracy boundary, ensuring improved defensive stability under varying adversarial conditions.

Chapter 6

Quantum Machine Learning: Quantum SVM Algorithms for Efficient Defense Against Gradient-Based Adversarial Attacks

This section addresses challenge 2 and has been published in NaNA 2025 International Conference on Networking and Network.

6.1 Introduction

Quantum computing has applications across a wide range of fields, including chemistry, physics, artificial intelligence, and data mining. Quantum game theory, which extends classical game theory into the quantum domain, has attracted growing interest from researchers in these disciplines. In this framework, quantum strategies that leverage quantum mechanics properties such as superposition, entanglement, and interference are adopted rather than traditional classical strategies.

The advancement and increasing accessibility of quantum computers have provided researchers with greater opportunities to explore and experiment with quantum systems. This accessibility has accelerated investigations into quantum strategies within conventional games, such as the Prisoner's Dilemma and various two-player games. While zero-sum finite games may not always exhibit equilibrium in pure strategy settings, they often reveal equilibrium when players adopt mixed strategies.

More recently, quantum game theory has gained significant attention in the field of cybersecurity, where researchers are evaluating whether classical cryptographic systems are resilient enough to withstand emerging threats posed by quantum computing technologies [104] [105].

6.1.2 Qubits and the Quantum Computing Paradigm

Quantum computing represents a fundamentally different model of computation compared to classical computing. In a classical computer, information is processed through electrical signals, where high voltage represents a 1 and low voltage represents a 0. These values form the basis of bits, the basic units of classical information, which operate under a deterministic logic given an input, the output is always predictable.

Quantum computers are built on the principles of quantum mechanics, which introduces non-determinism and probabilistic behaviour into computation. The basic unit of information in a quantum computer is the qubit. Unlike classical bits, which can only exist in one of two states, a qubit can exist in a superposition of both states at the same time.

Qubits are represented using Dirac notation. The two fundamental states of a qubit are written as $|0\rangle$ and $|1\rangle$ which are numbers but vectors in a mathematical space known as a Hilbert space a 2-dimensional complex vector space [106] [47].

A qubit's state can be any combination of these two basic states, expressed as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

Where α and β are complex number amplitudes that describe the probability of measuring the qubit in either state, or they must satisfy the condition:

$$|\alpha|^2 + |\beta|^2 = 1.$$

This ensures the state is normalized, meaning the probabilities of all possible outcomes add up to 1. These qubit states can also be expressed in the computational basis as:

$$|0\rangle = [1, 0] \text{ and } |1\rangle = [0, 1].$$

Quantum computers manipulate these qubit states through specialized operations that respect quantum principles. What makes qubits powerful is both their ability to hold more than one value at a time and also their ability to become entangled a property where the state of one qubit depends on another, even across distance. These quantum properties

superposition, entanglement, and quantum parallelism are what enable quantum computers to solve certain problems much more efficiently than classical computers.

In a two-qubit system, each individual qubit can exist in either the state $|0\rangle$ or $|1\rangle$. When considering the combined system of both qubits, this results in four possible configurations: both qubits can be in the $|0\rangle$ state, the first in $|0\rangle$ and the second in $|1\rangle$, the first in $|1\rangle$ and the second in $|0\rangle$, or both in the $|1\rangle$ state. These four combinations $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ form what is known as the computational basis of the two-qubit system. This basis spans a four-dimensional complex vector space, which arises from taking the tensor product of the two individual qubit spaces.

Quantum entanglement arises when two or more qubits become linked in such a way that the state of one qubit cannot be described independently of the state of the other(s). This interaction produces a unique combined state for the entire system, one that cannot be decomposed into separate, individual qubit states. In this work, we focus specifically on the characteristics of entangled states that emerge from the combination of quantum states in two-qubit systems [107].

Bell states are a specific set of four maximally entangled two-qubit states that form a basis for the space of entangled qubit pairs that can be created through the application of specific quantum gates. For instance, when a Hadamard gate is applied to the first qubit in the initial state $|00\rangle$, followed by a CNOT gate with the first qubit as the control and the second as the target, the system evolves into an entangled state known as the Bell state:

$$|\Phi^+\rangle = (|00\rangle + |11\rangle) / \sqrt{2}.$$

The state inherently has a property that measurement of one qubit instantly determines the outcome of the other, regardless of the distance between them. This non-classical correlation is at the heart of many quantum algorithms and protocols, including quantum teleportation and quantum cryptography.

In general, a two-qubit quantum state $|\psi\rangle$ can be written as a linear combination of the four computational basis states:

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle.$$

where α, β, γ, and δ are complex probability amplitudes that satisfy the normalization condition:

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1.$$

Whether or not such a state is entangled depends on whether it can be factored into the tensor product of two single-qubit states. If not, the state is entangled.

The tensor product, denoted by the symbol ⊗, is a mathematical operation used to combine quantum states. If the first qubit is in state $|0\rangle$, represented as the column vector [1, 0], and the second qubit is in state $|1\rangle$, represented as [0, 1], then the tensor product $|0\rangle \otimes |1\rangle$ yields a new state vector $|01\rangle$, which corresponds to [0, 1, 0, 0] in the four-dimensional space. This structure is essential in quantum computing because it not only allows us to represent independent combinations of qubit states but also forms the mathematical foundation for describing entangled states, which have no classical equivalent [94] [108]. Understanding the tensor product and the computational basis is crucial for working with multi-qubit systems and for harnessing the full potential of quantum information processing.

6.1.3 Quantum Gates and State Transformations

Another fundamental aspect of quantum computing is the ability to deliberately transform the states of qubits. In this framework, transformations are carried out by quantum gates, which are mathematically represented as unitary matrices. For two-qubit systems, these gates act on vectors within a four-dimensional complex vector space.

The simplest way to construct a two-qubit gate is by taking the tensor product of two single-qubit gates; two one-qubit gates, denoted U_1 and U_2, and two one-qubit states, $|\psi_1\rangle$ and $|\psi_2\rangle$. By applying the tensor product of the gates, we can create a two-qubit gate that acts on the combined state $|\psi_1\rangle \otimes |\psi_2\rangle$. Thanks to the property of linearity in quantum mechanics, this operation can be naturally extended to any linear combination of two-qubit states.

Mathematically, the two-qubit gate associated with these operations is the tensor product of the individual one-qubit matrices, expressed as:

$$U_1 \otimes U_2 .$$

This resulting matrix is itself unitary, preserving the essential properties required for quantum evolution, and therefore rightly deserves the title of a quantum gate.

More generally, a unitary operator U in quantum mechanics is simply a matrix that, when applied to a quantum state $|\psi\rangle$, results in a new state $|\psi'\rangle$. The operation is described by:

$$|\psi'\rangle = U|\psi\rangle.$$

This means that applying a unitary matrix to a quantum state transforms it within its Hilbert space without changing its overall probability norm. As illustrated in Figure 3, the unitary operator Uf acts on the initial state $|\psi\rangle$ and produces the transformed state $|\psi'\rangle$.

Figure of Hadamard gate and qubit

For $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$,

Hadamard gate applied to a qubit transforms it to a state of superposition. Hence for state 0, $H|0\rangle = {}^{1}/_{\sqrt{2}}(|0\rangle + |1\rangle)$, and state 1, $H|1\rangle = {}^{1}/_{\sqrt{2}}(|0\rangle - |1\rangle)$. The Hadamard gate and the operation on qubit state $|0\rangle$ and $|1\rangle$ is represented in matrix form as:

$$H|0\rangle = {}^{1}/_{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = \begin{pmatrix}{}^{1}/_{\sqrt{2}}\\ -1/\sqrt{2}\end{pmatrix}$$

$$H|1\rangle = {}^{1}/_{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \begin{pmatrix}{}^{1}/_{\sqrt{2}}\\ 1/\sqrt{2}\end{pmatrix}.$$

A CNOT gate applies an X-gate to a target qubit only if we measure the control qubit as 1. This gate takes in t 2 inputs, a control qubit and a target qubit. If the control qubit is $|0\rangle$ then nothing happens but if the control qubit is $|1\rangle$ then the CNOT-gate applies an X-gate on the target qubit essentially flipping its state.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

6.1.4 Entangled Strategy Simulation for Adversarial Threat Modelling

The quantum circuit is initialized with two qubits, which represent the roles of a defender and an attacker in an adversarial interaction. The marginal probability assigned to the first qubit is set using an RY rotation, which prepares it in a superposition state corresponding to the defender's likelihood of adopting a defensive strategy.

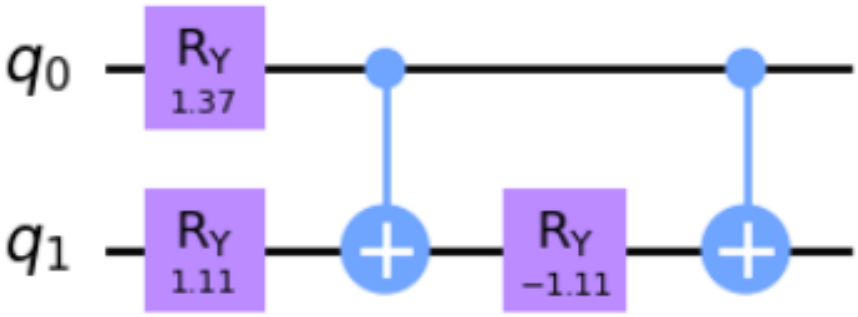

Figure 6. 1 Circuit diagram for quantum entanglement and rotation of 2 Qubits

By adjusting the rotation angle based on the defender's probability of success or choice, the qubit's quantum state encodes uncertainty in the defender's response to an attack. This reflects the realistic nature of cybersecurity, where a defender's actions are probabilistic rather than deterministic, depending on detection capabilities, resource allocation, and strategic priorities.

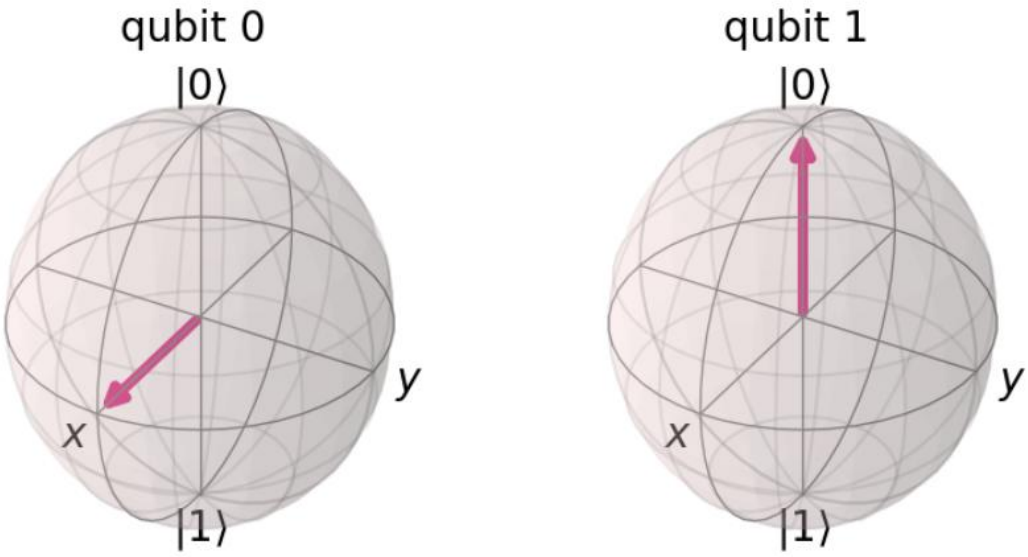

Figure 6. 2 Blochs Sphere showing the rotation of qubit 0 and qubit 1

Following the preparation of the defender's qubit, a controlled RY gate (CRY) is applied between the defender's and the attacker's qubits. This operation introduces a conditional relationship between their behaviours: the attacker's strategic move, represented by the second qubit, is influenced by the state of the defender's qubit. The conditional probability encoded by the CRY gate models a dependency where the attacker's success is not independent but tied to the defender's initial preparedness. If the defender prepares poorly, low defensive probability, the attacker's probability of a successful breach increases, and vice versa. In quantum terms, the CRY operation entangles the two players' states partially, creating a system where the measurement outcomes for one player affect the probability distribution for the other.

Finally, simulating and measuring the quantum circuit allows observation of the joint probability distribution over attacker and defender outcomes. This reflects a quantum game setting where strategies are probabilistically entangled rather than separate. From an adversarial learning perspective, this simple two-qubit model captures dynamic and strategic dependencies, highlighting how defence mechanisms condition an attacker's pathway. It serves as a basic yet powerful demonstration of how quantum computing can model

adversarial behaviour; not simply by independent actions, but through structured, entangled probability spaces that better reflect real-world security dynamics. As quantum computation scales, such models could extend into more complex quantum game frameworks, helping design more robust, adaptive cybersecurity strategies against quantum-enabled threats.

We present a toy example context of quantum game theory, modelling the interaction between an adversarial attacker and a defender using a two-qubit quantum system reveals unique advantages. The setup involves representing 2 events; event A associated with the attacker's strategy and event B associated with the defender's strategy as marginal probabilities on individual qubits. A qubit 0 is manipulated to represent the probability of the attacker's strategy with a marginal probability of 0.6, while qubit 1 represents the probability of the defender's reaction with a marginal probability of 0.4. Using quantum RY and CRY gates to rotate the qubit probabilities are encoded directly into the quantum states. The critical state 0011 where both qubits are 1 captures the overlap of these strategies, resulting in a joint probability of 0.24 (0.6×0.4). This setup mirrors the real-world situation where the success of the attacker depends on their own action and the likelihood of an effective defensive response occurring simultaneously.

The construction of the quantum circuit further enriches this strategic model by controlling how probabilities are transferred and recombined. Initially, the RY gate splits the probability amplitudes across different states depending on the rotation angle determined by event A. When the CRY gate is applied next, it further refines the distribution by conditionally rotating qubit 1 based on the state of qubit 0. This creates a fine-grained control over the states, allowing the model to differentiate cases where the attacker succeeds or fails based on whether the defender has activated their countermeasure. For instance, the application of the CRY gate results in a correction: states where qubit 0 is 0, indicating no attack, have their probability moved back into the baseline (state 0000), while when qubit 0 is 1 (attack initiated), the effect on qubit 1 simulates the likelihood of defence activation. Importantly, after applying a second CNOT gate, the probabilities of states are realigned so that the joint probability (state 0011) accurately reflects the simultaneous occurrence of both attacker and defender actions. This progression ensures that the outcome measurement of a single qubit captures the full dynamics of the encounter.

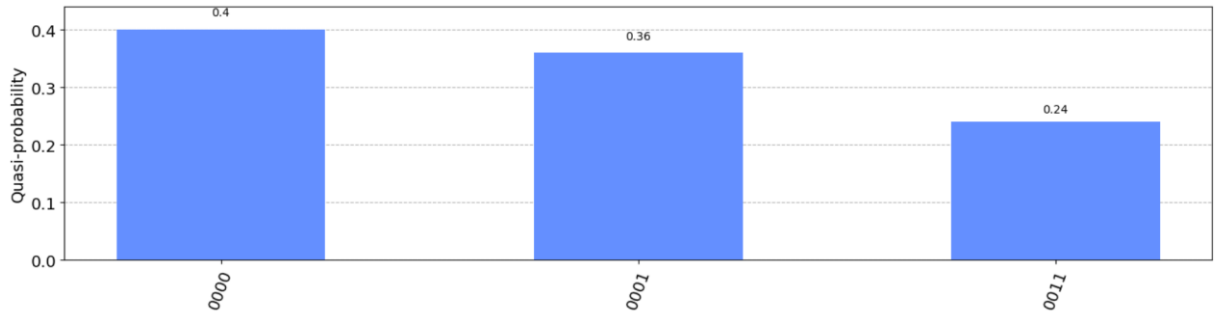

Figure 6. 3 States probability distribution of 4 states representation of 2 qubits after entanglement and measurement of Marginal probability

From a strategic standpoint, this quantum setup gives the defender a precise method to focus resources only where needed for instance when an attack is likely. The controlled RY gate allows transformations to occur only when qubit 0 (the attack event) is active, meaning that the defender does not waste computational "energy" on unnecessary states. Similarly, the attacker's likelihood of success can be modelled and adjusted based on their choice of strategies encoded in the qubit rotations. Unlike classical probability models that split all possibilities equally, quantum circuits using controlled operations like CRY gates provide fine-tuned control over the quantum state space. This better mirrors real-world adversarial settings, where strategic actions and reactions are highly conditional and intertwined. As a result, the use of quantum circuits with controlled rotations elegantly captures the essence of adversarial dynamics: selective influence, strategic overlap, and outcome dependence, making it a powerful framework for quantum-based cybersecurity simulations.

6.1.5 Adversarial Learning in the Design of Quantum Games

The paradigms of quantum computing have drawn from the structures and parameters of classical game theory. Quantum mechanics enables the reinterpretation of game-theoretic models in environments where strategies exist in superposition and outcomes may be entangled. This gives rise to quantum games, where decision-making reflects probabilistic outcomes and quantum interactions. Classical game theory provides a foundation for

structuring these interactions - distinguishing between symmetric and asymmetric roles, zero-sum and non-zero-sum outcomes, games with or without Nash equilibria or Pareto optimality. However, not all classical game categories apply to quantum computing. Some types - such as extensive-form, combinatorial, or imperfect-information games - are difficult to implement in quantum circuits due to constraints and differences in state evolution.

To address the complexity of quantum game implementation, adversarial learning has emerged as a framework. One technique, adversarial GRAPE (a-GRAPE), models quantum control design as a game between a control agent and an adversary introducing uncertainty or noise. Both sides optimize strategies in a game-theoretic sense - seeking an equilibrium where the controller maintains fidelity despite perturbations. This models adversarial dynamics where players operate under uncertainty. Quantum generative adversarial networks (QuGANs) adapt classical GANs to quantum systems, creating a competition between a generator and a discriminator. The generator produces quantum states similar to real data, while the discriminator attempts to detect forgeries. This process moves both sides toward a minimax equilibrium.

Adversarial learning improves the robustness of quantum strategies by integrating simulated perturbations during training. Quantum classifiers, like classical ones, can be vulnerable to adversarial examples designed to cause misclassification. Training on such examples improves model performance. The learning process, framed as a repeated game between a classifier and an attacker, helps define decision boundaries that generalize in noisy or adversarial conditions. This mirrors classical Nash equilibrium scenarios, where no player can improve their outcome alone.

By embedding adversarial dynamics into quantum game design, researchers can simulate hostile environments, identify weaknesses, and find optimal strategies.

6.1.6 Adversarial and Defender Decision Strategies

In adversarial and defender decision-making, quantum games allow both players to adapt strategies based on prior interactions. A non-cooperative quantum game can be described by the tuple $\langle N, \Omega, P \rangle$, where N is the number of players, Ω the strategy set for each player, and P the payoff function mapping strategies to outcomes. The adversary seeks to reduce

the defender's effectiveness through perturbations or misleading moves. The defender adjusts their approach to maintain system performance. These interactions form a loop where participants refine strategies through quantum operations, using feedback from entangled states and payoff outcomes.

Strategies involve quantum measurement, probabilistic modeling, and unitary transformations, where each move depends on expected utility and observed responses. The goal is to reach a quantum Nash equilibrium, where no player gains by changing strategy alone. This setup mirrors minimax optimization, where each player responds to the best possible action of the other. Adversarial learning allows for iterative improvement of strategies in quantum systems.

The transformation from classical to quantum algorithms introduces challenges, including new implementation techniques such as qutrits and Hadamard gates. Hadamard transformations entangle quantum states, enabling multiple overlays and allowing quantum games to use computational parallelism. These features expand the strategic scope of adversarial decision-making. Ongoing debate on implementation reflects the evolving nature of quantum game design. While quantum systems may resolve a range of decision strategy problems, constructing architectures to support these strategies remains a work in progress.

6.1.7 Quantum Nash Equilibrium

Nash equilibrium in classical game theory describes a strategy profile where no player improves their outcome by changing strategy alone. In quantum game theory, this includes quantum phenomena such as superposition and entanglement, leading to a quantum Nash equilibrium. Each player's strategy, possibly involving unitary operations on entangled qubits, remains optimal relative to the others' strategies.

Quantum Nash equilibria differ from classical ones due to the expanded strategy space in quantum systems. Superposition and entanglement modify how payoffs are structured. A classical example is the Prisoner’s Dilemma, where both players defect to gain individually. This game also demonstrates pure strategies without randomness or probability. In quantum implementations, entanglement allows players to reflect on each other's strategies.

According to Van Enk and Wu, this transformation eliminates individualistic strategies and turns the game from non-cooperative to cooperative. This results in outcomes where both players may choose cooperation instead of mutual defection.

In the Eisert-Wilkens-Lewenstein (EWL) quantum version of the Prisoner's Dilemma, each player's strategy is a unitary operator acting on a shared entangled state. When players use quantum strategies instead of classical moves, new outcomes emerge. Hadamard transformations entangle qubits and generate multiple overlays, enabling quantum parallelism. These overlays increase strategic combinations, expand the decision space, and enable cooperation. Designing such quantum games involves implementation techniques like using qutrits and specific quantum gates.

In the classical version, if both players confess, each receives 3 years in prison. If one confesses and the other does not, the confessor goes free while the other gets 5 years. If neither confesses, both receive 1 year. This payoff matrix creates a clear incentive for defection, resulting in a Nash equilibrium at mutual confession. Quantum implementations alter this behavior through entanglement, potentially changing the payoff structure. Studies have explored transitions using mathematical models, comparisons to human strategies, and analysis of decoherence. Some models omit Hadamard gates; others include them to study entanglement and unitary operations. These variations reflect the evolving landscape of quantum game theory. By adjusting their operations, players can escape the classical dilemma and reach cooperation, forming a quantum Nash equilibrium.

The reversibility of unitary operations allows players to revise strategies to improve payoffs. Entanglement may act as a coordination mechanism, similar to contracts in abstract economics. The EWL framework formalizes classical games into quantum settings while preserving equilibrium properties.

In adversarial learning, understanding quantum Nash equilibria aids in designing algorithms that remain stable under competition or uncertainty. As players model threats and responses, equilibrium concepts guide consistent strategy development. This integration of game theory and quantum systems supports algorithm design and control.

Ensuring the robustness of machine learning classifiers against adversarial attacks has become a critical challenge. The vulnerability of these classifiers to adversarial data has attracted considerable interest in the use of machine learning in real-world applications. Adversarial attacks exploit this vulnerability by introducing perturbations that are imperceptible to humans but sufficient to cause a model to misclassify with high probability. In response to these susceptibility, quantum adversarial machine learning has emerged as a promising approach. By leveraging the principles of quantum computing, quantum adversarial machine learning has potential to enhance classifier robustness, offering more effective defense measures against adversarial perturbations and increasing overall robustness.

The advantages of quantum technologies in fields such as cryptography, simulation, and quantum computing have become well recognized by researchers. Recent advancements in quantum computing have enabled applications of quantum machine learning (QML) that were previously infeasible. While machine learning (ML) algorithms have shown significant potential, training these algorithms on large datasets, especially for applications like computer vision and genomics, presents challenges. Specifically, for classification tasks for which the data Hilbert-space dimension is large making such applications vulnerable to adversarial attack. The numerous real-life applications of large datasets to ML training result in increased vulnerability of ML models to adversarial perturbations attacks in areas like adversarial training, consequently robustness of ML classifiers is of critical concern.

Quantum machine learning presents an opportunity to exponentially improve performance of ML classifiers compared to classical ML techniques. This improvement has motivated the implementation of QML models, particularly in applications with limited resources. For example, deep quantum neural networks have demonstrated notable advantages in tasks like image recognition, showing improvements in system performance, accuracy, and robustness. In adversarial machine learning, quantum techniques are being

applied to study vulnerabilities in ML algorithms, providing new defense mechanisms against adversarial perturbations.

Adversarial attacks pose a significant threat to machine learning models, particularly around decision boundaries, where small perturbations can lead to misclassifications. Support Vector Machines (SVMs), which rely on kernel methods to map data into higher-dimensional spaces, are vulnerable to adversarial examples that exploit these decision boundaries. Research has shown that adversarial samples, generated by adding slight perturbations to natural data, can cross decision boundaries, resulting in high-probability misclassifications. These misclassifications are particularly challenging in high-dimensional spaces, where the trade-off between robustness and quantum advantages becomes complex.

Quantum machine learning models, such as SVM classifiers enhanced with quantum kernels, offer a promising solution. Quantum kernels enable the SVM to learn non-linear relationships in adversarial data by leveraging the dot product of input vectors in high-dimensional quantum spaces. This capability allows the model to capture intrinsic properties of adversarial perturbations that would be difficult for classical models to process.

This paper investigates the potential of quantum-enhanced SVMs in adversarial environments, focusing on the impact of quantum kernels in improving robustness and performance. We propose an adversarial attack specifically designed for image classification tasks, focusing on the perturbations introduced to the original data. Our gradient-based attack injects sufficient noise to facilitate the crossing of the decision boundary by the manipulated data [109] [110] [111] [110] [76] [112] [113] [114]. We assess the performance of the adversarial samples on a classical Support Vector Machine (SVM) model employing adversarial training, considering various kernel functions, including Radial Basis Function (RBF), linear, and polynomial kernels. Additionally, we develop a quantum kernel utilized by the Quantum Support Vector Machine (QSVM) and evaluate the performance of the adversarial data on this model. The feature mapping for the input data is conducted using ZZ-feature maps in the quantum circuit to better represent the non-

linearity of the adversarial data, enabling a comprehensive analysis of the robustness of quantum approach to adversarial training.

6.1.8 Adversarial training strategies for Machine Learning Algorithms

Previous works have shown that conventional methods of training may not be sufficient to guarantee the robustness of CNN algorithms. Methods such as data augmentation only provide partial solutions to misclassifications. Naveed et al. used empirical methods to demonstrate that dimensionality and image complexity impact a classifier's robustness against adversarial attacks in the real world. Hence, adversarial learning is essential for the development of CNN algorithms that are less susceptible to practical attack methods. Our study focuses on using adversarial training in a Stackelberg game to find a mixed equilibrium strategy that guarantees optimal accuracy and robustness for a CNN with fixed dimensions.

Game theory has been used in numerous works to model the interaction between a classifier and adversarial attacks to obtain optimal robust strategies. Humin et al. used the minimax theorem to find equilibrium for non-convex-non-concave games. Meunier et al. demonstrated that no deterministic pure Nash equilibrium exists in such interactions between a classifier and adversaries, while revealing that mixed strategies outperform their pure strategy counterparts in an infinite zero-sum game . Ya-Ping H. et al. developed practical and efficient game theory algorithm frameworks via a two-player game to compute mixed Nash equilibrium [115]. An adversarial example maxmin game between a classifier and an adversary derives an optimal adversary against Neural Networks, which can be utilized to better understand accuracy-robustness trade-offs for neural networks. Chivukula A S et al. formulated deep learning using a Stackelberg game with variational adversaries, but did not show the existence of an equilibrium [19]. Tanner Flez et al. and Chi Jin et al. studied the existence and properties of local optimality in sequential games. These works presented adversarial deep learning as a non-cooperative game; however, Lefeng et al. explained the concepts of adversarial attacks and defenses in a cooperative game setting. Our studies focus on non-cooperative interaction between the classifier and

the adversary, where the adversaries perpetually perturb the data to increase the misclassification error of the classifier.

To optimize a learner's defense mechanism for resilience towards adversarial attacks, it is important to understand how the attacks are developed. The essence of adversarial data generation is to understand different methods for which adversarial data can be created by a potential adversary. The adversary aims to perturb a valid data sample such that the perturbation is imperceptible to the human eye, but when presented to the machine learner, the data is misclassified to a wrong class. This is achieved by adding just enough perturbation to cross the decision boundary of the learner classifier. If the value of the perturbation is too large, the data becomes distorted and nonsensical to the human eye and becomes obviously perturbed. Also, if the perturbation is too small, the data looks normal to the human but is not enough to cross the decision boundary and would not lead to misclassification by the learner. Carlini et al. proposed a technique that added a small vector to an input of a model such that the magnitude of the vector is equal to the sign of the gradients of the cost function of the model, which reliably causes a wide variety of classifiers to misclassify their input. The technique showed that by training the model with the worst-case adversarial perturbation rather than itself helps to regularize the model and generally makes it perform better even under adversarial attacks. Goodfellow et al. proposed the fast gradient method (FGSM) to generate perturbations that are added to examples. The work highlighted the importance of the direction of the gradient of the cost function in deriving appropriate perturbations. Madry et al. investigated the robustness of neural networks through min-max optimization with Projected Gradient Descent (PGD). The min-max formulation reflects adversarial training and attacks against constrained optimization models [116] [117] [118] [119].

To obtain optimal strategies, attack models need to be defined explicitly. There is no single learning strategy that can be unilaterally implemented for all attack models. Current neural networks and defenses are only effective against a few attacks, keeping the models vulnerable to other types of attacks. Indeed, there is a trade-off between accuracy and robustness in the implementation of defense against adversarial samples. The large number of scenarios of attacks and metrics such as L_0, L_1, L_2, and L_∞ makes it difficult to

generalize defenses since different levels of perturbations result in varying attack sensitivity and resulting adversarial accuracy. Therefore, there is a need for algorithms that generalize well over multiple attacks without trading off accuracy for the robustness of the network [120] [121] [110].

Huang et al. model a machine learning scenario as an interaction between a learner and an adversary. The learner's objective is to correctly predict the input data, while the adversary transforms the data to make the learner misclassify them to a wrong label or output. Adversarial learning presents a considerable level of cybersecurity threat in the domains of machine learning classifiers, including automated email spam filters, image classification algorithms for self-driving cars, medical imaging applications, etc [172-179]. Kantarcioglu et al. solved a classification problem using Stackelberg equilibrium with a simulated annealing algorithm to obtain an optimal set of attributes [122] [123] [124] [125] [126]. Fiez et al. also conducted similar work, but rather than assuming both players knew one another's payoff function, they showed that it's enough to know only the adversary's payoff function. Both works modeled the adversary as the leader who stochastically chooses his strategy, while the classifier is the follower and searches for an equilibrium after observing the adversary's choice. Madry et al. investigated the robustness of machine learning classifiers through robust optimization of mini-max theoretical frameworks [200 - 202]. The optimization method reflected the essence of adversarial training and attack methods against constrained optimization.

6.2 System Modelling and Analysis

6.2.1 Adversarial Samples

Given a dataset X in a subspace of $\mathbb{R}^d$, an SVM classifier $f: X \rightarrow Y$ is susceptible to adversarial attacks, where small perturbations to input data can lead to misclassification. An adversarial sample x^* is crafted by introducing a minimal perturbation to a valid input x such that $\| x^* - x \|$ is small, but $f(x^*) \neq f(x)$. In non-targeted attacks, the goal is any

misclassification, while targeted attacks aim for a specific label $f(x^*) = y^*$. The magnitude of the perturbation is typically measured using the L_p norm, with $p = 0,1,2,\infty$.

For example, the Fast Gradient Sign Method (FGSM) is a white-box attack commonly used against SVMs, where the adversary has access to model parameters. FGSM modifies the input data x based on the gradient of the loss function $L_\theta(x, y)$, generating an adversarial sample as:

$$x^* = x + \epsilon \cdot \text{sign}\big(\nabla_x L_\theta(x, y)\big),$$

where ϵ controls the size of the perturbation, and $\nabla_x L_\theta(x, y)$ is the gradient of the model's loss for the true label y. For instance, consider an SVM trained on the MNIST dataset to distinguish between digits 3 and 8. By applying FGSM, a small perturbation to a digit 3 image can cause the classifier to misclassify it as an 8, despite the negligible visual difference.

In contrast, black-box attacks do not rely on model parameters. These attacks either transfer adversarial examples generated from another model or query the SVM to learn about its decision boundary. Although more difficult to craft, black-box attacks are more generalizable and can affect various models. Both white-box and black-box attacks highlight the vulnerabilities of SVMs, as even small, targeted perturbations can lead to significant classification errors.

6.2.2 Support Vector Machine

The Support Vector Machine (SVM) is a machine learning algorithm for classifying two disjoint categories. The SVM primary objective is to identify a hyperplane that effectively separates the two categories into their respective classes. For a given input training dataset $D = \{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ is the input feature vector, $y \in \{-1, +1\}$ is the output label, N is the number of samples, and d is the dimension of the input space; the plane is defined as a normal vector $w \in \mathbb{R}^d$ indicating the plane's orientation and a scala $b \in \mathbb{R}$ representing the intercept such that $w^T \cdot x + b = 0$. The hyperplane serves as a

boundary that distinguishes between positive and negative class samples by applying the principle of structural risk minimization.

We aim to achieve a margin between each class and the boundary, which requires finding the optimal parameters w and b, such that:

$$w^T \cdot x_i + b \geq 1 \quad \text{for } y_i = +1$$

$$w^T \cdot x_i + b \leq -1 \quad \text{for } y_i = -1 .$$

The hyperplane for these conditions can be expressed as:

$$w^T \cdot x_i + b = 0 .$$

A soft margin formulation of an SVM, used to obtain a hyperplane from training data, is formulated below by solving the following optimization problem:

$$\text{Minimize} \quad \frac{1}{2} \| w \|^2 + C \sum_j \xi_j$$

subject to:

$$y_j(w \cdot x_j + b) \geq 1 - \xi_j, \quad \xi_j \geq 0 .$$

The value $C > 0$ is a hyperparameter to tune the model. The larger the value of C, the less tolerant the model is to the training samples that fall either inside the margin or on the wrong side of the hyperplane. By using the Lagrangian multiplier method, the dual problem is solved as a convex quadratic programming problem with inequality constraints and the parameter α can be obtained. The soft-margin optimization problem is expressed in terms of parameters α_j as follows:

$$\text{Maximize} \quad \sum_j \alpha_j - \frac{1}{2} \sum_{j,k} y_j \, y_k \alpha_j \alpha_k (x_j \cdot x_k)$$

subject to the constraints:

$$0 \leq \alpha_j \leq C, \quad \sum_j \alpha_j \, y_j = 0 .$$

Once the α_j values are obtained, we can return to the original formulation and compute b and w. Specifically, w can be computed as:

$$w = \sum_j \alpha_j \, y_j x_j \, ,$$

w only depends on the training points x_j where $\alpha_j \neq 0$. We can also obtain b by selecting any x_j that lies on the margin boundary and solving the equation:

$$b = 1 - \left(\sum_{i=1}^{N} \alpha_j \cdot y_j \cdot x_i^T \right) \cdot x^{(N)}, \quad \text{for } y^{(N)} = 1 \, .$$

Based on the result, we decide if x belongs to the positive or negative class, depending on whether the output is greater than zero or not.

6.2.3 SVM Kernel Trick

This method involves transforming the data from its original space $\mathbb{R}^n$ into a higher-dimensional space $\mathbb{R}^N$, such that the resulting hyperplane can separate the data in that new space. This higher-dimensional space is called the feature space, and the function $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^N$, which maps the original data to the feature space, is known as the feature map. Let x be a vector from the input space of dimension D, and let $\{\phi_j(\mathrm{x})\}_{j=1}^{\infty}$ represent a set of nonlinear functions that map from a D-dimensional space to an infinite-dimensional feature space. In this feature space, the hyperplane can be defined as:

$$w^T \cdot \phi(x) + b = 0 \, ,$$

where $\phi(x)$ is the feature vector in the infinite-dimensional space, and w is the weight vector in that same space. With N_s representing the number of support vectors, we can express the weight vector as:

$$w = \sum_{i=1}^{N_s} \alpha_i \cdot y_i \cdot \phi(x_i) \, .$$

However, we don't need the weight vector itself; all we require is the decision boundary, which can be expressed as:

$$\sum_{i=1}^{N_s} \alpha_i \cdot y_i \cdot \phi^T(x_i)\phi(x) = 0\,.$$

What we need are the inner products between the support vectors, $\langle \phi(x_i), \phi(x) \rangle$. These inner products can be calculated using a kernel function:

$$k(x, x_i) = \phi^T(x_i)\phi(x) = \langle \phi(x_i), \phi(\mathrm{x}) \rangle\,.$$

By specifying the kernel $k(x, x_i)$, we can avoid explicitly computing the weight vector w . With the kernel, we can fully leverage the fact that:

$$\sum_{j=1}^{\infty} w_j \cdot \phi_j(x) + b = \sum_{i=1}^{N_s} \alpha_i \cdot y_i \cdot k(x, x_i) + b = 0\,.$$

In the dual problem, the scalar product $x_i^T x_j = \langle x_i, x_j \rangle$ is replaced by the kernel $k\big(\mathrm{x}_i, x_j\big)$, while everything else remains the same. Given the training sample $\{x_i, y_i\}_{i=1}^N$ with $y_i \in \{-1, +1\}$, we want to find the Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ that maximize:

$$\sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\, \alpha_j y_i y_j k\big(x_i, x_j\big)$$

subject to the constraints:

$$\sum_{i=1}^{N} \alpha_i\, y_i = 0, \quad 0 \le \alpha_i \le C, \quad i = 1,2, \dots, N\,,$$

where C is a user-specified positive parameter. To compute the output, we follow these steps:

6.2.4 SVM Adversarial Attack

In this section, we describe a method for generating adversarial examples that target Support Vector Machines (SVM) using a Radial Basis Function (RBF) kernel. The attack iteratively perturbs an input x to alter the SVM's classification by leveraging the decision function $f(x)$ and its gradient $\nabla_x f(x)$. The perturbation is designed to be proportional to the function value and gradient, ensuring that the attack effectively drives the input across the decision boundary, resulting in a misclassification.

Let $f(x)$ represent the SVM decision function, defined as:

$$f(x) = \sum_{i=1}^{N} \alpha_i \, y_i K(x, x_i) + b \, .$$

where α_i are the Lagrange multipliers associated with the support vectors, $y_i \in \{-1,1\}$ are the labels of the support vectors x_i, and $K(x, x_i) = \exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right)$ is the RBF kernel with width parameter σ. The SVM assigns a class label to an input x based on the sign of $f(x)$. The goal of the adversarial attack is to find a perturbed input x_{adv} such that the sign of $f(x_{adv})$ is different from the sign of $f(x)$, thereby causing a misclassification.

To achieve this, we iteratively perturb the input in the direction that maximizes the change in the decision function. The gradient of the decision function with respect to x is given by:

$$\nabla_x f(x) = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \alpha_i \, y_i K(x, x_i)(x - x_i) \, .$$

This gradient points in the direction of the steepest change in $f(x)$. The perturbation at each step is calculated as follows:

$$\epsilon = \frac{f(x)}{\| \nabla_x f(x) \|^2} \nabla_x f(x) \, .$$

This formulation ensures that the magnitude of the perturbation is proportional to the decision function value $f(x)$, scaled by the squared magnitude of the gradient $\| \nabla_x f(x) \|^2$,

and applied in the direction of the gradient. This adaptive perturbation allows the adversarial example to efficiently cross the decision boundary while keeping the perturbation as small as possible.

The adversarial example x_{adv} is updated iteratively using the perturbation:

$$x_{adv} = x_{adv} + \epsilon \, .$$

At each step, the perturbation is recalculated using the updated x_{adv}, and the process is repeated until the classification of x_{adv} differs from the original classification of x, i.e., until $\text{sign}\big(f(x_{adv})\big) \neq \text{sign}\big(f(x)\big)$.

The intuition behind this attack is that by computing the perturbation as a function of both $f(x)$ and $\nabla_x f(x)$, we ensure that the adversarial modification is aligned with the SVM decision boundary, leading to a more efficient and targeted attack. This approach also minimizes the overall perturbation applied to the input by adapting the step size based on the function value and gradient magnitude, making the adversarial example less detectable while still achieving misclassification.

The iterative nature of the attack allows for a controlled traversal of the decision boundary, ensuring that the adversarial example is incrementally modified until the desired misclassification is achieved. By applying the perturbation iteratively, the attack can finely tune the adversarial example to achieve the minimal necessary perturbation for misclassification. This approach provides a balance between the effectiveness of the attack and the perceptibility of the perturbation, making it well-suited for generating adversarial examples that are both effective and subtle.

Input: Input sample x, SVM decision function $f(x)$, gradient of decision function $\nabla_x f(x)$, support vectors x_i, labels of support vectors y_i, Lagrange multipliers α_i, kernel width σ Initialize: $x_{adv} \leftarrow x$ Compute the gradient $\nabla_x f(x_{adv}) = -\frac{1}{\sigma^2}\sum_{i=1}^{N} \alpha_i \, y_i K(x_{adv}, x_i)(x_{adv} - x_i)$ Compute the perturbation $\epsilon = \frac{f(x_{adv})}{\|\nabla_x f(x_{adv})\|^2} \nabla_x f(x_{adv})$ Update the adversarial example $x_{adv} = x_{adv} + \epsilon$ Output: Adversarial example x_{adv}.

6.2.5 Quantum Kernels for Adversarial Support Vector Machines

Quantum computing enhances classical machine learning models, particularly through the use of quantum kernels in Support Vector Machines (SVMs). Quantum kernels rely on quantum feature maps such as the ZZ features maps, which transform classical data into high-dimensional quantum spaces through parameterized quantum circuits. These circuits make use of quantum mechanical properties such as superposition and entanglement, allowing quantum systems to represent and process adversarial data in ways that are difficult for classical SVMS to replicate during adversarial training. The quantum transformation enables SVMs to classify complex, non-linearly separable data with adversarial perturbations in an efficient manner.

Classical SVMs utilize the Radial Basis Function (RBF) kernel to classify non-linear data by mapping the input into a higher-dimensional feature space. While effective, the RBF kernel requires tuning of parameters such as σ, which controls the kernel width. Additionally, its computational cost can increase significantly for adversarial training, as the process of explicitly mapping adversarial data into higher-dimensional spaces becomes resourceful. For adversarial training, where the SVM model must handle adversarial data x_{adv} in addition to natural data $x_{natural}$, the computational cost is impacted by the increased complexity of mapping data into higher-dimensional feature spaces. The kernel function must now handle both natural and adversarial data:

$$K(x_{natural}, x_{adv}) = \exp\left(-\frac{\| x_{natural} - x_{adv} \|^2}{2\sigma^2}\right).$$

As adversarial data is generated to be near the decision boundary, the distance $\| x_{natural} - x_{adv} \|$ may be small, increasing the difficulty of classification. For illustration, suppose $A_{natural}$ represent the accuracy of the SVM on natural data, A_{adv} represent the accuracy of the SVM on adversarial data generated using the model, and f_{svm} denote the SVM model pretrained on natural data. If the $x_{natural}$ be the natural data samples, and x_{adv} be the adversarial data samples generated from f_{svm}, then the accuracy on natural data can be defined as:

$$A_{natural} = \mathbb{E}_{x_{natural}}[\mathbb{I}(f_{svm}(x_{natural}) = y_{natural})] .$$

where $y_{natural}$ is the true label of the natural data, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the prediction is correct and 0 otherwise.

Similarly, the accuracy on adversarial data is given by:

$$A_{adv} = \mathbb{E}_{x_{adv}}[\mathbb{I}(f_{svm}(x_{adv}) = y_{adv})] ,$$

where y_{adv} represents the true label of the adversarially perturbed data.

Since adversarial data is designed to deceive the model, the relationship between the two accuracies can be expressed as:

$$A_{adv} < A_{natural} ,$$

which indicates that the SVM model performs less accurately on adversarial data than on natural data.

In contrast, quantum kernels automatically encode classical data into an exponentially large quantum feature space, without requiring manual parameter tuning. The feature space has a dimension of 2^m, where m is the number of qubits. The quantum circuits encode data into these large feature spaces through superposition and entanglement, allowing quantum kernels to represent complex relationships between data points through quantum interactions. Quantum feature maps are used to encode classical data into quantum states by leveraging quantum circuits composed of layers of quantum gates. For instance, the Pauli feature map, proposed by Havlicek et al., which is specifically designed to be difficult to simulate classically. These quantum circuits consist of layers of Hadamard gates, interleaved with entangling gates such as CNOT gates. A Pauli feature map of depth d transforms a classical input vector x of dimension $z = m$ into a quantum state through a sequence of quantum gate operations. This transformation can be expressed as:

$$U_\phi(x) = \prod_d U_\phi\,(x) H^m ,$$

In this equation, Pauli gates $P_j \in \{I, X, Y, Z\}$ are used, and the index S determines the interactions between qubits, defining how they are connected. Adversarial data is encoded into the function $\phi_S(x_{adv})$, which captures the interactions between both single and multi-qubit systems. This encoding process allows the quantum SVM to represent data in a high-dimensional quantum feature space, facilitating classification.

A fundamental property of quantum circuits is their reversibility. Given an input state $|0\rangle^m$, a parameterized quantum circuit $U_\phi(x_{adv})$ maps the adversarial data into a quantum feature space. Applying the inverse of this circuit, $U_\phi^\dagger(x_{adv})$, restores the original state:

$$U_\phi^\dagger(x_{adv})U_\phi(x_{adv})|0\rangle^m = |0\rangle^m ,$$

This reversible nature allows for a comparison of different data points. If two adversarial data points, x_{adv} and y_{adv}, are compared by first applying the quantum circuit for x_{adv} and then applying the inverse circuit for y_{adv}, the system can be measured to determine how similar the two points are. If the points are similar, the probability of measuring the state $|0\rangle^m$ will be high. On the other hand, if the points are dissimilar, this probability decreases:

$$U_\phi^\dagger(y_{adv})U_\phi(x_{adv})|0\rangle^m ,$$

This forms the basis of the quantum kernel. In classical SVMs, kernel functions compute the similarity between data points by evaluating inner products in a feature space. A quantum kernel follows a similar approach but instead computes the inner product between quantum states associated with the classical data points. For two points x_{adv} and y_{adv}, the quantum kernel is expressed as:

$$k(x_{adv}, y_{adv}) = |\langle\phi(x_{adv})|\phi(y_{adv})\rangle|^2 = |\langle 0^m|U_\phi^\dagger(y_{adv})U_\phi(x_{adv})|0^m\rangle|^2 ,$$

The kernel value is estimated by measuring the quantum state multiple times and recording the number of systems that collapses to the state $|0\rangle^m$. The number of times provides

an estimate of the similarity between the points x_{adv} and y_{adv}, with a higher kernel value indicating greater similarity.

The process of leveraging quantum kernels in SVMs integrates the quantum kernel computation with classical kernel methods. The quantum kernel is used in place of traditional kernels by generating the Gram matrix (or kernel matrix) using a quantum computer (or simulator). The remaining computations, such as solving the optimization problem in the SVM, are carried out on a conventional computer. One common approach is to pass the quantum kernel function to a classical algorithm, or alternatively, to precompute the training and testing kernel matrices. The internal representations of the quantum kernel itself are often hidden from the external user, as the quantum system directly provides the necessary kernel evaluations.

The Pauli Feature Map and the ZZ Feature Map serve as concrete implementations of quantum kernels, enabling non-linear interactions between features to be modeled through entangling operations between qubits. The ZZ Feature Map is a specific case of the Pauli feature map that uses Z gates to perform entangling operations between pairs of qubits. These transformations allow for complex data representations that are not feasible in classical feature spaces [201-205].

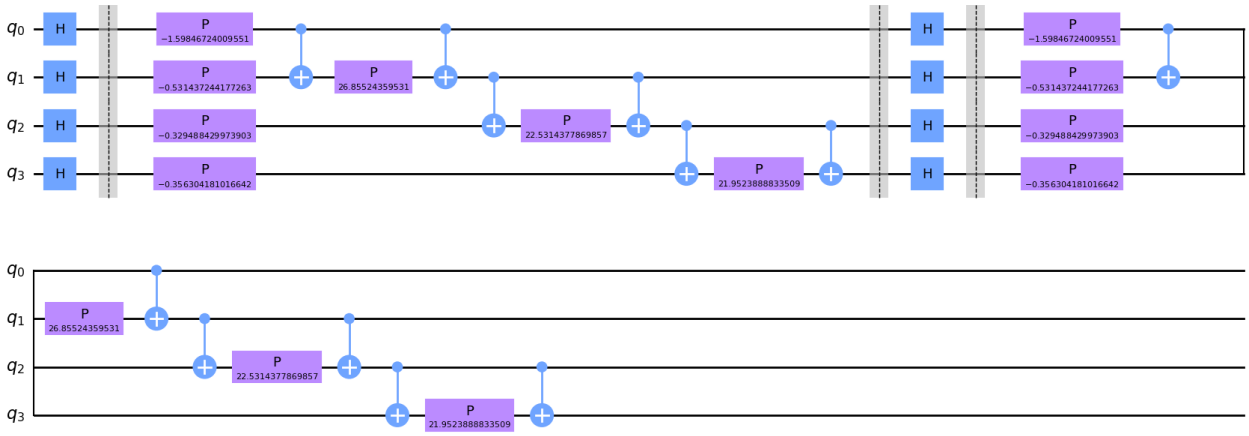

Figure 6. 4 Circuit diagram of 4 qubit states ZZ feature mapping and entanglement operation for Quantum Kernel

6.2.5 Enforcing correlation between Defender and adversarial attackers

Considering the payoff matrix of the defender and the adversarial attack where the defender has 2 strategies $D_i(i = \{1,2\})$ using adversarial trained classical *advSVM* or *QSVM* while the attacker has 2 strategies $T_i(i = \{1,2\})$ whether to use a strong perturbation attack data x_{adv} that is more resourceful quantified as λ, and risks detection or use a milder perturbation attack which is less resourceful and less visually perceptible correlated as the distance d between the original image $x_{natural}$ and perturbed image x_{adv}. The *advSVM* is trained on the worst-case scenario of adversarial perturbation available to aggressive an adversary based on the *k(k=300)* value of its PGD attack. The payoff of the stealthy adversary is the same as the aggressive adversary but discounted with the distance between the original image and the perturbed image. The discount penalizes images that are too disparate from the original in terms of the level of perturbation and measured as the average distance from the original. Given that x_{adv} is the generated adversarial sample by the attacker, and $\mathcal{D}$ is the size of the test dataset the discount factor is defined as follows

$$d = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}}\frac{||x' - x||}{||x||_2}$$

$$Payoff_{adversary} = error_{learner} - \frac{\lambda}{d}.$$

Table 6. 1 Payoff Matrices for Defender and Adversarial Attacker for Quantum Game

	T_1	T_2
D_1	(a, w)	(b, x)
D_2	(c, y)	(d, z)

Given that the defender payoff $c > a > d > b$ and similarly $y > w > z > x$ is the adversary payoff hence D_2T_2 is the dominant strategy for both players and the Nash equilibrium for the game. Such that if the defender selects a QSVM model corresponding to D_1, the attacker having complete information will select a stealthy strategy T_2, that still results in attack success however less costly. If the defender selects adversarial SVM corresponding to strategy D_2 the attacker still selects T_2 making this a dominant strategy for the attacker. However, the strategy D_1T_1 has the highest payoff and appeals to the defender but cannot be achieved using the classical game theory framework. In this situation, the defender prefers a dominant strategy of deploying an adversarial trained classical SVM not only because of accuracy on adversarial dataset x_{nat} but also for the relative low resource in training on any modern CPU/GPU, however for robustness against adversarial attack and high-dimensional data the defender prefers to deploy QSVM requiring quantum simulators with potential exponential speed-ups.

Rather than classically choosing between pure strategies probalistically as mixed strategies, the quantum game player applies unitary operator $U_{(\varphi,\alpha,\theta)}$ that superimposes both parties' strategies and control gate J that entangles the defender and adversarial qubit manipulation such that strategies become correlated.

Quantum Game

The game is initially set up in the qubit state $|00\rangle$, and the players select their quantum strategies U_i corresponding the rotation θ_i of the qubit. The quantum strategies of the players $i = 1,2$ is the set of 2×2 matrices $U_{(\varphi_i,\alpha_i,\theta_i)}$ thereby leading the system from the initial state to a final state $|\Psi\rangle$ such that

$$|\Psi\rangle = (U_1 \otimes U_2)|00\rangle = U_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes U_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

where $U_{\varphi,\alpha,\theta} = \begin{pmatrix} e^{i\varphi}\cos\frac{\theta}{2} & e^{i\alpha}\sin\frac{\theta}{2} \\ -e^{-i\alpha}\sin\frac{\theta}{2} & e^{-i\varphi}\cos\frac{\theta}{2} \end{pmatrix}$ $0 \le \varphi, \alpha \le 2\pi, 0 \le \theta \le \pi$

$$|\Psi\rangle = \begin{bmatrix} [U_1]_{11}\,[U_2]_{11} \\ [U_1]_{11}\,[U_2]_{21} \\ [U_1]_{21}\,[U_2]_{11} \\ [U_1]_{21}\,[U_2]_{21} \end{bmatrix} = \begin{bmatrix} \propto \\ \beta \\ \gamma \\ \delta \end{bmatrix}$$

$$= \propto |00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle\,.$$

The amplitudes α, β, δ, d are complex numbers that are determined by the players' selected strategy given that $|\propto|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. The payoff P for each player is calculated as:

$$P_{def} = a|\propto|^2 + b|\beta|^2 + c|\gamma|^2 + d|\delta|^2$$

$$P_{adv} = w|\propto|^2 + x|\beta|^2 + y|\gamma|^2 + z|\delta|^2\,.$$

The payoff of both players is a classical mixed strategies where the defender chooses to use the QSVM classifier with a probability $[U_1]_{11}$ and use adversarial trained classical SVM with probability $[U_1]_{21}$. To obtain distinct payoff different from the classical, the qubits are entangled using and entanglement J operator before the manipulation by both parties after which the final state is given by $U_1 \otimes U_2 J|00\rangle$

$$|\psi'\rangle = CNOT.(U_1 \otimes U_2)|00\rangle$$

$$|\psi'\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} [U_1]_{11}\,[U_2]_{11} & \cdots & [U_1]_{12}\,[U_2]_{12} \\ \vdots & \ddots & \vdots \\ [U_1]_{21}\,[U_2]_{21} & \cdots & [U_1]_{22}\,[U_2]_{22} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \propto' \\ \beta' \\ \gamma' \\ \delta' \end{bmatrix}.$$

The resulting state $|\psi'\rangle$ is entangled, since the product of their outer probability amplitude of the vector is not equal to the product of the inner probability amplitude. The entanglement leads to correlation between both parties such that if the defender selects a strategy there are guaranteed that the adversarial will select a particular corresponding strategy without any prior communication between the players.

6.3 Experiment

We demonstrated the effectiveness of the quantum SVM kernel defense, by implementing a radial basis function gradient descent attack on MNIST and CIFAR-10 image datasets.

In our experiment we considered a classical SVM with RBF kernel and SVM with a quantum kernel, using two classes from both MNIST and CIFAR-10 dataset, one as benign data the other as adversarial data. The datasets were split into training and test sets, the dimensionality of each data was reduced to 8, using PCA to make them compatible with ZZ feature mapping of the quantum circuit. The RBF kernel function is used to compute the perturbation using the coefficient of the support vectors expressed in the equation to generate the respective adversarial sample. The number of steps in the direction of the deepest descent is controlled by k, which correlates to the strength of adversarial attack, the table below shows the accuracy of attack declines with increase value of k for both MNIST and CIFAR-10 datasets. Using adversarial training, the classical SVM model is trained to obtain an SVM model that is more robust to the attack. The accuracy results show some improvement compared to the natural model. For the QSVM, the adversarial dataset is reduced to 4 features using PCA corresponding to the number of qubits to be used in the experiment and normalized to between I and -1.

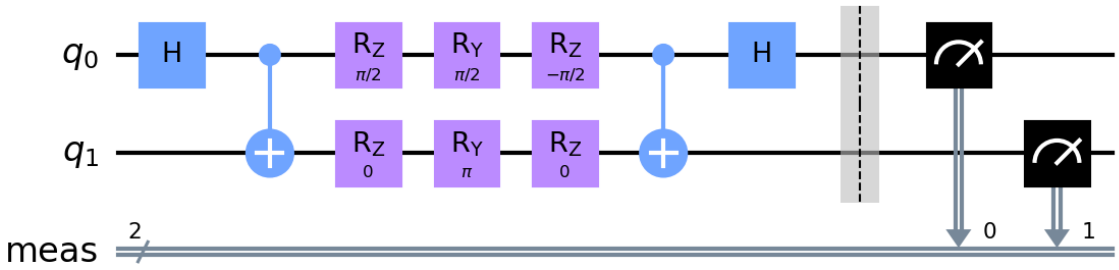

Figure 6. 5 Entanglement Operation for Quantum Game formulation

Using Qiskit, we encoded the adversarial data into the quantum state space by utilizing the quantum feature map. The ZZ feature maps the classical adversarial feature vector to the quantum state by applying a unitary operation $U_{\emptyset}$ on the initial qubit state $|0\rangle^n$ representing the encoded adversarial data shown in the circuit diagram. The quantum feature maps result in a quantum kernel which corresponds to the similarity measure for each pair of datapoints in the training adversarial data x_{adv} and y_{adv} we used the training and test kernel matrices in the classical SVM algorithm. The strength of the adversarial data x_{adv}

was varied in the adversarial attack to observe the robustness of the QSVM shown in table below.

Table 6. 2 Table showing the accuracy on of the advSVM and QSVM model after PGD adversarial using various value of k for FGSM dataset

k	AdvSVM	QSVM	d
25	0.81	0.7	0.1290
30	0.51	0.73	0.1489
35	0.51	0.73	0.1635
45	0.49	0.62	0.1871
100	0.51	0.61	0.2508
150	0.48	0.52	0.2651
200	0.44	0.51	0.2687
250	0.44	0.47	0.2702
300	0.46	0.49	0.2705

Table 6. 3 Table showing the accuracy of the advSVM and QSVM model after PGD adversarial using various value of k for CIFAR-10 dataset

k	AdvSVM	QSVM	d
25	0.72	0.74	0.1610
30	0.45	0.62	0.1523
35	0.44	0.59	0.1810
45	0.40	0.59	0.2012
100	0.40	0.55	0.2723
150	0.39	0.52	0.2946
200	0.35	0.51	0.3001
250	0.35	0.48	0.2901
300	0.32	0.47	0.3105

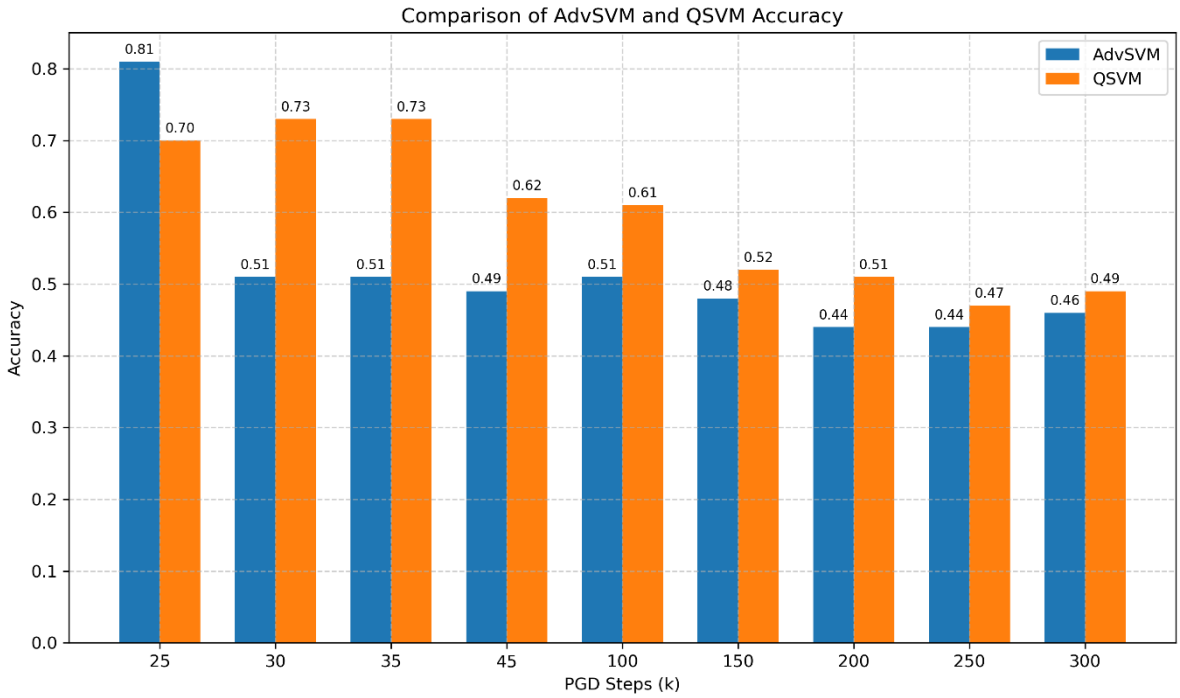

Figure 6. 6 Robust Accuracy for MNIST dataset on AdvSVM and QSVM under PGD Attack=[25,300]

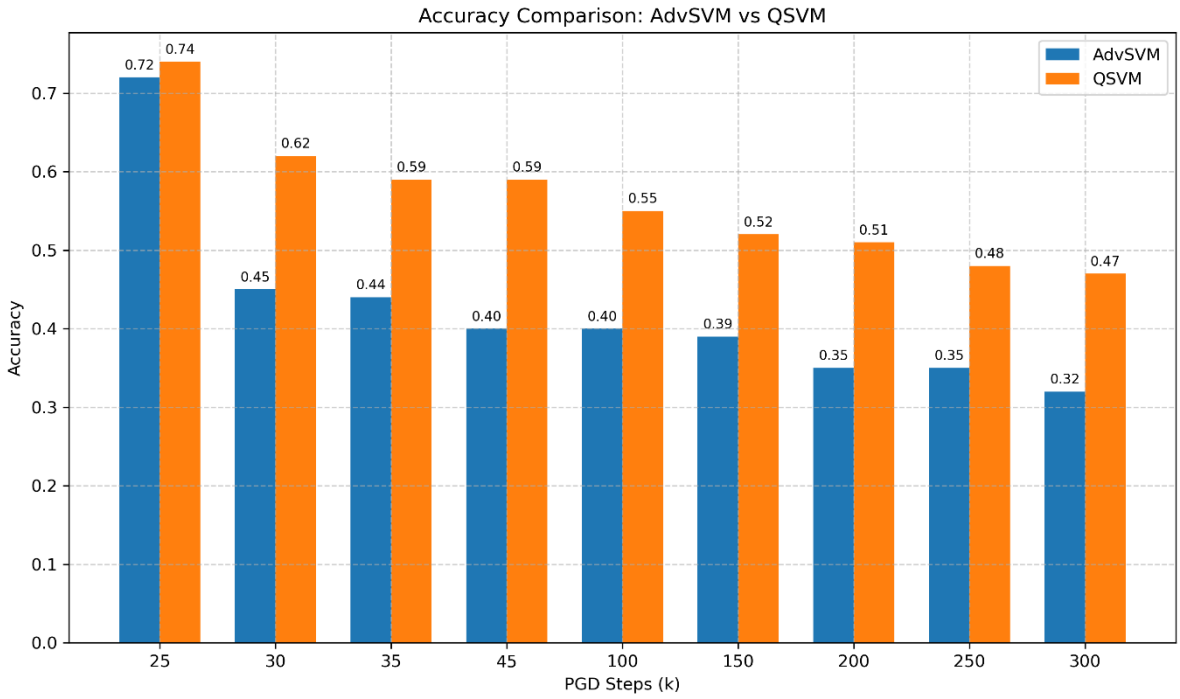

Figure 6. 7 Robust Accuracy forCIFAR-10dataset onAdvSVM and QSVM

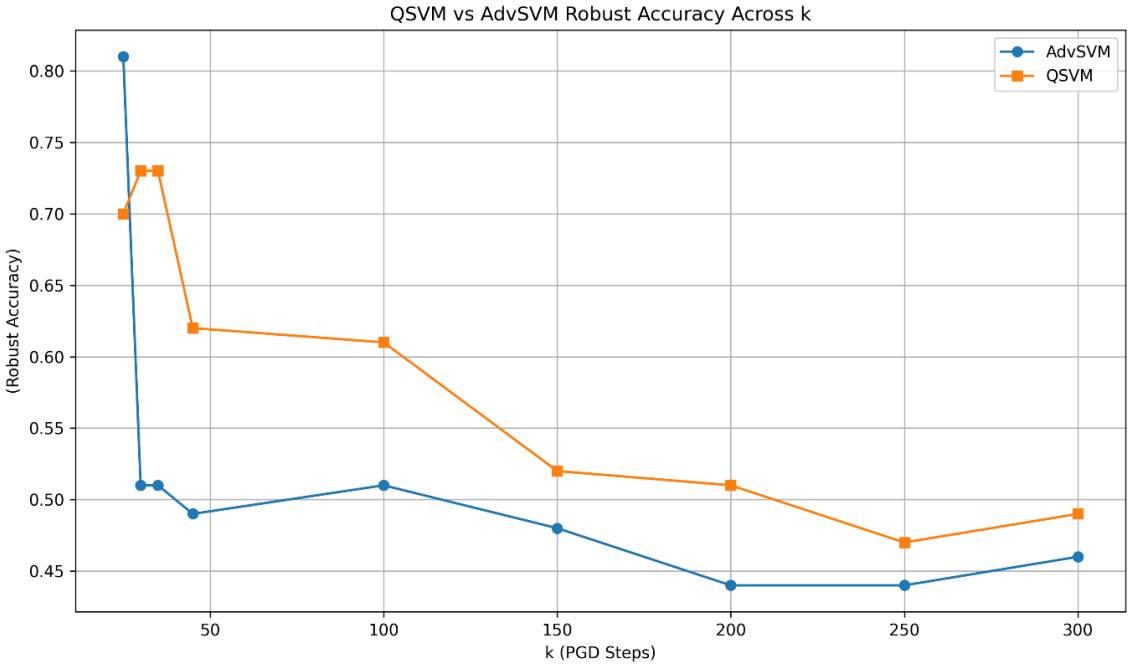

Figure 6. 8 Adversarial Robustness for AdvSVM and QVSM under PGD Attack

Using quantum games, we correlated the adversarial attack and defense strategies to ensure the defender selects the optimal defensive state for maximum accuracy. By varying the angle θ_i the strength of entanglement between both qubits varies, allowing higher payoffs through modifying the amplitudes affecting the final state probabilities.

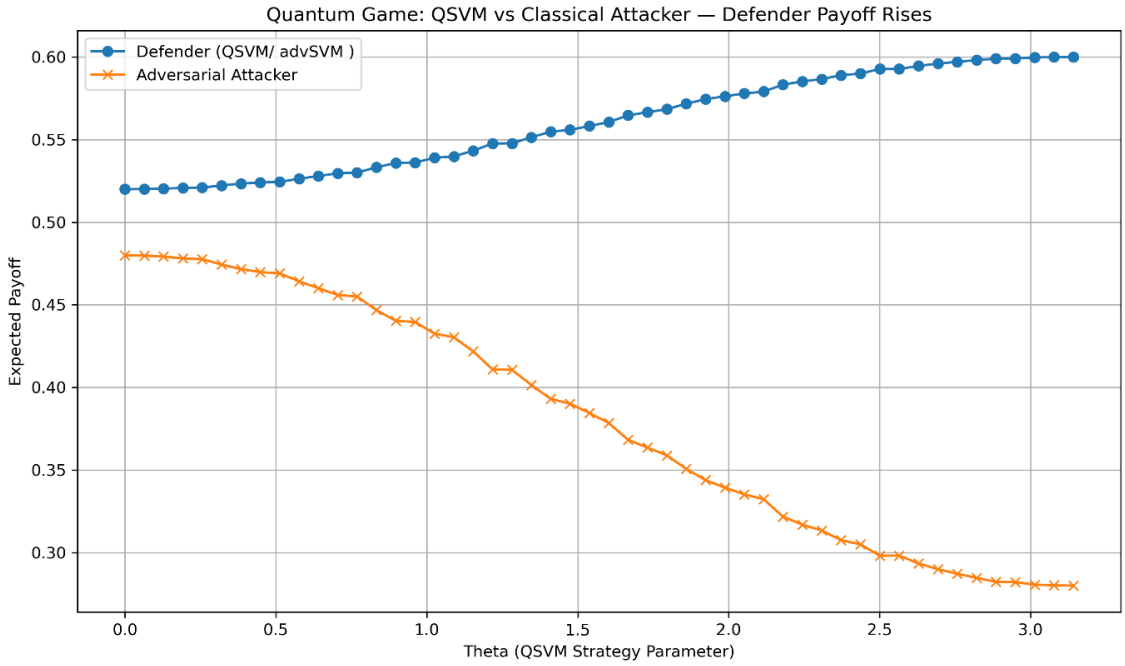

Figure 6. 9 Simulation of Payoff for Defender and Attacker in a Quantum Game Formulation

6.4 Discussion

From our experiment, we observed that the generated adversarial samples increase in strength as the PGD steps increase. The impact of the adversarial attack is observed in fig which shows the gradual reduction of accuracy of both models advSVM and QSVM model. However, the QSVM model showed more robustness to the same attack and its accuracy was consistently higher than those of the advSVM model for the MNIST and CIFAR-10 dataset. Hence, the QSVM model is more robust in the face of adversarial attack compared to an advsQSVM trained on adversarial data. We evaluated the efficacy of quantum enhanced strategies in the adversarial setting by simulating a quantum game

between a defender implementing a QSVM and advSVM against an gradient based adversarial attacker to obtain optimal payoff for dynamic interaction between both players. We used a 2-qubit quantum circuit each qubit representing the strategy of the players, qubit 0 for the defender and qubit 1 for the adversarial attacker. The initial state of the qubit $|00\rangle$ is entangled using a Hadamard and CNOT gate, followed by the application of quantum strategies parameterized by angles θ from 0 to π, with $\phi = \frac{\pi}{2}$. The attacker applies a fixed classical strategy with $\theta = \pi$ and $\phi = 0$. After the strategy is selected, an inverse entanglement operation $J^{\dagger}$ is applied before measurement. The simulation was iterated over 50 evenly spaced values of θ, and the expected payoffs for both players were calculated at each point. Results in Figure 3 show a clear payoff gradient favouring quantum strategies. The defender's strategy is parameterized by an angle θ , varied from to 0 to π . The attacker uses a fixed classical strategy with $\theta = \pi$. The circuit includes an entanglement step (Hadamard and CNOT), player strategies, and disentanglement before measurement.

For one iteration at $\theta = {}^{\pi}/_{2}$, the measured output was:

$$'00' : 320, \quad '01' : 180, \quad '10' : 140, \quad '11' : 384$$

Probabilities:

$P_{00} = 0.312,\ P_{01} = 0.176,\ P_{10} = 0.137, P_{11} = 0.375$

Payoff matrix:

$$Payoff(D_i, T_j) = \begin{cases} (0.62, 0.38)\ if\ (i,j) = (0,0) \\ (0.28, 0.60)\ if\ (i,j) = (0,1) \\ (0.60, 0.28)\ if\ (i,j) = (1,0) \\ (0.52,\ 0.48)\ if\ (i,j) = (1,1) \end{cases}$$

Expected payoffs:

$$P_D = 0.312 \times 0.62 + 0.176 \times 0.28 + 0.137 \times 0.60 + 0.375 \times 0.48 = 0.442$$

This shows that at $\theta = {}^{\pi}/_{2}$, the defender receives a higher payoff. The entangled quantum strategy allows the QSVM to outperform the classical strategy under the defined

matrix, which reflects asymmetric outcomes for cooperative and defective behavior. Repeating the simulation for different values of θ reveals that the payoff improves as the defender shifts from classical to quantum strategies. This escape from classical equilibrium traps highlights the strategic advantage of quantum correlations in adversarial scenarios.

6.5 Findings Summary

The study demonstrated the robustness and advantage of QSVM in adversarial settings. Evaluating adversarial perturbed samples for MNIST and CIFAR-10 datasets, the QSVM consistently outperformed its adversarial trained advSVM counterpart, maintaining higher accuracy with increasing perturbation strength. Furthermore, by modelling the interaction between a defender and an attacker, where the defender has 2 strategies namely deploying QSVM and advsm, the quantum enhances strategies enable the defender to escape and achieve higher payoffs. These findings underscore the potential of quantum learning models to adaptively and optimally respond to adversarial threats, establishing QSVM as a resilient defense paradigm in defending against adversarial attacks. The improved robustness observed after adversarial retraining supports the hypothesis that weighted Stackelberg reinforcement enhances resilience by emphasizing high-risk samples during optimization. However, the slight drop in clean-data accuracy confirms the classical robustness–generalization trade-off.

Models such as WARS and QSVM demonstrate that integrating game-theoretic and quantum principles can mitigate this trade-off by redistributing learning focus adaptively. This behaviour aligns with theoretical expectations of mixed-strategy equilibria, where probabilistic defences yield improved average-case robustness against heterogeneous adversaries

Chapter 7

Conclusion and Future Work

7.1 Research Summary

In this paper, we present an adversarial training method formulated as a Weighted Adversarial Stackelberg game, developed to enhance the robustness of MobileNet CNN models. Our method incorporates both a strategic game-theoretic framework and reinforcement learning. We demonstrate how the Stackelberg equilibrium contributes to reducing MobileNet's vulnerability to adversarial perturbations. We further improve this defense mechanism by integrating the SARSA reinforcement learning algorithm, which enables the model to adaptively refine its parameters during training. The combination of the Stackelberg game and SARSA algorithm strengthens MobileNet's ability to handle input manipulations and extends the model's applicability to other CNN architectures.

Our Stackelberg game formulation focuses on assigning different weights to clean and adversarial samples during model training. These weights are adjusted to give higher emphasis to adversarial data during testing. This prioritization effectively minimizes the misclassification rate caused by adversarial interference. Through this setup, we are able to derive a deterministic strategy model with learning parameters optimized using the Stackelberg equilibrium conditions. The result is a MobileNet variant that exhibits better generalization and maintains classification accuracy in the presence of both targeted and untargeted attacks.

Beyond MobileNet, we validate our method across other CNN models, demonstrating consistent improvements in robustness. This suggests that the proposed framework can serve as a general defense approach in adversarial learning. Our contributions also

highlight the potential for combining reinforcement learning with game-theoretic strategies to build adaptable, scalable, and attack-resilient deep learning systems.

We develop a Bayesian Stackelberg game framework in which the defender, modeled as a machine learning classifier, optimizes its strategy in response to multiple intelligent adversaries. The defender does not have complete knowledge of the specific type of adversary it will face but holds prior beliefs about the distribution of possible adversary types. Each adversary, in turn, has access to a range of attack strategies. This setting reflects real-world security scenarios where adversaries are diverse and unpredictable.

Traditional adversarial learning approaches often focus on defending against a single, static type of attacker with a fixed strategy and payoff function. In contrast, our framework accounts for uncertainty and variability in adversarial behavior. Using a nested Stackelberg game structure, the defender first models' adversarial transformations on input data and then searches for an optimal mixed strategy in a single-leader, multi-follower setting. This allows the defender to prepare for a wider spectrum of potential attacks. We evaluate different attack and defense strategies, including adversarial training. In our experiments, models trained using the FGSMR method achieved higher accuracy on clean MNIST data compared to those trained with PGD, while also requiring less training time per epoch (Tianjin H. et al., 2020).

To further enhance the framework, we introduce a variant of the Bayesian Stackelberg game where the defender, as the leader, directly anticipates and responds to the expected behavior of adversaries. Unlike classical game-theoretic models that assume perfect information or rely on equilibrium-based solutions, we incorporate uncertainty in the defender's knowledge of the environment. Our contribution includes the use of the Decomposed Optimal Bayesian Stackelberg Solver (DOBSS), which allows the defender to compute an optimal leader strategy without requiring a full Nash equilibrium. This method identifies high-reward, non-equilibrium strategies by solving a single mixed-integer linear program, thereby improving computational efficiency (Paruchuri, 2008; Zhou et al., 2016).

The model enables the defender to compute a mixed strategy that maximizes expected payoff against a distribution of possible attackers. This is especially relevant in real-world environments where adversaries vary in behavior, knowledge, and goals.

The formulation of this Bayesian Stackelberg game involves solving the payoff matrices associated with different defender and adversary strategies, using a probabilistic framework to reflect uncertainty. We derive a mixed strategy solution that allows the defender to switch probabilistically between CNN models or decision rules, depending on the type of attacker encountered. This approach increases resilience to both known and unseen adversarial tactics. Our results demonstrate that the mixed strategy approach derived from the Bayesian Stackelberg equilibrium provides more robust performance in classification tasks under adversarial conditions. It enables better generalization across different attack profiles while minimizing overall classification error.

The integrated PGD is an efficient generator of adversarial samples compared to other methods on CIFAR, MNSIT and ImageNet dataset [49]. Using the PGD attack for each of the distance metric, more adversarial samples were obtained compared to other state of the arts methods. The l_0 and l_2 attack found adversarial samples with lower distortion than the other previously published attack methods and performed with a 100% success. The l_∞ attacks with the PGD method have a higher success rate and quality compared to previous works. For instance, on the ImageNet dataset l_∞ attacks have high attacks success rate that by only flipping the lowest bit of each pixel one can change the classification of an output label without changing the visual perception of the output image. Hence, the PGD is a better baseline for developing an effective adversarial sample to be integrated with the JND image masking algorithm for an overall improved attack sample. Furthermore, JND generated images can successfully fool an already trained Inception v3 image classifier as well as a RetinaNet object detector, results from experiments show that JND adversarial images have higher quality compared to state-of-the-art generators especially when input images with high resolutions are used. Therefore, our combination of the JND and PGD algorithm as a method to develop adversarial samples with improved quality will result to an efficient and less time-consuming process of adversarial attack generation.

The study demonstrated the robustness and effectiveness of quantum support vector machines (QSVM) when subjected to adversarial scenarios. Using adversarially perturbed samples from the MNIST and CIFAR-10 datasets, QSVM consistently achieved higher classification accuracy than the adversarially trained classical support vector machine (advSVM), particularly as the strength of the perturbations increased. The QSVM maintained performance across various perturbation magnitudes, indicating its ability to preserve decision boundaries and withstand input manipulations that degrade classical models.

The analysis modeled the interaction between a defender and an attacker, where the defender selected between two strategies using QSVM or using an adversarially trained SVM. The attacker employed gradient-based perturbations. Results showed that quantum-enhanced strategies enabled the defender to break free from suboptimal equilibrium positions common in classical frameworks, achieving higher payoff in a game-theoretic context. This advantage stemmed from the entangled nature of the strategies in the quantum domain, which allowed the defender to influence the attacker's outcome without explicit coordination.

Overall, the findings support the view that quantum learning models are capable of adaptively responding to adversarial threats with enhanced resilience. The QSVM approach offers a path toward more secure and reliable classification systems in adversarial environments. By exploiting quantum feature spaces and leveraging entanglement, QSVM establishes itself as a viable defense paradigm for machine learning applications exposed to adversarial interference.

7.2 Future work

Future work will explore applying the proposed Stackelberg reinforcement learning framework to transformer-based models and evaluating its effectiveness beyond CNNs. The approach can be extended to online learning scenarios where the defender adapts to changing adversarial behavior in real time. Incorporating this framework into federated learning setups will allow multiple agents to train collaboratively while handling adversarial threats without centralized data sharing. Future studies can also investigate

integrating quantum classifiers into the Stackelberg model to assess the impact on model performance under adversarial conditions. Additional work will examine how the framework performs against perceptually tuned adversarial examples generated using methods like PGD combined with JND masking.

Although the results demonstrate promising robustness improvements, real-world deployment presents practical challenges. Quantum hardware still faces qubit-decoherence and scalability issues that limit large-scale quantum adversarial training. Similarly, adversarially retrained models on edge devices may encounter energy and memory constraints, requiring lightweight adaptations.

Moreover, integration into safety-critical domains such as healthcare and autonomous systems introduces regulatory, privacy, and interpretability requirements. Addressing these socio-technical considerations will be essential for transitioning adversarially robust quantum models from laboratory environments to operational deployment.

A practical limitation of adversarial training is its computational cost, particularly for embedded or edge systems where processing and power resources are constrained. Each PGD iteration requires multiple gradient computations, resulting in quadratic growth in training time.

Optimization strategies include:
• Gradient approximation – using single-step or stochastic perturbations to approximate PGD;
• Quantization-aware training – reducing precision to lower memory and computation without compromising robustness;
• Model pruning and knowledge distillation – transferring adversarial robustness from large to lightweight models; and
• Quantum-inspired parameter optimization, which can exploit quantum parallelism for faster convergence.

Combining these methods can substantially reduce complexity while maintaining robust performance suitable for edge deployment. Testing the method across domains such as autonomous systems, medical data, and cybersecurity will help validate its

generalizability. Finally, integrating multi-objective optimization into the Stackelberg solver could improve its ability to balance model performance, robustness, and resource usage during training.

References

[1] H. Quadri, B. Gu, H. Wang and S. Islam, "Advancing MobileNet Security: Weighted Adversarial Learning in Convolutional Neural Networks," *11th International Conference on Machine Intelligence Theory and Applications (MiTA),* 2024.

[2] H. Quadri, B. Gu, H. Wang and S. Islam, "Mixed Bayesian Stackelberg Strategies for Robust Adversarial Classifiers," *EAI Endorsed Transactions on Scalable Information Systems,* vol. 12, no. 1, pp. 1-10, 2025.

[3] H. Wang, Y. Zhang, J. Cao and V. Varadharajan, "Achieving Secure and Flexible M-Services through Tickets," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 33, pp. 697 - 708, 2003.

[4] H. Wang, Y. Zhang and J. Cao, "Effective Collaboration with Information Sharing in Virtual Universities," *IEEE Trans. Knowl. Data Eng.,* vol. 21, pp. 840-853, 2009.

[5] Y. Wang, Y. Shen, H. Wang, J. Cao and X. Jiang, "Mtmr: Ensuring mapreduce computation integrity with merkle tree-based verifications," *IEEE Transactions on Big Data,* vol. 4, pp. 418--431, 2016.

[6] Y.-F. Ge, E. Bertino, H. Wang, J. Cao and Y. Zhang, "Distributed Cooperative Coevolution of Data Publishing Privacy and Transparency," *ACM Transactions on Knowledge Discovery from Data,* vol. 18, no. 10.1145/3613962, 2023.

[7] Y.-F. Ge, M. Orlowska, J. Cao, H. Wang and Y. Zhang, "MDDE: multitasking distributed differential evolution for privacy-preserving database fragmentation," *The VLDB Journal,* vol. 31, pp. 1-19, 2022.

[8] Y.-F. Ge, H. Wang, E. Bertino, Z.-H. Zhan, J. Cao, Y. Zhang and J. Zhang, "Evolutionary Dynamic Database Partitioning Optimization for Privacy and Utility," *IEEE Transactions on Dependable and Secure Computing,* pp. 1-17, 2023.

[9] Y.-F. Ge, H. Wang, E. Bertino, J. Cao and Y. Zhang., "Multiobjective Privacy-Preserving Task Assignment in Spatial Crowdsourcing," in *IEEE Transactions on Cybernetics (2025).*, 2025.

[10] Y.-F. Ge, W.-J. Yu, J. Cao, H. Wang, Z.-H. Zhan, Y. Zhang and J. Zhang, "Distributed Memetic Algorithm for Outsourced Database Fragmentation," *IEEE Transactions on Cybernetics,* vol. PP, pp. 1-14, 2020.

[11] X. Sun, H. Wang, J. Li and Y. Zhang, "Injecting purpose and trust into data anonymisation," in *Computers Security*, 2011.

[12] X. Sun, M. Li, H. Wang and A. Plank., "An efficient hash-based algorithm for minimal k-anonymity.," in *In Conferences in research and practice in information technology (CRPIT), vol. 74, pp. 101-107. 2008.*, 2008.

[13] X. Sun, H. Wang, J. Li and Y. Zhang, "Satisfying privacy requirements before data anonymization," in *The Computer Journal 55, no. 4 (2012): 422-437*, 2012.

[14] J. Yin, W. Hong, H. Wang, J. Cao, Y. Miao and Y. Zhang, "A compact vulnerability knowledge graph for risk assessment," in *ACM Transactions on Knowledge Discovery from Data 18, no. 8 (2024): 1-17*, 2024.

[15] L. Chen, W. Liu, H. Wang, S.-W. Jeon, Y. Jiang and Z. Zheng., "Consistency-guided adaptive alternating training for semi-supervised salient object detection," in *IEEE Transactions on Circuits and Systems for Video Technology (2025).*, 2025.

[16] S. Jahan, Y.-F. Ge, H. Wang and E. Kabir, "Adaptive-parameter memetic algorithm for privacy-preserving trajectory data publishing: A multi-objective optimization approach:," in *Computing 107, no. 7 (2025): 151.*, 2025.

[17] A. M. Alvi, M. J. Khan, N. T. Manami, Z. A. Miazi, K. Wang, S. Siuly and H. Wang, "XCR-Net: a computer aided framework to detect COVID-19.," in *EEE Transactions on Consumer Electronics (2024). doi: 10.1109/TCE.2024.3446793*, 2024.

[18] A. Athalye, N. Carlini and D. Wagner, Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples, arXiv preprint arXiv:1802.00420, 2018.

[19] A. Chivukula, X. Yang, W. Liu, T. Zhu and W. Zhou, "Game Theoretical Adversarial Deep Learning with Variational Adversaries," *IEEE Transactions on Knowledge and Data Engineering,* pp. 3568-3581, 2020.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, Towards Deep Learning Models Resistant to Adversarial Attacks, arXiv preprint arXiv:1706.06083, 2017.

[21] A. N. Bhagoji, W. He, B. Li and D. Song, "Practical Black-Box Attacks on Deep Neural Networks Using Efficient Query Mechanisms," *Computer Vision – ECCV 2018 : 15th European Conference, Munich, Germany Proceedings,* vol. 11216, pp. 158-174 (17p), 2022.

[22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *In International Conference on Learning Representations, Vancouver, BC, Canada,* 2018.

[23] M. Amini and A. Lechner, "Revisiting the adversarial robustness accuracy," *IEEE Robotics and Automation Letters, Robotics and Automation Letters, IEEE,* vol. 8, no. 3, pp. 1595-1602, 2023.

[24] A. Mathias and L. Alexander, "Revisiting the Adversarial Robustness-Accuracy Tradeoff in Robot," *IEEE Robotics and Automation Letters, Robotics and Automation Letters,* vol. 8, no. 3, pp. 1595-1602 (8p), 2022.

[25] A. Athalye, N. Carlini and D. Wagner, Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples, arXiv preprint arXiv:1802.00420, 2018.

[26] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. ˇSrndi´c, P. Laskov, G. Giacinto and F. Roli, "Evasion attacks against machine learning at test time," *In Joint European conference on machine learning and knowledge discovery in databases,* no. Springer, 2013, p. 387–402, 2013.

[27] B. Wu, Y. Lin, Q. Song and M. Liu, "Attacks in Adversarial Machine Learning: A Systematic Survey from the Life-cycle Perspective," in *arXiv preprint arXiv:2302.09457*, 2023.

[28] B. Yisen, W. Xingjun and M. James, "On the Convergence and Robustness of Adversarial Training," *arXiv:2112.08304v2,* 2021.

[29] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto and F. Roli, Pattern Recognition Systems under Attack: Design Issues and Research Challenges, International Journal of Pattern Recognition and Artificial Intelligence, 2014.

[30] A. Shafahi et al, "Adversarial Training for Free!," *NeurIPS,* vol. 32, pp. 3358-3369, 2022.

[31] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *preprint arXiv:1706.06083*, 20170619.

[32] J. Sathya and M. P. Chitra, "Frequency-Selective Adversarial Attack in WSN Detected using Deep Learning Algorithms," *3rd International Conference on*

Sustainable Computing and Data Communication Systems (ICSCDS), pp. 1870-1874 (5p), 20250806.

[33] S. H. Ahmed, M. Abdulaziz Bamadhaf and A. Mehmood, "Adversarial Attacks on Machine Learning Models in Cybersecurity," *International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA),* pp. 1-6 (6p), 20250807.

[34] H. Biecek and B. P., "Adversarial Attacks and Defenses in Explainable Artificial Intelligence: A Survey," in *arXiv preprint arXiv:2306.06123*, 2023.

[35] B. Biggio, B. Nelson and P. Laskov, "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," in *Pattern Recognition, 84, pp. 317–331*, 2018.

[36] J. Bose and G. Gidel, "Adversarial example games," *Advances in neural information processing systems,* vol. 33, pp. 8921-8934, 2020.

[37] S. C. Benjamin and P. M. Hayden, "Multi-player quantum games," *Physical Review A,* vol. 64, p. 030301, 2001.

[38] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. Ghaoui and J. MI, "Theoretically principled trade-off between robustness and accuracy," *AcoInternational Conference on Machine Learning (ICML),* no. 190108573. 2019b, 2019.

[39] C. Xie, Y. Wu, L. v. d. Maaten, A. Yuille and K. He, "Feature Denoising for Improving Adversarial Robustness," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 501-509 (9p), 2019.

[40] C. Zhai, R. Lin and J. Zhang, "Adversarial Robustness of Quantum Neural Networks: Challenges and Perspectives," in *IEEE Trans. Quantum Eng*, 2023.

[41] C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille and Q. V. Le, "Adversarial Examples Improve Image Recognition," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 816-825 (10p), 2020.

[42] H. Hwang, S.-H. Kim, M. Cha, M.-H. Choi, K. Lee and H.-J. Lee, "Analysis of the Effect of Feature Denoising from the Perspective of Corruption Robustness," *International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC),* Vols. 1-6, pp. 20230625, (6p), 2023.

[43] D. Dietterich and T. Hendrycks, Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, arXiv preprint arXiv:1903.12261, 2019.

[44] X. Li, Y. Chen, Y. Zhu, S. Wang, R. Zhang and H. Xue, "ImageNet-E: Benchmarking Neural Network Robustness via Attribute Editing," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 20371-20381 (11p), 2023.

[45] C. Zhang, F. Pan, J. Kim, I. S. Kweon and C. Mao, "ImageNet-D: Benchmarking Neural Network Robustness on Diffusion Synthetic Object," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 21752-21762 (11p), 20240616.

[46] L. Dritsoula, P. Loiseau and J. Musacchio, "A game-theoretical approach for finding optimal strategies in an intruder classification game," *51st IEEE Conference on Decision and Control (CDC),* pp. 7744-7751 (8p), 2012.

[47] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd, Quantum machine learning, Nature, 549(7671), 195–202.

[48] G. Chen, M. Wang, S. Han, J. Yin, H. Wang and J. Cao, "Deep Reinforcement Learning-Based Cloud-Edge Offloading for WBANs," *IEEE Transactions on Consumer Electronics,* no. 10.1109/TCE.2024.3504545, pp. 4053 - 4064, 2024.

[49] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[50] E. Rice and W. Leslie, "Fast is better than free: Revisiting adversarial training," *arXiv:2001.03994v1,* 2020.

[51] S. Reza, V. Shmatikov, C. Song and M. Stronati, "Membership Inference Attacks Against Machine Learning Models," *proceedings of the IEEE Symposium on Security and Privacy, 2017,* vol. arXiv:1610.05820, no. 10.48550, 2017.

[52] R. S. S. Kumar, M. Nystrom, J. Lambert and A. Marshall, "Adversarial Machine Learning-Industry Perspectives," *IEEE Security and Privacy Workshops (SPW),* vol. 3, no. arXiv:2002.05646, 2020.

[53] R. Shokri, M. Stronati, C. Song and V. Shmatikov, Membership Inference Attacks Against Machine Learning Models, IEEE Symposium on Security and Privacy (SP), 2017.

[54] M. K. Y Vorobeychik, Adversarial machine learning, Morgan & Claypool Publishers, 2018.

[55] A. S. Chivukula and L. Wei, Adversarial Deep Learning Models with Multiple Adversaries, IEEE Transactions on Knowledge and Data Engineering, 2019.

[56] V. Mnih, K. Kavukcuoglu and D. Silver, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, no. 7549, pp. 529-533 (5p), 2015.

[57] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction (2nd ed., MIT Press, 2018.

[58] B. Li, V. François-Lavet, T. Doan and J. Pineau, "Domain Adversarial Reinforcement Learning," in *arXiv:2102.07097v1 Machine Learning (cs.LG); Artificial Intelligence (cs.AI)*, 2021.

[59] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv preprint arXiv:1412.6572,* 2015.

[60] Y. Zhou, M. Kantarcioglu, B. Thuraisingham and B. Xi, "Adversarial support vector machine learning," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data minin*, 2012.

[61] S. M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[62] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, Towards Deep Learning Models Resistant to Adversarial Attacks, arXiv preprint arXiv:1706.06083, 2017.

[63] M. T. West, E. R. Peterson and K. Temme, "Towards Quantum Enhanced Adversarial Robustness in Machine Learning," in *arXiv preprint arXiv:2306.12688*, 2023.

[64] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, F. Tramèr and C. Malossini, High-Dimensional, Model-Agnostic, Targeted Attacks on Machine Learning, Proceedings of the 37th International Conference on Machine Learning (ICML) pp. 4682–4691, 2020.

[65] S. Lu and S. L. Braunstein, Quantum decision tree classifier, Quantum Information Processing, 2014.

[66] I. J. Goodfellow, "Explaining and Harnessing Adversarial Examples," in *arXiv preprint arXiv:1412.6572*, 2014.

[67] J. Ren, D. Zhang, Y. Wang, L. Chen, Z. Zhou, Y. Chen, X. Cheng, X. Wang, M. Zhou, J. Shi and Q. Zhang, "Towards a Unified Game-Theoretic View of Adversarial

Perturbations and Robustness," in *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021.

[68] J. Su, D. V. Vargas and K. Sakurai, One Pixel Attack for Fooling Deep Neural Networks, IEEE Transactions on Evolutionary Computation, 2019.

[69] M. Staib and S. Jegelka, "Distributionally Robust Optimization and Generalization in Kernel Methods," *Advances in Neural Information Processing Systems,* vol. 35, pp. 10815-10823, 2019.

[70] A. Dong and S. Jegelka, "Efficient Adversarial Training via Differentiated Input Transformations," in *Advances in Neural Information Processing Systems*, 2023.

[71] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin and N. Papernot, "High-Dimensional, Model-Agnostic, Targeted Attacks on Machine Learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML), 4682–4691. https://proceedings.mlr.press/v119/jagielski20a.html*, 2020.

[72] J. Su, D. V. Vargas and K. Sakurai, One Pixel Attack for Fooling Deep Neural Networks, IEEE Transactions on Evolutionary Computation, 2019.

[73] J. Yin, M. Tang, J. Cao and H. Wang, "Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description," in *Knowledge-Based Systems*, 2020.

[74] A. Kehoe, P. Wittek, Y. Xue and Pozas-Kerstjens, Defence against adversarial attacks using classical and quantum enhanced boltzmann machines, Machine Learning: Science and Technology, 2021.

[75] A. Kurakin, I. Goodfellow and S. Bengio, "Adversarial machine learning at scale," 2017.

[76] L. Wang et al, "Ensemble Adversarial Training via Knowledge Transfer," in *IEEE Trans. Neural Netw. Learn. Syst*, 2024.

[77] T. Chang, W. Liu and D. Huang, "Research on Adversarial Sample Defense Method Based on Image Denoising," *6th International Conference on Computer Vision, Image and Deep Learning (CVIDL),* pp. 445-450 (6p), 2025.

[78] M. Ledoux, The concentration of measure phenomenon, American Mathematical Soc, 2001.

[79] Y.-C. Lin, M.-Y. Liu and M. Sun, "Detecting Adversarial Attacks on Neural Network Policies with Visual Foresight," *arXiv,* 2017.

[80] Y.-F. Chen, W.-Y. Shih, H.-C. Lai, H.-C. Chang and J.-L. Huang, "Pairs Trading Strategy Optimization Using Proximal Policy Optimization Algorithms," *IEEE International Conference on Big Data and Smart Computing (BigComp),* pp. 40-47 (8p), 2023.

[81] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *arXiv preprint arXiv:1409.1556*, 2014.

[82] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE CVPR, 770–778.*, 2016.

[83] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv,* 2017.

[84] J. Deng, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE CVPR, 248–255*, 2009.

[85] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572,* 2015.

[86] Y. Zhang, Y. Li and B. Gong, "A Survey on Adversarial Attacks and Defenses in Support Vector Machines," in *ACM Computing Surveys, 53(3), 1–26*, 2020.

[87] S. Liang, H. Li and K. Ren, "Enhancing Adversarial Robustness of SVM via Joint Margin Maximization," in *Proceedings of the AAAI Conference on Artificial Intelligence, 36(1), 777–785. https://doi.org/10.1609/aaai.v36i1.19983*, 2022.

[88] C. Xiao, B. Li, J. Zhu, W. He, M. Liu and D. Song, "Adversarial Examples for SVMs and Tree Ensembles," in *Proceedings of the 27th ACM Conference on Computer and Communications Security (CCS), 628–644*, 2019.

[89] X. Li, J. Wu and C. Xu, "Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics," in *Proceedings of the IEEE ICCV, 5764–5772. https://doi.org/10.1109/ICCV.2019.00586*, 2019.

[90] E. J. Candès, X. Li, Y. Ma and J. Wright, "Robust Principal Component Analysis?," in *Journal of the ACM, 58(3), 1–37. https://doi.org/10.1145/1970392.1970395*, 2011.

[91] W. Xu, D. Evans and Y. Qi, "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," in *Proceedings of the Network and Distributed System Security Symposium (NDSS). https://doi.org/10.14722/ndss.2018.23198*, 2018.

[92] L. Banchi, J. Pereira and S. Pirandola, Generalization in Quantum Machine Learning: A Quantum Information Standpoint, PRX Quantum, 2021.

[93] M. Schuld and F. Petruccione, Supervised Learning with Quantum Computers, Springer, 2018.

[94] H. Y. Huang, R. Kueng and J. Preskill, Information-Theoretic Bounds on Quantum Advantage in Machine Learning, Nature Physics, 17(7), 864–869. https://doi.org/10.1038/s41567-021-01287-z, 2021.

[95] D. Ristè, M. da Silva, C. Ryan, A. Cross, A. Córcoles, J. Smolin, J. Gambetta, J. Chow and Johnson, Demonstration of quantum advantage in machine learning, Quantum Information, 2017.

[96] J. Romero, J. P. Olson and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, Quantum Science and Technology, 2017.

[97] W. Ren, Experimental quantum adversarial learning with programmable superconducting qubits, Nature Computational Science, 2022.

[98] X. Yuan, P. He, Q. Zhu and X. L, Adversarial Examples: Attacks and Defenses for Deep Learning, IEEE Transactions on Neural Networks and Learning Systems, 2019.

[99] X. Yuan, "Adversarial Examples: Attacks and Defenses for Deep Learning," in *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[100] Y. Dong, "Boosting Adversarial Attacks with Momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[101] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu and J. Zhu, Boosting Adversarial Attacks with Momentum, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[102] Yulong Wang et al., "Adversarial Attacks and Defenses in Machine Learning-Powered Networks: A Contemporary Survey," in *arXiv preprint arXiv:2303.06302*, 2023.

[103] H. Zeng, Zhu and C. Goldstein, "Are Adversarial Examples Created Equal? A Learnable Weighted Minimax Risk for Robustness under Non-uniform Attacks," *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 35, pp. 10815-10823, 2021.

[104] M. Caro, E. Gil-Fuster, J. Meyer, J. Eisert and R. Sweke, Encoding-dependent generalization bounds for parametrized quantum circuits, arXiv, Quantum, 5, 582, 2021.

[105] M. Caro, H. Huang, M. Cerezo, K. Sharma, A. Sornborger, C. L. and P. Coles, Generalization in quantum machine learning from few training data, Nat. Comm, 2022.

[106] J. Biamonte, P. Wittek, N. Pancotti and P. Rebentrost, Quantum machine learning., Nature, 2017.

[107] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, Physical Review, 2018.

[108] H. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng and Preskill, Quantum advantage in learning from experiments., Science, 376, pp. 1182–1186, 2022.

[109] A. Iqbal, W. Kong, Y. Iqbal, U. S. Khan, S. F. Khan and I. Hussain, "Enhancing fast adversarial training with momentum-driven initialization and max-norm regularization for robust deep learning models," *The Visual Computer: International Journal of Computer Graphics,* vol. 41, no. 13. Springer Nature Journals, pp. 11389-11406 (18p), 2025.

[110] B. Ji, M. Liu and S. Mumtaz, "Optimization of Meta-Learning Algorithms Based on Adversarial Training in Vehicular and Mobile Communications," *IEEE Transactions on Vehicular Technology, Vehicular Technology,* vol. 74, no. 10, pp. 16169-16177 (9p), 2025.

[111] K. Keeratitanankul, E. Pacharawongsakda and D. Jitkongchuen, "Deep Learning with Adversarial Training for Credit Scoring," *6th International Conference on Big Data Analytics and Practices (IBDAP),* pp. 214-220 (7p), 2025.

[112] B. Lyu and Z. Zhu, "Analyzing the Implicit Bias of Adversarial Training From a Generalized Margin Perspective," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 47, no. 9, pp. 8025-8039 (15p), 2025.

[113] B. Tong, H. Lai, Y. Pan and J. Yin, "On the Zero-shot Adversarial Robustness of Vision-Language Models: A Truly Zero-shot and Training-free Approach," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 19921-19930 (10p), 2025.

[114] L. Yan, Q. Zhu and X. Zhai, "Federated Adversarial Defense with Adversarial Training and Personalized Evaluation," *2nd International Conference on Digital*

Media, Communication and Information Systems (DMCIS), pp. 121-124 (4p), 2025.

[115] Y.-P. Hsieh, C. Liu and V. Cevher, Finding Mixed Nash Equilibria of Generative Adversarial Networks, Machine Learning (cs.LG); Computer Science and Game Theory (cs.GT); Machine Learning (stat.ML), 2018.

[116] D. Angioni, L. Demetrio and M. Pintor, "Robustness-Congruent Adversarial Training for Secure Machine Learning Model Updates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 47, no. 9, pp. 7457-7469 (13p), 2025.

[117] F. N. Azizah, A. Sulistya and B. R. Lidiawaty, "A Comparative Evaluation of SMOTE Placement Strategies for Multi-Class Traffic Complaint Classification Using SVM," *International Electronics Symposium (IES), Electronics Symposium (IES),* pp. 599-606 (8p), 2025.

[118] L. P. B, K. Parashar and P. T, "An Interpretable and Scalable Diabetes Detection Model Leveraging GWO-Optimized Features and SVM with Advanced Data Preprocessing," *5th International Conference on Soft Computing for Security Applications (ICSCSA),* pp. 871-878 (8p), 2025.

[119] A. Bamdad, A. Owfi and F. Afghah, "Adaptive Meta-learning-based Adversarial Training for Robust Automatic Modulation Classification," *2025 IEEE International Conference on Communications Workshops (ICC Workshops),* pp. 292-297 (6p), 2025.

[120] S. Ahmed, M. Al-Imran and S. I. B. Ameer, "Optimized Ensemble Architecture Integrating LSTM, FFNN, and SVM for Binary Prediction Tasks," *6th International Conference on Big Data Analytics and Practices (IBDAP),* vol. (6p), pp. 233-238, 2025.

[121] W. Huang, X. Tian, D. Deng, P. Ma and B. Song, "Cross-Devices Side-Channel Attacks: An Attack Paradigm Based on Adversarial Training," *5th International Conference on Machine Learning and Intelligent Systems Engineering (MLISE),* pp. 166-170 (5p), 2025.

[122] C. Zeng, B. Ren, H. Liu and J. Chen, "Applying the Bayesian Stackelberg Active Deception Game for Securing Infrastructure Networks," *Entropy,* p. 909, 2019.

[123] T. Wu, Z. Xin and S. Chen, "Adversarial Feature Training for Few-Shot Object Detection," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 35, no. 9 , pp. 9324-9336 (13p), 2025.

[124] P. Gangwani, J. Soni and A. Almonte, "Adversarial AI Training to Mitigate Cyber Attacks," *IEEE International Conference on Artificial Intelligence Testing (AITest),* pp. 54-61 (8p), 2025.

[125] M. Gao, "DSCAT: Dual Smoothing-constrained Adaptive Step Adversarial Training," *8th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE),* pp. 720-724 (5p), 2025.

[126] N. Geng, Z. Bai, X. Song and Y. Song, "Multi-Source Domain Adaptive Adversarial Training Network Emotion Recognition," *37th Chinese Control and Decision Conference (CCDC),* pp. 1514-1519 (6p), 2025.

[127] "https://github.com/chenyaofo/pytorch-cifar-models/tree/master," 2021.

[128] D. A. Meyer, "Quantum strategies," *Physical Review Letters,* vol. 82, pp. 1052--1055, 1999.

[129] Z. R. Sania Naveed, Convolutional Autoencoder-Based Adversarial Defense for Robust Face Recognition under FGSM White-Box Attacks, Global knowledge Academy, 2023.

[130] Florian Tramèr et al, "Stealing Machine Learning Models via Prediction APIs," in *25th USENIX Security Symposium*, 2016.

[131] Matthew Jagielski et al, "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning," in *IEEE Symposium on Security and Privacy (SP)*, 2018.

[132] Brendon G. Anderson et al, "An Overview and Prospective Outlook on Robust Training and Certification of Machine Learning Models," in *arXiv preprint arXiv:2208.07464*, 2022.

[133] Elie Alhajjar et al, "Adversarial Machine Learning in Network Intrusion Detection Systems," in *arXiv preprint arXiv:2004.11898*, 2020.

[134] X. Yuan, "Adversarial Examples: Attacks and Defenses for Deep Learning," in *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[135] X. Yuan, P. He, Q. Zhu and X. Li, Adversarial Examples: Attacks and Defenses for Deep Learning, IEEE Transactions on Neural Networks and Learning Systems, 2019.

[136] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu and J. Zhu, Boosting Adversarial Attacks with Momentum, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[137] E. Nowroozi, A. Dehghantanha, R. M. and K.-k. R. Choo, "A survey of machine learning techniques in adversarial image forensics," *Computers & Security,* vol. 100, no. 102092, 2021.

[138] J. Eisert, M. Wilkens and M. Lewenstein, "Quantum games and quantum strategies," *Physical Review Letters,* vol. 83, pp. 3077--3080, 1999.

[139] H. Wang, J. Cao and Y. Zhang, "A flexible payment scheme and its role-based access control," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 17, pp. 425- 436, 2005.

[140] C. Schmid, A. P. Flitney, W. Wieczorek, N. Kiesel, H. Weinfurter and L. C. L. Hollenberg, "Experimental implementation of a four-player quantum game," *New Journal of Physics,* vol. 10, p. 033011, 2008.

[141] P. Rebentrost, M. Mohseni and S. Lloyd, "Quantum support vector machine for big data classification," *Physical Review Letters,* vol. 113, p. 130503, 2014.

[142] K. heng, L. Wang, Y. Shen, H. Wang, Y. Wang, X. Jiang and H. Zhong, "Secure k-NN Query on Encrypted Cloud Data with Multiple Keys," *IEEE Transactions on Big Data,* vol. PP, pp. 1-1, 2017.

[143] Y. Zhang, Y. Shen, H. Wang, J. Yong and X. Jiang, "On Secure Wireless Communications for IoT Under Eavesdropper Collusion," *IEEE Transactions on Automation Science and Engineering,* vol. 13, pp. 1-13, 2015.

[144] J. Zhang, H. Li, X. Liu, Y. Luo, F. Chen and H. Wang, "On Efficient and Robust Anonymization for Privacy Protection on Massive Streaming Categorical Information," *IEEE Transactions on Dependable and Secure Computing,* vol. PP, pp. 1-1, 2015.

[145] J. Shu, X. Jia, K. YANG and H. Wang, "Privacy-Preserving Task Recommendation Services for Crowdsourcing," *IEEE Transactions on Services Computing,* vol. PP, pp. 1-1, 2018.

[146] A. Shafahi and Ghiasi, "Label Smoothing and Logit Squeezing: A Replacement for Adversarial Training?," *Proc. of the IRE,* 2019.

[147] M. Schuld, A. Bocharov, K. M. Svore and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A,* vol. 101, p. 032308, 2020.

[148] N. Carlini and A. A., "Simple Black-box Adversarial Attacks," *arXiv preprint arXiv,* 2019.

[149] S. Kotyan and Danilo Vasconcellos, "Adversarial robustness assessment: Why in evaluation both~L0~and~L∞~attacks are necessary," 2022.

[150] C. Szegedy, W. Zaremba and I. Sutskever, "Intriguing properties of neural networks," in *ICLR 2014, OpenReview*, 2014.

[151] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert and F. Roli, Support Vector Machines under Adversarial Label Contamination, arXiv preprint arXiv:2206.00352, 2022.

[152] T. Guo and M. Tan, "Improving Robustness with Adversarial Pretraining and Normalization," in *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.

[153] P. Liu and Y. Zhang, "A Survey of Adversarial Machine Learning in Cybersecurity," in *IEEE Access*, 2023.

[154] J. Lin, C. Zhang and Z. Yu, "Certified Defenses against Adversarial Attacks with Provable Guarantees," in *IEEE Trans. Dependable Secure Comput*, 2023.

[155] X. Yuan, Y. He and X. Li, "Recent Advances in Adversarial Attacks and Defenses in Computer Vision," in *IEEE Signal Process. Mag.*, 2023.

[156] N. Carlini and David Wagner, Towards Evaluating the Robustness of Neural Networks, IEEE Symposium on Security and Privacy (SP), 2017.

[157] N. Liu and P. Wittek, Vulnerability of quantum classification, Phys. Rev., 2020.

[158] Y. Du, M.-H. Hsieh, T. Liu, D. Tao and N. Liu, Quantum noise protects quantum classifiers against adversaries, Physical Review Research, 2021.

[159] M. Weber, N. Liu, B. Li, C. Zhang and Z. Zhao, Optimal provable robustness of quantum classification via quantum, Quantum Information, 2021.

[160] L. Zheng, Tanner Fiez, Z. Alumbaugh, B. Chasnov and L. J. Ratliff, "Stackelberg Actor-Critic: A Game-Theoretic Perspective," in *Association for the Advancement of Artificial*, 2021.

[161] H. Hirano, A. Minagi and K. Takemoto, Universal adversarial attacks on deep neural networks for medical image classification, Springer Nature, 2021.

[162] A.-A. Stoica, V. Y. Nastl and M. Hardt, Causal Inference from Competing Treatments, arXiv:2406.03422, 2024.

[163] P. Paruchuri, J. P. Pearce, S. Kraus and M. Janusz Marecki, "Playing Games for Security: An Efficient Exact Algorithm," *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Sys,* vol. 2, pp. 895-902, 2009.

[164] Y. Zhang et al., "Adversarial Robustness Assessment: Why Evaluation Must Include Both L_o and $L\infty$ Attacks," in *Proc. of CVPR*, 2023.

[165] Y. Wu and S. Nowozin, "Improving Adversarial Training by Distributionally Robust Optimization," in *Proc. ICLR*, 2022.

[166] Z. Yue, Z. He, H. Zeng and J. McAuley, "Black-Box Attacks on Sequential Recommenders via Data-Free Model Extraction," *RecSys 2021 - 15th ACM Conference on Recommender Systems,* no. Scopus, pp. 44-54 (11p), 2021.

[167] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning," in *Proceedings of the IEEE*, 2018.

[168] N. Carlini et al, "On Evaluating Adversarial Robustness," in *arXiv preprint arXiv:1902.06705*, 2023.

[169] K. Grosse, D. Pfaff and M. Smith, "The limitations of model uncertainty in adversarial settings," *arXiv preprint arXiv:181202606,* 2018.

[170] Z. Hou and Y. Zhao, "Negative Sampling Algorithm for Cybersecurity Knowledge Graph Based on Combination of GMM and SVM," *International Conference on Communication Networks and Smart Systems Engineering (ICCNSE),* pp. 207-211, 2025.

[171] L. Dritsoula, P. Loiseau and J. Musacchio, "A Game-Theoretic Analysis of Adversarial Classification," *IEEE Transactions on Information Forensics and Security, Information Forensics and Security,* vol. 12, pp. 3094-3109 (16p), 2017.

[172] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann and R. Wolf, "Training deep quantum neural networks," *Nature Communications,* vol. 1, no. 1, pp. 1-6, 2020.

[173] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," *In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security,* pp. 3--14, 2017.

[174] V. Havlíček, A. Córcoles, K. Temme, A. Harrow, A. Kandala, J. Chow and J. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature, 2019.

[175] L. Zhang, T. Zhu, F. Khadeer, D. Ye and W. Zhou, "A Game-Theoretic Method for Defending Against Advanced Persistent Threats in Cyber Systems," *IEEE Transactions on Information Forensics and Security,* vol. 18, pp. 1349-1364, 2023.

[176] W. He, J. Wei, X. Chen, N. Carlini and D. Song, Adversarial Example Defenses: Ensembles of Weak Defenses Are Not Strong, 11th USENIX Workshop on Offensive Technologies (WOOT 17), 2017.

[177] Y. Qian, C. Zhao and Z. Gu, "Feature-Focusing Adversarial Training via Disentanglement of Natural and Perturbed Patterns," *IEEE Transactions on Knowledge and Data Engineering,* vol. 37, no. 9, pp. 5201-5213 (13p), 2025.

[178] W. Gong and D. Deng, "Universal adversarial examples and perturbations for quantum classifiers," *National Science Review,* p. 2021, 2022.

[179] K. Murat, S. Christopher, G. Alexandros and V. S. Dimakis, "CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training," *arXiv,* 2017.

[180] L. M. D. P and G. M, "Robustness Analysis of Machine Learning Models Against Data Poisoning and Evasion in Diabetes Classification," *3rd International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS),* pp. 1197-1203 (7p), 2025.

[181] G. Wang, J. Tang, Z. Ding and Y. Tian, "A Gradient Direction Consistency-Based Dynamic Iterative Adversarial Training," *3rd International Conference on Big Data and Privacy Computing (BDPC),* pp. 148-153 (6p), 2025.

[182] S. Yu and Y. Zhou, "Quantum Adversarial Machine Learning for Robust Power System Stability Assessment," *IEEE Power & Energy Society General Meeting (PESGM),* pp. pages 1-5 (5p), 2024.

[183] W. Yu-Hang, J. Guo, A. Liu, K. Wang, Z. Wu, Z. Liu, W. Yin and J. Liu, "TAET: Two-Stage Adversarial Equalization Training on Long-Tailed Distributions," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 5476-15485 (10p), 2025.

[184] Y. Zhao and X. Guan, "Adversarial Training for Robustness Enhancement in LLM-Based Code Vulnerability Detection," *IEEE 7th International Conference on Communications, Information System and Computer Engineering (CISCE),* pp. 1147-1152 (6p), 2025.

[185] E. Celik and M. K. Gullu, "Enhanced Training with Adaptive Vertex Mixup Against Adversarial Attacks in Federated Learning," *33rd Signal Processing and Communications Applications Conference (SIU),* pp. 1-4 (4p), 2025.

[186] M. elShehaby, A. Kotha and A. Matrawy, "Adaptive Continuous Adversarial Training (ACAT) to Enhance ML-NIDS Robustness," *ICC 2025 - IEEE International Conference on Communications,* pp. 1091-1096 (6p), 2025.

[187] C. Fan, W. Guo and L. Xu, "Enhancing Model Robustness and Accuracy via Learnable Adversarial Training," *28th International Conference on Computer Supported Cooperative Work in Design (CSCWD),* pp. 1501-1506 (6p), 2025.

[188] J. Guan, W. Fang and M. Ying, "Robustness verification of quantum classifiers," *arXiv,* 2020.

[189] J. Yin, G. Chen, W. Hong, J. Cao, H. Wang and Y. Miao., "A heterogeneous graph-based semi-supervised learning framework for access control decision-making.," in *World Wide Web 27, no. 4: 35.*, 2024.

[190] J. Yin, G. Chen, W. Hong, H. Wang, J. Cao and Y. Miao, "Empowering vulnerabilityprioritization: A heterogeneous graph-driven framework for exploitabilityprediction," *International conference on web information systems engineering,* pp. 289-299 , Springer Nature Singapore, 2023.

[191] M. You, Y.-F. Ge, K. Wang, H. Wang, J. Cao and G. Kambourakis, "Hierarchical adaptive evolution framework for privacy-preserving data publishing.," in *World Wide Web 27, no. 4 (2024): 49.*, 2024.

[192] M. N. A. Tawhid, S. Siuly, K. Wang and H. Wang., "GENet: a generic neural network for detecting various neurological disorders from EEG.," in *IEEE Transactions on Cognitive and Developmental Systems 16, no. 5 (2024): 1829-1842.*, 2024.

[193] Y.-F. Ge, H. Wang, J. Cao, Y. Zhang and X. Jiang, "Privacy-preserving data publishing: an information-driven distributed genetic algorithm.," in *World Wide Web 27, no. 1 (2024): 1.*, 2024.

[194] C.-Q. Huang, Q.-H. Huang, X. Huang, H. Wang, K.-J. L. Ming Li and Y. Chang, "XKT: toward explainable knowledge tracing model with cognitive learning theories for

questions of multiple knowledge concepts.," in *IEEE Transactions on Knowledge and Data Engineering 36, no. 11 (2024): 7308-7325.*, 2024.

[195] E. Huang, Z. Zhao, J. Yin, J. Cao and H. Wang, "Transformer-Enhanced Adaptive Graph Convolutional Network for Traffic Flow Prediction.," in *ACM Transactions on Intelligent Systems and Technology (2025). DOI 10.1145/3729244*, 2025.

[196] R. Nowrozy, K. Ahmed, A. S. M. Kayes, H. Wang and T. R. McIntosh, "Privacy preservation of electronic health records in the modern era: A systematic survey," in *ACM Computing Surveys 56, no. 8 (2024): 1-37.*, 2024.

[197] H. Wang and L. Sun., "Trust-involved access control in collaborative open social networks," in *Fourth International Conference on Network and System Security, pp. 239-246. IEEE,*, 2010.

[198] R. Li, D. Zhang, Y. Wang, Y. Jiang, Z. Zheng, S.-W. Jeon and H. Wang., "Open-Vocabulary Multi-Object Tracking with Domain Generalized and Temporally Adaptive Features.," in *IEEE Transactions on Multimedia (2025). vol. 27, pp. 3009–3022. DOI: 10.1109/TMM.2025.3557619*, 2025.

[199] M. You, Y.-F. Ge, K. Wang, H. Wang, J. Cao and G. Kambourakis, "Tlef: two-layer evolutionary framework for t-closeness anonymization," in *In International conference on web information systems engineering, pp. 235-244. Singapore: Springer Nature Singapore*, Singapore, 2023.

[200] H. Quadri, H. Wang, I. Sardar and L. Bo, "Quantum SVM Algorithms for Efficient Defense Against Gradient-Based Adversarial Attacks," *NaNA International Conference on Networking and Network,* 2025.

[201] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert and R. Sweke, Encoding-dependent generalization bounds for parametrized quantum circuits, Quantum 5, 582, arXiv, 2021.

[202] V. Nguyen, L. Nguyen, R. Hwang, B. Canberk and T. Duong, "Quantum Machine Learning for 6G Network Intelligence and Adversarial Threats," *IEEE Communications Standards Magazine, Communications Standards Magazine,* vol. 9, pp. 40-48 (9p), 2025.

[203] C.-H. Lin, C.-Y. Kuo and S.-S. Young, "Quantum Adversarial Learning for Hyperspectral Remote Sensing," *IGARSS - IEEE International Geoscience and Remote Sensing Symposium,* pp. 7807-7811 (5p), 2024.

[204] W. Cheng, S. Zhang and Y. Lin, "Study on the Adversarial Sample Generation Algorithm Based on Adversarial Quantum Generation Adversarial Network," *3rd*

International Symposium on Computer Technology and Information Science (ISCTIS), pp. 238-243 (6p), 2023.

[205] Q. Ma, C. Hao, N. Si, G. Chen, J. Zhang and D. Qu, "Quantum adversarial generation of high-resolution images," *EPJ Quantum Technology, Springer Nature Journals,* vol. 12, no. 1, 2025.