Robust Models in the Presence of Uncertainty and Adversarial Attacks

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
by Research

by Sreenivasan M

20173013
sreenivasan.m@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
October 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Robust Models in the Presence of Uncertainty and Adversarial Attacks** " by **Sreenivasan M**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Naresh Manwani

To my daughter **Harini S**

Acknowledgments

First and foremost, I would like to express my gratitude to my thesis advisor, Dr. Naresh Manwani of the Machine Learning Lab at IIIT Hyderabad, for his unwavering support and guidance throughout my postgraduate research. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I would also like to thank my professors, non-teaching staff, and friends for their contributions, which is only possible with this work.

Finally, I must express my heartfelt appreciation to my parents and family for their unconditional support and encouragement over the last five years. This achievement would not have been possible without their assistance.

Abstract

With the increased data availability, the biggest problem is getting accurate labeled data for training models. However, most of the time, we get the missing values. It is preferred to consider weak supervised learning methods to avoid the cost of quality labeled data. One such technique is called partial labeled data classification. We present a novel second-order cone programming framework to create robust classifiers that can tolerate uncertainty in the observations of partially labeled multiclass classification problems. Only the presence of second-order moments is necessary for the proposed formulations independent of the underlying distribution. The case of missing values in observations for multiclass classification problems is thus specifically addressed by these formulations.

Adversarial examples are machine learning model inputs that an attacker has purposefully constructed to cause the model to make a mistake. Recent research aimed to improve the computational efficiency of adversarial training for deep learning models. Projected Gradient Descent (PGD) and Fast Gradient Sign Method (FGSM) are popular current techniques for generating adversarial examples efficiently. There is a trade-off between these two regarding robustness or training time. Adversarial training with the PGD is considered as one of the most efficient adversarial defense techniques to achieve moderate adversarial robustness. However, PGD requires high training time since it takes multiple iterations to generate perturbations. On the other hand, adversarial training with the FGSM takes much less training time since it takes one step to generate perturbations but fails to increase adversarial robustness. On the CIFAR-10/100 datasets, our approach outperforms PGD strong adversarial training techniques in terms of robustness to adversarial training and speed.

While deep learning systems trained on medical images have demonstrated cutting-edge performance in various clinical prediction tasks, recent research indicates that cleverly created hostile images can fool these systems. Deep learning-based medical image classification algorithms have been questioned regarding their practical deployment. To attack this problem, we provide an unsupervised learning technique for detecting adversarial attacks on medical images. Without identifying the attackers or reducing classification performance, our suggested strategy FNS (Features Normalization and Standardization), can detect adversarial attacks more effectively than earlier methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the case of classification problems under supervised machine learning methods, the input training data must be labeled. The model's accuracy depends on how correctly the input data are labeled and how much data captures different scenarios. With the availability of big data, a good amount of data is available for training. However, the problem comes with the availability of labeled data and whether all the captured data is complete (which means no missing values). Different approaches for labeling data, like using multiple annotators, etc., can compromise cost but impact model prediction. There are scenarios where we cannot capture complete data due to sensor issues. One way to handle these is by using partial-label data learning methods. So far, to the best of our knowledge, no one has solved the problem of partial label data with missing attributes.

Also, with the recent developments in deep networks, [62] showed a significant improvement in clinical predictions, especially for X-ray and color fundus photographs. However, deep networks can be fooled on the other side, which is unacceptable in high-security systems like medical systems. In defending against these attacks, various factors must be considered, like model retraining, training time, etc. All these have been considered, and proposed adversarial defense techniques for the same.

1.1 Robustness in handling missing values and partial label data

The machine learning model's performance is accurate when the provided data precisely covers the domain for which the model is designed and organized following the model's features. Because most of the data is unstructured or only very weakly structured, a machine learning technique called weak supervision is used to continue annotating this type of data. Weak supervision is a machine learning component that uses unorganized or inaccurate data to provide labels for a significant amount of unsupervised data, allowing for a large amount of data in machine learning or supervised learning. Figure 1.4 illustrates scenarios for completely supervised and partially labeled data machine learning for further understanding.

A set of labeled and unlabelled examples are available to the learner in semi-supervised learning [87]. Each example is given numerous labels, which may be true in the case of multi-label learning [9] [72].

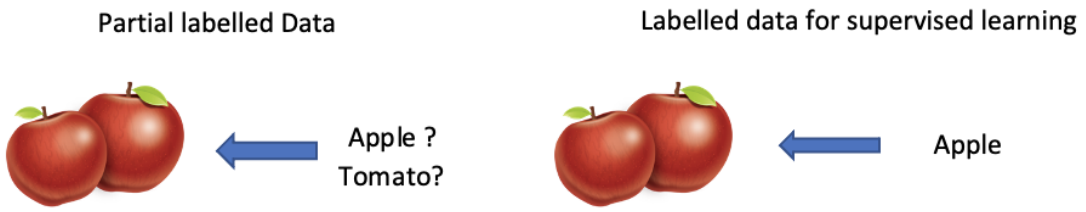

Figure 1.1 Range of supervision in classification.

Examples are not individually labeled in multi-instance learning [20] [1] [73], but rather organized into sets that either contain at least one positive example or exclusively negative examples. Each example is given many labels, but only one is accurate in partially labeled learning, also known as ambiguously labeled learning.

Photographs with multiple faces in each image and captions that list the names of the people in the photo without identifying which face belong to which person are common examples. Figure 1.5 [14] below shows how each face in this situation is ambiguously labeled with the names taken from the caption.

Figure 1.2 Two examples of partial labeling scenarios for naming faces

It is possible to identify every face in the accompanying photographs for the top row in the above figure using a screenplay. However, it is unclear who is who in every other face (green labels). Images in photo collections and websites are frequently labeled ambiguously with multiple potential names in

the caption or neighboring text for the bottom row in the above figure. In both situations, the objective is to learn a model from instances with unclear labels to clarify the training labels and generalize to new examples. The performance of several applications, including image retrieval and video summarization, can be enhanced by learning exact models for face and object recognition from such erroneously labeled images and videos.

Despite being halfway between fully supervised and fully unsupervised learning, this partially labeled setting differs qualitatively from the semi-supervised setting, where both labeled and unlabelled data are available. This partially labeled problem has been discussed in numerous papers, but to the best of our knowledge, it is assumed that there was no uncertainty considered in the input data in all the cases.

1.2 Robustness in Adversarial attack detection

Medical imaging has seen extensive use of convolutional neural networks (CNNs). However, imperceptible to human professionals, adversarial attacks by perturbations can be made against medical images. In clinical practice, CNN-based applications face serious security threats and difficulties. Chest X-rays and color fundus images, two prominent diagnostic and prognostic imaging modalities in patient care [41][5]. Manual analysis of medical images by human professionals, such as radiologists and ophthalmologists, is time-consuming and error-prone in practical practice. Therefore, it has been suggested that automated machine-learning systems could enhance clinical decision-making and aid in human diagnosis [85] [6]. Recent advancements in deep learning network (DNN) technology, including convolutional neural networks (CNNs) [35], have allowed several deep learning (DL) systems to demonstrate performance on a variety of medical image detection tasks that are on par with or better than that of human experts. Figure 1.6 illustrates that CNNs are susceptible to adversarial attacks [4][28].

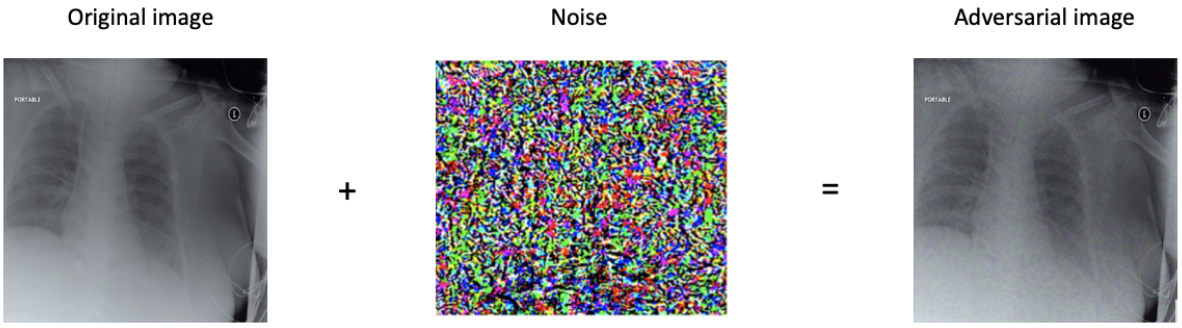

Figure 1.3 Adversarial attack examples

The above figure shows an original Chest X-Ray (CXR) and its adversarial example. Adversarial perturbations are generated by projected gradient descent (PGD) [19] with 50 iterations with radius $\epsilon = 8/255$ with step size $\alpha = 1.25 * \epsilon$

Images are attacked by adding a small adversarial perturbation to the original images, and this perturbation is imperceptible to humans yet misleads a standard CNN model resulting in incorrect classifications, with a substantial decline in its predictive performance [79][80]. Two types of adversarial attacks (white-box and black-box) are categorized by the level of information to which the "attacker" has access. In the white-box settings, the attacker has direct access to the target model parameters [45]. In contrast, the attacker cannot access the model parameters in the black-box settings. Therefore, black-box attacks are more applicable in many realistic scenarios. CNNs under adversarial attacks will fail to assist by misleading human clinicians. Importantly, such a vulnerability also poses severe security risks and prevents deploying automated CNN-based systems in real-world use, especially in the medical domain where accurate diagnostic results are highly important in patient care. Several works of literature on adversarial machine learning exist, but most have focused on natural images [46]. The scope of studies on medical images has been rather constrained; however, several have proven that medical DL systems are vulnerable to adversarial attacks by conventional or image-specific attacking strategies across medical specialties [25][82]; others have proposed techniques to automatically identify adversarial attacks solely in white-box scenarios [45].

Despite these attempts, it is still difficult for the machine-learning research community to develop a general detection system that can accurately identify a variety of adversarial perturbations, especially under black-box attacks. Furthermore, to increase the robustness of CNNs(medical) and improve diagnostic performance, we need to be aware of any prior work examining the underlying factors that mask CNNs' susceptibility to adversarial attacks.

The high dimensionality of training data [11], where each image often comprises hundreds or even tens of thousands of pixels, is viewed in most prior research as contributing to adversarial attack vulnerability. Accordingly, the current state-of-the-art defense technique, adversarial training (AT) [46][78], initially makes use of CNNs to produce adversarial attack instances in each training phase before using these examples as training data to increase model robustness. CNNs performance is solely dependent on the adversarial attack samples created during training.

1.2.1 Adversarial Training time

A robust classifier correctly labels images that have been subjected to adversarial disturbance. As an alternative, robustness can be attained by detecting and rejecting hostile examples [44] [48] [81]. A whole set of supposedly strong defenses was recently defeated by [2], leaving adversarial training - in which the defender supplements each mini-batch of training data with adversarial examples [46] as one of the few remaining defenses that are resistant to adversarial attacks.

In addition to the gradient calculation required to update the network parameters, adversarial training takes time because stochastic gradient descent (SGD) iterations need several gradient computations to generate adversarial images. Contrary to popular belief, building a robust network with adversarial training takes far longer than building a non-robust version. In short, the real slowing factor relies on how many gradient steps are employed to generate adversarial examples.

Although adversarial training is still one of the most reliable defenses, it is time-consuming and can take several days, even on relatively sized datasets like CIFAR-10 and CIFAR-100. We propose an effective adversarial training method that overcomes the problem of high training time and low model accuracy due to adversarial attacks.

1.3 Contribution

The main goal of this thesis is to address the uncertainty in partial label data multiclass classification problems, improve the adversarial attacks' robustness, and identify methods to reduce the adversarial-based training time by not compromising the adversarial defense.

The first takes into account the missing values in the partial label data. The current literature on learning a multiclass classifier with a second-order cone programming approach assumes that the family of distributions considered have the same second-order moments. In this context, we assume that the uncertainty is only in the input patterns where some components may be missing, and the output labels are known precisely. One of the main contributions of this thesis is to generalize the results of [63] partially. In the case of classification problems under supervised machine learning methods, the input training data must be labeled. The model's accuracy depends on how correctly the input data are labeled and how much data captures different scenarios. With the availability of big data, a good amount of data is available for training. However, the problem comes with the availability of labeled data and whether all the captured data is complete (which means no missing values). Different approaches for labeling data, like using multiple annotators, etc., can compromise cost but impact model prediction. There are scenarios where we cannot capture complete data due to sensor issues. One way to handle these is by using partial-label data learning methods. So far, to the best of our knowledge, no one has solved the problem of partial label data with missing attributes.

Also, with the recent developments in deep networks, [62] showed a significant improvement in clinical predictions, especially for X-ray and color fundus photographs. However, deep networks can be fooled on the other side, which is unacceptable in high-security systems like medical systems. In defending against these attacks, there are various factors to be considered, like model retraining, training time, etc. All these have been considered and proposed adversarial defense techniques for the same.

1.4 Robustness in handling missing values and partial label data

The machine learning model's performance is accurate when the provided data precisely covers the domain for which the model is designed and organized following the model's features. Because most of the data is unstructured or only very weakly structured, a machine learning technique called weak supervision is used to continue annotating this type of data. Weak supervision is a machine learning component that uses unorganized or inaccurate data to provide labels for a significant amount of unsupervised data, allowing for a large amount of data in machine learning or supervised learning.

Figure 1.4 illustrates scenarios for completely supervised and partially labeled data machine learning for further understanding.

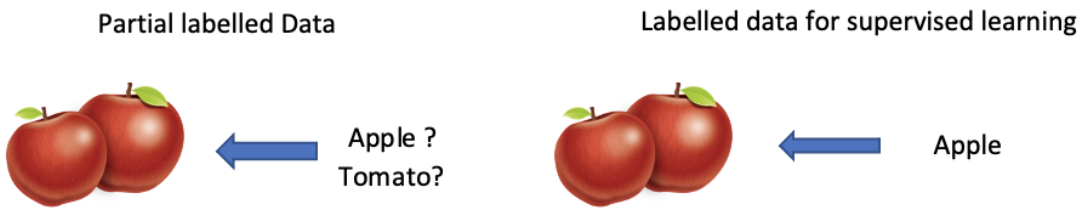

Figure 1.4 Range of supervision in classification.

A set of labeled and unlabelled examples are available to the learner in semi-supervised learning [87]. Each example is given numerous labels, which may be true in the case of multi-label learning [9] [72]. Examples are not individually labeled in multi-instance learning [20] [1] [73], but rather organized into sets that either contain at least one positive example or exclusively negative examples. Each example is given many labels, but only one is accurate in partially labeled learning, also known as ambiguously labeled learning.

Photographs with multiple faces in each image and captions that list the names of the people in the photo without identifying which face belongs to which person are common examples. Figure 1.5 [14] below shows how each face in this situation is ambiguously labeled with the names taken from the caption.

It is possible to identify every face in the accompanying photographs for the top row in the above figure using a screenplay. However, it is unclear who is who in every other face (green labels). Images in photo collections and websites are frequently labeled ambiguously with multiple potential names in the caption or neighboring text for the bottom row in the above figure. In both situations, the objective is to learn a model from instances with unclear labels to clarify the training labels and generalize to new examples. The performance of several applications, including image retrieval and video summarization, can be enhanced by learning exact models for face and object recognition from such erroneously labeled images and videos.

Despite being halfway between fully supervised and fully unsupervised learning, this partially labeled setting differs qualitatively from the semi-supervised setting, where both labeled and unlabelled data are available. This partially labeled problem has been discussed in numerous papers, but to the best of our knowledge, it is assumed that there was no uncertainty considered in the input data in all the cases.

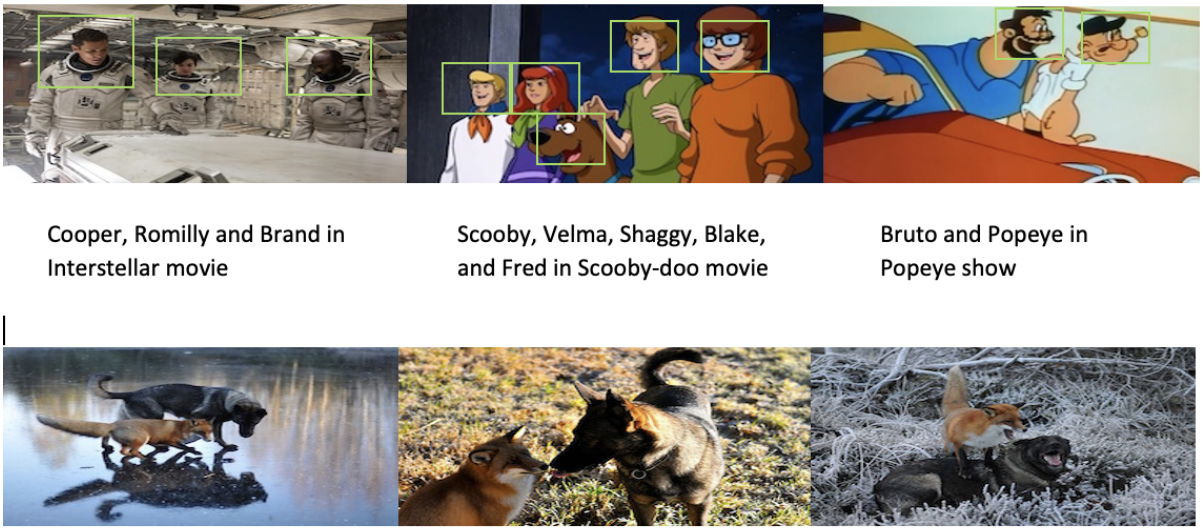

Figure 1.5 Two examples of partial labeling scenarios for naming faces

1.5 Robustness in Adversarial attack detection

Medical imaging has seen extensive use of convolutional neural networks (CNNs). However, imperceptible to human professionals, adversarial attacks by perturbations can be made against medical images. In clinical practice, CNN-based applications face serious security threats and difficulties. Chest X-rays and color fundus images, two prominent diagnostic and prognostic imaging modalities in patient care [41][5]. Manual analysis of medical images by human professionals, such as radiologists and ophthalmologists, is time-consuming and error-prone in practical practice. Therefore, it has been suggested that automated machine-learning systems could enhance clinical decision-making and aid in human diagnosis [85] [6]. Recent advancements in deep learning network (DNN) technology, including convolutional neural networks (CNNs) [35], have allowed several deep learning (DL) systems to demonstrate performance on a variety of medical image detection tasks that are on par with or better than that of human experts. Figure 1.6 illustrates that CNNs are susceptible to adversarial attacks [4][28].

The above figure shows an original Chest X-Ray (CXR) and its adversarial example. Adversarial perturbations are generated by projected gradient descent (PGD) [19] with 50 iterations with radius $\epsilon = 8/255$ with step size $\alpha = 1.25 * \epsilon$

Images are attacked by adding a small adversarial perturbation to the original images, and this perturbation is imperceptible to humans yet misleads a standard CNN model resulting in incorrect classifications, with a substantial decline in its predictive performance [79][80]. Two types of adversarial attacks (white-box and black-box) are categorized by the level of information to which the attacker has access. In the white-box settings, the attacker has direct access to the target model parameters [45]. In contrast, the attacker cannot access the model parameters in the black-box settings. Therefore, black-

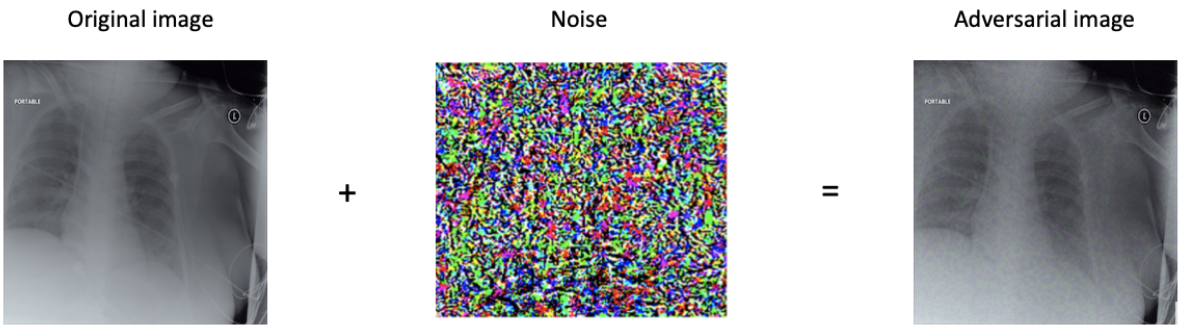

Figure 1.6 Adversarial attack examples

box attacks are more applicable in many realistic scenarios. CNNs under adversarial attacks will fail to assist by misleading human clinicians. Importantly, such a vulnerability also poses severe security risks and prevents deploying automated CNN-based systems in real-world use, especially in the medical domain where accurate diagnostic results are highly important in patient care. Several works of literature on adversarial machine learning exist, but most of them have focused on natural images [46]. The scope of studies on medical images has been rather constrained; however, several have proven that medical DL systems are vulnerable to adversarial attacks by conventional or image-specific attacking strategies across medical specialties [25][82]; others have proposed techniques to automatically identify adversarial attacks solely in white-box scenarios [45].

Despite these attempts, it is still difficult for the machine-learning research community to develop a general detection system that can accurately identify a variety of adversarial perturbations, especially under black-box attacks. Furthermore, to increase the robustness of CNNs(medical) and improve diagnostic performance, we need to be aware of any prior work examining the underlying factors that mask CNNs' susceptibility to adversarial attacks.

The high dimensionality of training data [11], where each image often comprises hundreds or even tens of thousands of pixels, is viewed in most prior research as contributing to adversarial attack vulnerability. Accordingly, the current state-of-the-art defense technique, adversarial training (AT) [46][78], initially makes use of CNNs to produce adversarial attack instances in each training phase before using these examples as training data to increase model robustness. CNNs performance is solely dependent on the adversarial attack samples created during training.

1.5.1 Adversarial Training time

A robust classifier properly labels images that have been subjected to adversarial disturbance. As an alternative, robustness can be attained by detecting and rejecting hostile examples [44] [48] [81]. A whole set of supposedly strong defenses was recently defeated by [2], leaving adversarial training - in which the defender supplements each mini-batch of training data with adversarial examples [46] as one of the few remaining defenses that are resistant to adversarial attacks.

In addition to the gradient calculation required to update the network parameters, adversarial training takes time because stochastic gradient descent (SGD) iterations need several gradient computations to generate adversarial images. Contrary to popular belief, building a robust network with adversarial training takes far longer than building a non-robust version. In short, the real slowing factor relies on how many gradient steps are employed to generate adversarial examples.

Although adversarial training is still one of the most reliable defenses, it is time-consuming and can take several days, even on relatively sized datasets like CIFAR-10 and CIFAR-100. We propose an effective adversarial training method that overcomes the problem of high training time and low model accuracy due to adversarial attacks.

1.6 Contribution

The main goal of this thesis is to address the uncertainty in partial label data multiclass classification problems, improve the adversarial attacks' robustness, and identify methods to reduce the adversarial-based training time by not compromising the adversarial defense.

The first takes into account the missing values in the partial label data. The current literature on learning a multiclass classifier with a second-order cone programming approach assumes that the family of distributions considered have the same second-order moments. In this context, we assume that the uncertainty is only in the input patterns where some components may be missing, and the output labels are known precisely. One of the main contributions of this thesis is to generalize the results of [63] to partially labeled data multiclass classification problems. Here, we use the Chebyshev inequalities for multivariate random variables to obtain a second-order cone programming (SOCP), one of this thesis's main contributions.

The second and third part focuses on robustness in identifying adversarial attacks, especially for medical images, and reducing adversarial training time with the best adversarial defense test accuracy.

1.7 Thesis Organization

In this thesis, we targeted three problems which are mentioned below. Chapter 4 will discuss the uncertainty in the partial label data multiclass classification problem and present our novel algorithm, which deals with missing attributes.

Chapter 5 discusses the problem of improving the adversarial training time without compromising on the adversarial defense. Here, we propose an algorithm that effectively defends the Projected gradient descent method, known as the most effective iterative adversarial attack algorithm. The time taken for training with our algorithm is less when compared to the one trained with the Projected gradient descent method. We tested our algorithm on CIFAR-10 and CIFAR-100 datasets.

Chapter 6 investigates and discusses various adversarial attacks and defense algorithms for medical images in the literature. This also deals with our novel algorithm, which effectively detects adversarial

attacks like Projected gradient descent, Fast gradient sign method, Basic iterative method, and Momentum iterative fast gradient sign methods on X-ray datasets.

Chapter 7 discusses some intriguing future directions related to the problems addressed in the thesis.

The second and third part focuses on robustness in identifying adversarial attacks, especially for medical images, and reducing adversarial training time with the best adversarial defense test accuracy.

1.8 Thesis Organization

In this thesis, we targeted three problems which are mentioned below.

Chapter 4 will discuss the uncertainty in the partial label data multiclass classification problem and present our novel algorithm, which deals with missing attributes.

Chapter 5 discusses the problem of improving the adversarial training time without compromising on the adversarial defense. Here, we propose an algorithm that effectively defends the Projected gradient descent method, known as the most effective iterative adversarial attack algorithm. The time taken for training with our algorithm is less when compared to the one trained with the Projected gradient descent method. We tested our algorithm on CIFAR-10 and CIFAR-100 datasets.

Chapter 6 investigates and discusses various adversarial attacks and defense algorithms for medical images in the literature. This also deals with our novel algorithm, which effectively detects adversarial attacks like Projected gradient descent, Fast gradient sign method, Basic iterative method, and Momentum iterative fast gradient sign methods on X-ray datasets.

Chapter 7 summarizes the thesis with some concluding remarks about the contributions made and intriguing future directions.

Chapter 2

Literature Survey On Adversarial Attacks And Defenses

Adversarial attacks are methods that purposefully alter inputs to deceive machine learning algorithms and produce false predictions. These attacks frequently entail adding subtle changes to input data that are carefully planned to result in misclassifications. Contrarily, adversarial defenses work to strengthen the resistance of machine learning models to such attacks. A brief literature survey on existing adversarial attacks and their defenses will be discussed in the following sections.

2.1 Survey on Adversarial attacks and Adversarial defenses

There are a few theories that researchers have put out to explain the existence of adversarial examples. Few [54] believe these examples are due to deep neural networks' excessive non-linearity, while others [68] believe that due to the over or under-fitting of the model. Nevertheless, Goodfellow et al. [27] showed that adding perturbation to the input of a regularization model and a linear model with sufficient dimensions did not significantly increase the model's ability to defend against adversarial attacks. According to Goodfellow et al., [27], the linear behavior in high-dimensional space is the root of adversarial samples. Each input feature in a high-dimensional linear classifier is normalized so that modest changes to any one input dimension will not significantly alter the classifier's overall prediction. However, small changes to all input dimensions will have an impact.

The three fundamental characteristics of adversarial examples are transferability, regularization effect, and adversarial instability [84], which are explained below. Transferability - As long as the adversary is aware of model M2 and M2 is trained to carry out the task that model M1 carries out, it is not necessary to learn the architecture or parameters of M1 while building adversarial samples for an attack on one target model M1. Adversarial instability - Target models will correctly classify adversarial examples if they have undergone physical transformations like translation. Regularization effect - An adversarial training method can be considered, but it is more expensive.

2.1.1 Adversarial attacks

This sub-section discusses the ways how adversaries can attack the model. Broadly, it can be at the training and testing phase of the model. During the training phase, it can be at data injection where the adversary can add new data with wrong labels, corrupting the model. It can also happen if the adversary has access to the model and training data, where the model's logic can be corrupted before training the model. During the testing phase, the adversary forces the model to predict wrong outputs without modifying the model. This can be broadly classified into two categories: white-box and black-box attacks.

White-box attacks: Here, the adversary has complete knowledge of the model, logic, dataset, etc. With these, adversaries can generate adversarial examples where the model is most vulnerable. As per Papernot et al., [54], as shown in Figure 2.1, an adversary can generate adversarial examples using direction sensitivity estimation and perturbation selection, i.e., the adversary tries to identify the direction and the magnitude of change in input feature to affect the model prediction. The changes are placed so that they are not recognizable by human eyes.

$$\mathbf{X}_{adv} = \mathbf{X} + \arg\min_{\delta \mathbf{X}} \|\delta \mathbf{X}\| \tag{2.1}$$

such that model predicts wrong output for input data $\mathbf{X}_{adv}$

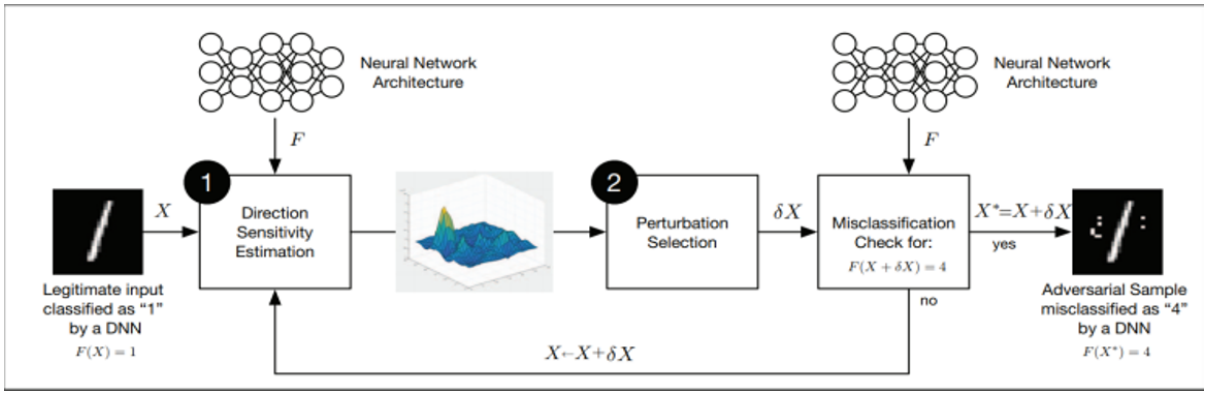

Figure 2.1 Framework for construction of Adversarial sample (Image credit: Papernot et al. [54])

Below are some of the well-known techniques used to generate adversarial samples based on Direction sensitivity estimation:

1. L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno): To find an adversarial sample, Szegedy [66] first formulated the minimization issue as indicated in the above equation (2.1). They solved a reduced version by determining the least loss function additions to prevent the neural network from making a false classification. This turned the difficult problem into a convex optimization problem. Even while this strategy performs well, calculating the adversarial samples is expensive.
2. FGSM (Fast Gradient Sign Method): A Fast Gradient Sign approach was put forth by Goodfellow

et al. [27] to determine the gradient of the cost function to the input. The following formula generates adversarial samples.

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon * sign\left(\nabla_x L(\mathbf{x}, y)\right)$$

3. One-step Target class method: This is a variation of the FGSM that maximizes the probability P($y_{target}|\mathbf{x}$) of a particular target class y_{target}, which is not the true class for the input $\mathbf{x}$. The following formula generates adversarial samples:

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon * sign\left(\nabla_x L(\mathbf{x}, y_{target})\right)$$

4. BIM (Basic Iterative Method): This is an extension of FGSM, which repeatedly applies FGSM multiple times with a small step size.

$$\mathbf{x}_i = Clip\left[\mathbf{x}_{i-1} + \alpha.sign\left(\nabla_x L(\mathbf{x}_{i-1}, y)\right)\right]$$

where i denotes the iteration number for iterative attack and the $Clip$ function clips all the values between 0 and 1.

5. PGD(Projected Gradient Descent): A more potent foe is the multi-step variation, which is projected gradient descent (PGD) on the negative loss function. PGD [46] perturbs a clean data $\mathbf{x}$ for T steps with smaller step sizes. After each step of perturbation, PGD projects the adversarial example back onto the ϵ-ball of $\mathbf{x}$ if it goes beyond:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha * sign\left(\nabla_x L(\mathbf{x}_{i-1}, y)\right)$$

where α is the step size, and $\mathbf{x}_i$ is the adversarial example at the i^{th} step ($\mathbf{x}_0 = \mathbf{x}$). The step size is usually set to $\epsilon/T \leq \alpha < \epsilon$ for overall T steps of perturbations.

6. JSMA (Jacobian-based Saliency Map): Here, Papernot et al. [55] suggested a technique called models Jacobian matrix for determining the sensitivity direction to create adversarial examples. The adversarial examples are created using the complex saliency map approach that provides the gradient of each output component to each input component. Only a small number of the image's pixels must be changed because this strategy restricts the perturbations to l_0 norm.

7. One-pixel attack: Su et al. [65] proposed that an adversarial sample is generated by changing only a one-pixel value in the image. To implement the adversarial attacks, they iteratively modify each pixel using the differential evolution process to create a sub-image, compare it to the parent image, and keep the sub-image with the best attack effect.

8. Deepfool: For a given image, Moosavi-dezfooli et al. [52] suggested computing a minimal norm adversarial perturbation iteratively to locate the decision boundary that is most similar to the normal sample and identify the fewest adversarial examples across the boundary. They employed various linear approximations to approach the broad nonlinear decision functions and resolve the challenge of picking

the parameter under the FGSM [27]. They showed that the perturbations they produce have a similar deception rate and are less than those produced by FGSM.

When determining which aspect of target misclassification is most likely to produce the least disruption, an adversary can exploit the target model's sensitivity to input information. There are two sorts of techniques for altering input dimension:

1. A way to interfere with each input dimension was put out by Goodfellow et al. [27]. However, the amount of interference needed to compute the gradient sign direction via the FGSM approach is negligible. Using this technique, the distance between the original samples and their associated adversarial samples is effectively shortened in Euclidean space. This approach is effective for quickly producing many adversarial samples but is more easily identified because of its relatively substantial disturbance.

2. Papernot et al. [55] chose to perturb a smaller number of input dimensions and a more intricate procedure involving the saliency map. The disturbance caused by the input features when producing adversarial samples is effectively decreased by this technique.

Black-box attacks: Adversaries cannot access the model. However, they can see the outputs associated with particular inputs and create a model similar to the target model by running sample images on the target model and observing it. The input and querying times for the target model significantly impact this attack's effectiveness.

2.1.2 Adversarial defenses

Researchers have proposed many adversarial defense strategies to strengthen the robustness of neural networks against adversarial attacks. These strategies can be categorized into three main groups: changing data, changing models, and employing auxiliary tools.

Changing Data: These techniques, which include adversarial training, gradient hiding, preventing transferability, data compression, and data randomization, modify the training dataset during the training phase or the input data during the testing phase.

a. Adversarial training: Different methods were proposed here, but it is unrealistic to introduce all unknown attack samples into adversarial training.

b. Gradient Hiding: Here, consider the models which are not non-differential, like random forests, etc., then adversaries will fail to use gradient-based techniques to generate adversarial examples. Nevertheless, these models can be fooled [54] with black box attacks using proxy models.

c. Blocking the Transferability: Even if the classifiers have a distinct design or were trained on a separate dataset, the transferability feature still holds for black-box attacks. Hosseini et al. [32] proposed a three-step NULL labeling method as shown in Figure 2.2.

Step1: Initial training with clean images,

Step2: Model F to predict the probability of the amount of adversarial perturbation by generating adversarial examples from the validation data using Greedy methods and

Step 3: The classifier is trained with clean and adversarial samples. For adversarial samples, where the classifier is trained to assign a high probability vector to NULL label, i.e., it adaptively identifies

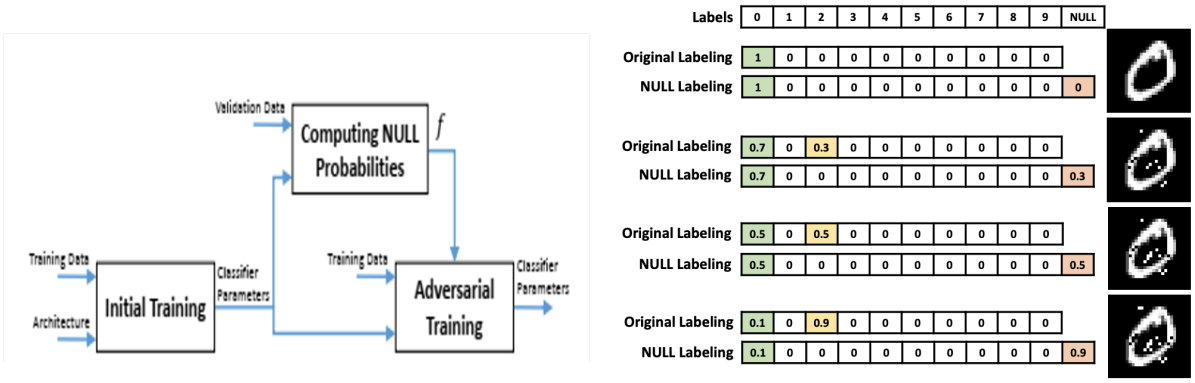

Figure 2.2 Three steps NULL labeling method (Image credit: Hosseini et al. [32])

problematic direction around each input data and corrects itself as shown in Figure 2.2.

d. Data Compressions: JPG/JPEG [17] compression method improves many model recognition accuracy from FGSM/Deep Fool attacks, but they are ineffective with powerful attacks. The biggest limitation of these categories is that a large amount of compression will decrease the accuracy of original image classification.

e. Data Randomization: Here, it mainly deals with modifying the data. For example, if the adversarial sample is modified like resized, its impact to fool the deep networks can be reduced. Want et al. [76] conducted data expansion operations in the training process, such as some Gaussian randomization processing which could slightly improve the robustness of the network model.

Changing the model: These techniques include regularization, defensive distillation, feature squeezing, deep contractive network, and mask defense methods, which are detailed below:

1. Regularization method: Regularization is one approach to reduce the data's exposure during training. The algorithm's robustness is increased and promises good results for adversarial attacks.

2. Defensive Distillation: As per [56], as shown in Figure 2.3, first, train the network with a SoftMax Temperature T (not set to 1), and then use the probability vector to train a distilled network at a temperature T=1 on the same data. However, the effectiveness of defense distillation cannot be guaranteed in black-box attacks.

3. Feature Squeezing method: From [81], it is a model enhancement technique that reduces the complexity of data representation. There are two heuristic methods To employ a smooth filter on the image and to decrease the color depth at the pixel level. However, while this method successfully resists adversarial attempts, it also reduces the accuracy of the model.

4. DCN (Deep Contractive Network) method: Here, [29] uses contractive autoencoders concept and implements to layer-wise in the network to reduce the variation of each layer due to changes in input and was provided to have specific defensive effects against attacks such as L-BFGS.

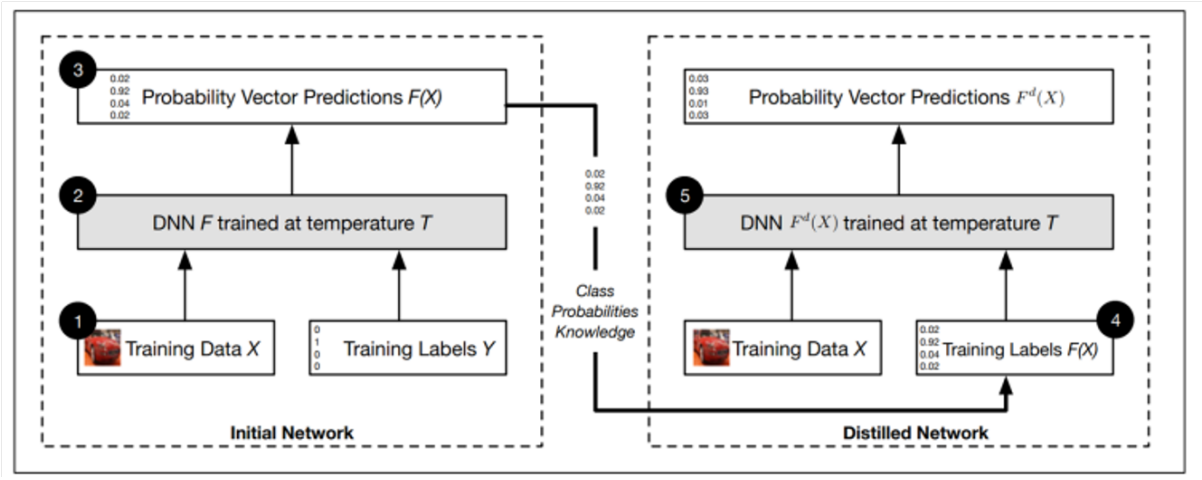

Figure 2.3 Defensive Distillation algorithm (Image credit: Papernot et al. [56])

5. Mask Defense method: The basic idea is to remove unnecessary features for generating adversarial samples. Gao et al. [26] proposed the DeepCloak method, as shown in Figure 2.4, which identifies unnecessary features introduced in the image and masks them, see Figure 2.4.

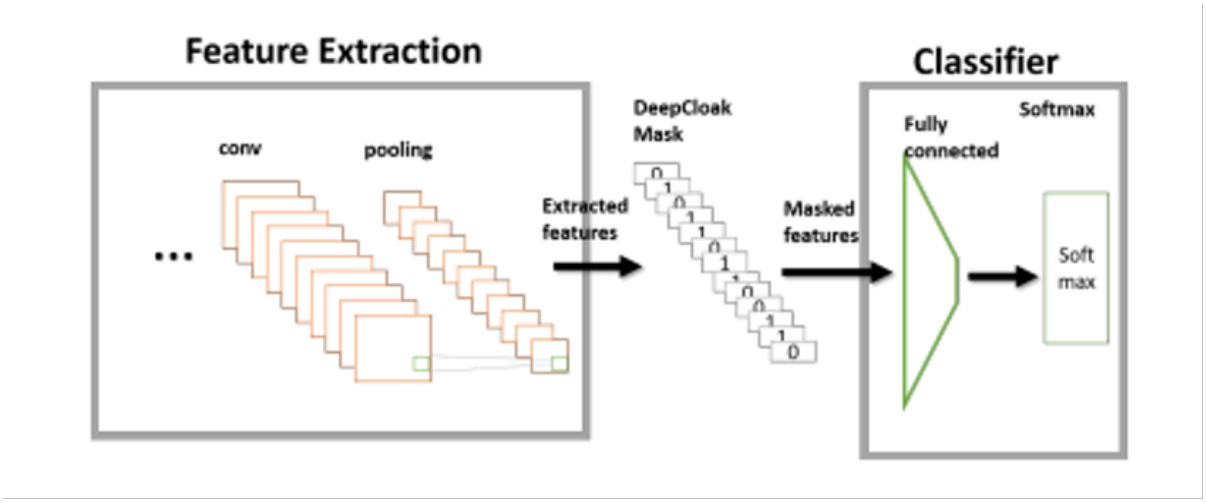

Figure 2.4 Mask Defense algorithm (Image credit: Gao et al. [26])

a. Extract the features of all original images ($\mathbf{x}$) and their respective adversarial images($\mathbf{x}'$).
b. Accumulate their differences for all the training data.
c. Mask top k entries in the above-accumulated difference values to 0, and the rest are masked with 1.
d. Introduce the above mask into the network before the classification layer.

Using Auxiliary tools: Below are a few techniques for defending adversarial attacks.

1. MagNet: A framework called MagNet was presented by Meng et al. [48] that reads the output of

the classifier's final layer as a black box without reading any data from the inner layer or changing the classifier itself. As shown in Figure 2.5, an autoencoder is used in Detect and Reform modules. Detect module is trained with original samples where an autoencoder is designed to minimize reconstructed divergence between input and output. Similarly, the autoencoder is trained to produce output with reduced reproduction error for the Reform module. Detect module is used to differentiate whether input data is the original or adversarial sample and if the Jensen-Shanon divergence value is greater than the threshold. If it is less than the threshold value, it is passed to the reformer module to reconstruct the approximated original sample from the adversarial before passing it to the classifier. This method performs well for black box attacks only. 2. Defense-GAN: Here, Wasserstein GANs are considered and

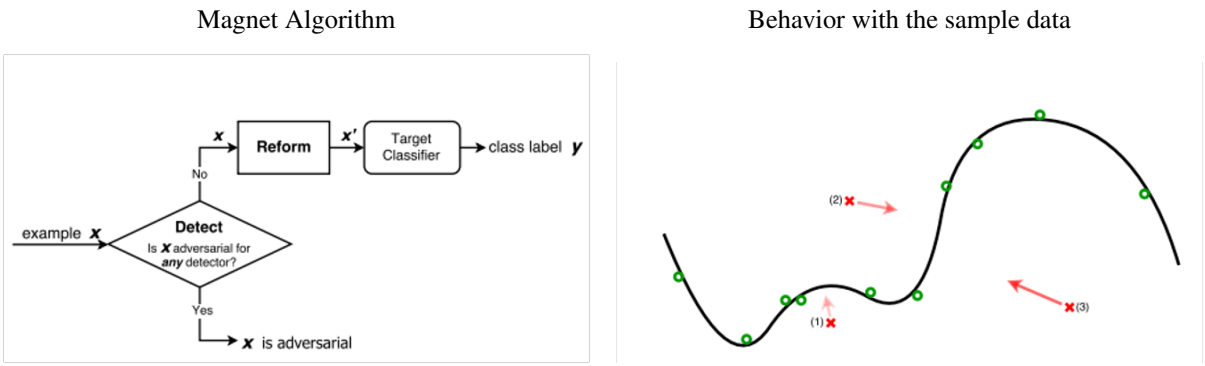

Figure 2.5 Magnet: Curved plot represents original samples, and red spots depict adversarial samples. Arrows indicate the nearest transformation of the sample using a reformer. (Image credit: Meng et al. [48])

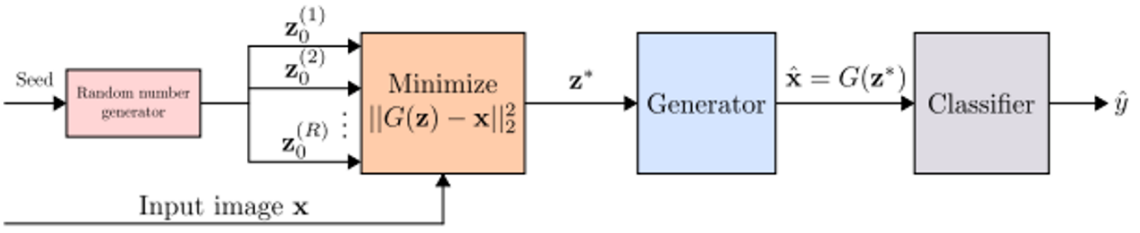

Figure 2.6 Defense GAN algorithm (Image credit: Samangouei et al. [60])

as proposed by Samangouei et al. [60] trained on original samples (both generator and discriminator). During testing time, the adversarial example is fed to the generator, and $\mathbf{z}$ (random latent vector) is updated for L steps using gradient descent (GD) to reduce the reconstruction error $\|G(\mathbf{z}) - \mathbf{x}\|_2$ before feeding $\mathbf{z}$ to the classifier. As GANs are trained on original data and can represent it, the reconstruction of the data with adversarial images should be consistent with the actual data distribution. This method is capable of handling both white and black box attacks.

Chapter 3

Literature Survey On Machine Learning Using Chance Constraint Optimization Methods

Chance constraint optimization methods are a class of mathematical techniques used in various areas to handle uncertainty and risks associated with decision-making processes. In these methods, constraints are formulated probabilistically, allowing decision-makers to account for the uncertainty in the system or environment explicitly. Instead of enforcing hard constraints, chance constraints enable the optimization process to maintain a certain feasibility level with a specified probability. This method is very helpful when uncertainty is present due to various factors, such as noisy data, incomplete information, or random fluctuations. In machine learning, chance constraint optimization methods are crucial in ensuring model reliability and robustness, especially in applications where uncertainty in data or model parameters can impact decision-making accuracy. By integrating uncertainty into the optimization process, chance constraint methods provide a powerful tool to make more informed and risk-aware decisions in diverse applications.

Given the numerous applications of CCP, we have specifically focused on the relevant literature survey required for multiclass classification scenarios in the below sections.

3.1 Chance-Constrained Programs for Link Prediction

The paper [23] introduces a novel framework for link prediction in evolving networks. The link prediction problem involves predicting potential future interactions between nodes in a network given a partial current state. Traditional prediction methods suffer from low accuracy due to extreme class skew and many potential links. The authors propose learning algorithms based on chance-constrained programs (CCPs) that exhibit the necessary properties for an effective link predictor to address this. The CCP framework allows for preferential bias towards positive or negative classes, handles data skewness, and scales to large networks. Two formulations are presented in the paper [23]: Clustering-based Second Order Cone Program (CBSOCP) and Max-Margin Formulation with specified Lower Bounds on

Accuracy (LBSOCP). Below are the notations:

$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$: The training dataset, where $\mathbf{x}_i \in \mathbb{R}^n$ is a feature vector defined between two nodes, and $y_i \in \{-1, +1\}$ is the corresponding label representing the presence or absence of an edge between the two nodes.

$\mathbf{w} \in \mathbb{R}^n$: The weight vector of the classifier.

$b \in \mathbb{R}$: The bias term of the classifier.

$\xi_i \geq 0$: Slack variables used to handle noise in the data during optimization.

κ_1, κ_2 : Scaling factors used to convert probabilistic constraints to deterministic ones.

where $\kappa_i = \sqrt{\frac{\eta_i}{1-\eta_i}}$ for $i = 1, 2$, where η_i represents the lower bound on classification accuracy for the respective class.

η_i : Lower bound on classification accuracy for the respective class.

k_1 and k_2 : The number of components in the mixture model for the positive and negative classes, respectively.

$\boldsymbol{\mu}_i$ and σ_i : The second-order moments (mean and standard deviation) of the clusters for the positive class, and $\boldsymbol{\mu}_j$ and σ_j for the negative class.

C : A hyperparameter representing the trade-off between maximizing the margin and minimizing the classification error.

Formulation 3.1.1 - Clustering-based SOCP (CBSOCP):

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{k} \xi_i \\
\text{subject to} \quad & y_i(\mathbf{w}^T \boldsymbol{\mu}_i - b) \geq 1 - \xi_i + \kappa_1 \sigma_i W, \quad \forall i = 1, \ldots, k_1 \\
& y_j(\mathbf{w}^T \boldsymbol{\mu}_j - b) \geq 1 - \xi_j + \kappa_2 \sigma_j W, \quad \forall j = 1, \ldots, k_2 \\
& W \geq \|\mathbf{w}\|_2, \\
& \xi_i \geq 0, \quad \forall i = 1, \ldots, k_1 + k_2
\end{aligned}
$$

Formulation 3.1.2 - Max-Margin Formulation with specified Lower Bounds on Accuracy (LBSOCP):

$$
\begin{aligned}
\text{minimize:} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\
\text{subject to:} \quad & t \geq \|\mathbf{w}\|^2 \\
& \mathbf{w}^T \boldsymbol{\mu}_1 - b \geq 1 + \kappa_1 C_1^T \mathbf{w} \\
& b - \mathbf{w}^T \boldsymbol{\mu}_1 \geq 1 + \kappa_2 C_2^T \mathbf{w}; \text{ where } \Sigma_1 = C_1 C_1^T \text{ and } \Sigma_2 = C_2 C_2^T
\end{aligned}
$$

The authors evaluate the proposed algorithms on real-world co-authorship networks in the experiments, including the Digital Bibliography & Library Project (DBLP), Genetics, and Biochemistry datasets. The results demonstrate that both CBSOCP and LBSOCP significantly outperform traditional classifiers like Support Vector Machines (SVMs) and Perceptron with Uneven Margins (PAUM) regarding precision and recall. Notably, LBSOCP performs better than CBSOCP. Moreover, LBSOCP and CBSOCP achieve a high recall within the top-k predictions, making them effective for applications like recommendation and collaborative filtering. The formulations are scalable and computationally efficient, enabling their application to large networks. The chance-constrained uncertain classification approach presented in the paper [23] offers a promising solution to the link prediction problem, addressing challenges posed by data skewness and potential links in evolving networks.

3.2 Interval Data Classification under Partial Information

The paper [7] introduces a novel methodology for constructing maximum-margin classifiers that are robust to interval-valued uncertainty in examples. The key idea is to employ chance constraints to ensure that uncertain examples are classified correctly with high probability. The main novelty lies in using Bernstein bounding schemes to relax the chance-constrained program into a convex second-order cone program (SOCP). Unlike existing methods, the proposed relaxations require only the knowledge of the support and mean of the uncertain examples and make no assumptions about the underlying uncertainty distributions. As a result, the classifiers built using this approach model interval-valued uncertainty less conservatively, leading to improved generalization compared to state-of-the-art methods. Experimental results on synthetic and real-world datasets demonstrate the superiority of the proposed classifiers in handling interval-valued uncertainty.

The paper [7] addresses the increasing interest in analyzing interval-valued data in the learning community. In many real-world problems, data cannot be precisely described, making intervals a more appropriate representation. Examples include cancer diagnosis, where features of tumorous tissue vary among cells, and gene-expression data, which often involve noisy experiments with replicates. Representing uncertainty as interval data has been shown to yield accurate learning algorithms. The proposed maximum-margin classification formulation uses means and bounding hyper-rectangles of interval-valued training examples to build the decision function. By modeling interval-valued uncertainty using Chance-Constrained Programming (CCP), the paper [7] approximates the CCP as a SOCP using Bernstein schemes. Geometric interpretation reveals that the proposed classifier views each uncertain example as a region of intersection between its bounding hyper-rectangle and an ellipsoid centered at its mean, making it less conservative than methods using bounding hyper-rectangles alone. The classifier's expected generalization is emphasized as it directly relates to its conservativeness and margin achieved.

Formulations: The chance-constrained optimization problem for the proposed interval data classifier (Interval Classifier with Maximum Margin and Bernstein Relaxations (IC-MBH)) is given by:

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{Subject to:} \quad y_i(\mathbf{w}^\top \boldsymbol{\mu}_i' - b) + z_i^\top \boldsymbol{\mu}_i \geq 1 - \xi_i + \kappa\|z_i\|_1 + \kappa\left\|\sum_i (y_i L_i \mathbf{w} + \mathbf{z}_i)\right\|_2$$

where $\kappa = \sqrt{2\log\left(\frac{1}{\varepsilon}\right)}$

$\boldsymbol{\mu}_{ij}' = E[\mathbf{x}_{i1}]$

$\boldsymbol{\mu}_i' = \begin{bmatrix}\boldsymbol{\mu}_{i1}' & \cdots & \boldsymbol{\mu}_{in}'\end{bmatrix}^\top$ is the vector of deviations of the expected values from the midpoints for the i^{th} training example.

$L_i = \text{diag}\left(\begin{bmatrix} l_{i1} & \cdots & l_{in}\end{bmatrix}\right)$

where l_{i1} is half of the difference between the upper and lower bound of uncertainty interval of feature 1 of i^{th} example

$\mathbf{z}_i : [z_{i1}, \ldots, z_{in}]^\top$ is the vector of deviation variables for the i^{th} training example.

Σ_i : diag($[\sigma(\boldsymbol{\mu}_{i1}), \ldots, \sigma(\boldsymbol{\mu}_{in})]$) is the diagonal matrix of deviations of the expected values from the midpoints for the i^{th} training example, represented by the functions $\sigma(\boldsymbol{\mu}_{ij})$

m : Number of training examples.

n : Number of features in each training example.

$\mathbf{w}$: Weight vector.

b : Bias term.

ξ_i : Slack variable for the i^{th} training example.

C : Regularization parameter.

y_i : Label (+1 or -1) for the i^{th} training example.

ϵ : user specified upper bound representing the probability of misclassification on interval data

The paper [7] uses two error measures: Nominal Error (NomErr) and Optimistic Error (OptErr). Nominal Error (NomErr) is used to evaluate the classification error on nominal (non-uncertain) examples. It measures how well the classifier performs on conventional, non-uncertain data points. Optimistic Error (OptErr) quantifies the classification error on uncertain examples. It evaluates the classifier's performance, specifically on interval-valued data, providing insights into how well the classifier handles uncertainty. In conclusion, the proposed IC-MBH classifier significantly advances handling interval-valued uncertainty in learning algorithms. The combination of chance constraints, Bernstein bounding schemes, and the convex SOCP formulation enable improved generalization and robustness

to uncertainty. The paper's [7] contributions and promising results pave the way for further research and applications in real-world problems with imprecise data measurements.

3.3 Multiclass classification using Chance Constraint Optimization

Shivaswamy [63] proposed robust classifiers using second-order cone programming approaches for binary and multiclass classification problems under uncertain conditions. Specifically, they have proposed Support Vector Machine (SVM) formulations when missing attributes are in the input data.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the feature space from which the instances are drawn and $\mathcal{Y} = \{1, \ldots, L\}$ be the label space. The ground-truth label associated with $\mathbf{x}$ is denoted by lowercase y. The set of labels not equal to the candidate label is denoted by $\bar{y}$ (obviously, $y \cup \bar{y} = [L]$ where $[L] = \{1, \ldots, L\}$). Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be the training set. The objective is to learn a classifier $h : \mathcal{X} \to \mathcal{Y}$. Assume that $W \in \mathbb{R}^{d \times L}$ is a matrix of weights parameterizing the classifier $h(\mathbf{x})$ and $\mathbf{w}_i$ be the i^{th} column vector of W, denotes the parameter vector corresponding to the ith class. Let us assume that the training examples $\mathbf{x}_i,\ i = c+1 \ldots, n$ have missing feature values. To fill the missing feature values in $\mathbf{x}_i\ (i = 1, \ldots, c)$, we use some imputation method. Let $\bar{\mathbf{x}}_i$ be the feature vector we get after imputing missing values in $\mathbf{x}_i$.

$$
\begin{aligned}
\min_{\mathbf{w},\xi} \quad & \frac{1}{2}\sum_{j=1}^{L} \|\mathbf{w}_j\|^2 + C\sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad & \left[\mathbf{w}_{y_i} - \mathbf{w}_j\right]^\top \boldsymbol{\mu}_{y_i} \geq 1 - \xi_i + \gamma_i \left|\left| \Sigma_{y_i}^{\frac{1}{2}} \left(\mathbf{w}_{y_i} - \mathbf{w}_j\right) \right|\right|; \\
& \qquad\qquad\qquad\qquad \forall j \in \bar{y}_i; \quad i = c+1 \ldots n \\
& \left[\mathbf{w}_{y_i} - \mathbf{w}_j\right]^\top \mathbf{x}_i \geq 1 - \xi_i; \quad \forall j \in \bar{y}_i; \quad i = 1 \ldots c \\
& \xi_i \geq 0; \quad i = 1 \ldots c
\end{aligned}
\tag{3.1}
$$

where $\gamma_i = \left(\frac{\kappa_i}{1-\kappa_i}\right)^{1/2}$ and κ_i denotes the probability that the sample lie on the correct side of the classification region within the margin as given below:

$$
p\left(\left(\mathbf{w}_{y_i} - \mathbf{w}_j\right)^\top \bar{\mathbf{x}}_i \geq 1 - \xi_i\right) \geq \kappa_i,\ \forall j \in \bar{y}_i; \quad i = c+1 \ldots n
$$

Note that for $i = c+1 \ldots n$, the separability constraints force the mean $\boldsymbol{\mu}_{Y_i}$ of candidate label y_i to achieve higher score with $\mathbf{w}_{y_i}$ compared to any other $\mathbf{w}_y\ (y \notin y_i)$.

Similarly, for the kernelized formulations is given below:

$$
\begin{aligned}
&\min_{\xi} \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad &\left[\boldsymbol{\alpha}^{y_i} - \boldsymbol{\alpha}^{j}\right]^{\top} \tilde{K}(\boldsymbol{\mu}_{y_i}) \geq 1 - \xi_i + \gamma_i \left|\left| \Sigma_{\bar{y}_i}^{\frac{1}{2}} \left(\boldsymbol{\alpha}^{y_i} - \boldsymbol{\alpha}^{j}\right) \right|\right|; \\
&\qquad\qquad\qquad\qquad \forall j \in \bar{y}_i,\ i = c+1 \ldots n \\
&\left[\boldsymbol{\alpha}^{y_i} - \boldsymbol{\alpha}^{j}\right]^{\top} \tilde{K}(\mathbf{x}_i) \geq 1 - \xi_i; \quad \forall j \in \bar{y}_i,\ i = 1 \ldots c \\
&\sum_{j=1}^{L} \|\boldsymbol{\alpha}^{j}\|^2 \leq \Lambda; \qquad \xi_i \geq 0; \quad \forall 1 \leq i \leq n;
\end{aligned}
\tag{3.2}
$$

where $\gamma_i = \sqrt{\frac{\kappa_i}{1-\kappa_i}}$ and $\boldsymbol{\alpha}^y = [\alpha_1^y, \ldots, \alpha_n^y]^{\top}$. $\mathbf{w}_1, \ldots, \mathbf{w}_L$ are constructed as follows.

$$
\mathbf{w}_y = \sum_{i=1}^{c} \alpha_i^y \phi(\mathbf{x}_i) + \sum_{i=c+1}^{n} \bar{\alpha}_i^y \phi(\bar{\mathbf{x}}_i), \quad y = 1 \ldots L
$$

[43] proposed second order cone programming formulations for multiclass SVM for one vs one and one vs rest methods. They also proposed a novel algorithm for multiclass svm using the same as mentioned below.

$$
\begin{aligned}
&\min_{\mathbf{w_i}, b_i} \quad \frac{1}{2} \sum_{j=1}^{L} \|\mathbf{w}_j\|^2 + \sum_{i=1}^{L} \sum_{j=1}^{i-1} \|\mathbf{w}_i - \mathbf{w}_j\|^2 \\
&\text{s.t.} \qquad \left[\mathbf{w}_i - \mathbf{w}_j\right]^{\top} \boldsymbol{\mu}_i - (b_i - b_j) \geq 1 + \gamma_{ij} \left|\left| \Sigma_i^{\frac{1}{2}} \left(\mathbf{w}_i - \mathbf{w}_j\right) \right|\right|; \quad i, j = 1 \ldots L; \quad i \neq j;
\end{aligned}
\tag{3.3}
$$

where $\gamma_{ij} = \sqrt{\frac{\kappa_{ij}}{1-\kappa_{ij}}}$

Chapter 4

Robust Learning With Missing Feature Values And Partial Labels

4.1 Introduction

In machine learning, multiclass classification is a well-researched problem. Most state-of-the-art algorithms assume the training data contains actual labels for each example. However, data labeling is costly and time-consuming, which makes access to actual class labels challenging. An alternative to a true label is a set of candidate labels called partial labels. Multiclass Learning with partial labels [42] is the name of this configuration. Learning can happen successfully if the partial label set contains the true label. Partially labeled data is more accessible and offers a less expensive option for learning with precise labels.

For learning with partial labels, [14] describes a general approach for risk minimization. This approach can adjust any common convex loss function to work in the partial label environment. Since the ground-truth label is not available for a specific instance, the loss is determined by taking the average of the candidate label set's scores as a stand-in.

Let $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ denote the patterns and accompanying labels, respectively. The conventional machine learning formula only considers the situation in which $(\mathbf{x}, y)$ are given precisely. However, this is not always the case. Real-world data is often noisy, partial, or ambiguous due to a variety of reasons, including measurement errors, data gathering restrictions, or inherent variability. Ignoring or overlooking this ambiguity can lead to biased or inaccurate models. For instance, when there are missing values, we may be able to estimate the values of the missing variables, albeit with some uncertainty (using a secondary estimation procedure). Other times, the observations might be purposefully suppressed. In other cases, the data may represent an entire class of equivalent observations (e.g., in optical character recognition, all digits, their translations, small rotations, slanted versions, etc., bear the same label). Therefore, designing estimators with the potential range of such data in mind is only natural. In the situation of missing variables, what we propose in the present work goes beyond the conventional imputation method. Instead, we incorporate the incomplete determination of some observations into the optimization problem, resulting in convex programming formulations.

Multiclass learning for the partially labeled data with missing values remained an unaddressed problem. This paper's key contribution is to propose a support vector machine (SVM) formulation for learning efficient multiclass classifiers with missing attributes and partial labels [15]. In this formulation, we use chance-constrained programming, which turns out to be a second-order cone program (SOCP) [43]. This approach can learn classifiers for any distribution with finite mean and covariance. We call the proposed formulation RPL-SVM, which stands for robust partial label support vector machine. We compare RPL-SVM with state-of-the-art baselines to show its effectiveness.

4.2 Multiclass Classification Using Partially Labeled Data (PL-SVM)

Let us consider the problem of multiclass classification given a partially labeled training set. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the feature space from which the instances are drawn and $\mathcal{Y} = \{1, \ldots, L\}$ be the label space. In the partial label setting, every instance $\mathbf{x} \in \mathcal{X}$ is associated with a candidate label set $Y \subseteq \mathcal{Y}$. The set of labels not present in the candidate label set is denoted by $\bar{Y}$. The ground-truth label associated with $\mathbf{x}$ is denoted by lowercase y. It is assumed that the actual label y lies within the set Y (i.e., $y \in Y$). Let $\{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_n, Y_n)\}$ be the training set. The objective is to learn a classifier $h : \mathcal{X} \to \mathcal{Y}$. Let $W \in \mathbb{R}^{d \times L}$ be the weight matrix of the classifier $h(\mathbf{x})$. The i^{th} column vector $\mathbf{w}_i$ of W denotes the weight vector of the i^{th} class. The maximum margin approach to learning a multiclass classifier using partial labels solves the following optimization problem [13] [58] [70].

$$\begin{aligned} \min_{\mathbf{w},\xi} \; & \frac{1}{2} \sum_{j=1}^{L} \|\mathbf{w}_j\|^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \; & \left[\frac{1}{|Y_i|} \sum_{y \in Y_i} \mathbf{w}_y - \mathbf{w}_j \right]^\top \mathbf{x}_i \geq 1 - \xi_i; \; \forall j \in \bar{Y}_i \quad i = 1 \ldots n \\ & \xi_i \geq 0; \quad i = 1 \ldots n \end{aligned} \tag{4.1}$$

Where $\bar{Y}_i = \mathcal{Y} \setminus Y_i$. In the above formulation, C is used set a trade-off between sum of errors $\sum_{i-1}^{n} \xi_i$ and complexity term $\sum_{j=1}^{L} \|\mathbf{w}_j\|^2$.

4.3 RPL-SVM Linear: Robust Formulation For Linear Classifiers With Partial Labels and Missing Values

Here, we propose an RPL-SVM formulation that can learn robust linear classifiers. The above formulation for partial label learning assumes that example $\mathbf{x}$ does not suffer from missing values and other noise processes. However, in many practical systems, the data is affected by such issues (e.g., sensor data, images, etc.).

Let us assume that the training examples $\mathbf{x}_i,\ i = c+1\ldots,n$ have missing feature values. We use an imputation method to fill in these missing feature values. Let $\bar{\mathbf{x}}_i$ be the feature vector we get after imputing missing values in $\mathbf{x}_i$. The imputation method may add some feature noise in $\mathbf{x}_i$, which can cause separability constraints hard to satisfy. Thus, we would like to satisfy the separability constraints with a very high probability. In this case, we replace the separability constraint involving $\mathbf{x}_i$ with a probabilistic constraint as below.

$$p\left(\left(\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right)^\top \bar{\mathbf{x}}_i \geq 1-\xi_i\right) \geq \kappa_i,\ \forall j \in \bar{Y}_i; \quad i = c+1\ldots n$$

The probability is concerning the distribution of $\bar{\mathbf{x}}_i$. Here, we require that $\bar{\mathbf{x}}_i$ be correctly classified with probability greater than κ_i. κ_i ranges in the interval (0,1]. A large value of κ_i will result in a good classifier with a low risk of making mistakes. Directly solving this probability constraint is challenging. We assume that the second moment of $\bar{\mathbf{x}}_i$ exists to solve this. Note that Y_i is the partial label set for $\bar{\mathbf{x}}_i$ and each label in Y_i is an equally good candidate for the class label of $\bar{\mathbf{x}}_i$. Thus, to consider the first and second moment of $\bar{\mathbf{x}}_i$, we need to consider all the examples having Y_i as a partial label set. Let $\boldsymbol{\mu}_{Y_i}$ and Σ_{Y_i} be the mean and covariance matrix corresponding to the partial label set Y_i. We want the above inequality to hold even for the worst-case distribution having mean $\boldsymbol{\mu}_{Y_i}$ and covariance Σ_{Y_i}. Thus, we want the following to hold for $i = c+1\ldots n$.

$$\inf_{\bar{\mathbf{x}}_i\in(\boldsymbol{\mu}_{Y_i},\Sigma_{Y_i})} p\left(\left(\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right)^\top \bar{\mathbf{x}}_i \geq 1-\xi_i\right) \geq \kappa_i,\ \forall j \in \bar{Y}_i$$

We get the following conditions using multivariate Chebyshev inequality [47, 37, 10].

$$\sup_{\bar{\mathbf{x}}_i\in(\boldsymbol{\mu}_{Y_i},\Sigma_{Y_i})} p\left(\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_{y_i} - \mathbf{w}_j\right]^\top \bar{\mathbf{x}}_i \leq 1-\xi_i\right) = \frac{1}{1+d^2} \leq 1-\kappa_i; \qquad (4.2)$$
$$\forall j \in \bar{Y}_i,\ i = c+1\ldots n$$

where $d^2 = \inf_{\left[\frac{1}{|Y_i|}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right]^\top \bar{\mathbf{x}}_i \leq 1-\xi_i} (\bar{\mathbf{x}}_i - \boldsymbol{\mu}_{Y_i})^\top \Sigma_{Y_i}^{-1}(\bar{\mathbf{x}}_i - \boldsymbol{\mu}_{Y_i})$. This constraint always holds for a family of distributions with the same second-order moments. In the worst case, equality is reached. Constraint in eq.(4.2) is rewritten as

$$\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right]^\top \boldsymbol{\mu}_{Y_i} - (1-\xi_i) \geq \gamma_i \left\|\Sigma_{Y_i}^{1/2}\left(\frac{1}{|Y_i|}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right)\right\|$$

where $\gamma_i = \left(\frac{\kappa_i}{1-\kappa_i}\right)^{1/2}$. The derivation of the above is provided in Appendix A.1. Thus, the robust optimization problem for linearly separable partial label data with missing input values is as follows.

$$
\begin{aligned}
\min_{\mathbf{w},\xi} \quad & \frac{1}{2}\sum_{j=1}^{L}\|\mathbf{w}_j\|^2 + C\sum_{i=1}^{n}\xi_i \\
\text{s.t.} \quad & \left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right]^\top \boldsymbol{\mu}_{Y_i} \geq 1 - \xi_i + \gamma_i \left\| \Sigma_{Y_i}^{\frac{1}{2}} \left(\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right)\right\|; \\
& \qquad\qquad \forall j \in \bar{Y}_i; \quad i = c+1 \ldots n \\
& \left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right]^\top \mathbf{x}_i \geq 1 - \xi_i; \quad \forall j \in \bar{Y}_i; \quad i = 1 \ldots c \\
& \xi_i \geq 0; \quad i = 1 \ldots c
\end{aligned}
\tag{4.3}
$$

Note that for $i = c+1 \ldots n$, the separability constraints force the mean $\boldsymbol{\mu}_{Y_i}$ of partial label set Y_i to achieve higher score with $\frac{1}{Y_i}\sum_{y_i\in Y_i}\mathbf{w}_{y_i}$ compared to any other $\mathbf{w}_y$ $(y \notin Y_i)$. On the other hand, examples $i = 1, \ldots, c$ are not facing any missing attribute. Thus, for $i = 1, \ldots, c$, we require that the example $\mathbf{x}_i$ achieve higher score with $\frac{1}{\bar{Y}_i}\sum_{y_i\in Y_i}\mathbf{w}_{y_i}$ compared to any other $\mathbf{w}_y$ $(y \notin Y_i)$.

4.4 RPL-SVM Nonlinear: Robust Formulation For Nonlinear Classifiers With Partial Labels

In this section, we propose an RPL-SVM formulation for learning nonlinear classifiers. We first assume a feature map $\phi : \mathbb{R}^d \to \mathcal{H}$ corresponding to a Mercer's kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$. Where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS) corresponding to the kernel K. We assume that each $\mathbf{w}_y$ $(y = 1 \ldots L)$ is a linear combination of all the examples.

$$
\mathbf{w}_y = \sum_{i=1}^{c}\alpha_i^y\phi(\mathbf{x}_i) + \sum_{i=c+1}^{n}\bar{\alpha}_i^y\phi(\bar{\mathbf{x}}_i), \quad y = 1 \ldots L \tag{4.4}
$$

Where $\bar{\mathbf{x}}_i$ is achieved by imputing missing values of $\mathbf{x}_i$. Let $\gamma_i = \sqrt{\frac{\kappa_i}{1-\kappa_i}}$,
$\tilde{K}(\bar{\mathbf{x}}_i) = [K(\mathbf{x}_1, \bar{\mathbf{x}}_i), \ldots, K(\mathbf{x}_c, \bar{\mathbf{x}}_i), K(\bar{\mathbf{x}}_{c+1}, \bar{\mathbf{x}}_i), \ldots, K(\bar{\mathbf{x}}_n, \bar{\mathbf{x}}_i)]^\top$, $\boldsymbol{\alpha}^y = [\alpha_1^y, \ldots, \alpha_c^y, \bar{\alpha}_{c+1}^y, \ldots, \bar{\alpha}_n^y]^\top$,
$\tilde{K}(\boldsymbol{\mu}_{Y_i}) = [K(\mathbf{x}_1, \boldsymbol{\mu}_{Y_i}), \ldots, K(\mathbf{x}_c, \boldsymbol{\mu}_{Y_i}), K(\bar{\mathbf{x}}_{c+1}, \boldsymbol{\mu}_{Y_i}), \ldots, K(\bar{\mathbf{x}}_n, \boldsymbol{\mu}_{Y_i})]^\top$ and $\Sigma_{Y_i}$is the covariance of $\tilde{K}(\mathbf{x}_i)$ which is in $\tilde{K}$-space. Then, RPL-SVM solves the following optimization problem to learn

nonlinear classifiers using partial label data with missing attributes.

$$
\begin{aligned}
&\min_{\xi} \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad &\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\boldsymbol{\alpha}^y - \boldsymbol{\alpha}^j\right]^\top \tilde{K}(\boldsymbol{\mu}_{Y_i}) \geq 1 - \xi_i + \gamma_i \left\| \Sigma_{Y_i}^{\frac{1}{2}} \left(\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\boldsymbol{\alpha}^y - \boldsymbol{\alpha}^j \right) \right\|; \\
&\qquad\qquad \forall j \in \bar{Y}_i,\ i = c+1 \ldots n \\
&\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\boldsymbol{\alpha}^y - \boldsymbol{\alpha}^j\right]^\top \tilde{K}(\mathbf{x}_i) \geq 1 - \xi_i; \quad \forall y \in \bar{Y}_i,\ i = 1 \ldots c \\
&\sum_{j=1}^{L} \|\boldsymbol{\alpha}^j\|^2 \leq \Lambda; \qquad \xi_i \geq 0; \quad \forall 1 \leq i \leq n;
\end{aligned}
\tag{4.5}
$$

The derivation of the above formulation is provided in Appendix A.2.

4.5 Imputing Missing Values

We use the expectation-maximization (EM) based algorithm proposed in [18] to impute the missing values. This approach is a specific implementation of the EM algorithm to handle missing attributes. The algorithm proceeds iteratively, starting with the initial mean and covariance estimates. These initial estimates are used to impute the missing values for the entire dataset, taking into account any relationships or dependencies between the observed and missing components.

Once the missing values are imputed, the algorithm calculates the likelihood of the observed and imputed data, considering the uncertainty introduced by the missing values. This likelihood is used to update the mean and covariance estimates, incorporating the information gained from the imputed values. The imputation and parameter estimation process is repeated iteratively until the algorithm converges, meaning that the estimates stabilize and no significant changes occur between iterations. By iteratively imputing missing values and refining parameter estimates, the algorithm leverages the available information to infer the underlying patterns and structure in the data, even in the presence of missing or incomplete information. Below are the steps followed in our experiments using the Dempster algorithm [18] for handling the missing attributes:

We start by calculating each partial label set's sample mean and covariance from the available observations. We consider any missing variables necessary with a linear model and Expectation Maximisation (EM). Let $(\mathbf{x}, Y)$ be composed of the parts $\mathbf{x}_a$ and $\mathbf{x}_p$, which stand for the missing and available components, respectively with a mean ($\boldsymbol{\mu}_{\mathbf{x}}$) and covariance ($\Sigma_Y$) and its decomposition calculated from the

partially labeled data set Y is given below.

$$\boldsymbol{\mu}_{\mathbf{x}} = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{x}_p} \\ \boldsymbol{\mu}_{\mathbf{x}_a} \end{bmatrix} \qquad \Sigma_Y = \begin{bmatrix} \Sigma_{Y_{pp}} & \Sigma_{Y_{pa}} \\ \\ \Sigma_{Y_{pa}}^\top & \Sigma_{Y_{aa}} \end{bmatrix}$$

The imputed means and covariances are calculated using the below formulas.

$$E\left[\mathbf{x}_a\right] = \boldsymbol{\mu}_{\mathbf{x}_a} + \Sigma_{Y_{ap}} \Sigma_{Y_{pp}}^{-1} (\mathbf{x}_p - \boldsymbol{\mu}_{\mathbf{x}_p})$$
$$E\left[\mathbf{x}_a \mathbf{x}_a^T\right] - E\left[\mathbf{x}_a\right] E\left[\mathbf{x}_a\right]^\top = \Sigma_{Y_{aa}} - \Sigma_{Y_{ap}} \Sigma_{Y_{pp}}^{-1} \Sigma_{Y_{ap}}^T$$

In this algorithm, we start with initial mean and covariance estimates, using them to impute the missing values. Then iterates by re-estimating the mean and covariance until convergence.

4.6 Experiments

In this section, we show the experimental results of the proposed approach. We perform experiments on Wine, Glass4, IRIS, Dermatology, Waveform, SatImage, and Shuttle 5 datasets, available on the UCI repository [8].

4.6.1 Baselines Used

For benchmarking, we use the following three state-of-the-art approaches.

1. Partial label multiclass SVM (PL-SVM) [14]: This is SVM based approach for multiclass classification using partial labels. It uses kernels for nonlinear classifiers.

2. Partial label KNN (PL-KNN) [34]: This approach uses a generalization of KNN for learning with partial labels. We use $k = 5$ for our experiments.

3. PICO [74]: This is a deep learning approach for learning with partial labels.

4.6.2 Data Preparation

We divide each dataset into train and test data with a 67:33 ratio.

4.6.2.1 Introducing Missing Features

We ensure that 50% of training samples per class contain missing feature values. In a feature vector, some $R\%$ of feature values are deleted to create missing attribute data. Based on the datasets, we considered different values for the percentage of missing attributes (R). We have tested with different percentages of missing attributes (25%, 50%, and 75%)

4.6.2.2 Creating Partial Labels

We create partial labels for training data as follows. Let k be the total number of classes. To create a partial label set of size c, first, we keep the original label in the set. The rest of the $c-1$ labels from the remaining $k-1$ labels are chosen randomly. We considered partial label sets of sizes 2 and 4 for Dermatology, SatImage, and Shuttle5 datasets. We considered partial label set sizes of 2 and 3 for the Glass4 dataset. We considered partial label sets size 2 for IRIS, Wine, and Waveform datasets.

4.6.3 Experimental Setup

Once missing values are filled using the imputation algorithm [18], we learn classifiers using the proposed algorithm RPL-SVM and the baseline algorithms. We compare the test accuracy of RPL-SVM with PL-SVM, PL-KNN, and PICO. We calculated each algorithm's average test data classification accuracy by repeating the experiments 50 times. For RPL-SVM, we have considered various κ values. We used cvxpy [19], an open-source Python-embedded modeling language for convex optimization, with an MOSEK solver for solving the optimization problem. The experiments are conducted on a 2.6 GHz 6-Core Intel Core i7 processor with 16 GB 2667 MHz DDR4 memory.

4.6.4 Performance Comparison Results of RPL-SVM with Baselines

Table 4.1 provides average test error rates observed for different datasets with different configurations. We have considered kernel-based formulations in the proposed RPL-SVM and baseline PL-SVM approach. For RPL-SVM, we have reported the results with the best κ value for comparison. For RPL-SVM and PL-SVM, we consider polynomial kernel with degree 2 and Gaussian kernel in our experiments. We report all algorithms' average test error rates and standard deviation values. We only represented standard deviation values up to two decimals and ignored the rest.

We make the following observations. As the partial label set size increases or the percentage of missing attributes increases, the performance of RPL-SVM almost remains the same. Thus, it shows robustness against missing attributes and partial labels. RPL-SVM outperforms PL-SVM and PL-KNN for all configurations of missing attributes and partial label set sizes. Compared to PICO, RPL-SVM outperforms IRIS, Waveform, Wine, Glass4, and Dermatology datasets for all configurations of missing attributes and partial label set sizes. For the Shuttle5 dataset, RPL-SVM outperforms PICO for a partial label set of size two and all configurations of missing attribute percentages.

4.6.5 Effect of κ on RPL-SVM

Figure 4.1 provides the average test error rate for RPL-SVM with different partial label set sizes, percentages of missing attributes, and different κ values. Ideally, as the κ value increases, the constraints will be satisfied with high probability, resulting in better performance. Thus, ideally, test error should decrease with increasing value of κ. We observe this pattern for IRIS. However, this may not always

happen due to the following reason. For a given function class, the data may not satisfy the probabilistic constraints with high probability for small values of ξ. Thus, ξ values will increase, adding further to errors.

Dataset	**Partial Label Size**	**Missing Attributes**	**RPL-SVM**	**PL-SVM**	**PL-KNN**	**PICO**
IRIS	2	25%	**0.04 ± 0.02**	0.05± 0.03	0.1±0.04	0.04±0.03
		50%	**0.05 ± 0.02**	0.06± 0.03	0.1	0.06
Wave--form	2	25%	**0.14± 0.01**	0.15± 0.04	0.27	0.14± 0.02
		50%	**0.14± 0.01**	0.15	0.26	0.14± 0.02
		75%	**0.14± 0.01**	0.15± 0.01	0.27	0.14± 0.02
Wine	2	25%	**0.05± 0.02**	0.08± 0.02	0.31	0.16± 0.01
		50%	**0.06± 0.03**	0.09± 0.03	0.34	0.18
		75%	**0.08± 0.03**	0.11± 0.04	0.36	0.17
Glass4	2	25%	**0.04± 0.03**	0.1± 0.12	0.05	0.08
		50%	**0.06± 0.04**	0.25± 0.12	0.06	0.08
	3	25%	**0.1± 0.07**	0.24± 0.11	0.11	0.27
		50%	**0.1± 0.07**	0.32± 0.12	0.13	0.37
Derma--tology	2	10%	**0.02**	0.03	0.04	0.02± 0.01
		25%	**0.02**	0.03	0.04	0.08
	4	10%	**0.03± 0.01**	0.04± 0.02	0.1	0.3± 0.04
		25%	**0.03± 0.01**	0.04	0.1	0.3± 0.0.5
SatImage	2	25%	0.14± 0.01	0.17± 0.09	0.15	**0.12± 0.01**
		50%	0.14± 0.01	0.17± 0.09	0.15	**0.12± 0.01**
		75%	0.14	0.17± 0.09	0.15	**0.11**
	4	25%	0.22	0.4	0.24	**0.15**
		50%	0.22± 0.02	0.4± 0.04	0.24	**0.15± 0.01**
		75%	0.22± 0.02	0.4± 0.04	0.24	**0.17**
Shuttle5	2	25%	**0.02± 0.02**	0.13± 0.03	0.03	0.05
		50%	**0.02± 0.12**	0.13± 0.03	0.03	0.08
		75%	**0.02± 0.02**	0.13± 0.03	0.04	0.05
	4	25%	0.35± 0.05	0.45± 0.07	0.47	**0.08**
		50%	0.35± 0.05	0.45± 0.07	0.48	**0.08**
		75%	0.35± 0.06	0.45± 0.08	0.48	**0.07**

Table 4.1 Experiment datasets description

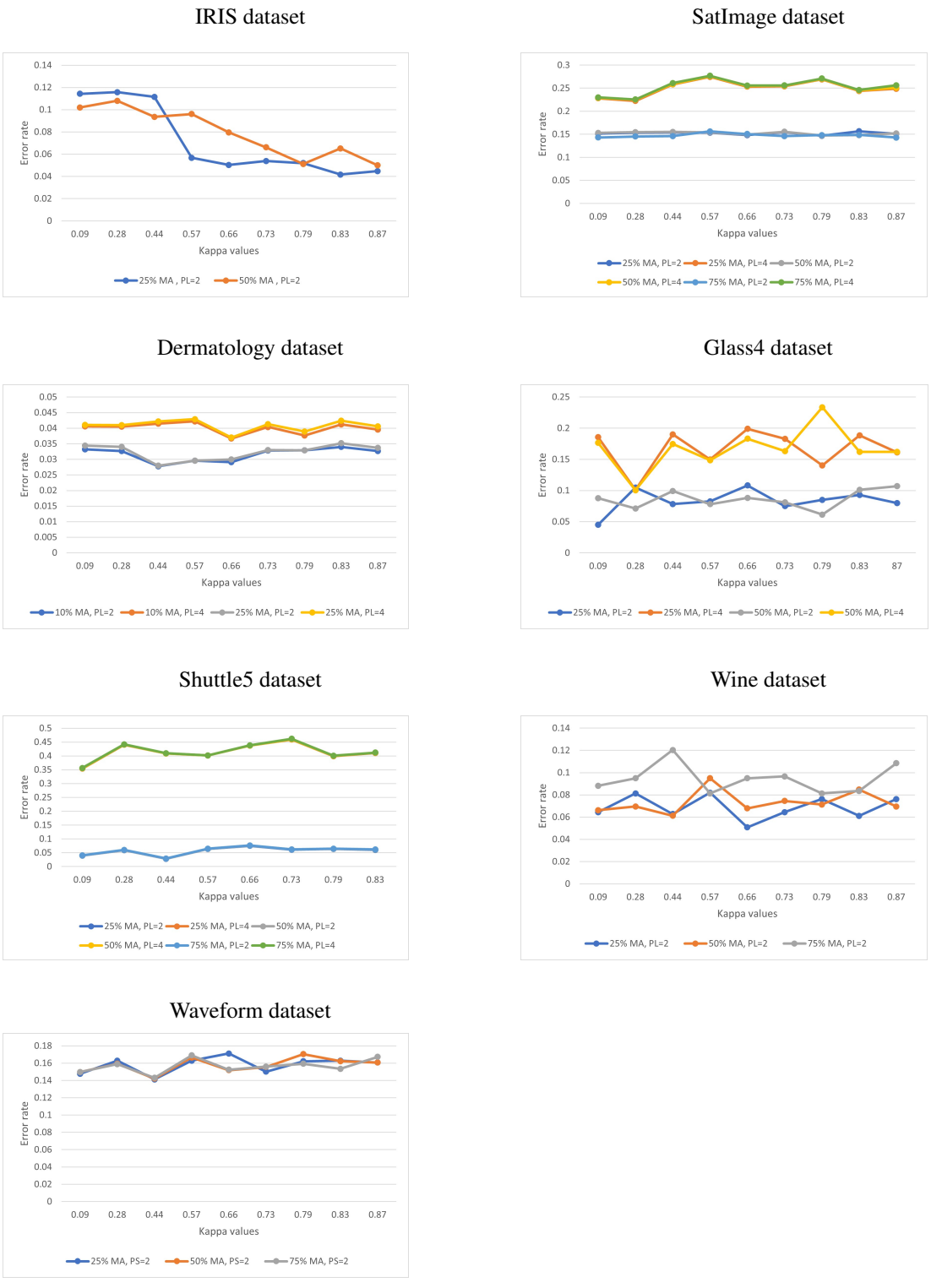

Figure 4.1 Average error rates for different κ values using RPL-SVM algorithm. MA- Missing attributes, PL- Partial label set size

4.7 Conclusions

We propose a novel approach, RPL-SVM, a multiclass SVM classification algorithm with missing attributes and partially labeled data. This is useful in real-life scenarios when multiple annotators give different labels for the same instance. We provide an experimental comparison of all the algorithms on balanced and unbalanced datasets. The results show that the proposed algorithm RPL-SVM outperforms PL-KNN, PICO, and PL-SVM methods. This can be extended further for bigger datasets requiring faster optimizers for solving SOCP formulation and multi-label classification problems.

Chapter 5

Momentum Iterative Gradient Sign Method Outperforms PGD Attacks

5.1 Introduction

Adversarial examples are machine learning model inputs that an attacker has purposefully constructed to cause the model to make a mistake. A good adversarial example from MIT is the one that produced the 3D-printed turtle that, when viewed from almost any angle, modern ImageNet trained image classifiers misclassify as a rifle [3]. Other research has found that making imperceptible changes to an image can fool a medical imaging system into correctly classifying a benign mole as malignant with 100 percent certainty [25] and that a few pieces of tape can fool a computer vision system into incorrectly classifying a stop sign as a speed limit sign [24]. A network is specifically trained on hostile samples as one of the few defenses against adversarial attacks that can withstand strong attacks. Unfortunately, traditional adversarial training on a large-scale dataset like ImageNet is impracticable due to the enormous expense of providing solid adversarial samples. A recent line of work focused on making adversarial training computationally efficient for deep learning models. Projected Gradient Descent (PGD) and Fast Gradient Sign Method (FGSM) are popular current techniques for generating adversarial examples efficiently. There is a trade-off between these two regarding robustness or training time.

Among these defense techniques, adversarial training with the PGD is considered one of the most effective ways to achieve moderate adversarial robustness. However, PGD requires too much training time since it takes multiple iterations to generate perturbations. On the other hand, adversarial training with the FGSM takes much less training time since it takes one step to generate perturbations but fails to increase adversarial robustness. When combined with random initialization, Eric Wong [77] showed that adversarial training with the FGSM is as effective as PGD-based training with a lower computation time cost. We propose our Momentum Iterative gradient sign methods with Reuse of Perturbations (MIRP). These perturbations introduced between epochs perform better than PGD, which was previously believed to be ineffective, rendering the method no more costly than standard training in practice. Our algorithm achieves better robustness to PGD adversarial training on CIFAR-10, CIFAR-100 datasets and is faster than PGD string adversarial training methods.

Deep neural networks (DNNs) have recently achieved state-of-the-art performance on image classification datasets such as ImageNet. For example, He et al.[30] have achieved 96.43 $\%$ top-5 test accuracy with their ResNet architecture, which is difficult to exceed for humans. [67] discovered that a network can misclassify an image by adding a very small perturbation to it, which humans will not notice the difference. These perturbed inputs are termed adversarial examples. The existence of adversarial examples threatens the security of systems incorporating machine learning models. For example, autonomous cars could be forced to crash, intruders could confound face recognition software or SPAM filters, and other filters can be bypassed. Also, adversarial examples can often be generated not only with access to the models architecture and parameters but also without it. Therefore, it is important to understand the weakness of our models and come up with ways to defend them against potential adversaries. One popular approximation method successfully used in adversarial training is the PGD attack, as it remains empirically robust to this day.

5.2 Background

Neural networks are trained with first-order gradient descent algorithms. There exist two families of gradient descent algorithms: accelerated stochastic gradient descent (SGD) and adaptive learning rate. SGD-based algorithms include Momentum and Nesterov-based methods. Adaptive learning rates include Adam, Adadelta, and Adabelief. DNNs trained with SGD-based algorithms have a strong generalization ability with a low convergence rate which is vice-versa in adaptive family-based algorithms. In general, solutions based on these optimization families are considered for improving the transferability of adversarial examples.

Among many attempts [49, 22, 53, 36, 39, 71, 57] , adversarial training is the most extensively investigated way to increase the robustness of DNNs [36, 71, 28]. However, running a strong PGD adversary within an inner loop of training is expensive, and some earlier work on this topic found that taking large but fewer steps did not always significantly change the resulting robustness of a network [75]. Thus, an inherent trade-off appears between computationally efficient approaches which aim at solving the optimum perturbation value in few iterations as possible and approaches which aim at solving the same problem more accurately but with more iterations. To be more specific, the PGD attack uses multiple steps of projected gradient descent (PGD), which is accurate but computationally expensive, whereas Fast Sign Gradient Method (FGSM) uses only one iteration of gradient descent, which is computationally efficient.

Recent research has examined "Fast and Free Adversarial training," where modified FGSM adversarial training achieves an accuracy closure to the model trained with PGD attacks. This approach aims to reduce the extra computational overhead of the PGD defense. These improvements include top performing training methods from DAWNBench competition [12] can train CIFAR-10 and CIFAR-100 architectures to standard benchmark metrics in mere minutes, using only a modest number of computational resources. Although some of the techniques can be quite problem specific for achieving bleeding

edge performance, more general techniques such as cyclic learning rates [64] and half-precision computations [50] have been quite successful in the top ranking submissions and can also be useful for adversarial training.

5.3 MIRP

With the Fast Adversarial training [77], updated FGSM training combined with random initialization is as effective as defense with PGD-based training. The key difference here is to use perturbation from the previous iteration as the initial starting point for the next iteration, i.e., starting with the previous mini batch's perturbation or initializing from a uniformly random perturbation allows FGSM adversarial training to succeed at being robust to full-strength PGD adversarial attacks. FGSM with random initialization algorithm is shown below:

Algorithm 1 FGSM with random initialization adversarial training for R epochs, given some radius ϵ, step size α, and a dataset of size M for a network f_θ

for $t = 1...R$ **do**
  for $i = 1...M$ **do**
    δ = Uniform (-δ,δ)
    $\delta = \delta + (\alpha * sign(\nabla_\delta l(f_\theta(x_i + \delta); y_i))$
    $\delta = max(min(\delta; \epsilon); -\epsilon)$
    $\theta = \theta - (\nabla_\delta l(f_\theta(x_i + \delta); y_i))$
  end for
end for

Also, with Boosting algorithm [21] (which does not use the sign(.) method) achieves better performance than FGSM but with PGD attacks, the accuracy drops to 0. On the above basis, we propose a new algorithm to use Momentum iterative gradient sign adversarial training with random initialization for the perturbation, as shown below.

Algorithm 2 MIRP adversarial training for R iterations, T epochs, given some radius ϵ, N PGD steps, step size α, and a dataset of size M for a network f_θ

for $t = 1...T$ **do**
  for $i = 1...M$ **do**
    for $j = 1...R$ **do**
      $m_{j+1} = \mu * m_j + \frac{(\nabla_\delta l(f_\theta(x_i+\delta);y_i))}{\|(\nabla_\delta l(f_\theta(x_i+\delta);y_i))\|_{L_1}}$
      $\delta = \delta + (\alpha * sign(m_{j+1}))$
      $\delta = max(min(\delta; \epsilon); -\epsilon)$
    end for
    $\theta = \theta - (\nabla_\delta l(f_\theta(x_i + \delta); y_i))$
  end for
end for

5.3.1 Results

Below tables 5.1, 5.2 shows the test accuracy and the training time of the different models tested on different datasets.

Method	Accuracy with PGD attack / Training time
MIRP	50.59 / 47 min
Std Momentum Iterative	0 / 47.12 min
FGSM with random initialization	47.53 / 24.6 min
PGD	49.75 / 89.6 min

Table 5.1 Standard and robust performances of various adversarial training methods on CIFAR-10 dataset

Method	Accuracy with PGD attack / Training time
MIRP	28.22 / 44.55 min
Std Momentum Iterative	0 / 44.16 min
FGSM with random initialization	24.88 / 23.19 min
PGD	27.36 / 92.16 min

Table 5.2 Standard and robust performances of various adversarial training methods on CIFAR-100 dataset

Here, the Standard Momentum Iterative method is based on [21], FGSM with random initialization is based on [77]. We used the same configuration settings like step size, cyclic learning rate, and mixed precision arithmetic as per Eric Wong [77]. All experiments using MIRP are carried out with random starting points and step size alpha = 1.25*epsilon. All PGD adversaries used at evaluation are run with ten random restarts for 50 iterations with ϵ = 8/255. Speedup with mixed precision was incorporated with the Apex amp package at the O2 optimization level without loss scaling for CIFAR-10/100 experiments. All experiments are tested on a Nvidia T4 machine.

Reuse between epochs: We tested multiple combinations between momentum iterative gradient values and perturbation, and below are the inferences drawn:

Decaying Factor (μ): Tested various values of Decaying factor from 0.1 to 0.6 and could see that model starts overfitting for any value beyond 0.1 as shown below:

m , δ	Accuracy with PGD attack / Training time
Non-zero, Non-zero	50.59 / 47 min
Non-zero, Zero	0 / 47.5 min
Zero , Non-zero	50.39 / 45.8 min
Zero , Zero	0 / 47.12 min

Table 5.3 Performance of MIRP on CIFAR-10 dataset with varying m and δ

Decaying factor	Accuracy with PGD attack / Training time
0.1	50.59 / 47 min
0.6	49.19 / 50.8 min

Table 5.4 Performance of MIRP on CIFAR-10 dataset with varying decaying factor

Here, Non-zero refers to the values retained between batches, and Zero refers to initializing values to zero between batches.

5.4 Conclusions

Our findings show that MIRP adversarial training can be more effective when used with random initialization than the more costly PGD adversarial training. As a result, we can learn adversarially robust classifiers for CIFAR10/100 in minutes. Leveraging these significant reductions in time to train robust models will allow future work to iterate even faster and accelerate research in learning models resistant to adversarial attacks.

Chapter 6

Features Normalisation And Standardisation (FNS) - An Unsupervised Approach For Detecting Adversarial Attacks For Medical Images

6.1 Introduction

Deep learning-based medical imaging systems have significantly improved the accuracy and efficiency of clinical prediction tasks, thanks to the development of deep learning algorithms and the availability of high-quality labeled medical imaging datasets. For example, [16] extracts feature from X-rays for lung disease categorization, [61] uses computed tomography (CT) scans to detect lung cancer, and [59] uses magnetic resonance imaging (MRI) scans to establish an early diagnosis of prostate cancer. Several healthcare start-ups, including Zebra Medical Vision and Aidoc have recently secured FDA certifications for their AI medical imaging systems. According to these FDA clearances, deep learning-based medical imaging systems could shortly be used for clinical diagnosis.

Parallel to advancements in deep learning-based medical imaging systems, so-called adversarial images have shown flaws in these systems in various clinical areas [25]. Adversarial images are purposely generated inputs to deep learning models to deceive image categorization. The method falsely labels "Pleural Thickening" as "Pneumothorax" when only minor perturbations are added to a clean X-ray image. As a result, users of such systems may be exposed to unforeseen harmful scenarios, such as diagnostic errors, medical reimbursement fraud, and so on, if sufficient safeguards are not in place. As a result, an adequate defense strategy must be devised to deploy these devices securely.

Several defensive strategies have been offered in response to the threat. Adversarial training, which enlarges the training dataset with adversarial images to improve the resilience of the trained Convolutional Neural Network (CNN) model, is a standard method in the natural imaging domain. Many diverse adversarial images are included in the training dataset, which can dramatically reduce classification accuracy. [45] develops a logistic regression classifier based on characteristics extracted from a trained CNN model to distinguish adversarial images from clean images. However, the usefulness of this approach is limited to a set of predefined attack methods. These issues are addressed in the following way.

[69] adds a radial basis mapping kernel to CNN models, which translates data onto a linearly well-separated manifold to improve class separation and lessen the impact of perturbations. Global dependencies and contextual information can be leveraged to strengthen resilience, according to [31]. To guard against adversarial attacks, they suggest a non-local context encoder in medical picture segmentation systems. Although both strategies improve robustness by changing the network design, the trade-off between accuracy and robustness [83] may degrade system performance in practice.

We propose a robust adversarial image detection technique that effectively counters adversarial attacks on deep learning-based medical image classification systems. We focus on unsupervised anomalous detection utilizing features retrieved from a trained CNN classifier, as inspired by [86] and [38]. Our method successfully detects adversarial images and can effectively defend against unseen attacks, whether white-box or black-box, because it makes no assumptions about prior attack method knowledge. To demonstrate the success of our suggested defense strategy, we conduct extensive experiments using a publicly available X-ray dataset. We have considered the X-ray dataset to validate our algorithm. As per [62], X-ray and color fundus photographs are common diagnostic and prognostic imaging modalities in patient care.

6.2 Background

The adversarial image is created by subtly altering the original image; as a result, the perturbations appear as noise at the pixel level, obstructing human detection. Conversely, such noise is visible at the feature level of CNN models. According to [33], adversarial perturbations are difficult to detect by human eyes, resulting in significant noise at the feature level. Furthermore, the convolution-pooling techniques performed in CNN models during forward propagation might increase this" noise," resulting in misclassification. On the other hand, because the size of perturbations rises layer by layer, high-level characteristics can easily distinguish between clean and adversarial images [78].

The adversarial disturbance produced by the DeepFool assault is relatively minor compared to FGSM and other attacks [40]. We restricted our scope to iterative methods as they have high success rates than others.

6.3 Proposed method - Features Normalisation and Standardisation (FNS)

We propose a new adversarial image detection module for the medical image classification system, independent of the attacker's method, and does not require model retraining. Figure 6.1 shows the proposed method where a Multivariate Gaussian Model (MGM) is created with the extracted features of clean (original) images just before the classification layer of DenseNet-121 [33]. We have considered the output of the last Dense block for modeling MGM as the high-level characteristics can be distinguished easily between clean and adversarial images as per [78]. As mentioned in figure 6.1, these extracted features were normalized and standardized before the creation of MGM. Once the model is

created, it is used to identify the clean and adversarial images during the testing phase, as shown in figure 6.2. Only clean images are passed to the classification layer of DensetNet-121 for disease classification.

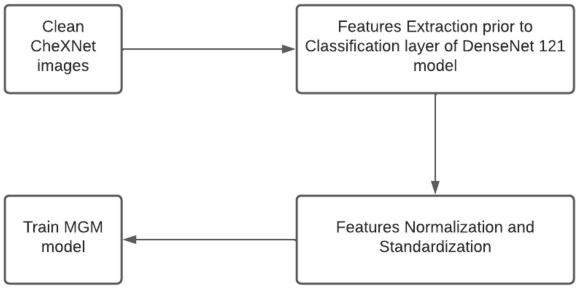

Figure 6.1 Proposed model for training MGM

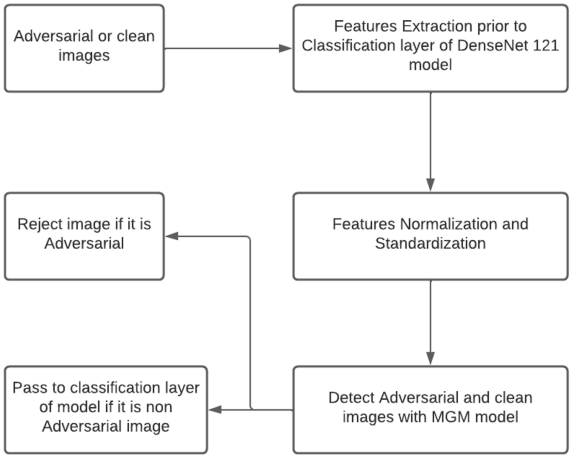

Figure 6.2 Testing phase

Before being modeled using MGM: $y \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $y = H(\mathbf{x})$ represents the feature extracted using the final fully connected layer given a clean input image $\mathbf{x}$, the high-level feature distribution of clean images is normalized and standardized. The $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d \times d}$ are mean vector and covariance matrix, where d represents the dimension of MGM. Considering features extracted from clean training images $Y = \{y_1, ..., y_n\}$, estimate $\boldsymbol{\mu} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} y_i$ and $\Sigma = \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{\mu})^{\mathrm{T}} (y_i - \boldsymbol{\mu}) + \lambda I$, where λI is the non-negative regularization added to the diagonal of the covariance matrix and n is the number of input samples. After modeling MGM, for a given image (can be clean or adversarial), compute the probability of $y^* = H(\mathbf{x}^*)$ belonging to the clean image distribution by

$$p(y^*) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{d}{2}}} \exp\left(\frac{-1}{2} (y_i - \boldsymbol{\mu})^{\mathrm{T}} \Sigma^{-1} (y_i - \boldsymbol{\mu})\right).$$

However, in reality, p(y^*) is computationally expensive due to the high dimension (d = 1024), and because its value is so near to zero, arithmetic underflow results. We re-parameterize the covariance matrix using Cholesky decomposition to get around these technical challenges, i.e., $\Sigma = \mathrm{RR}^T$ and rewrite the probability density function into log form:

$$\log p(y^*) = \frac{1}{2}[\,2 \times (\sum_{i=1}^{d} R_{ii}) + \|R^{-1}(\,y^* - \boldsymbol{\mu})\|^2 + d \log 2\pi]].$$

Finally, as shown in figure 6.2, $\mathbf{x}^*$ will be detected as an adversarial image and rejected if $\log p(y^*)$ is lower than a threshold. The threshold value is determined by considering 95% clean images during training.

6.4 Experiments

We verify the performance of our proposed defense approach by conducting experiments on a large public chest X-ray dataset. The NIH ChestX-ray14 [28] contains 112,120 frontal-view chest X-rays taken from 30,805 patients, where around 46% images are labeled with at least one of 14 pathologies. The features extracted from the complete clean training and validation datasets are used for training and validating our proposed detection module. We created an MGM (detection model) with the features of 95% clean images and extracted the features of FGSM, BIM, PGD, and MIM-based adversarial attack images for the whole dataset. For generating PGD and BIM adversarial attack images, we ran the iteration count of 7 (7 PGD steps). For generation MIM adversarial attack images, we considered a decay factor of 0.1 and an iteration count of 3 [51].

Figure 6.3 shows the distribution of log probability scores of MGM (trained on the features of 95% clean images) for the features from clean (100 % originals) images, FGSM, BIM, MIM, and PGD-based adversarial images. The zoomed-in version of images gives more clarity about the interference of adversarial attacks on original images. Figure 6.5 shows the visualization of features extracted before the classification layer of DenseNet-121 for Original images, FGSM, PGD, BIM, and MIM adversarial attacks.

To the best of our knowledge, data normalization and standardization are applied at the input level (layer 0) or the intermediate layer but not on the features extracted before the classification layer of a neural network. We extracted the features (1000 features) just before the classification layer of the DenseNet-121 and performed normalization and standardization of the features before generating an MGM. Our proposed method creates an MGM model with the above normalized and standardized features with 95% of clean images and randomly tests with 1000 clean images, FGSM, BIM, MIM, and PGD adversarial attack images. Figure 6.4 shows the distribution of log probability scores above MGM on clean images, FGSM, BIM, PGD, and MIM attack images. From the zoomed-in version of the images, it can notice that there is no interference of adversarial attack images on the original images. We can identify the adversarial images 100% with a proper threshold value, as shown in Table 6.1. All the above experiments are performed on an Nvidia T4 machine with CUDA Version 11.2.

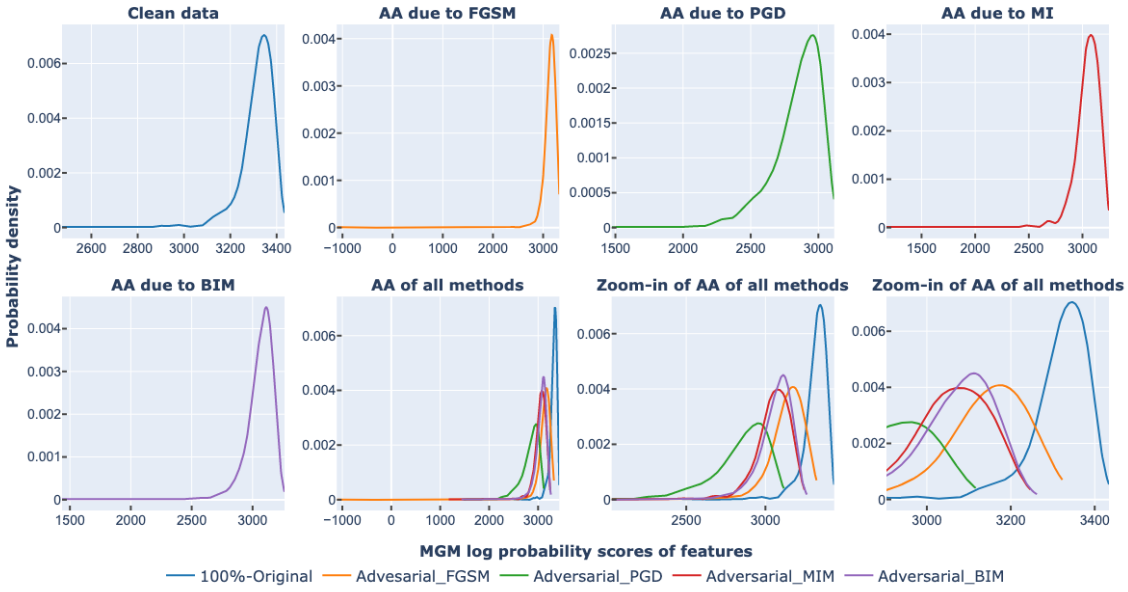

Figure 6.3 MGM model trained without FNS using 95% clean data (AA - Adversarial Attack) on X-ray dataset

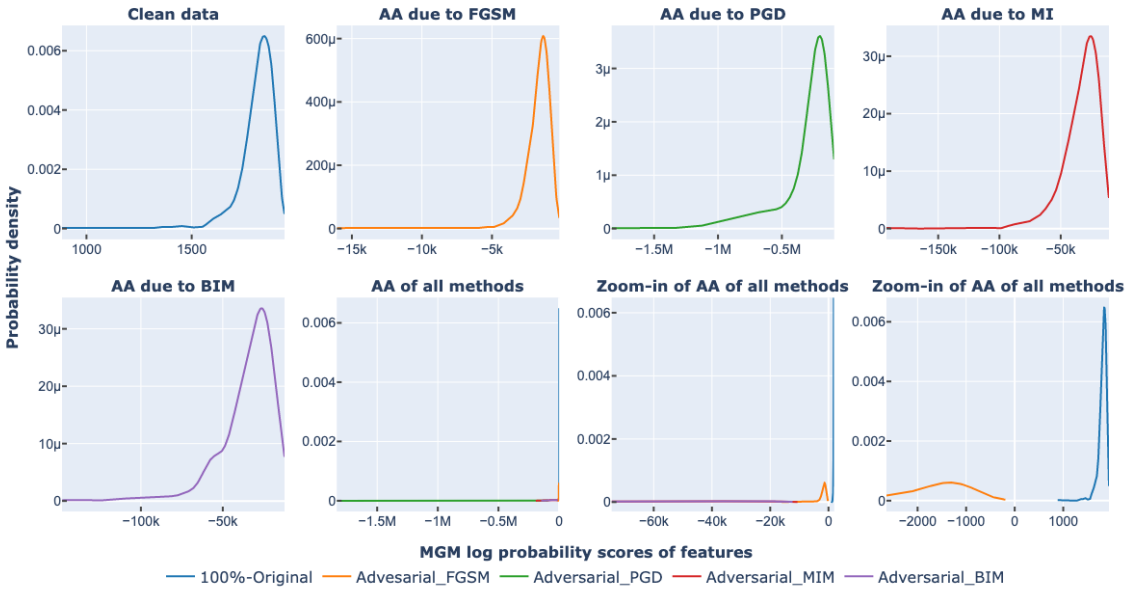

Figure 6.4 MGM model trained with FNS using 95% clean data (AA - Adversarial Attack) on X-ray dataset

Approach	Threshold value	FGSM	PGD	MIM	BIM
MGM without feature normalization	3260	88.5	92.18	92.13	92.18
MGM with feature normalization (our method)	882	100	100	100	100
Isolation Forest*	-	83.8	87.4	87.4	87.4
One class Support Vector Machine*	-	87	93.1	93.1	93.1

Table 6.1 F1 score in detecting Adversarial image for X-ray dataset
* refers the results captured from [38]

Figure 6.5 2D t-SNE visualization of features X-ray images

6.5 Conclusions

We propose a Features Normalisation and Standardisation unsupervised approach for detecting adversarial images. This is very useful in real-life scenarios where it does not require attackers' methods or retraining the model. We provide an experimental comparison of the iterative adversarial attack algorithms on the X-ray dataset. The results show that our proposed algorithm accurately determines adversarial images. This can be extended for further medical image datasets.

Chapter 7

Future Work

In this thesis, we have addressed three primary challenges in machine learning: a) learning multiclass classifiers using partially labeled data with missing feature values, b) improving robustness in identifying adversarial attacked images, particularly in the medical domain, and c) reducing adversarial training time while achieving improved adversarial defense test accuracy. We have extensively explored various learning and adversarial algorithms using medical and common datasets, providing valuable insights into their performance and effectiveness and addressing the challenges associated with partial label sets.

7.1 Partial label data multiclass classification under uncertainty conditions

As the proposed RPL-SVM algorithm exhibits promising results in handling partially labeled data with missing attributes, future work can focus on enhancing its scalability and efficiency for larger datasets. Research efforts could explore faster optimization techniques or parallel computing approaches to efficiently solve the Second Order Cone Programming (SOCP) formulation, enabling the algorithm to handle more extensive datasets effectively. Additionally, extending RPL-SVM to accommodate multi-label classification problems would be valuable, considering that real-world scenarios often involve instances with multiple labels. Investigating the challenges and opportunities of multi-label classification within the context of partially labeled data can lead to a more versatile and widely applicable algorithm. Additional studies can be conducted to expand the application of Chance Constraint Programming (CCP) optimization approaches by integrating Deep Learning models.

7.2 MIRP outperforms PGD attacks

While the MIRP adversarial training method significantly reduces training time compared to PGD adversarial training, future research can further evaluate its generalizability across different datasets and network architectures. Assessing the robustness of MIRP on more complex datasets and larger networks

can shed light on its effectiveness in diverse scenarios. Moreover, exploring various random initialization strategies and their impact on MIRP's performance could lead to improved training techniques for enhancing adversarial robustness. Additionally, understanding the trade-offs between training time and adversarial robustness achieved by MIRP can provide valuable insights for selecting suitable training strategies based on specific application requirements.

7.3 FNS approach for detecting Adversarial attacks for Medical Images

Extending the FNS approach to handle a wider range of complex and diverse medical image datasets is crucial to enhance its applicability in real-world medical scenarios. Conducting research to evaluate the FNS approach's performance on multimodal medical image data and investigating techniques to adapt to different imaging modalities can improve the detection of adversarial attacks in medical imaging applications. Furthermore, assessing the generalization capabilities of the FNS approach across various non-iterative adversarial attacks, including those designed to evade detection, will contribute to the robustness and reliability of medical image analysis systems.

Appendix A

A.1 Derivation of Constraints for RPL-SVM Linear Formulation

We rewrite the constraint in eq.(4.2) as follows.

$$d^2 = \inf_{\mathbf{C}^\top \mathbf{Z} \geq f} \mathbf{Z}^\top \mathbf{Z} \tag{A.1}$$

where $\mathbf{Z} = \Sigma_{Y_i}^{-\frac{1}{2}}(\bar{\mathbf{x}}_i - \boldsymbol{\mu}_{Y_i})$, $\mathbf{C}^\top = \left[\frac{1}{|Y_i|}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right]^\top \Sigma_{Y_i}^{\frac{1}{2}}$ and $f = 1 - \xi_i - \left[\frac{1}{|Y_i|}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right]^\top \boldsymbol{\mu}_{Y_i}$. The Lagrangian of the optimization problem in eq.(A.1) is written as $L(\mathbf{Z}, \lambda) = \mathbf{Z}^\top\mathbf{Z} + \lambda(f - \mathbf{C}^\top\mathbf{Z})$, where λ is the Lagrangian multiplier corresponding to the inequality constraint. KKT optimality conditions are: $2\mathbf{Z} = \lambda\mathbf{C}; \quad f = \mathbf{C}^\top\mathbf{Z}$. Substituting $\mathbf{Z}$ in f, we get $f = \mathbf{C}^\top\mathbf{Z} = \frac{\lambda}{2}\mathbf{C}^\top\mathbf{C}$. Thus, $\lambda = \frac{2f}{\mathbf{C}^\top\mathbf{C}}$ and $\mathbf{Z} = \frac{f\mathbf{C}}{\mathbf{C}^\top\mathbf{C}}$. From eq.(4.2), we know that $d^2 \geq \frac{\kappa_i}{1-\kappa_i}$. Substituting $\mathbf{Z}$ value in eq.(A.1) yields, $d^2 = \mathbf{Z}^\top\mathbf{Z} = \frac{(f\mathbf{C})^\top(f\mathbf{C})}{(\mathbf{C}^\top\mathbf{C})^\top(\mathbf{C}^\top\mathbf{C})} = \frac{|f|^2}{(\mathbf{C}^\top\mathbf{C})}$. Thus, $d = \frac{|f|}{\|\mathbf{C}\|}$. Using $d \geq \left(\frac{\kappa_i}{1-\kappa_i}\right)^{1/2}$, we get following constraint by substituting f and $\mathbf{C}$ values.

$$\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right]^\top \boldsymbol{\mu}_{Y_i} - (1-\xi_i) \geq \left(\frac{\kappa_i}{1-\kappa_i}\right)^{1/2} \left\| \Sigma_{Y_i}^{1/2}\left(\frac{1}{|Y_i|}\sum_{y\in Y_i}\mathbf{w}_y - \mathbf{w}_j\right)\right\|$$

A.2 Derivation of RPL-SVM Nonlinear Formulation

We reformulate the optimization problem in eq.(4.1) as a SOCP by removing $\|\mathbf{w}\|^2$ from the objective and adding a constraint $\|\mathbf{w}\|^2 \leq \Lambda$ as follows.

$$\begin{aligned}
&\min_{\xi} \sum_{i=1}^{n} \xi_i \\
\text{s.t.} \quad &\left[\frac{1}{\mid Y_i \mid} \sum_{y_i \in Y_i} \mathbf{w}_{y_i} - \mathbf{w}_y\right]^\top \phi(\mathbf{x}_i) \geq 1 - \xi_i; \quad \forall y \in \bar{Y}_i, \;\; i = 1 \ldots c \\
&\left[\frac{1}{\mid Y_i \mid} \sum_{y_i \in Y_i} \mathbf{w}_{y_i} - \mathbf{w}_y\right]^\top \phi(\bar{\mathbf{x}}_i) \geq 1 - \xi_i; \quad \forall y \in \bar{Y}_i, \;\; i = c+1 \ldots n \\
&\sum_{i=1}^{L} \|\mathbf{w}_i\|^2 \leq \Lambda; \quad \xi_i \geq 0; \quad \forall 1 \leq i \leq n
\end{aligned} \tag{A.2}$$

The optimization problem in eq.(A.2) can be shown equivalent to the optimization problem in eq.(4.1) for appropriate choices of C and Λ. Substituting eq.(4.4) in the constraints of the optimization problem in eq.(A.2) will be as follows.

$$\begin{aligned}
&\left\langle \frac{1}{|Y_i|} \sum_{y \in Y_i} (\sum_{k=1}^{c} \alpha_k^y \phi(\mathbf{x}_k) + \sum_{k=c+1}^{n} \bar{\alpha}_k^y \phi(\bar{\mathbf{x}}_k),\ \phi(\mathbf{x}_i) \right\rangle - \left\langle \sum_{k=1}^{c} \alpha_k^j \phi(\mathbf{x}_k) + \sum_{k=c+1}^{n} \bar{\alpha}_k^j \phi(\bar{\mathbf{x}}_k),\ \phi(\mathbf{x}_i) \right\rangle \\
&\geq 1 - \xi_i;\ \forall j \in \bar{Y}_i,\ i = 1 \ldots c \\
\Rightarrow &\frac{1}{\mid Y_i \mid} \sum_{y \in Y_i} \left(\sum_{k=1}^{c} \alpha_k^y K(\mathbf{x}_k, \mathbf{x}_i) + \sum_{k=c+1}^{n} \bar{\alpha}_k^y K(\bar{\mathbf{x}}_k, \mathbf{x}_i) \right) - \sum_{k=1}^{c} \alpha_k^j K(\mathbf{x}_k, \mathbf{x}_i) - \sum_{k=c+1}^{n} \bar{\alpha}_k^j K(\bar{\mathbf{x}}_k, \mathbf{x}_i) \\
&\geq 1 - \xi_i,\ \forall j \in \bar{Y}_i,\ i = 1 \ldots c \\
\Rightarrow &\frac{1}{\mid Y_i \mid} \sum_{y \in Y_i} \langle \boldsymbol{\alpha}^y, \tilde{K}(\mathbf{x}_i) \rangle - \langle \boldsymbol{\alpha}^j, \tilde{K}(\mathbf{x}_i) \rangle \geq 1 - \xi_i; \forall j \in \bar{Y}_i,\ i = 1 \ldots c
\end{aligned}$$

where $\tilde{K}(\mathbf{x}_i) = [K(\mathbf{x}_1, \mathbf{x}_i), \ldots, K(\mathbf{x}_c, \mathbf{x}_i), K(\bar{\mathbf{x}}_{c+1}, \mathbf{x}_i), \ldots, K(\bar{\mathbf{x}}_n, \mathbf{x}_i)]^\top$ and $\boldsymbol{\alpha}^y = [\alpha_1^y, \ldots, \alpha_c^y, \bar{\alpha}_{c+1}^y, \ldots, \bar{\alpha}_n^y]^\top$. Similarly, constraints for $i = c+1 \ldots n$ will be

$$\frac{1}{\mid Y_i \mid} \sum_{y \in Y_i} \langle \boldsymbol{\alpha}^y, \tilde{K}(\bar{\mathbf{x}}_i) \rangle - \langle \boldsymbol{\alpha}^j, \tilde{K}(\bar{\mathbf{x}}_i) \rangle \geq 1 - \xi_i; \forall j \in \bar{Y}_i,\ i = c+1 \ldots n$$

where $\tilde{K}(\bar{\mathbf{x}}_i) = [K(\mathbf{x}_1, \bar{\mathbf{x}}_i), \ldots, K(\mathbf{x}_c, \bar{\mathbf{x}}_i), K(\bar{\mathbf{x}}_{c+1}, \bar{\mathbf{x}}_i), \ldots, K(\bar{\mathbf{x}}_n, \bar{\mathbf{x}}_i)]^\top$. The nonlinear version of the RPL-SVM is obtained by considering the uncertainty in $\tilde{K}(\mathbf{x}_i)$. We assume that examples $\mathbf{x}_i,\ i = c+1 \ldots n$ have missing feature values. Probabilistic constraint takes the below form under noisy attributes in $\tilde{K}(\mathbf{x}_i)$.

$$p\left(\left[\frac{1}{\mid Y_i \mid} \sum_{y \in Y_i} \boldsymbol{\alpha}^y - \boldsymbol{\alpha}^j \right]^\top \tilde{K}(\mathbf{x}_i) \geq 1 - \xi_i \right) \geq \kappa_i,\ \forall j \in \bar{Y}_i,\ i = c+1 \ldots n$$

We continue to regard $\tilde{K}(\mathbf{x}_i)$ as a random vector, just like in the original problem. Using a similar approach as in the linear case, we can show that these probabilistic constraints lead to the following equivalent constraints.

$$\left(\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\boldsymbol{\alpha}^y - \boldsymbol{\alpha}^j\right]^\top \tilde{K}(\boldsymbol{\mu}_{Y_i})\right) \geq 1-\xi_i+\gamma_i\left\|\Sigma_{Y_i}^{\frac{1}{2}}(\frac{1}{\mid Y_i \mid}\sum_{y_i\in Y_i}\boldsymbol{\alpha}^y-\boldsymbol{\alpha}^j)\right\|;$$
$$\forall j\in\bar{Y}_i,\ i=c+1\ldots n$$

where $\tilde{K}(\boldsymbol{\mu}_{Y_i}) = [K(\mathbf{x}_1,\boldsymbol{\mu}_{Y_i}),\ldots,K(\mathbf{x}_c,\boldsymbol{\mu}_{Y_i}),K(\bar{\mathbf{x}}_{c+1},\boldsymbol{\mu}_{Y_i}),\ldots,K(\bar{\mathbf{x}}_n,\boldsymbol{\mu}_{Y_i})]^\top$. Σ_{Y_i} is the covariance of $\tilde{K}(\mathbf{x}_i)$ which is in $\tilde{K}$-space. Thus, the nonlinear formulation of RPL-SVM is as follows.

$$\begin{aligned}
&\min_{\xi}\sum_{i=1}^{n}\xi_i\\
\text{s.t.}\quad &\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\boldsymbol{\alpha}^y-\boldsymbol{\alpha}^j\right]^\top\tilde{K}(\boldsymbol{\mu}_{Y_i})\geq 1-\xi_i+\gamma_i\left\|\Sigma_{Y_i}^{\frac{1}{2}}\left(\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\boldsymbol{\alpha}^y-\boldsymbol{\alpha}^j\right)\right\|;\\
&\qquad\qquad\forall j\in\bar{Y}_i,\ i=c+1\ldots n\\
&\left[\frac{1}{\mid Y_i \mid}\sum_{y\in Y_i}\boldsymbol{\alpha}^y-\boldsymbol{\alpha}^j\right]^\top\tilde{K}(\mathbf{x}_i)\geq 1-\xi_i;\quad \forall y\in\bar{Y}_i,\ i=1\ldots c\\
&\sum_{j=1}^{L}\|\boldsymbol{\alpha}^j\|^2\leq\Lambda;\qquad \xi_i\geq 0;\quad \forall 1\leq i\leq n;
\end{aligned}\tag{A.3}$$

where $\gamma_i=\sqrt{\frac{\kappa_i}{1-\kappa_i}}$ and $\boldsymbol{\alpha}^y=[\alpha_1^y,\ldots,\alpha_n^y]^\top$.

Related Publications

1. Mohandas, S.; Manwani, N. and Dhulipudi, D. (2022). Momentum Iterative Gradient Sign Method Outperforms PGD Attacks. In Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 3, ISBN 978-989-758-547-0, ISSN 2184-433X, pages 913-916. DOI: 10.5220/0010938400003116 (ICAART-2022).

2. Mohandas, S. and Manwani, N. (2023). Features Normalisation and Standardisation (FNS): An Unsupervised Approach for Detecting Adversarial Attacks for Medical Images. In Proceedings of the 15th International Conference on Agents and Artificial Intelligence - Volume 3, ISBN 978-989-758-623-1, ISSN 2184-433X, pages 140-145 (ICAART-2023).

3. Mohandas, S. and Manwani, N. (2023). RPL-SVM: Making SVM Robust Against Missing Values and Partial Labels. In proceedings of the 20th Pacific Rim International Conference on Artificial Intelligence (PRICAI-2023).

Bibliography

[1] S. Andrews and T. Hofmann. Multiple-instance learning via disjunctive programming boosting. In *NIPS*, 2003.

[2] A. Athalye, N. Carlini, and D. A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ArXiv*, abs/1802.00420, 2018.

[3] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples, 2018.

[4] X. Bai, X. Wang, X. Liu, Q. Liu, J. Song, N. Sebe, and B. Kim. Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments. *Pattern Recognit.*, 120:108102, 2021.

[5] C. Barata, M. E. Celebi, and J. S. Marques. Explainable skin lesion diagnosis using taxonomies. *Pattern Recognit.*, 110:107413, 2021.

[6] A. Basavanhally, S. Ganesan, S. C. Agner, J. P. Monaco, M. D. Feldman, J. E. Tomaszeweski, G. Bhanot, and A. Madabhushi. Computerized image-based detection and grading of lymphocytic infiltration in her2+ breast cancer histopathology. *IEEE Transactions on Biomedical Engineering*, 57:642–653, 2010.

[7] S. Bhadra, S. Jagarlapudi, A. Ben-Tal, and C. Bhattacharyya. Interval data classification under partial information: A chance-constraint approach. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2009.

[8] C. Blake. Uci repository of machine learning databases. 1998.

[9] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognit.*, 37:1757–1771, 2004.

[10] S. P. Boyd and L. Vandenberghe. Convex optimization. *IEEE Transactions on Automatic Control*, 51:1859–1859, 2006.

[11] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Trans. Intell. Technol.*, 6:25–45, 2021.

[12] C. A. Coleman, D. Narayanan, D. Kang, T. Zhao, J. Zhang, L. Nardi, P. Bailis, K. Olukotun, C. Ré, and M. A. Zaharia. Dawnbench : An end-to-end deep learning benchmark and competition. 2017.

[13] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP*, 2002.

[14] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *J. Mach. Learn. Res.*, 12:1501–1536, 2011.

[15] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, 2001.

[16] Z. A. Daniels and D. N. Metaxas. Exploiting visual and report-based information for chest x-ray analysis by jointly learning visual classifiers and topic models, 2019.

[17] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *ArXiv*, abs/1705.02900, 2017.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em - algorithm plus discussions on the paper. 1977.

[19] S. Diamond and S. P. Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of machine learning research : JMLR*, 17, 2016.

[20] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89:31–71, 1997.

[21] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[22] Y. Dong, H. Su, J. Zhu, and F. Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *ArXiv*, abs/1708.05493, 2017.

[23] J. R. Doppa, J. Yu, and P. Tadepalli. Chance-constrained programs for link prediction. 2009.

[24] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning models, 2018.

[25] S. G. Finlayson, J. Bowers, J. Ito, J. Zittrain, A. Beam, and I. S. Kohane. Adversarial attacks on medical machine learning. *Science*, 363:1287 – 1289, 2019.

[26] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi. Deepcloak: Masking deep neural network models for robustness against adversarial samples. *arXiv: Learning*, 2017.

[27] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

[28] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

[29] S. S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. *CoRR*, abs/1412.5068, 2014.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[31] X. He, S. Yang, G. Li, H. Li, H. Chang, and Y. Yu. Non-local context encoder: Robust biomedical image segmentation against adversarial attacks. In *AAAI*, 2019.

[32] H. Hosseini, Y. Chen, S. Kannan, B. Zhang, and R. Poovendran. Blocking transferability of adversarial examples in black-box learning systems. *ArXiv*, abs/1703.04318, 2017.

[33] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[34] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. *Intell. Data Anal.*, 10:419–439, 2005.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[36] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2017.

[37] G. R. G. Lanckriet, L. E. Ghaoui, C. Bhattacharyya, and M. I. Jordan. A robust minimax approach to classification. *J. Mach. Learn. Res.*, 3:555–582, 2002.

[38] X. Li and D. Zhu. Robust detection of adversarial attacks on medical images. *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1154–1158, 2020.

[39] Y. Li and Y. Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *ICML*, 2017.

[40] H. Liang, E. He, Y. Zhao, Z. Jia, and H. Li. Adversarial attack and defense: A survey. *Electronics*, 2022.

[41] G. J. S. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[42] L.-P. Liu and T. G. Dietterich. A conditional multinomial mixture model for superset label learning. In *NIPS*, 2012.

[43] J. López and S. Maldonado. Multi-class second-order cone programming support vector machines. *Inf. Sci.*, 330:328–341, 2016.

[44] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. N. R. Wijewickrema, M. E. Houle, G. R. Schoenebeck, D. X. Song, and J. Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *ArXiv*, abs/1801.02613, 2018.

[45] X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao, J. Bailey, and F. Lu. Understanding adversarial attacks on deep learning based medical image analysis systems, 2021.

[46] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2018.

[47] A. W. Marshall and I. Olkin. Multivariate chebyshev inequalities. *Annals of Mathematical Statistics*, 31:1001–1014, 1960.

[48] D. Meng and H. Chen. Magnet: A two-pronged defense against adversarial examples. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[49] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. *ArXiv*, abs/1702.04267, 2017.

[50] P. Micikevicius, S. Narang, J. Alben, G. F. Diamos, E. Elsen, D. García, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed precision training. *ArXiv*, abs/1710.03740, 2018.

[51] S. Mohandas, N. Manwani, and D. P. Dhulipudi. Momentum iterative gradient sign method outperforms pgd attacks. In *ICAART*, 2022.

[52] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2015.

[53] T. Pang, C. Du, and J. Zhu. Robust deep learning via reverse cross-entropy training and thresholding test. *ArXiv*, abs/1706.00633, 2017.

[54] N. Papernot, P. Mcdaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2016.

[55] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387, 2015.

[56] N. Papernot, P. Mcdaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, 2015.

[57] N. Papernot, P. Mcdaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, 2016.

[58] G. Rätsch, A. Smola, and S. Mika. Adapting codes and embeddings for polychotomies. In *NIPS*, 2002.

[59] I. Reda, B. O. Ayinde, M. M. Elmogy, A. M. Shalaby, M. T. El-Melegy, M. A. El-Ghar, A. A. El-Fetouh, M. Ghazal, and A. S. El-Baz. A new cnn-based system for early diagnosis of prostate cancer, 2018.

[60] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ArXiv*, abs/1805.06605, 2018.

[61] A. Shaffie, A. Soliman, H. A. Khalifeh, M. Ghazal, F. Taher, A. S. Elmaghraby, R. S. Keynton, and A. S. El-Baz. Radiomic-based framework for early diagnosis of lung cancer, 2019.

[62] X. Shi, Y. Peng, Q. Chen, T. D. L. Keenan, A. T. Thavikulwat, S. Lee, Y. Tang, E. Y. Chew, R. M. Summers, and Z. Lu. Robust convolutional neural networks against adversarial attacks on medical images. *Pattern Recognit.*, 132:108923, 2022.

[63] P. K. Shivaswamy, C. Bhattacharyya, and A. Smola. Second order cone programming approaches for handling missing and uncertain data. *J. Mach. Learn. Res.*, 7:1283–1314, 2006.

[64] L. N. Smith and N. Topin. Super-convergence: very fast training of neural networks using large learning rates. In *Defense + Commercial Sensing*, 2019.

[65] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23:828–841, 2017.

[66] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[67] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2014.

[68] K. Taga, K. Kameyama, and K. Toraichi. Regularization of hidden layer unit response for neural networks. *2003 IEEE Pacific Rim Conference on Communications Computers and Signal Processing (PACRIM 2003) (Cat. No.03CH37490)*, 1:348–351 vol.1, 2003.

[69] S. A. Taghanaki, A. Das, and G. Hamarneh. Vulnerability analysis of chest x-ray image classification against adversarial attacks. In *MLCN/DLF/iMIMIC@MICCAI*, 2018.

[70] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, 2003.

[71] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. Mcdaniel. Ensemble adversarial training: Attacks and defenses. *ArXiv*, abs/1705.07204, 2018.

[72] G. Tsoumakas, I. M. Katakis, and I. P. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, 2010.

[73] P. A. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2005.

[74] H. Wang, R. Xiao, Y. Li, L. Feng, G. Niu, G. Chen, and J. J. Zhao. Pico: Contrastive label disambiguation for partial label learning. *ArXiv*, abs/2201.08984, 2022.

[75] J. Wang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6628–6637, 2019.

[76] Q. Wang, W. Guo, K. Zhang, A. Ororbia, X. Xing, C. L. Giles, and X. Liu. Learning adversary-resistant deep neural networks. *ArXiv*, abs/1612.01401, 2016.

[77] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training, 2020.

[78] C. Xie, Y. Wu, L. van der Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 501–509, 2019.

[79] K. Xu, H. Chen, S. Liu, P.-Y. Chen, T.-W. Weng, M. Hong, and X. Lin. Topology attack and defense for graph neural networks: An optimization perspective. *ArXiv*, abs/1906.04214, 2019.

[80] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin. Adversarial t-shirt! evading person detectors in a physical world. In *ECCV*, 2020.

[81] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *ArXiv*, abs/1704.01155, 2017.

[82] T. K. Yoo and J. Y. Choi. Outcomes of adversarial attacks on deep learning models for ophthalmology imaging domains. *JAMA ophthalmology*, 2020.

[83] H. R. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. *ArXiv*, abs/1901.08573, 2019.

[84] J. Zhang and C. Li. Adversarial examples: Opportunities and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 31:2578–2593, 2018.

[85] Q. Zhang, C. Lin, and F. Li. Application of binocular disparity and receptive field dynamics: A biologically-inspired model for contour detection. *Pattern Recognit.*, 110:107657, 2021.

[86] Z. Zheng and P. Hong. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In *NeurIPS*, 2018.

[87] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. In *Introduction to Semi-Supervised Learning*, 2009.