Adversarial Defences Against Attacks on Machine Learning Models

University of Turku
Department of Computing
Bachelor's Thesis
Computer Science
March 2026
Patrick Parve

UNIVERSITY OF TURKU
Department of Computing

PATRICK PARVE: Adversarial Defences Against Attacks on Machine Learning Models

Bachelor's Thesis, 23 p.
Computer Science
March 2026

Adversarial attacks against machine leaning ML models in life-critical areas pose a significant threat, by being able to make an input that can deceive a ML model to make an incorrect decision. The goal of this thesis is to examine and evaluate the current situation between adversarial attacks and defences. The thesis examines different types of adversarial defence methods and strategies. Those include more established defensive methods that have been around longer and are more commonly used such as adversarial training, feature squeezing and the trapdoor defence. Also, the thesis explores some more recently proposed defences and strategies like Targeted Manifold Manipulation, cryptographic techniques, DeepDefence and Large Language Model Adversarial Defence. Findings show that an adversarial defence that cannot be defeated by an adversarial attack does not exist, nor is close to being made. A lot of defensive techniques and methods exist, but none can guarantee 100% success rate when it comes to being able to defend a ML model from every possible adversarial attack.

Keywords: machine learning, adversarial defence, adversarial attack

TURUN YLIOPISTO
Tietotekniikan laitos

PATRICK PARVE: Adversarial Defences Against Attacks on Machine Learning Models

TkK-tutkielma, 23 s.
Tietojenkäsittelytiede
Maaliskuu 2026

Vihamieliset hyökkäykset (engl. adversarial attacks) koneoppimismalleja vastaan elintärkeillä sovellusalueilla muodostavat merkittävän uhan, sillä niillä voidaan luoda syöte, joka saa koneoppimismallin tekemään virheellisen päätöksen. Tämän kandidaatintutkielman tavoitteena on tarkastella ja arvioida vihamielisten hyökkäysten ja puolustusmenetelmien nykytilannetta. Tutkielmassa käsitellään erilaisia vihamielisiä puolustusmenetelmiä ja -strategioita. Näihin kuuluvat vakiintuneemmat puolustusmenetelmät, jotka ovat olleet käytössä pidempään ja joita käytetään yleisemmin, kuten koneoppimismallin kouluttaminen vihamielisillä esimerkeillä (engl. adversarial training), ominaisuuksien puristaminen (engl. feature squeezing) ja takaporttipuolustusmenetelmä (engl. trapdoor). Lisäksi tutkielmassa tarkastellaan joitakin viime aikoina ehdotettuja puolustusmenetelmiä ja -strategioita, kuten kohdistettu sarjamanipulaatio (engl. Targeted Manifold Manipulation), kryptografiset menetelmät, DeepDefence ja suurten kielimallien puolustusmenetelmä vihamielisiä hyökkäyksiä vastaan (engl. Large Language Model Adversarial Defence). Tulokset osoittavat, ettei ole olemassa eikä lähitulevaisuudessa odotettavissa sellaista puolustusmenetelmää, jota ei voitaisi murtaa vihamielisellä hyökkäyksellä. Puolustustekniikoita ja -menetelmiä on kehitetty runsaasti, mutta mikään niistä ei voi taata 100%:n onnistumisprosenttia koneoppimismallin suojaamisessa kaikilta mahdollisilta vihamielisiltä hyökkäyksiltä.

Asiasanat: koneoppiminen, vihamielinen puolustus, vihamielinen hyökkäys

Contents

List of Figures

1 INTRODUCTION

Computing power has drastically increased, which has made using deep learning (DL) popular for machine learning (ML) tasks [1]. ML tasks include diagnosing diseases, self-driving vehicles, inspecting checking frauds [2], and natural language processing [3]. The popularity and large adoption of ML models has made the models' weakness to adversarial attacks visible. Input data with a small amount of distortion can lead to an incorrect outcome, which in life-critical areas is a serious concern. [2] More academic attention has gone towards the accomplishments of ML models completing complex tasks than towards security in ML [3]. Attention towards the concerns of ML safety is needed. Currently there are no effective defensive mechanisms against inputs that try to mislead the ML models into giving wrong outputs [2]. The relationship between adversarial attacks and defences can be called an arms race. When a new method gets proposed a counter to it will soon follow. Adversarial attacks are considered to be ahead in this race, as creating a defence that can protect against all types of different attacks has proven to be very difficult. [3]

Due to the importance of security, this thesis will go over adversarial defences. Some of the different defensive strategies that have been around longer will be explained first, such as adversarial training, feature squeezing and the trapdoor defence. More recently proposed defensive solutions like targeted manifold manipulation, cryptographic techniques, DeepDefence and Large Language Model Adversarial Defence will be covered after. New solutions and more innovation is what is needed,

to have more secure ML model's that we can safely use. By the end this thesis will have had answer to the following research questions:

RQ1: What are some different defensive methods that have been used to defend ML models against adversarial attacks?

RQ2: What are the weaknesses of those defensive methods?

RQ3: What new possible methods are currently being proposed?

This thesis was conducted as a literature review. Web of Science and IEEE Xplore were the two different databases that were used to gather relevant resources for this thesis. The search phrase used to find the intended material was: "adversarial attack*" AND ("machine learning" OR "deep learning") AND (defense* OR robust* OR detection OR prevention). The publication year was set to 2019-2025 at the start and later to 2024-2025 when searching for publications on newer proposed defensive methods. The language was also set to English, after which there were 3,463 results on Web of Science and 4,873 on IEEE Xplore. After which I narrowed down the results based on the title and then later based on the abstract. A total of twenty-seven articles were gathered. From there, a total of seven articles were chosen to be used. Also seven additional articles were found, that were got from either the reference lists of the first seven articles chosen, articles that cited one of the chosen articles or from a modified search to find more specific articles. In total, fourteen different articles were used in this thesis. The research process is laid out in Figure 1.1.

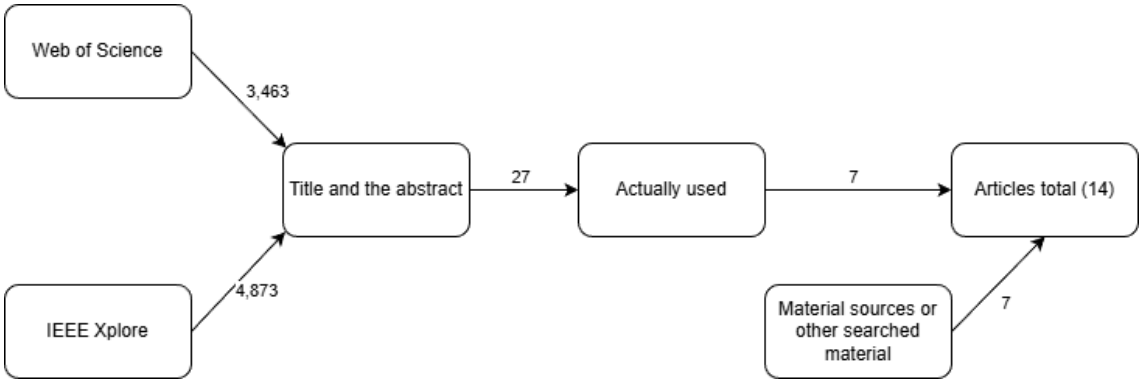

Figure 1.1: Research process

The thesis is structured in the following way: Chapter 2 goes over a few examples of popular adversarial attacks that are regularly used for testing defences, with a usage example shown for one of the attacks. Any other background information needed to understand the content will also be covered. Chapter 3 introduces three different adversarial defensive strategies and explains how they work and the weaknesses they have. Chapter 4 explores newer defensive methods and also goes over them. Chapter 5 contains the conclusion of the thesis, where answers to the research questions are separately given.

2 BACKGROUND

ML is a technique that is data driven. It has an abundance of input and output pairs. It learns based on the data to predict outputs from given inputs. Data's quality and amount reflect on the ML algorithm's quality. The better and more data there is to train a ML model, the higher quality the model will have. [4]

There are three main types of learning for ML algorithms: supervised ML, unsupervised ML and reinforcement learning. In supervised learning, the model has the input and the output for it. The data is labelled and the model learns the relation between them. Contrarily, in unsupervised learning there is no information with the given input, only data. The model tries to find patterns that could link the inputs. Reinforcement learning is where the ML model learns from its past predictions which it got feedback on. Based on the feedback, the model will make changes to its strategy. Feedback is either a reward or penalty, and the model learns to maximise the reward to be able to produce better results. [4]

An adversarial example is a ML model's input that is deliberately constructed to cause a ML model to make an incorrect prediction. To humans, adversarial examples still seem like valid inputs. [3] The making of adversarial examples and the manipulated data itself are examples of adversarial attacks [5]. Adversarial attacks are acts that strive to feed inaccurate information to the ML models with a goal of making the models end up with an incorrect decision or even a wrong classification. The attacks work on different types of ML algorithms. They weaken

the performance of the models in actions like image recognition, speech recognition, and route identification. [5]

An evasion attack is a type of adversarial attack that occurs after the model's training process, where input data is manipulated to cause the model to misclassify it and output a false result [6]. The attack can be targeted or untargeted. When the attack is targeted, the goal is for the output to have a certain classification. In untargeted attacks, the classification just needs to be something other than the correct class. [7]

Adversarial defences are the methods that are being used to defend against adversarial examples [1]. Currently, available defensive methods either make the ML model robust against a certain amount of perturbation or detect the attack [8] before the adversarial attack gets inserted into the ML model [5].

For testing adversarial defences and attacks, there are benchmark datasets that are commonly used to train ML models. Datasets can vary depending on the domain of the ML model. There are datasets for image classification, natural language and audio processing. [9] Possibly two of the most popular datasets for image classification are MNIST and CIFAR-10 [2]. MNIST dataset has handwritten digits that can be seen in Figure 2.1. There are 60,000 examples for training the model and 10,000 for testing it. CIFAR-10 has 60,000 32 x 32 colour images with 10 classes. [9]

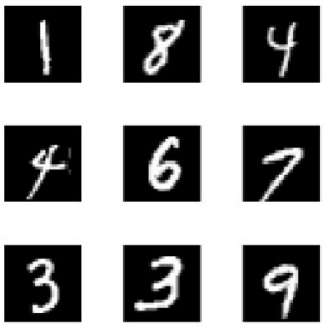

Figure 2.1: MNIST dataset [10].

Black-box, grey-box, and white-box models are the three main threat models for adversarial attacks and defences. The models are based on the attacker's knowledge of the targeted ML model. In the black-box model, the structure of the network or the parameters of the model are not known to the attacker, but the attacker is still able to submit inputs to the model and examine the outputs given. In the grey-box model, the architecture of the model is known to the attacker. Additionally, the target model can be used in the grey-box scenario. The white-box threat model is the worst scenario for the defence, as the attacker has full access to the target model. The attacker can adjust the attacks and make precise adversarial examples because of the knowledge available about all the parameters. Defensive methods that have been shown to be effective against black-box or grey-box attacks are susceptible to adaptive attacks in a white-box situation. [1]

One famous example of an adversarial example was made by the Fast Gradient Sign Method (FGSM) that was used on a picture of a panda. Figure 2.2 shows how GoogleNet classified the images. The panda in the original image on the left was correctly classified as a panda. After perturbation was added to the image, the new image on the right was misclassified by GoogleNet as a gibbon, with a high confidence score of 99.3 %. [3]

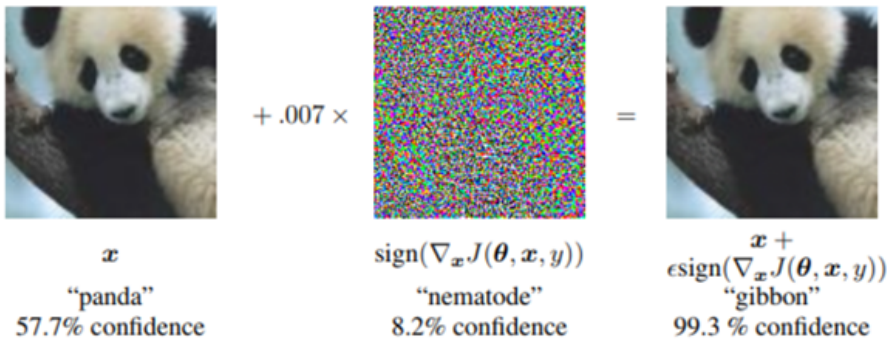

Figure 2.2: Picture of a panda where perturbation that is added changes the image's classification. The perturbation has been amplified for visualisation purposes only [3].

FGSM is a single-step gradient-based [3] untargeted evasion attack [7] in a white-box model [3]. It is used to generate adversarial examples [6] by looking for the input's perturbation direction of the gradient that causes the model's success rate to drop the most, therefore causing the chance of misclassifying inputs [3]. FGSM's performance can be improved by random perturbing before using FGSM, which will diversify the adversarial samples [1].

There are iterative alternatives to FGSM like Basic Iterative Method (BIM) [1] and Projected Gradient Descent (PGD). They do multiple actions of gradient descent with a smaller step size [11]. This way adversarial examples will be more effective by iteratively improving the perturbation [6]. The difference is that PGD starts from a random perturbation [11] and is therefore seen as a generalised version of BIM [1].

Another type of attack is an optimisation-based attack like Carlini and Wagner Attack (C&W). The goal is to find adversarial perturbation that is keeping the adversarial loss and the distance between normal data and the adversarially manipulated data to a minimum. [12] This attack is also stronger than the previously mentioned FGSM and BIM attacks. [11]

3 DEFENCES

3.1 Adversarial Training

Adversarial training is one of the most extensively researched techniques for defending against adversarial attacks where the input is manipulated in a way to mislead the ML system [6]. Against a range of perturbations, it is a common defensive strategy that is used to improve ML models' robustness against adversarial attacks [8].

Adversarial training is a defence method where adversarial examples that are carefully crafted [6], are inserted into the training data at every training iteration. Adversarial examples can be generated using adversarial attack methods like FGSM or BIM. [3] The adversarial examples are made to deceive the model. The model becomes more durable against adversarial perturbations when being trained with both clean and adversarial examples. The adversarial examples in the training dataset need to be correctly labelled during adversarial training as otherwise the model can get deceived by the adversarial examples. [6]

Adversarial training performs appropriately well and is relatively simple to put into use [3]. However, it does have one shortcoming. The effectiveness of adversarial training is bound by the attack method used during the adversarial example's generation process. Those examples are put into the training data. An adversarially trained model will not assure that the model will be robust against other adversaries that use different attack methods. For instance, adversarial training on adversarial

examples generated using single-step methods creates models, which are less robust against iterative gradient-based attacks like BIM. Nonetheless, if a ML model is trained on examples made from single-step gradient-based attacks like FGSM, the model will display robustness against attacks of the same category. [3] Adversarial training can also obscure the class boundaries between real classes, which can have an impact on the accuracy of clean data. Currently, adversarial training with PGD-perturbed adversarial samples is the most effective form of adversarial training. Unfortunately, this training process is very expensive and regularly fails to establish across new data. [8]

3.2 Feature Squeezing

Feature squeezing is a defensive method that is made to detect an adversarial example in the input phase before being allowed into the ML model. It removes unnecessary features from an input by squeezing it and compares the predictions between squeezed and unsqueezed inputs. If the predictions' difference is larger than a certain threshold value, the input will be labelled as adversarial and rejected. [3] When the squeezing is too aggressive, the model's performance can downgrade even with valid inputs [6].

Adversarial examples from attacks like FGSM, BIM and a few more are able to be detected by feature squeezing on MNIST, CIFAR10 and ImageNet datasets when the attacker does not know what defensive method is being used. Some squeezers work against certain attacks better than other squeezers, which work better against different attacks. Therefore, there is a need to join two squeezers like colour depth reduction and spatial smoothing method to be prepared against different types of adversarial examples. [3] The process with two squeezers is shown in Figure 3.1.

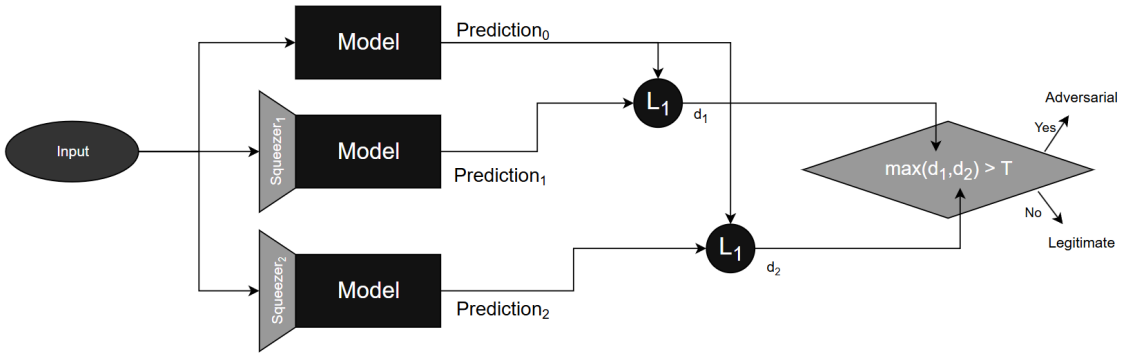

Figure 3.1: Illustration of feature squeezing. This figure is a remade version of the original that is used in [3].

The feature squeezing method is a powerful strategy against static adversaries, but when an attacker has full knowledge of the squeezers, it can be liable to adversarial examples. Thankfully, this method of defence is independent, so it can be used together with other defence techniques to protect a model better. [3]

Feature squeezing is one of the most commonly used techniques to make creating adversarial examples difficult while still maintaining model accuracy [6]. Although feature squeezing alone is not a reliable defence [3], when used together with other defensive techniques, it can be effective [6]. It can also be used to increase adversarial robustness alone or alongside adversarial training [3].

3.3 Trapdoor

The trapdoor defensive method used in image recognition is an adversarial detection type of defence that uses modified data to catch adversarial examples. The data that is being used to catch the attacks is called a trapdoor. They can be added for a single class label or multiple labels, the differences are shown in Figure 3.2. Trapdoors have an imperceptible impact on the ML model's performance to classify

clean data. For every label that is being protected, a trapdoor is made specifically for it. There is a detection threshold for each label. Once detected the input will be seen as an adversarial attack. With trapdoors, the attackers are unable to find genuine weaknesses in a ML model. [12]

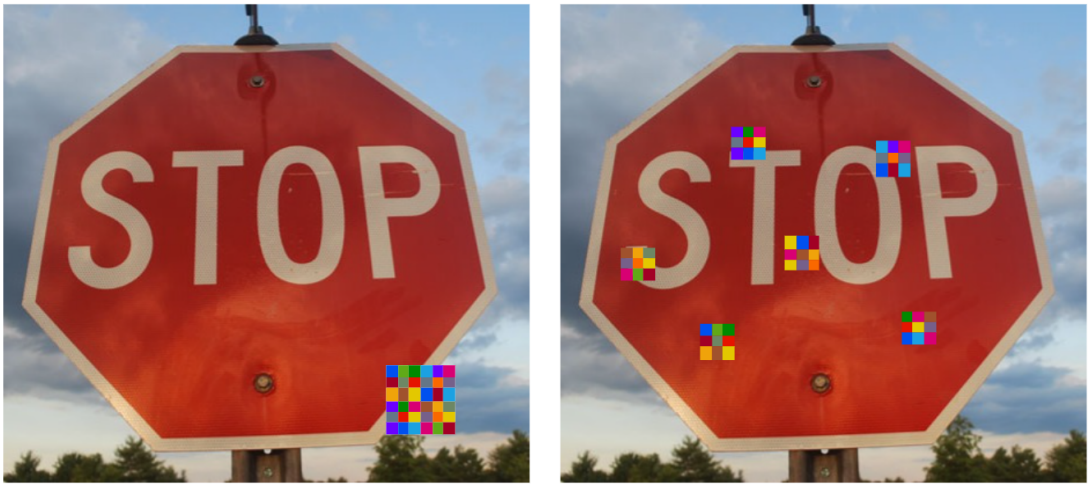

Figure 3.2: Trapdoor examples used in trapdoor defence. The perturbation in the image is amplified to be more visible. The single-label defence contains a single 6 x 6 square and the all-label defence has five 3 x 3 squares. Inspiration for the figure is from [12].

Trapdoor training datasets are made by adding trapdoor perturbation to a normal input chosen randomly and adding it to the original training dataset. The added perturbation will be affiliated with a new label, and the image will be a trapdoor image. The trapdoor can be completely customized. The customizable features include the location, size, pixel intensity and dimensions. [12]

A trapdoored model will then be made by training the model with the trapdoor training dataset. The model will have high accuracy in classifying normal data, which will be on par with a model with no trapdoored data. It will also have a

high success rate in labelling images containing the trapdoor as trapdoor images. The model's trapdoor(s) and parameters do however need to be configured during training to get the best results. The model's training session will last a little longer with trapdoors in the data. In Figure 3.3 the overview of the trapdoor defence is shown. [12]

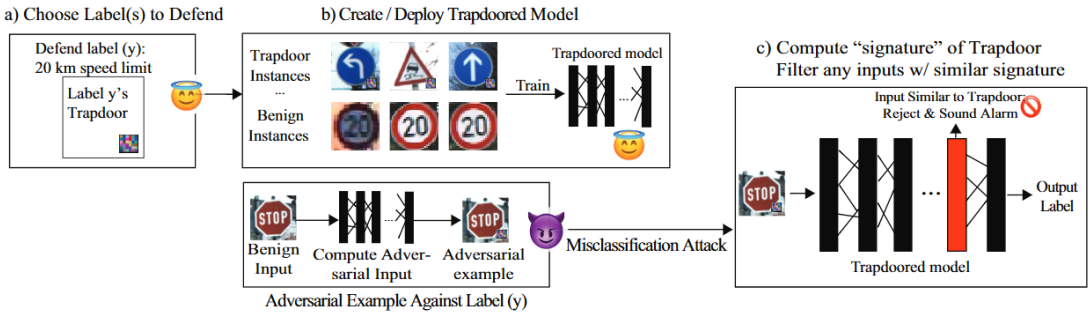

Figure 3.3: Step by step how the trapdoor defence is run [12].

When all labels are being defended, the model is slightly worse at adversarial detection than when a single label is being defended. As more trapdoors are implemented into the model, there is a greater chance for them to interfere with each other. This can be improved by being careful with the placements of the trapdoors. Nonetheless, when defending all the labels the defence performs well with many different models and against multiple adversarial attacks like PGD, FGSM and C&W. The trapdoor defence also outperforms feature squeezing against all attacks besides C&W and ElasticNet (EN). [12]

4 NEW METHODS

4.1 Targeted Manifold Manipulation

Targeted Manifold Manipulation (TMM) is a defence method that detects attacks and makes it harder for attackers to succeed. This defensive method makes a separate class for data that is not seen as any other class in the ML model. This method works by locally modifying the gradient in a way that perturbed inputs get directed towards traps. Traps are around the genuine data, and the traps form a trap-ring. Traps are created by adding a small modification to the data the trap is going to be surrounding. Multiple traps are made for a single class, and they are added around the real data. Trap-rings start as close to the genuine data point as possible. A visualisation of two classes and the trap rings surrounding them is shown in Figure 4.1 When data falls into one of the traps in a trap-ring, its class gets labelled as a Trapclass. The ML model therefore sees that data as an attack. [8]

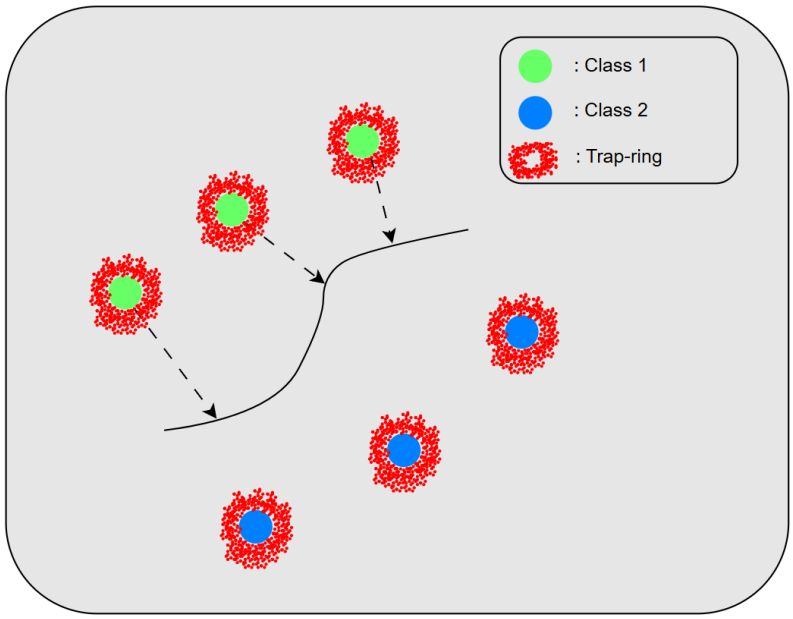

Figure 4.1: TMM where classes are surrounded by the trap ring. The black dotted arrows show the attack from Class 1 to Class 2 [8]. This figure is a remake of the original from [8].

TMM-Defence uses three filters to detect attacks. They are the trapclass, entropy and OOD filters. The entropy filter tries to catch attacks that generate a low-confidence perturbation, like C&W. An attack that generates a high-confidence perturbation like PGD gets detected by the OOD filter. [8]

When adversarial attacks are untargeted, the attacks mostly get caught by the trapclass filter. For targeted attacks it depends on the attack used. The combination of the filters is what will protect the ML model, as each filter is more effective against specific types of attack. [8] The process of an input going through a ML model using TMM defence is shown in Figure 4.2.

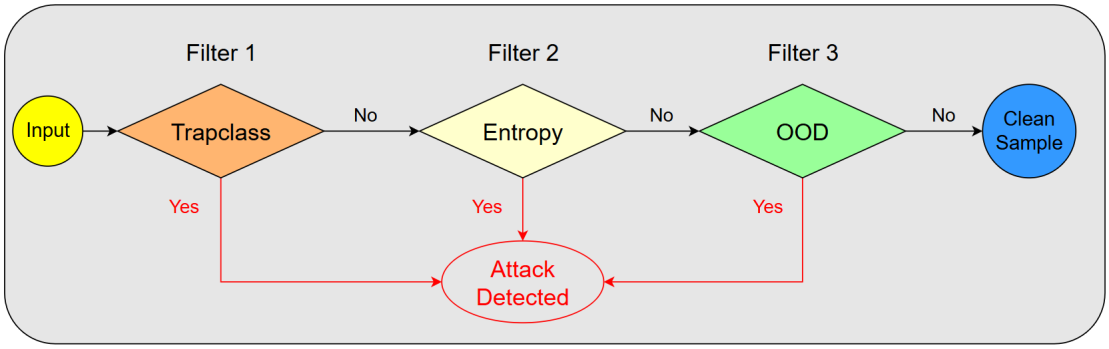

Figure 4.2: The adversarial attack detection method of TMM defence. Inspiration for the figure is taken from [8].

TMM-Defence performs better when compared with other state-of-the-art attack detectors like Local Intrinsic Dimensionality (LID), Mahalanobis and Trapdoor. TMM-Defence's detection accuracy was higher against all adversarial attacks (targeted and untargeted) on all datasets that were used in testing except for one statistical outlier. When compared with adversarial training, TMM-Defence does not need to know the attack method that is going to be used. [8]

TMM-Defence provides universal defence and has a very low impact on the accuracy of clean data. It is computationally cheaper to run than other detectors, because it does not require a separate classifier. Therefore, this defence can be highly scalable and thus have many classes. [8]

4.2 Robustness Through Cryptographic Techniques

Using cryptographic techniques can improve security and resistance for a trained ML model for when it is active by lessening adversarial attacks. The following approach uses homomorphic encryption technique and secure multi-party computation, which are from the cryptographic field. Homomorphic encryption allows computation on

encrypted data without having to decrypt it first. Secure multi-party computation in a ML model case means that many parties can operate the model without having access to each other's data. This works both in training and in actual usage cases. [2]

By using these techniques, the proposed method follows the flow shown in Figure 4.3. The method advances by first encrypting the data it receives by using homomorphic encryption. This makes the data protected and private. The data is securely transmitted to the ML model to be analysed. Based on the ML model parameters, calculations and computations are directly done on the encrypted data. Ciphertext is generated by the ML model that gets decrypted and becomes the ML model's output. [2]

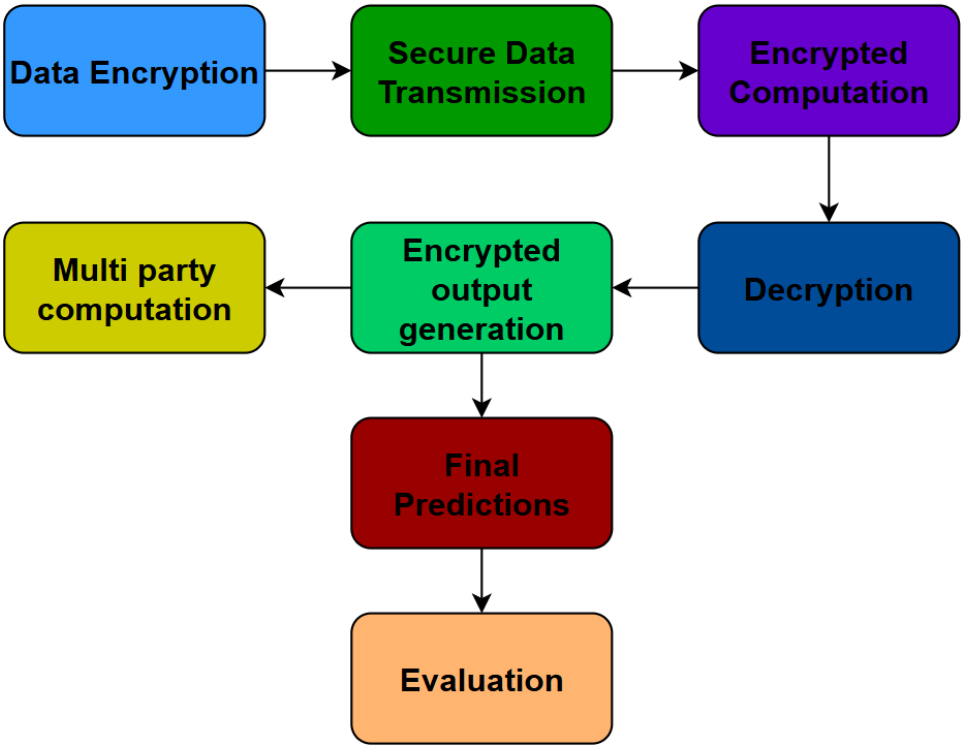

Figure 4.3: Flowchart of the proposed defensive method that uses cryptographic techniques. Remade based on the original from [2].

This model in a normal ML procedure can efficiently detect classes with a high accuracy while keeping them separated and not mixing the classes up. It improves accuracy and performs better than other existing defensive solutions against adversarial attacks when tested on MNIST and CIFAR10 datasets. Adversarial training aims were at 85 percent accuracy, but this method of defence presents more than 90 percent accuracy overall. The model will take longer and need more resources to perform, but it is a promising method to strengthen ML models against adversarial attacks. [2]

4.3 DeepDefence

DeepDefence is a defensive framework to defend against adversarial perturbations. It uses Gradient-Feature Alignment (GFA), which is a process where the input's gradient will get aligned with internal feature representations. This is used across multiple layers to increase the resistance to adversarial attacks. By using GFA to block adversarial perturbation across multiple layers, the adversarial perturbation will not start adding up across the different layers in the neural network, therefore making the defence scalable and robust. It makes the perturbations move in directions where they are less effective. [13]

Models trained with GFA become more robust towards a wide range of adversarial attacks, including gradient-based attacks [13] like FGSM or BIM [3] and optimization-based attacks [13] like C&W [3]. Perturbations need to be strong and more visible to fool a DeepDefence trained model. An ML model trained with DeepDefence makes attackers search for ways to mislead the model along the radial where the perturbations are massively suppressed by either the structure of the neural network or activation functions. The decision boundaries of the model get improved due to the training. [13]

DeepDefence is lightweight and compatible with standard architectures, which make it scalable and a practical solution for building more robust DL systems. In testing it outperformed adversarial training that used PGD adversarial examples. Unstructured black-box attacks pose a challenge to DeepDefence, and it is important to note that this defence against the chosen attacks in testing achieved an average performance score of 70.55 % with an average +/- being 3.33 % through the five reruns. [13] This defence is better when compared to other defences used in the testing [13], but it is not close to perfect. Possibly combining DeepDefence with other defences in the future could improve a model's robustness.

4.4 LLMAD

Large Language Model Adversarial Defence (LLMAD) is a new proposed defensive method that solely focuses on defending large language models (LLM). Current adversarial defensive models struggle with improving robustness while keeping classification accuracy on clean data. The effectiveness between different tasks can also be inconsistent and possibly lead to weakening a model's robustness. LLM is separate from the visual domain in the sense that most adversarial attack and defence methods in the image domain cannot be utilised in the text domain. [14]

LLMAD consists of two modules, which are the perturbation detection module and the other is perturbation correction module. [14] The step-by-step process of an adversarial sample being input into a ML model that is using LLMAD is shown in Figure 4.4. It first detects and then corrects adversarial attacks by utilising LLMs semantic understanding and text processing capabilities. By using these two modules together, it lessens the impact on the model's clean data. [14]

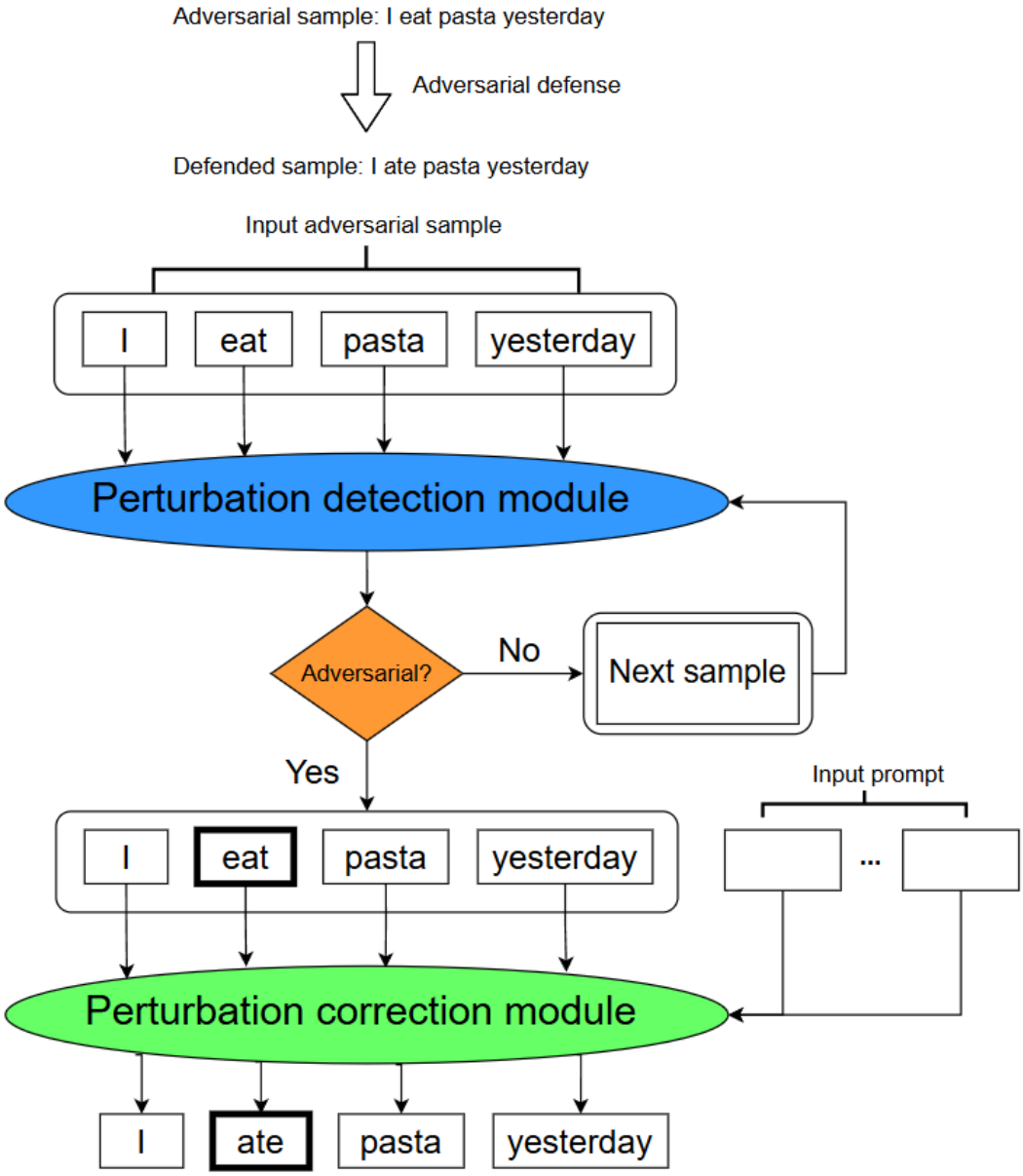

Figure 4.4: LLMAD model detecting and correcting an adversarial input. Inspiration for the figure is taken from [14].

The detection module in the LLMAD is designed to differentiate if an input is an adversarial attack/example or a clean sample. It improves the efficiency of the defence and does not modify clean samples. It detects perturbations, and when found, it pushes the adversary to the correction module. [14]

The perturbation detection module is trained with data pairs. These data pairs consist of an error-containing sequence and the corresponding sequence without the error. The goal of training is to minimise loss to improve performance in perturbation detection. [14]

The correction module receives the adversarial inputs that were detected. It inputs them into a prompt template to find a possible correction and correct the input for the LLM. Changes to the sentence length should be minimal so it accounts for real-world scenarios where one or two words are misspelt or do not fit the sentence context. [14] It resembles what spellcheck is in some text editors.

Using LLMAD improves a model's classification accuracy by an average of 13.4% when compared to other defences. LLMAD has stable and superior detection performance as it outperforms other defences in the majority of dataset and attack combination scenarios. [14] While the results are better compared to other defences, they are not perfect and the accuracy and effectiveness go over over 90% only once.

5 CONCLUSION

In this thesis, different adversarial defensive methods were considered and evaluated based on their effectiveness. These methods ranged from traditional approaches to more recently suggested techniques. In the following paragraphs, the research questions of the study will be answered based on the information gathered in this thesis.

RQ1: Three different defensive strategies were introduced: Adversarial training, Feature Squeezing and the Trapdoor defence. Each defence works in a different way.

Adversarial training, which is one of the most popular adversarial defensive methods, works by adding perturbated data into the training datasets at every training session. That makes the ML model able to recognise what an adversarial example looks like and therefore not get tricked by it.

Feature squeezing is a detection based defence that goes through the inputs before they are sent through to the ML model. The defence works by removing features from the input and then examining if the input is possibly an adversarial example.

Trapdoor defence adds trapdoor perturbation to clean inputs, which are given a different label and then added to the training data. The perturbated data is then used to catch adversarial examples.

RQ2: None of the three mentioned defences work against every attack 100% of the time. Adversarial training has a big weakness, where it is not as effective against

adversarial attacks that it was not trained with. When a different attack is used, the defence effectiveness is not guaranteed. Feature squeezing is not reliable alone and especially when the attacker has knowledge of the defence and what squeezer is being used. With the trapdoor defence there are limitations on how many trapdoors can be added as they can start interfering with each other. This makes it harder to defend when there are a lot of different labels, as all labels should be defended against adversarial attacks.

RQ3: New defensive methods or strategies that this thesis found and covered are TMM, robustness through cryptographic techniques, DeepDefence and LLMAD. These defences have minimal testing results as they are very new, so it can be hard to say how good they can be in any scenario.

TMM is similar to trapdoor defence, but it modifies the models gradient to make perturbed inputs get caught in the traps. The TMM-Defence contains three filters to detect different types of attacks like C&W and PGD. The impact of this method on accuracy of clean data is low while performing better than the trapdoor defence. The defence can have many classes, while being cheap to run and highly scalable.

Using cryptographic techniques like homomorphic encryption and multi-party computation can lessen adversarial attacks and therefore improve security and resistance. By using those techniques the accuracy improves. In testing it performed better than adversarial training. When used the model does need more resources and will take more time, which is not ideal, but a promising proposal nonetheless.

DeepDefence makes models more robust by not letting perturbation to add up across different layers in the neural network. It is scalable, lightweight and also outperformed adversarial training. Still it does not reach the 100% detection rate, but an average of 70.55% across all attacks that were tested.

LLMAD is a method directed towards defending LLM's. It has a trained detection module and a correction module. Detection performance is superior and accuracy of a model using LLMAD increase more when compared to other defences.

There still is no adversarial defence that can defend a ML model against all adversarial attacks with a 100% success rate. This has proven to be a challenging task, but due to the popularisation of ML, it is a area of research that needs to continue. New methods and ideas are constantly being proposed, but no breakthrough has been made as of yet.

The new methods in this thesis are hard to compare to others, due to the fact that people use different attacks and dataset combinations. Unless compared directly to one another in the articles it is difficult to say which is the best defence overall. Each defence is better against certain attacks, but there always is a attack that can get through. There are mentions of combining different types of defences to possibly strengthen the defence even more, but no study or experiments of that kind were found. That could be an area of future research. The baseline or standard of adversarial defence seems to be adversarial training, which is the most used defence for comparisons.

In testings, ready made datasets like MNIST and CIFAR-10 were always used to train the ML models. If the datasets were bigger and higher quality, along with more powerful computers to run the models, maybe the results for the defences would have been better as well, as more data will improve ML model's quality. It is important to mention that there are a lot of different scenarios in the real world and they all cannot be captured to be put into a dataset. Therefore, it is expected for a ML model to make mistakes and hard to make a model that is ready for every new scenario. Defences can help, but especially in critical sectors currently, neither the defence or the ML model should be relied upon. More future research and innovation is needed for that not to be the case.

References

[1] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning", en, *Engineering (Beijing)*, vol. 6, no. 3, pp. 346–360, Mar. 2020.

[2] V. D. Gowda, P. S. Kumar, P. Damacharla, M. Tarambale, P. K. Lakineni, and K. Sripathi, "Enhancing machine learning robustness against adversarial attacks through cryptographic techniques", *J. Inf. Optimiz. Sci.*, vol. 46, no. 4-A, pp. 927–937, 2025.

[3] R. R. Wiyatno, A. Xu, O. Dia, and A. de Berker, "Adversarial examples in modern machine learning: A review", 2019. eprint: `1911.05268` (cs.LG).

[4] R. Upreti, P. G. Lind, A. Elmokashfi, and A. Yazidi, "Trustworthy machine learning in the context of security and privacy", en, *Int. J. Inf. Secur.*, vol. 23, no. 3, pp. 2287–2314, Jun. 2024.

[5] Y. L. Khaleel, M. A. Habeeb, and H. Alnabulsi, "Adversarial attacks in machine learning: Key insights and defense approaches", *Applied Data Science and Analysis*, vol. 2024, pp. 121–147, Aug. 2024.

[6] R. Muthalagu, J. Malik, and P. M. Pawar, "Detection and prevention of evasion attacks on machine learning models", en, *Expert Syst. Appl.*, vol. 266, no. 126044, p. 126 044, Mar. 2025.

[7] M. Standen, J. Kim, and C. Szabo, "Adversarial machine learning attacks and defences in Multi-Agent reinforcement learning", en, *ACM Comput. Surv.*, vol. 57, no. 5, pp. 1–35, May 2025.

[8] B. Ghosh, H. Harikumar, S. Venkatesh, and S. Rana, "Targeted manifold manipulation against adversarial attacks", in *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, Copenhagen, Denmark: IEEE, Apr. 2025, pp. 427–438.

[9] A. Abomakhelb, K. A. Jalil, A. G. Buja, A. Alhammadi, and A. M. Alenezi, "A comprehensive review of adversarial attacks and defense strategies in deep neural networks", en, *Technologies (Basel)*, vol. 13, no. 5, p. 202, May 2025.

[10] E. Yocam, A. Rizi, M. Kamepalli, V. Vaidyan, Y. Wang, and G. Comert, "Quantum adversarial machine learning and defense strategies: Challenges and opportunities", 2024. eprint: `2412.12373` (quant-ph).

[11] A. Aldahdooh, W. Hamidouche, S. A. Fezza, and O. Déforges, "Adversarial example detection for DNN models: A review and experimental comparison", en, *Artif. Intell. Rev.*, vol. 55, no. 6, pp. 4403–4462, Aug. 2022.

[12] S. Shan, E. Wenger, B. Wang, B. Li, H. Zheng, and B. Y. Zhao, "Gotta Catch'Em all: Using honeypots to catch adversarial attacks on neural networks", in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, Virtual Event USA: ACM, Oct. 2020, pp. 67–83.

[13] C. Lin, T. Yeap, I. Kiringa, and B. Zhang, "DeepDefense: Layer-wise gradient-feature alignment for building robust neural networks", Nov. 2025. arXiv: `2511.13749 [cs.LG]`.

[14] L. Che, C. Wu, and Y. Hou, “Large language model text adversarial defense method based on disturbance detection and error correction”, en, *Electronics (Basel)*, vol. 14, no. 11, p. 2267, May 2025.