International Journal of Network Dynamics and Intelligence Scilight

Survey/Review Study

On the Formal Evaluation of the Robustness of Neural Networks and Its Pivotal Relevance for AI-Based Safety-Critical Domains

Mohamed Ibn Khedher [1,*], Houda Jmila [2], and Mounim A. El-Yacoubi [2]

[1] IRT-SystemX, 2 Bd Thomas Gobert, Palaiseau 91120, France
[2] Samovar, Telecom SudParis, Institut Polytechnique de Paris, 19 place Marguerite Perey, Palaiseau 91120, France
[*] Correspondence: ibnkhedhermohamed@hotmail.com

Received: 11 July 2023
Accepted: 31 October 2023
Published: 21 December 2023

Abstract: Neural networks serve as a crucial role in critical tasks, where erroneous outputs can have severe consequences. Traditionally, the validation of neural networks has focused on evaluating their performance across a large set of input points to ensure desired outputs. However, due to the virtually infinite cardinality of the input space, it becomes impractical to exhaustively check all possible inputs. Networks exhibiting strong performance on extensive input samples may fail to generalize correctly in novel scenarios, and remain vulnerable to adversarial attacks. This paper presents the general pipeline of neural network robustness and provides an overview of different domains that work together to achieve robustness guarantees. These domains include evaluating the robustness against adversarial attacks, evaluating the robustness formally and applying defense techniques to enhance the robustness when the model is compromised.

Keywords: neural network verification; adversarial attacks; defense techniques; formal robustness guaranties

1. Introduction

In the past decade, artificial intelligence (AI), machine learning and particularly deep learning, have achieved unprecedented levels of performance in many applications. Deep neural networks (DNNs) are cutting-edge algorithms that have transformed machine learning and set new benchmarks. The state-of-the-art performance of DNNs has left an indelible impact on a wide range of demanding application fields with numerous achievements reported in computer vision [1–5], cybersecurity [6], robotics [7], and autonomous system control [8].

Despite DNNs' remarkable capability of addressing critical applications, recent studies have revealed their concerns about vulnerability. It has been demonstrated that even small perturbations in the input space can lead to incorrect decisions by DNNs [8]. This process, known as adversarial attacks, involves careful modification of the input data and the possible triggering of the mis-prediction by the network. These findings have significantly heightened awareness regarding the imperative need to ensure the intended behaviour of machine learning systems, particularly DNNs, when faced with perturbed inputs. Consequently, addressing this fundamental challenge has become a pressing concern in the field.

Adversarial examples are generated by introducing subtle perturbations to correctly classify input data by a neural network. The objective is to modify the prediction of the network in such a way that the perturbed input is incorrectly classified. These adversarial examples are not randomly generated, but are carefully crafted through specific computational methods. There are various techniques to generate such examples, but most of them focus on minimizing the distance between the original and adversarial examples, while ensuring that the network's prediction becomes incorrect. Some adversarial examples require access to all the parameters of the classifier, namely white-box attacks. In contrast, black-box attacks only require access to the prediction function. These different examples highlight the varying levels of the information required to generate adversarial examples and underscore the importance of understanding and mitigating the vulnerability of neural networks. Various techniques for generating adversarial examples have been proposed in the literature [9–11]. In situations where human lives are at peril, adversarial examples can pose a significant threat to human security.

Studying the robustness of a DNN involves examining its ability to consistently make correct decisions for all perturbations applied to a correctly classified input. This verification process aims to assess the DNN's resilience against external attacks, commonly known as adversarial attacks. The objective is to ensure that the DNN remains resistant to such attacks, maintaining its capacity to deliver accurate and reliable results even in the presence of adversarial perturbations. By evaluating the robustness of the DNN against these challenges, we can determine its capacity to withstand and overcome potential threats, reinforcing its trustworthiness and effectiveness in critical applications.

In many tasks, neural networks are becoming the backbone of the prediction framework, and inaccurate outputs can be costly. The validation method for neural networks has traditionally required examining the network's performance on a wide range of input data samples, and deciding if the outputs match the desired outcomes. Testing all possible inputs, however, is impossible due to the effectively unlimited input space. Even networks that perform well on a wide sample of inputs may struggle to generalize appropriately to new contexts, and remain vulnerable to adversarial attacks. Formal verification results of DNNs over the input space are examined in this study. Based on analysis methods, these algorithms can be split into three categories: search, reachability, and optimization. In the category of optimization, these methods employ optimization strategies to disprove assertions by treating the neural network's represented function as a constraint within the optimization process. Reachability-based methods utilize layer-by-layer reachability analysis of the network to establish the validity of properties. Search-based methods, on the other hand, explore input cases to invalidate assertions. Search techniques are often combined with reachability or optimization approaches, as the two approaches provide valuable directions to potential search paths. By reviewing and comparing these methods, this article aims to provide insights into the formal verification results of DNNs and their potential for enhancing network trustworthiness and reliability in real-world applications.

In cases where the model is not robust against adversarial attacks and lacks robustness guarantees, defense techniques can be applied to enhance its resilience against disturbances. These defense techniques primarily aim to intervene during the learning phase to make the model more robust.

The contributions of our paper are given as follows.

● In literature, the fields of robustness, including formal robustness evaluation, local robustness evaluation and robustness improvement, are addressed separately. To our knowledge, we are the first to present these three aspects within the same pipeline. We call this pipeline the *robustness pipeline*.

● As these fields of robustness are usually treated separately, there are some state-of-the-art results focused on specific domains where authors present numerous solutions. This can sometimes make it challenging for readers to choose the best method suitable for their specific use case. In this paper, we primarily present the most popular methods to help readers understand the general principles of selecting from various approaches.

● Generally, robustness is approached from mathematical foundations with a focus on the theory. In this paper, we aim to explain robustness in a more accessible way. Specifically, we briefly touch on the mathematics, and show how robustness is applied concretely in our daily lives and why robustness is essential to our safety.

Figure 1 depicts the general pipeline of robustness and illustrates the interconnection between different components, namely the evaluation of the model against adversarial attacks, the formal evaluation of obtaining formal guarantees of robustness, and the application of defense techniques to enhance robustness.

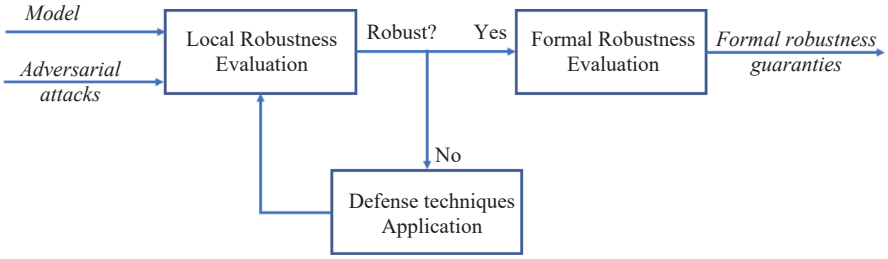

Figure 1. General pipeline of robustness evaluation.

The remaining of the paper is structured as follows. Section 2 demonstrates the vulnerability of neural networks to adversarial attacks. In section 3, the necessity of adopting a neural network verification approach is discussed. Sections 4, 5 and 6 present the popular approaches of NNV, respectively, based on reachability, optimization and search. In Section 7, we briefly discuss defense methods that aim to make neural networks robust. Section 8 discusses formal method limitations and provides an overall discussion of our study. Section 9 concludes and provides some perspectives.

2. Adversarial Attacks and Robustness

The ability of neural networks to withstand adversarial attacks is essential for ensuring their reliability and secu-

rity. Adversarial attacks involve intentionally manipulating neural networks by making small changes to input data, which results in incorrect outputs. In this section, we discuss the robustness of neural networks through an example of adversarial attacks, and then we describe some of the most popular attacks.

2.1. Robustness

To illustrate the concept of robustness, let us consider the example shown in Figure 2. The figure demonstrates an instance where an input has been altered to completely changes the network's decision. This particular attack is known as the fast gradient sign method (FGSM), which involves adding noises to the image based on the gradient sign of the cost function.

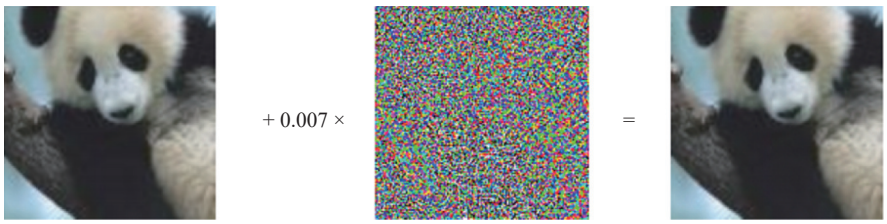

Figure 2. Impact of a FGSM attack on image classification [9]. From left to right: an original image classified as panda with 57.7% of confidence. A noise generated using the FGSM technique. A resulting image classified as Gibbon with 99.3% of confidence.

In the figure, we observe that the original input, which is correctly classified as a panda with a confidence level of 57.7%, undergoes a complete label change after being perturbed. This original input is now classified as a gibbon, with 99.3% certainty, even though the perturbed image is visually indistinguishable from the original one. Such an error can have severe consequences depending on the context in which it occurs. Figure 2 shows the visual representation of the significant impact that can be produced by the input perturbation.

2.2. Adversarial Attacks

Adversarial attacks can be categorized into two types: the white-box attacks where the attacker has full access to the model's internal workings, and the black-box attacks where the attacker has limited information and can only query the model for prediction. Understanding these attack types is important for developing robust defense mechanisms and enhancing the security of neural networks against adversarial manipulations.

● *White-box attacks.* In white-box attacks, the attacker has complete access to the architecture and parameters of the classifier. This allows the attacker to exploit information, such as the gradients of the loss function associated with the model, to generate effective attacks.

● *Black-box attacks.* In a black-box attack, the attacker does not have complete access to the policy network. The model parameters are unknown in this scenario. Typically, for classification models, the attacker has access only to the model's output (decision).

Moreover, depending on the attacker's goal, two attack forms are listed as follows.

● *Targeted attacks.* Direct the ML algorithm to a specific class, where the adversary tricks the classifier into predicting all adversary examples as a specific target class.

● *Non-targeted attacks.* Aim to misclassify the input sample away from its original class, regardless of the new output class.

2.3. Examples of Neural Network Attacks

In the literature, several attacks have been proposed. In this section, we provide details of the most popular attacks for both black-box and white-box categories.

2.3.1. White-Box Attacks

Several attacks have been proposed in the literature, such as the FGSM [9], basic iterative method (BIM) [12], projected gradient descent (PGD)[13], Jacobian-based saliency map attack (JSMA) [10] and DeepFool (DF) [11].

The FGSM [9] is one of the most popular white-box attacks, and generates adversarial examples by introducing noises to the original sample along the gradient directions. Mathematically, in the FGSM, the adversarial samples can be obtained by using the following equation:

$$x' = x + \epsilon \cdot \text{sign}(\Delta_x J(x, y)) \tag{1}$$

where J represents the loss function of the trained model, Δ_x denotes the gradient of the model with respect to a normal sample x having a correct label y, and ϵ controls the magnitude of the perturbation. The FGSM attack is com-

putationally efficient, requires only one gradient evaluation, and can be applied directly to a batch of inputs. This makes FGSM a popular choice for adversarial training, where a large number of adversarial samples need to be generated. Additionally, the FGSM can be readily adapted to targeted attack scenarios by descending the gradient $\Delta_x J(x, y')$, where y' represents the target label.

2.3.2. Black-Box Attacks

In the black-box setting [14], an attacker has limited knowledge about the model and can only observe the labels or confidence scores of the outputs. The three categories of black-box attacks are given as follows based on the information available to the attacker.

- *Score-based attack.* In Score-based attacks, the attacker has access to the predicted scores or the probability of each predicted label belonging to the model's classification classes [15].
- *Decision-based attacks.* The attacker only has access to the final decision of the model without any confidence score [16, 17].
- *Transfer-based attacks.* The attacker has access to a portion or the entirety of the training dataset which is used to train another fully observable model known as the "substitute model". This substitute model is trained to emulate the behaviour of the attacked model, referred to as the "target model". Adversarial perturbations synthesized from the substitute model are then used to attack the target model.

3. Neural Network Verification Problem

In this section, we describe the need to verify neural network robustness. Additionally, we outline the general formulation of a neural network verification problem using a concrete example.

3.1. Validation of Neural Network

Neural networks are employed for critical tasks, and the consequences of incorrect outputs can be costly. Traditionally, the validation of neural networks primarily involves evaluating the network's performance on a substantial number of data samples in the input space, and verifying if the outputs align with the desired outcomes. However, due to the virtually infinite nature of the input space, it is impractical to check every possible input. Even networks (that exhibit strong performance on a large sample of inputs) may struggle to generalize accurately to new situations and remain susceptible to adversarial attacks. Hence, it is crucial to ensure that a neural network can accurately classify inputs even when they are perturbed. This process is known as network robustness verification. To evaluate robustness, a neural network verification (NNV) approach is required. The general principle of an NNV problem is described as follows. Perturbations, or attacks, can manifest in various forms and levels of intensity. Given an initially correctly classified sample, the verification of network robustness against attacks involves confirming that all perturbations generated by the attack process result in correct classification.

3.2. Need of an NNV Approach

The NNV approach is a valuable tool that finds applications in various tasks, ranging from non-critical applications like image recognition in non-sensitive domains to highly critical applications like cybersecurity [18] and autonomous vehicles. For example, a comprehensive verification process is required to assess the robustness of a neural network classifier designed for recognizing handwritten digit images against pixel illumination attacks. This involves generating all possible variations of the images resulting from different illumination conditions. Each perturbed image is then provided as an input to the neural network, and the resultant classification accuracy is examined to determine if the network can still correctly classify the distorted images. Based on these rigorous tests, the ability to maintain accurate classification can be evaluated in the presence of illumination variations.

Adversarial examples can pose a significant risk to human safety, especially in scenarios where lives are at stake. For instance, for self-driving vehicles, the robustness against environmental perturbations is crucial such as varying lighting conditions or reduced visibility. Misinterpreting these perturbations can lead to incorrect decisions, potentially resulting in catastrophic consequences. To ensure an acceptable level of robustness, autonomous driving systems must undergo rigorous evaluation in the face of numerous challenges such as occlusions, and partial/total absence of perception and lighting.

To provide a concrete example within the context of autonomous vehicles [19], let us examine two images in Figure 3. On the left, we have an initial image depicting a clear and recognizable speed limit sign of "20 km/h". On the right, a blurring perturbation is applied to the image. Despite the degraded quality of the second image, both images still appear to depict the same content to the human perception: a speed limit sign of "20 km/h". Surprisingly, the DNN specifically developed for traffic sign classification misclassifies the perturbed image as a speed limit sign of "80 km/h" [19]. This example demonstrates the vulnerability of DNNs to adversarial perturbations, highlighting

the discrepancy between human the perception and network interpretation. Such findings emphasize the critical need to address the robustness and reliability issues of AI systems (especially in safety-critical domains like autonomous vehicles) to ensure accurate and trustworthy decision-making in the face of adversarial inputs.

Figure 3. Adversarial attacks. left: initial image, right: Blurring [19].

3.3. General Formulation of NNV Problem

Given a DNN $N: x \rightarrow y$, a set P covering the inputs and a set Q covering the outputs, the NNV problem is to seek an answer to the question: is there an input x along with its output $y = N(x)$ that verifies P and fails Q?

In the example of Figure 4, verifying the robustness of the DNN is equivalent to checking the following constraint where $\forall(x_1 > -2) \wedge (x_1 < 2) \wedge (x_2 > -2) \wedge (x_2 < 2)$ and $(y > -5)$.

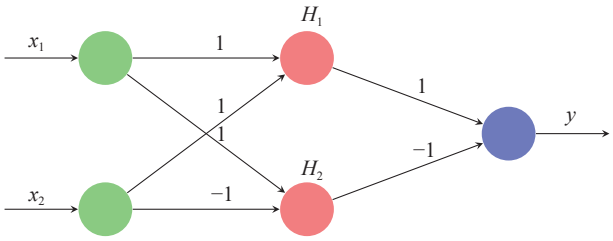

Figure 4. Example of neural networks.

3.4. Categorization of Approaches

NNV approaches can be categorized based on their formulation of three types of analysis: reachability, optimization, and search.

● *Reachability*. Methods in this category employ layer-by-layer reachability analysis to assess the network. The reachability problem can be formulated as follows: given a computational system with a set of allowed rules, decide whether a certain state of a system is reachable from a given initial state of the system. Popular approaches include ExactReach, MaxSens and AI^2.

● *Optimization*. These methods utilize optimization techniques to falsify assertions. The neural network's function is treated as a constraint within the optimization problem, resulting in a non-convex optimization setting. Primal optimization methods encode nonlinear activation functions as linear constraints. NSVerify, MIPVerify, and ILP are illustrative examples. Dual optimization simplifies the constraints through methods like Lagrangian dual techniques (e.g., duality, ConvDual, and Lagrangian decomposition) and semidefinite programming approaches (e.g., certify).

● *Search*. Methods in this category search for counter examples to disprove assertions. Search is often combined with reachability or optimization as these methods offer potential search directions. Representative methods that integrate search and reachability include ReLuVal, Neurify, DLV, Fast-Lin, Fast-Lip, CROWN, Nnenum, and VeriNet. For search and optimization, notable examples are ReLuplex, Marabou, Planet and SherLock.

In the following sections, we describe the approaches from the three categories. In summary, these categories (reachability, optimization, and search) encompass various analysis methods employed by algorithms in the verification of neural networks.

4. Reachability Approach

In computer science, the reachability problem is to determine whether a final situation is reachable from an initial situation. Given a neural network N and an input set X, the reachability set Y is defined as all the possible outputs: $Y = \{y, y = N(x), \forall x \in X\}$. If the reachable set is included in the desired set, the neural network is declared secure; otherwise, the NN is declared as not secure.

The approaches proposed in the literature consist of performing the exact [20] or approximate [21, 22] reachability analysis to determine the set of outputs associated with the set of neural network inputs.

Exact reachability is used for piecewise linear networks to compute the reachable set for each linear segment of the network. It is facilitated by a tool called ExactReach [20], which keeps track of all sets involved in the analysis. When dealing with the nonlinear region of the network, constraint refinement is necessary, requiring the set to be divided into pieces that only operate on the linear segments. The number of linear segments, however, grows exponentially with the increase in nodes and layers, making exact reachability computationally demanding. On the other hand, over-approximation methods are applicable to networks with any nonlinear activation function. They provide an overestimated approximation of the reachable set. Unlike exact reachability, these methods do not keep track of the reachable set for each linear segment, making them computationally tractable. However, there is a trade-off between the level of over-approximation introduced by the verification algorithm and its computational efficiency.

4.1. Exact Reachability

The approach presented in [20] consists of exactly representing the neural network inputs by the union of polyhedra. The computation of the reachability set is computed at each layer of the network using the polyhedron manipulation tools. ExactReach, as mentioned in [20], is a tool specifically designed to perform exact reachability analysis for neural networks utilizing either linear or rectified linear unit (ReLU) activations. When considering ReLU activation functions, ExactReach demonstrates that if the input set can be represented as a union of polytopes, then the resulting reachable set of outputs will also be a union of polytopes. This characteristic is particularly useful as it allows for a precise representation of the possible output ranges of ReLU-based networks. By leveraging this property, ExactReach enables a detailed exploration of the reachable set for networks employing ReLU activations. This analysis involves computing the exact boundaries of the reachable set, and considering all possible combinations of input values and activation functions. As a result, the tool provides a comprehensive understanding of the network's behaviour and helps determine the output ranges associated with specific input regions.

4.2. Approximate Reachability

The authors of [22] have introduced a novel verification scheme called the "abstract interpretation for AI" (AI^2). The primary objective of AI^2 is to provide a framework for modeling neural network inputs using the zonohedron-based geometric shape which is a specific type of polyhedron. The utilization of zonohedra allows for precise representation of input data within the neural network. To facilitate the analysis of neural network behaviours, a set of abstract operators is defined within the AI^2 framework. These operators enable the tracking and monitoring of the zonohedron's evolution as it traverses through different layers of the neural network. By applying these operators, the abstract interpretation approach offers insights into how the geometric representation of the input data propagates and transforms within the network.

An overview of the principle of the abstract interpretation is provided, and the ReLU activation function is taken as an example. The objective is to evaluate the robustness of a neural network subject to a correctly classified input which is denoted as A and characterized by its pixels: $A = (x_1, x_2, \cdots, x_n), x_i \in [0,1], \forall i \in [1,n]$. Each component x_i represents a pixel value normalized between 0 and 1. In this work, the authors focus on a specific attack known as the L_∞ attack. This attack involves constructing a ball centered around the original image A. The radius of this ball is denoted as ϵ, representing the intensity or magnitude of the attack. The input, originally represented as a vector in the range $[0,1]^n$, is transformed into an abstract domain denoted as X. This domain is defined as the Cartesian product $\times_{i=1}^{n}[l_i, u_i]$, where $l_i = x_i + \epsilon$ and $u_i = x_i - \epsilon$. These values, namely the upper bound and lower bound, determine the boundaries of the abstract domain. This abstract domain X encapsulates an “adversarial region” that encompasses all possible images generated through this attack type (Figure 5). The neural network is considered to be robust if it is capable of correctly classifying any element within this region. In other words, the neural network demonstrates resilience against adversarial attacks, as inputs originating from this region will retain the same label as the original input.

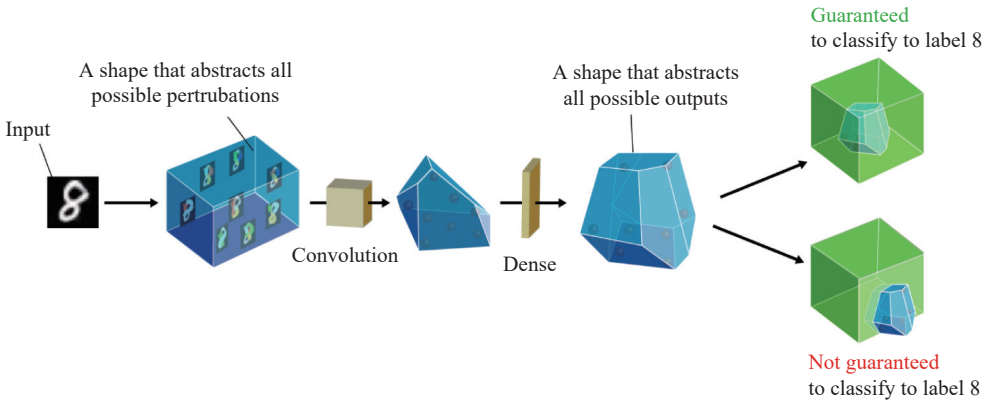

Figure 5. Abstract Interpretation for Neural Network Verification [22].

An essential step in the mathematical formulation of the abstract interpretation is to convert the layers of a neural network into abstract layers. In order to proceed with the abstraction process, each neuron is associated with 1) two *linear constraints* that define the polyhedron; and 2) two values that represent the range of values achievable to the neuron. These four equalities/inequalities completely define the abstract domain at each stage of the neural network processing. These equations are generated and processed by the abstract transformers. The abstract neural network of the example in Figure 4 is presented in Figure 6.

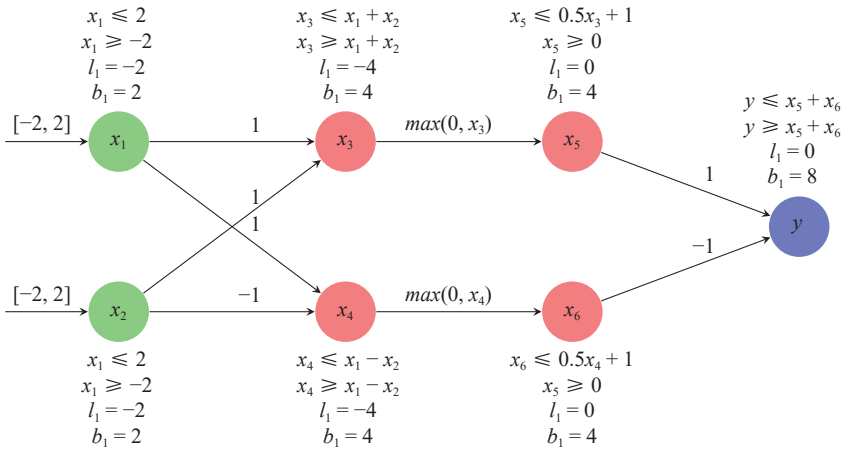

Figure 6. Example of abstract neural networks.

AI^2 is not suitable for all activation functions. The authors of [23] have adapted the method to evaluate neural networks by using the LeakyReLU function. This adaptation opens up new possibilities for scenarios where LeakyReLU is predominant.

5. Optimization Approach

Optimization techniques are employed to challenge the assertion. These methods consider the neural network's function as a constraint in the optimization process. Consequently, the optimization problem becomes non-convex. In primal optimization, various approaches have been developed to encode nonlinear activation functions as linear constraints. NSVerify [24], MIPVerify [25], Big-M [26] and ILP [27] are typical methods that simplify constraints using primal optimization. Dual optimization techniques offer an alternative way to simplify constraints. Lagrangian dual methods like the duality [18], ConvDual [19], and Lagrangian decomposition [20], and semidefinite programming methods such as Certify [21] and SDP [22] are representative approaches for dual optimization.

5.1. Primal Optimization

Primal optimization methods aim to challenge the assertion by incorporating the network structure as a constraint in the optimization problem. Existing methods are primarily applicable to networks with ReLU activations. To address this, various techniques have been developed to encode the network using linear constraints or mixed integer linear constraints, thereby leveraging the piecewise linearity inherent in the ReLU activation function. These methods enable the exploration of optimization solutions by considering the specific characteristics of ReLU-based networks.

The authors of [26] have considered the *bigM* technique as an automated encoder that linearizes *ReLU* which

is a non-linear activation function. This is a mixed integer linear programming transformation that exactly transforms non-linear constraints into linear inequalities.

Considering the example of Figure 4 where the neurons H_1 and H_2 are activated with ReLU, the NNV problem using $big-M$ can be formulated as a problem of type $\max\ Z = Constant$ subject to a set of constraints. These constraints represent the relationship between neurons of different layers. This approach involves linearizing the ReLU function by replacing it with a set of constraints. The relationship between the neuron H_1 and inputs x_1 and x_2 is formulated as follows:

$$
\begin{aligned}
&a_{out}(H_1) \geqslant (\theta_1 - 1)M + (x_1 + x_2)\\
&a_{out}(H_1) \leqslant (1 - \theta_1)M + (x_1 + x_2)\\
&a_{out}(H_1) \leqslant \theta_1 \times M\\
&x_1 + x_2 \geqslant (\theta_1 - 1) \times M\\
&x_1 + x_2 \leqslant \theta_1 \times M\\
&\theta_1 \in \{0, 1\}
\end{aligned}
$$

Here, a_{out} is the input of neuron H, and M is the largest value.

5.2. Dual Optimization

In primal optimization methods, various techniques have been devised to encode the constraints imposed by the network. Alternatively, dual optimization can be utilized to simplify the constraints present in the primal problem. Dual optimization typically results in simpler constraints compared to other techniques used in primal optimization. The objectives in dual optimization (which correspond to the constraints in primal optimization) tend to be more complex than those in the primal problem. The construction of the dual problem often involves incorporating relaxations. It is important to note that these approaches are considered incomplete due to relaxations. Indeed, dual optimization is characterized by using relaxations (approximations) of the linear equations to solve the optimization problem.

Duality [28] is a method that addresses the optimization problem by solving a Lagrangian relaxation problem to enable the derivation of output bounds. ConvDual [29] contains a dual approach which is used to estimate output bounds, and begins by performing a convex relaxation of the network within the primal optimization framework to simplify the dual problem. Certify [30] employs a semidefinite relaxation technique to compute over-approximated certificates or bounds.

6. Search

6.1. Search and Reachability

In order to achieve a desirable balance between computational efficiency and approximation accuracy, it is crucial for reachability methods to strike the right chord. By integrating reachability with search techniques, there is a promising prospect of enhancing both efficiency and precision. These approaches usually entail searching within either the input or hidden spaces to uncover counterexamples. Nevertheless, it is important to note that since reachability analysis tends to be over-approximated, these methods may encounter potential incompleteness issues even though they are deemed reliable.

Reachability methods (that utilize interval arithmetic) typically yield loose bounds since they do not consider dependencies among the hidden nodes. In contrast, symbolic interval propagation can offer tighter bounds by tracking these dependencies on a layer-by-layer basis. More accurate estimation of the bounds can be achieved by incorporating symbolic interval propagation and accounting for the interdependencies within the network's hidden nodes. The reachability analysis is done symbolically, while the search is done through iterative interval refinement. Indeed, ReluVal [31] employs a combination of symbolic interval analysis and search techniques to reduce the extent of over-approximations in the output sets. In its search procedure, ReluVal employs iterative interval refinement by iteratively dividing its input range. This iterative interval refinement process i also utilized in BaB [32].

6.2. Search and Optimization

Combining search with optimization can be achieved by searching in the input space or searching in the function space. When searching in the function space, the exploration is focused on finding potential activation patterns. An activation pattern refers to the assignment of values (e.g., on or off signals in the ReLU) to each activation function within the network. To facilitate this approach, methods often utilize satisfiability (SAT) or satisfiability module theory (SMT) solvers. By leveraging these solvers, effective search algorithms can be developed to explore and optimize activation patterns within the function space. In this section, we explore approaches that combine search and optimization methods to achieve efficient algorithms by taking advantage of the piecewise linearity in activation

functions.

Reluplex [33] utilizes the simplex algorithm to analyse ReLU networks, specifically focusing on the search for counterexamples. By applying the simplex algorithm to ReLU networks, Reluplex aims to identify scenarios where the network fails to meet the desired criteria or exhibits unexpected behaviours.

The simplex algorithm, originally developed for linear programming problems, is adapted within Reluplex to explore the input space of ReLU networks. It systematically searches for counterexamples which are input instances that violate certain properties or constraints specified for the network. By identifying counterexamples, Reluplex provides valuable insights into the limitations and vulnerability of ReLU networks, helping researchers and practitioners refine and improve their models.

7. Defense

Defense techniques are designed to enhance the resilience of a model against adversarial attacks. Three categories of defense techniques are highlighted in the literature, namely the approaches based on 1) modifying the input data; 2) modifying the classifier; 3) and adding an external model, as described below.

7.1. Modifying the Input Data

Instead of directly manipulating the training process or model architecture, these techniques focus on altering the training data or input data during testing. For instance, a technique known as "Gaussian data augmentation" [34] involves augmenting the original dataset by adding copies of the samples with Gaussian noises. The rationale behind this approach is that training the model to predict the same outcome (for both the original instance and its slightly perturbed version) can enhance its generalization ability and robustness against perturbations.

The simplicity (ease of implementation) and effectiveness of this method against both black-box and white-box attacks contribute to the widespread usage of the method.

7.2. Modifying the Classifier

The process of defense through classifier modification entails altering the original classification model by modifying loss functions, incorporating additional layers or sub-networks, and making other relevant adjustments. One notable example is the "gradient masking" technique, which aims to enhance the model's defense against adversarial attacks by concealing or masking its gradient information from potential attackers. While many adversarial attack methods rely on exploiting knowledge of the gradient, the gradient masking method works by modifying the machine learning model to obscure or hide its gradient, making it more difficult for attackers to manipulate or exploit. This technique aims to hide gradient information from potential attackers by manipulating the gradients during training, and prevents adversaries from easily crafting effective adversarial examples based on gradient information. For example, the authors of [35] have developed a gradient masking method to defend against C&W attacks.

7.3. Adding an External Model

In contrast to previous approaches that involve modifying the original model itself, the defense techniques discussed here maintain the integrity of the original model and introduce additional external models during the testing phase. This strategy aims to bolster the model's resilience against adversarial attacks.

An illustrative example has been presented in [36], where the authors have leveraged the framework of *generative adversarial networks (GANs)* to train the primary network alongside a generator network. The generator network's objective is to produce perturbations that can potentially deceive the primary network. By incorporating this additional model, the defense mechanism aims to enhance the model's ability to detect and counteract adversarial inputs.

The utilization of external models in conjunction with the original model during testing provides a supplementary layer of defense, and can help mitigate the impact of adversarial attacks. This approach showcases the potential benefits of combining different models and techniques to fortify the overall robustness of the system against adversarial threats.

8. Discussion

Formal methods for neural network verification have limitations when it comes to the scalability of large-scale neural networks typically used in industrial applications. The computational complexity involved in analysing and verifying these complex networks increases exponentially with the size of the model, making it challenging to apply formal methods efficiently. Additionally, formal methods often rely on specific assumptions and mathematical abstractions that may not capture the full complexity of different types of layers used in neural networks. Consequently, these methods may be limited in their applicability and effectiveness, as they may only work well with cer-

tain types of layers or network architectures. It remains an active area of research to overcome these limitations and develop scalable and comprehensive formal methods for large-scale neural networks in order to ensure reliable and robust deployment of neural networks in real-world settings.

In the literature, formal methods can be exact, particularly the methods based on mixed integer programming formulations. These methods provide precise guarantees, but are computationally intensive and time-consuming. In contrast, approximation-based methods, such as reachability-based approaches, offer faster computation by employing approximations to simplify the calculation. Nonetheless, due to over-approximation, the results obtained may not be exact. In other words, if non-robustness is detected using such methods, it could potentially be attributed to coarse over-approximation during the analysis process. Striking a balance between computational efficiency and precision level in formal verification methods remains an ongoing challenge in ensuring reliable and accurate assessments of neural networks' robustness. In spite of these limitations, methods that assess the robustness of neural networks can be leveraged in several ways to conceive clever defense attacks that enhance the models' immunity against adversarial attacks.

Regarding the local evaluation of robustness against adversarial attacks, there are numerous types of adversarial attacks. In this paper, we have attempted to enumerate the main attacks. However, for users with an AI model who wish to evaluate the model, they should choose attacks that are most suitable for different data types including images, tabular data, time series, etc. Additionally, users should decide whether they want to test white-box or black-box attacks. This can be selected based on the amount of model information available to the user, e.g., whether the model is their own model with access to all its parameters, or a model given by another party in a black-box version.

Regarding defense techniques, the user can choose the most suitable technique for their objective to make their model robust to random noises, Gaussian noises, adversarial attacks, and so on. As users, their task does not end there and after applying the defense technique, they should check if their model is indeed robust. This is done first by evaluating the model against adversarial attacks, and if this test passes, confirmation of robustness can be ensured through a formal evaluation.

9. Conclusions and Perspectives

The objective of this paper is to investigate the safety of neural networks subject to input perturbations in different environments, particularly uncertain environments. The primary focus is on addressing the challenge of utilizing neural networks in safety-critical systems by establishing formal guarantees.

Despite exhibiting strong performance on a wide range of input samples, neural networks have often faced with difficulties in accurately generalizing to unfamiliar scenarios, thereby remaining vulnerable to adversarial attacks. To address this limitation, this paper has reviewed some of the most popular methods that provide a formal verification approach capable of evaluating the properties of DNNs throughout the entire input space. These methods have been categorized based on their formulation of three fundamental analysis techniques: reachability, optimization, and search. By employing these techniques, researchers can gain a deeper understanding of the behaviours and limitations of neural networks, enabling them to identify vulnerability and bolster network robustness in critical applications. Furthermore, in order to mitigate the vulnerability of neural networks to adversarial attacks, defense techniques can be applied to make the network more robust during the classification of perturbed data.

The findings of this study contribute to the ongoing efforts in developing reliable and secure neural networks, thereby paving the way for safer and more trustworthy AI systems in the future.

Author Contributions: Mohamed Ibn Khedher: Investigation, Methodology, Writing-original draft, Writing—review & editing; **Houda Jmila:** Investigation, Writing-original draft, Writing—review & editing; **Mounim A. El Yacoubi:** Elaboration of the structure of the paper; Writing some parts; Reading and correction of the whole paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khedher, M.I.; Jmila, H.; El Yacoubi, M.A. Fusion of interest point/image based descriptors for efficient person re-identification. In *Proceedings of 2018 International Joint Conference on Neural Networks, Rio de Janeiro, Brazil, 8–13 July 2018*; IEEE: New York,

2018; pp. 1–7. doi:10.1109/IJCNN.2018.8489111
2. Qin, H.F.; El-Yacoubi, M.A. Finger-vein quality assessment based on deep features from grayscale and binary images. Int. J. Patt. Recogn. Artif. Intell., **2019**, *33*: 1940022. doi: 10.1142/s0218001419400226
3. Yu, N.X.; Yang, R.; Huang, M.J. Deep common spatial pattern based motor imagery classification with improved objective function. Int. J. Netw. Dyn. Intell., **2022**, *1*: 73−84. doi: 10.53941/ijndi0101007
4. Li, X.; Li, M.L.; Yan, P.F.; *et al*. Deep learning attention mechanism in medical image analysis: Basics and beyonds. Int. J. Netw. Dyn. Intell., **2023**, *2*: 93−116. doi: 10.53941/ijndi0201006
5. Dao, Q.; El-Yacoubi, M.A.; Rigaud, A.S. Detection of Alzheimer disease on online handwriting using 1D convolutional neural network. IEEE Access, **2023**, *11*: 2148−2155. doi: 10.1109/access.2022.3232396
6. Jmila, H.; Khedher, M.I.; Blanc, G.; *et al*. Siamese network based feature learning for improved intrusion detection. In *Proceedings of the 26th International Conference on Neural Information Processing, Sydney, NSW, Australia, 12–15 December 2019*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 377–389. doi:10.1007/978-3-030-36708-4_31
7. Khedher, M.I.; Mziou, M.S.; Hadji, M. Improving decision-making-process for robot navigation under uncertainty. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence, SciTePress, 4–6 February 2021*; SciTePress, 2021; pp. 1105–1113. doi:10.5220/0010323311051113
8. Bunel, R.; Turkaslan, I.; Torr, P.H.S.; *et al*. A unified view of piecewise linear neural network verification. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, Canada, 3–8 December 2018*; Curran Associates Inc.: Red Hook, 2018; pp. 4795–4804.
9. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In *Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015*; 2015.
10. Papernot, N.; McDaniel, P.; Jha, S.; *et al*. The limitations of deep learning in adversarial settings. In *Proceedings of 2016 IEEE European Symposium on Security and Privacy, Saarbruecken, Germany, 21–24 March 2016*; IEEE: New York, 2016; pp. 372–387.
11. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, 2016; pp. 2574–2582.
12. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial machine learning at scale. In *Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017*; OpenReview.net, 2017.
13. Madry, A.; Makelov, A.; Schmidt, L.; *et al*. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018*; OpenReview.net, 2018.
14. Bhambri, S.; Muku, S.; Tulasi, A.; *et al*. A survey of black-box adversarial attacks on computer vision models. arXiv:1912.01667, 2020.
15. Chen, P.Y.; Zhang, H.; Sharma, Y.; *et al*. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas Texas USA, 3 November 2017*; ACM: New York, 2017; pp. 15–26. doi:10.1145/3128572.3140448
16. Brendel, W.; Rauber, J.; Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018*; OpenReview.net, 2018.
17. Chen, J.B.; Jordan, M.I. Boundary attack++: Query-efficient decision-based adversarial attack. arXiv: 1904.02144, 2019.
18. Jmila, H.; Khedher, M.I. Adversarial machine learning for network intrusion detection: A comparative study. Comput. Netw., **2022**, *214*: 109073. doi: 10.1016/j.comnet.2022.109073
19. Aung, A.M.; Fadila, Y.; Gondokaryono, R.; *et al*. Building robust deep neural networks for road sign detection. arXiv: 1712.09327, 2017.
20. Xiang, W.M.; Tran, H.D.; Johnson, T.T. Reachable set computation and safety verification for neural networks with ReLU activations. arXiv: 1712.08163, 2017.
21. Xiang, W.M.; Tran, H.D.; Johnson, T.T. Output reachable set estimation and verification for multilayer neural networks. IEEE Trans. Neural Netw. Learn. Syst., **2017**, *29*: 5777−5783. doi: 10.1109/TNNLS.2018.2808470
22. Gehr, T.; Mirman, M.; Drachsler-Cohen, D.; *et al*. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *Proceedings of 2018 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–24 May 2018*; IEEE: New York, 2018; pp. 3–18. doi:10.1109/SP.2018.00058
23. El Mellouki, O.; Khedher, M.I.; El-Yacoubi, M.A. Abstract layer for leakyReLU for neural network verification based on abstract interpretation. IEEE Access, **2023**, *11*: 33401−33413. doi: 10.1109/ACCESS.2023.3263145
24. Lomuscio, A.; Maganti, L. An approach to reachability analysis for feed-forward ReLU neural networks. arXiv: 1706.07351, 2017.
25. Tjeng, V.; Xiao, K.Y.; Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. In *Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019*; OpenReview.net, 2019.
26. Ibn-Khedher, H.; Khedher, M.I.; Hadji, M. Mathematical programming approach for adversarial attack modelling. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence, SciTePress, 4–6 February 2021*; SciTePress, 2021; pp. 343–350. doi:10.5220/0010324203430350
27. Bastani, O.; Ioannou, Y.; Lampropoulos, L.; *et al*. Measuring neural net robustness with constraints. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016*; Curran Associates Inc.: Red Hook, 2016; pp. 2621–2629.
28. Dvijotham, K.; Stanforth, R.; Gowal, S.; *et al*. A dual approach to scalable verification of deep networks. arXiv: 1803.06567, 2018.
29. Wong, E.; Kolter, J.Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018*; PMLR, 2018; pp. 5283–5292.
30. Raghunathan, A.; Steinhardt, J.; Liang, P. Certified defenses against adversarial examples. In *Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018*; OpenReview.net, 2018.
31. Wang, S.Q.; Pei, K.X.; Whitehouse, J.; *et al*. Formal security analysis of neural networks using symbolic intervals. In *Proceedings of the 27th USENIX Conference on Security Symposium, Baltimore, MD, USA, 15–17 August 2018*; USENIX Association: Berkeley, 2018; pp. 1599–1614.
32. Bunel, R.; Turkaslan, I.; Torr, P.H.S.; *et al*. Piecewise linear neural network verification: A comparative study. arXiv: 1711.00455, 2017.

33. Katz, G.; Barrett, C.W.; Dill, D.L.; *et al*. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the International Conference on Computer Aided Verification, Heidelberg, Germany, 24–28 July 2017*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 97–117. doi:10.1007/978-3-319-63387-9_5
34. Zantedeschi, V.; Nicolae, M.I.; Rawat, A. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, Texas, USA, 3 November 2017*; ACM: New York, 2017; pp. 39–49. doi:10.1145/3128572.3140449
35. Nguyen, L.; Wang, S.; Sinha, A. A learning and masking approach to secure learning. In *Proceedings of the 9th International Conference on Decision and Game Theory for Security, Seattle, WA, USA, 29–31 October 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 453–464. doi:10.1007/978-3-030-01554-1_26
36. Lee, H.; Han, S.; Lee, J. Generative adversarial trainer: Defense to adversarial perturbations with GAN. arXiv: 1705.03387, 2023.

Citation: Khedher, M.; Jmila, H.; El-Yacoubi, M. On the Formal Evaluation of the Robustness of Neural Networks and Its Pivotal Relevance for AI-Based Safety-Critical Domains. *International Journal of Network Dynamics and Intelligence*. 2023, 2(4), 100018. doi: 10.53941/ijndi.2023.100018