On the Verification of ML Systems and Models

Greta Dolcetti[1], Vincenzo Arceri[2], Agostino Cortesi[1] and Enea Zaffanella[2]

[1]*Ca' Foscari University of Venice, Italy*
[2]*University of Parma, Italy*

Abstract

The role and impact of machine learning systems and models are growing in every economic and social sector. The problem of guaranteeing the reliability and correctness of the underlying software therefore becomes increasingly relevant. In this article we identify the elements that characterize these systems and that have a challenging impact on the application of state-of-the-art verification techniques and we highlight the advantages and limitations of a set of formal techniques that can be combined to achieve this goal. In principle, we advocate not only for a deeper adoption of formal methods in the machine learning development and deployment, but also for a more systematic and holistic approach.

Keywords

Machine Learning, Verification, Formal Methods

1. Introduction

The field of computer science has undergone a significant paradigm shift due to progress in Artificial Intelligence (AI) and Machine Learning (ML). This evolution has led to the adoption of ML systems in complex tasks such as natural language processing and image recognition. Their capacity to perform well in tasks previously deemed impractical to solve has led to their adoption in various contexts, including autonomous driving and healthcare, where failures could result in serious damage.

The adoption of these methods carries a risk associated with their safety, particularly in safety-critical domains. To address this, various verification frameworks have been proposed to provide a systematic approach to verifying the correctness and reliability of ML models, ensuring they are safe and effective in practical applications. The verification of these systems would not only increase the confidence and trustworthiness regarding their adoption but would also lower the risk of failures. However, the current state-of-the-art for ML systems verification poses many challenges, which will be discussed in the following sections of this paper.

This paper aims to highlight the challenges of verifying ML systems and models, covering a wide range of challenges and providing research directions for both the ML and formal methods communities. We advocate for a broader and more *holistic* application of verification techniques to offer formal guarantees about properties related to various aspects of the models and systems, especially those directly regarding human activities, decisions, and sensible information. We believe that a greater effort is required from both academia and industry to achieve higher trustworthiness and reliability of adopted models.

2. Background

In traditional software design, testing is a big component of the development pipeline and, although necessary, it is not sufficient to *guarantee* the correctness of the product. Similarly, in ML, metrics like accuracy, precision, and recall provide empirical measures of performance but do not offer formal

7th International Workshop on Artificial Intelligence and fOrmal VERification, Logic, Automata, and sYnthesis (OVERLAY 2025)

greta.dolcetti@unive.it (G. Dolcetti); vincenzo.arceri@unipr.it (V. Arceri); cortesi@unive.it (A. Cortesi); enea.zaffanella@unipr.it (E. Zaffanella)
0000-0002-2983-9251 (G. Dolcetti); 0000-0002-5150-0393 (V. Arceri); 0000-0002-0946-5440 (A. Cortesi); 0000-0001-6388-2053 (E. Zaffanella)

guarantees or prove safety. Therefore, especially in safety-critical contexts, it is necessary to formally verify that ML models comply with properties of interest, such as safety [1] (*i.e.*, ensuring that predictions do not violate specifications), robustness [2] (*i.e.*, maintaining performance and correctness despite variations, perturbations, or adversarial inputs) and fairness [3] (*i.e.*, ensuring that predictions do not depend on features that are considered sensible, with obvious ethical implications).

2.1. Verification

In machine learning verification systems, the property to verify is usually composed of a set of preconditions and a set of postconditions, that namely represent a relation that should hold between the input (preconditions) and the output (postconditions) of the model. Collectively, this set of preconditions and postconditions is referred to as *specifications*. These properties usually transcend the architecture of the model, which is later taken into consideration during the verification process, and simply describe some characteristics that the model should (not) have.

Some properties that are subject to verification are

- **Safety.** Given a trained model, the resulting predictions are safe if they do not violate any previously given specifications, often regarding safety-critical decisions. A case for the verification of this property is reported in [1], on a Deep Neural Network (DNN) trained for the Airborne Collision Avoidance Systems (ACAS) [4] task, for which a verification algorithm is adopted to prove some safety specifications regarding the model predictions.
- **Robustness.** Robustness refers to the network's ability to maintain its performance and correctness in the face of variations, perturbations, or adversarial inputs. A robust neural network should produce reliable and consistent predictions even when exposed to unexpected or challenging conditions. This property is crucial for ensuring the trustworthiness and resilience of neural networks in real-world applications and, in some scenarios, offers also interpretability perspectives [5]. Informally, robustness guarantees that in classification tasks, for small perturbations in the neighborhood of a given input, the output of the network should not change, resulting in a different class. Many different approaches, like [6] and [7], have been proposed, making robustness one of the most pervasive property in the verification area.
- **Fairness.** The notions of fairness can be different. In *group* fairness, a model is fair if different values of a given feature, considered sensible, do not result in different predictions, meaning that some sort of statistical parity exists for members of protected groups (*i.e.*, the demographics of the set of individuals receiving any classification are the same as the demographics of the underlying population) [8]. Other types of fairness like *individual* fairness [8] can guarantee that if two inputs (individuals) are similar, then the two outputs of the model must be similar too. Both of these concepts are worth investigating, either because they may seem in conflict and present trade-offs between the two of them, but also because they can be analyzed from different perspectives, such as politics, justice, and philosophy, as shown in [9], making fairness not only an interesting property but also an ethical hot-spot.

2.1.1. Methodologies

The strategies for machine learning verification can be categorized into two main types: complete and incomplete.

Complete. Complete methods leverage techniques such as Satisfiability Modulo Theory (SMT) [1, 10, 11] and Mixed Integer Linear Programming (MILP) [12, 13]. In the former, the verification is defined as a Constraint Satisfaction Problem (CSP) in which both the network and the specification are expressed as a set of constraints. The verification step is then applied to the network and the *negation* of the specification, so that if the CSP is satisfiable, then the property does not hold and the solution is a counter-example that violates the property itself; otherwise, if the CSP is unsatisfiable, then there is no possible solution that violates the property and therefore, the property holds. In the latter, the verification is expressed as a MILP problem usually solved via Branch-and-Bound strategies (B-a-B) [7], to prove the validity/absence of a given property, providing conclusive evidence when the property holds and, in case of non-compliance, often providing a counter-example. While these methods excel in delivering precise results, their drawback lies in the lack of scalability, particularly when dealing with large systems. Nonetheless, they offer the advantage of avoiding false negatives.

Incomplete. Incomplete methods adopt techniques such as Linear Programming (LP) [14], Semi-Definite Programming (SDP) [15], or Abstract Interpretation (AbsInt) [16, 2, 17] to assess the validity of desired properties within a given model. These approaches usually adopt abstractions that scale well in large models and are sound (meaning that if the property is said to hold, it actually holds), but they can raise false alarms, although they never result in missed violations.

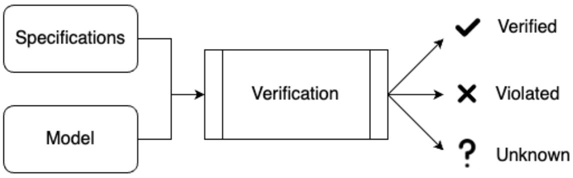

Figure 1: Diagram of the verification process of a machine learning model.

Concisely, as shown in Figure 1 a verification algorithm aims to provide a *guarantee* that a given property holds or not, also allowing for the case where, with the adopted technique, the result is unknown, for example because a timeout is reached. Usually, given a property to verify, the underlying method for the verification task is to check if the violation of the property is possible considering the specifications and the model, thus resulting in a negative approach w.r.t. the positive property expressed in the specifications. Practically speaking, the property is verified if its violation via a counterexample is infeasible; on the contrary, if a counterexample, eventually further analyzed using more refined models, allowing the violation of the original property is achievable, then the property does not hold. For example, if we concretize this abstract concept into the verification of the robustness property, the model is robust if there is no concrete input in the considered neighborhood that could result in a different output; on the other hand, the model is not robust if there exists at least an adversarial example for which the model output differs from the original one considered.

To complicate an already complex topic, the introduction of Foundation Models has further revolutionized the AI field thanks to their adaptability, enabling the use of pre-trained models that perform well across a wide range of tasks, which can then be fine-tuned for specific issues without requiring developing and training a model from scratch for each different task. Continuous improvements have led to larger, more capable, and general models. However, their size and complexity make them increasingly difficult to comprehend and explain without external techniques and tools.

3. Challenges

The field of ML verification is filled with difficult challenges to overcome, both internal and external to the verification task itself. These challenges will be discussed in the next paragraphs, highlighting how they affect the ML community and where the research on this topic, in our opinion, should focus.

Bias in the Data. ML models learn from data, making dataset creation and validation crucial. Unfortunately, few verification frameworks can prove interesting properties about the data, which can exhibit unwanted properties too. Data bias is a simple yet powerful example: biased datasets most likely train biased models. Furthermore, proving fairness and re-training the model requires more computational effort than training on an unbiased dataset. The problem lies in formally expressing properties of interest to prove on the datasets; moreover, datasets can be obtained from biased sources, such as sensors or manual annotation, which can be subject to errors and inaccuracies too.

Verified Models Are Not Real Models. The major issue with verifying ML models is that the majority of research is conducted on simple architectures and tasks, which inadequately reflect the complexity of real-world models. For example, benchmarks used in the Verification of Neural Networks Competition (VNN-COMP) [18] target models with a few hundred to a few million parameters, whereas newly developed Large Language Models (LLMs) can have hundreds of billions of parameters. Research often focuses on feed-forward networks with piecewise-linear activation functions, disregarding RNNs, LSTMs, and transformer-based models, for which few publications exist. There is, indeed, a necessity to extend the verification to more realistic systems and models, although this will probably result in an exponential increment to the computational cost of verification.

When analyzing the time necessary to complete verification tasks on ML models it is evident that these approaches struggle to scale because simple analyses can range from a few seconds to days to be completed. However, this should not discourage the adoption of verification steps. Although computationally expensive, the verification process is a one-time step that does not need to be repeated unless the model changes. To mitigate this cost, approaches using parallelization and GPU computing have been adopted [19]. Additionally, incremental [20] or continuous verification could be beneficial for models and systems that are regularly updated. This would allow the MLOps Lifecycle to be continuously integrated, deployed, and verified.

Supervised vs. Unsupervised Models. Most formal verification is applied to supervised or semi-supervised models. For unsupervised models, such as k-means [21], few formal verification approaches exist [22] due to the difficulty in expressing specifications and properties of interest. The main issue is that unsupervised models often lack a ground truth, making performance evaluation challenging. For example, clustering task validation is either internal (linked to cluster shape/distance) or external (linked to existing labels or domain knowledge), relying on techniques like clustering stability [23] or density-based validation [24]. Despite this, unsupervised models are widely used for clustering and association tasks, such as market and customer segmentation. Therefore, as stated in [25], "it is crucial to develop approaches towards fair unsupervised learning". This is closely linked to the issue of biased data discussed in Section 3, especially due to the absence of labels that can lead to undetected unfairness. In this scenario, formal methods could be used for cluster distance guarantees or proving robustness under a given perturbation.

Privacy. One of the major concerns surrounding ML and neural networks is the issue of privacy. Since these systems collect and process vast amounts of data, there is a risk that they may inadvertently compromise sensitive information or incentivize biases and discrimination. For example, a neural network trained on a dataset containing personal information may infer sensitive details about individuals, even if the data is anonymized [26]. The use of ML algorithms to make decisions about individuals raises questions about accountability and transparency. To address these concerns, it is essential to develop methods for protecting privacy and ensuring ML systems are transparent and accountable [27]. This may involve implementing robust data protection policies, developing techniques for anonymous data collection and processing, and establishing standards for the ethical use of ML and neural networks. On the other hand, techniques like differential privacy [28] and secure multi-party computation[29] can provide privacy guarantees, but they often introduce a privacy-utility tradeoff. In this context, the European Union's Artificial Intelligence Act[1] aims to protect citizen privacy by limiting some AI applications and enforcing obligations for high-risk systems. Similar approaches coming from regulators should be encouraged and incentivized.

[1] https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI(2021)698792

The Issue with Generative Models and LLMs. Generative models have been widely adopted and studied in recent times thanks to their ability to generate content: these models are extremely useful and powerful, allowing to perform tasks once considered infeasible, such as text-to-image or text-to-video. Since the performed activities differ greatly from ML classical tasks, like classification and regression, it is more difficult to identify, define and formalize the properties to verify. Robustness properties can be reinterpreted so that the aim is to obtain a similar output (intended as a generated content and not a label) for a given input neighborhood and some approaches have already been implemented [30]. However, new properties could emerge that focus on new issues: for example, verification of copyright infringement could be implemented in order to ensure that the model has not been trained on copyright-protected data. Furthermore, a plagiarism verification could be adopted to certify the novelty or, at least, the non-plagiarism nature of the generated content. It is also important to note the emerging trend of *jailbreaking* LLMs, where adversarial prompts are used to bypass safety mechanisms and force the model to generate restricted or harmful content. Introducing safeguards or defence mechanisms against jailbreaking, would allow for safer interactions.

LLMs have gained attention for their ability to mimic language and reasoning, offering glimpses of artificial general intelligence. However, their often closed-source nature and vast number of parameters make them challenging to verify. While some tests have been conducted to uncover vulnerabilities, these approaches typically involve empirical experiments on large amounts of data and queries, rather than relying on provably safe approaches based on formal methods. The Open Worldwide Application Security Project (OWASP) has identified top 10 vulnerabilities for LLMs,[2] highlighting the need for robust verification techniques. The inherent complexity of LLMs, coupled with the lack of transparency, emphasizes the need for further research. One major weakness affecting LLMs is *hallucinations*, where outputs may be incorrect, fake, or inappropriate. Researchers have attempted to mitigate this issue through fine-tuning, knowledge graphs, memory augmentations, and formal methods [31]. However, these approaches are not automated or widely applied in real-world applications. Breaking down verification, as in the Chain of Thought procedure [32], could bring benefits to the formal verification step; similarly, if a model can be decomposed into independent sub-models, the whole verification could be parallelized and sped up [3].

Verifying Models Not Systems. Few verification approach focus on the entire ML pipeline, particularly on the data processing step, which, indeed, is a crucial part of ML development and it should be treated as such. Verification or validation systems capable of extracting properties of the training data should be developed and applied as a first step of the ML pipeline. By gaining insights into the data, the model itself could be more reliable, and developers could address issues like data scarcity, imbalance, or bias, potentially using data augmentation or cleaning techniques. Cyber-physical systems, often used in safety-critical applications, consist of various components, including ML models, which can pose security issues [33]. Therefore, the system should be considered as a *whole*, verifying not only individual components but also their interactions. However, this poses a challenge, as each component may require a different kind of verification with a different framework and specifications. Integrated verification could result in an expensive and complex task. For example, the introduction of the Model Context Protocol (MCP)[3], a protocol that allows AI applications to interact with external resources, dynamic environments, and tools, demands careful consideration: not only because it consists of very different components, but also because it introduces new potential attack points within the interactions among these components.

Nevertheless, a system can only be declared safe if all its components and interactions are safe. It is essential to highlight that a system is as strong as its weakest component, and no aspect should be neglected.

[2] https://llmtop10.com

[3] https://modelcontextprotocol.io/

Regulations - Call for a Standard. Establishing an analogy with classic safety-critical software can be inspiring when analyzing how this field has evolved to provide further safety guarantees to both developers and users. Over the years, many standards and guidelines have been published, determining requirements that the products have to meet to be adopted[4]. Similarly, regulatory authorities and consumer organizations should require guarantees from entities that develop and produce ML systems to release them for commercial use. If the guarantees become mandatory, ML systems would result in safer, more secure, and reliable applications. Additionally, guidelines for research projects in ML, such as open-source model and dataset adoption, would allow for higher experimental reproducibility, which is now not always possible.

Verifying During Training vs. After Training. Most research on verifying neural networks (NNs) focuses on already trained models. The sequential execution and repetition of the training and verification steps can be computationally very expensive, requiring a lot of time. To address this, many approaches combine the two into guided training, such as adversarial training, which aims to formally minimize the worst-case loss for every possible input. Robust training relies on input perturbations, improving model robustness through data augmentation. These approaches primarily target formal robustness and often require white-box access to the models. Fairness, a data-related property, is also targeted during training, although often empirically and not formally. However, as highlighted in [34], training fair neural networks poses technical challenges, including the risk of overfitting and false senses of fairness during training. Nevertheless, [35] identified a trade-off between robustness and accuracy, while [36] showed how "existing certified training approaches improve worst-case certified accuracy at the cost of drastically reduced standard accuracy on clean inputs", limiting their adoption in practical applications.

ML for ML Verification. In the previous sections, we've seen formal methods and empirical experiments. However, it's profitable to adapt ML models to verify other ML models. Some approaches leveraging this idea exist: for example, [37] refines and verifies global robustness using generative models, and [38] uses generative models to discover adversarial examples. Similarly, [39] provides guarantees over the observation space approximated by a generative model by training a Generative Adversarial Network (GAN) to map states to plausible input images. While ML models are not yet widely used in verification tasks due to their black-box behavior, integrating formal methods with ML could be a promising perspective. ML models could generate candidate invariants, properties to verify, and counterexamples, which could then be fed into formal methods tools to formally check that they satisfy required properties. These candidate elements generated by ML components could be used to formally verify that they satisfy the required properties, making the verification process more automated and efficient.

Teaching. Due to AI's wide range of applicability, which led to a broader range of research perspectives and many cross-field advancements, nowadays it is common to find ML-related courses not only in computer science degrees. We are confident that formal methods can play a crucial role in teaching, infusing a verified-by-design culture at an early stage of learning, and leading to a higher security and safety awareness of these technologies. We think that this is even more important for training ML specialists in critical fields, such as healthcare. Furthermore, we believe that enriching these courses with formal methods, and the guarantees provided, would present a significant advantage for both educators and students, for example making the behavior of models and systems more explainable and easy to understand [40, 41]. In this sense, it is worth highlighting that, given the heterogeneous audience to whom these courses are offered, a higher degree of explainability and guarantees could provide a better comprehension of errors and bugs, which are usually hard to spot and debug in ML systems, especially if used as off-the-shelf tools.

[4] For example ISO 25000 and ISO 62304, which are issued by the International Organization for Standardization.

Emerging Issues - A Fast-Evolving World. An additional challenge in the verification of ML systems arises from the rapid evolution of this field, marked by the constant introduction of new applications and improved techniques. To match this pace, a joint community effort is essential. A notable example is the initiative by the LVE Project (https://lve-project.org/), where an open-source repository is maintained to track vulnerabilities concerning privacy, reliability, security and trust. This collaborative endeavor aims to enhance safety and awareness within the community by providing a comprehensive resource for identifying and addressing potential bugs in LLMs.

4. Constraints, Limitations and Assumptions

The quest for guaranteed ML models and systems comes with inevitable constraints, limitations, and assumptions. As stated in [42], good ML software should be robust and provide testing routines to verify code correctness. Our claim goes beyond testing routines, aiming for formal verification to obtain provable guarantees. However, accessing source code for verification is a major concern, as many recent models are closed-source. The verification step is also limited by a trade-off between accuracy and computational complexity. A careful integration into software production processes is required, considering both energy consumption and computational complexity.

5. Conclusion

The verification of ML models and systems should be a primary goal to reach in modern society. Succeeding in this task would give us guarantees that only formal verification can provide, resulting in the adoption of these models with higher reliability and trustworthiness, even in safety-critical applications, and lowering the risks of vulnerabilities, attacks, and malfunctions. However, it is not sufficient to tackle each one of the challenges presented in the previous sections singularly, thus the extent of the issue calls for an *holistic* approach. Indeed, looking at an isolated aspect could lead to systems that are safe and reliable under that aspect, but that can cause major damages in all the others, for example, a model could be robust but heavily biased. As part of this holistic approach, the safety of a model should be considered not only by technical metrics and issues but also in the light of ethical, jurisdictional, and legal features. In conclusion, providing users with clear and provable guarantees about all the aspects of the product offered should become the standard.

Acknowledgments

This work has been supported by SERICS (PE00000014 - CUP H73C2200089001) and iNEST (ECS00000043 – CUP H43C22000540006) projects funded by PNRR NextGeneration EU.

References

[1] G. Katz, C. W. Barrett, D. L. Dill, K. Julian, M. J. Kochenderfer, Reluplex: An efficient SMT solver for verifying deep neural networks, in: R. Majumdar, V. Kuncak (Eds.), Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I, volume 10426 of *LNCS*, Springer, 2017, pp. 97–117. doi:10.1007/978-3-319-63387-9_5.

[2] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, M. T. Vechev, AI2: safety and robustness certification of neural networks with abstract interpretation, in: 2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA, IEEE Computer Society, 2018, pp. 3–18. doi:10.1109/SP.2018.00058.

[3] C. Urban, M. Christakis, V. Wüstholz, F. Zhang, Perfectly parallel fairness certification of neural networks, Proc. ACM Program. Lang. 4 (2020) 185:1–185:30. doi:10.1145/3428253.

[4] K. D. Julian, J. Lopez, J. S. Brush, M. P. Owen, M. J. Kochenderfer, Policy compression for aircraft collision avoidance systems, in: 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), IEEE, 2016, pp. 1–10. doi:10.1109/DASC.2016.7778091.

[5] D. Banerjee, A. Singh, G. Singh, Interpreting robustness proofs of deep neural networks, CoRR abs/2301.13845 (2023). doi:10.48550/ARXIV.2301.13845.

[6] H. Salman, G. Yang, H. Zhang, C. Hsieh, P. Zhang, A convex relaxation barrier to tight robustness verification of neural networks, in: H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 9832–9842. URL: https://proceedings.neurips.cc/paper/2019/hash/246a3c5544feb054f3ea718f61adfa16-Abstract.html.

[7] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C. Hsieh, J. Z. Kolter, Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification, in: M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 2021, pp. 29909–29921. URL: https://proceedings.neurips.cc/paper/2021/hash/fac7fead96dafceaf80c1daffeae82a4-Abstract.html.

[8] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. S. Zemel, Fairness through awareness, in: S. Goldwasser (Ed.), Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012, ACM, 2012, pp. 214–226. doi:10.1145/2090236.2090255.

[9] R. Binns, On the apparent conflict between individual and group fairness, in: M. Hildebrandt, C. Castillo, L. E. Celis, S. Ruggieri, L. Taylor, G. Zanfir-Fortuna (Eds.), FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, ACM, 2020, pp. 514–524. doi:10.1145/3351095.3372864.

[10] G. Amir, H. Wu, C. W. Barrett, G. Katz, An smt-based approach for verifying binarized neural networks, in: J. F. Groote, K. G. Larsen (Eds.), Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings, Part II, volume 12652 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 203–222. doi:10.1007/978-3-030-72013-1_11.

[11] L. Pulina, A. Tacchella, An abstraction-refinement approach to verification of artificial neural networks, in: T. Touili, B. Cook, P. B. Jackson (Eds.), Computer Aided Verification, 22nd Int. Conf., CAV 2010, Edinburgh, UK, July 15-19, 2010. Proc., volume 6174 of *LNCS*, Springer, 2010, pp. 243–257. doi:10.1007/978-3-642-14295-6_24.

[12] M. Fischetti, J. Jo, Deep neural networks and mixed integer linear optimization, Constraints An Int. J. 23 (2018) 296–309. doi:10.1007/S10601-018-9285-6.

[13] V. Tjeng, K. Y. Xiao, R. Tedrake, Evaluating robustness of neural networks with mixed integer programming, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019. URL: https://openreview.net/forum?id=HyGIdiRqtm.

[14] W. Lin, Z. Yang, X. Chen, Q. Zhao, X. Li, Z. Liu, J. He, Robustness verification of classification deep neural networks via linear programming, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 11418–11427. doi:10.1109/CVPR.2019.01168.

[15] M. Newton, A. Papachristodoulou, Exploiting sparsity for neural network verification, in: A. Jadbabaie, J. Lygeros, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, M. N. Zeilinger (Eds.), Proceedings of the 3rd Annual Conference on Learning for Dynamics and Control, L4DC 2021, 7-8 June 2021, Virtual Event, Switzerland, volume 144 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 715–727. URL: http://proceedings.mlr.press/v144/newton21a.html.

[16] P. Cousot, R. Cousot, Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints, in: R. M. Graham, M. A. Harrison, R. Sethi (Eds.), Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los

Angeles, California, USA, January 1977, ACM, 1977, pp. 238–252. doi:10.1145/512950.512973.

[17] G. Singh, T. Gehr, M. Püschel, M. T. Vechev, An abstract domain for certifying neural networks, Proc. ACM Program. Lang. 3 (2019) 41:1–41:30. doi:10.1145/3290354.

[18] C. Brix, M. N. Müller, S. Bak, T. T. Johnson, C. Liu, First three years of the international verification of neural networks competition (VNN-COMP), Int. J. Softw. Tools Technol. Transf. 25 (2023) 329–339. doi:10.1007/S10009-023-00703-4.

[19] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, C. Hsieh, Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021. URL: https://openreview.net/forum?id=nVZtXBI6LNn.

[20] S. Ugare, D. Banerjee, S. Misailovic, G. Singh, Incremental verification of neural networks, Proc. ACM Program. Lang. 7 (2023) 1920–1945. doi:10.1145/3591299.

[21] S. P. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (1982) 129–136. doi:10.1109/TIT.1982.1056489.

[22] A. Maurer, D. A. Parletta, A. Paudice, M. Pontil, Robust unsupervised learning via l-statistic minimization, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 7524–7533. URL: http://proceedings.mlr.press/v139/maurer21a.html.

[23] U. von Luxburg, Clustering stability: An overview, Found. Trends Mach. Learn. 2 (2009) 235–274. URL: https://doi.org/10.1561/2200000008. doi:10.1561/2200000008.

[24] D. Moulavi, P. A. Jaskowiak, R. J. G. B. Campello, A. Zimek, J. Sander, Density-based clustering validation, in: M. J. Zaki, Z. Obradovic, P. Tan, A. Banerjee, C. Kamath, S. Parthasarathy (Eds.), Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014, SIAM, 2014, pp. 839–847. URL: https://doi.org/10.1137/1.9781611973440.96. doi:10.1137/1.9781611973440.96.

[25] F. Buet-Golfouse, I. Utyagulov, Towards fair unsupervised learning, in: FAccT ’22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022, ACM, 2022, pp. 1399–1409. doi:10.1145/3531146.3533197.

[26] M. Fredrikson, E. Lantz, S. Jha, S. M. Lin, D. Page, T. Ristenpart, Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing, in: K. Fu, J. Jung (Eds.), Proc. of the 23rd USENIX Security Symp., San Diego, CA, USA, August 20-22, 2014, USENIX Assoc., 2014, pp. 17–32. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/fredrikson_matthew.

[27] N. Balasubramaniam, M. Kauppinen, A. Rannisto, K. Hiekkanen, S. Kujala, Transparency and explainability of AI systems: From ethical guidelines to requirements, Inf. Softw. Technol. 159 (2023) 107197. doi:10.1016/J.INFSOF.2023.107197.

[28] C. Dwork, Differential privacy, in: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II, volume 4052 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 1–12. URL: https://doi.org/10.1007/11787006_1. doi:10.1007/11787006_1.

[29] O. Goldreich, Secure multi-party computation, Manuscript. Preliminary version 78 (1998) 1–108.

[30] M. Mirman, A. Hägele, P. Bielik, T. Gehr, M. T. Vechev, Robustness certification with generative models, in: S. N. Freund, E. Yahav (Eds.), PLDI ’21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, June 20-25, 2021, ACM, 2021, pp. 1141–1154. doi:10.1145/3453483.3454100.

[31] S. Jha, S. K. Jha, P. Lincoln, N. D. Bastian, A. Velasquez, S. Neema, Dehallucinating large language models using formal methods guided iterative prompting, in: IEEE Int. Conf. on Assured Autonomy, ICAA, Laurel, MD, USA, June 6-8, 2023, IEEE, 2023, pp. 149–152. doi:10.1109/ICAA58325.2023.00029.

[32] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: S. Koyejo, S. Mohamed,

A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Inf. Proc. Sys. 35 (NeurIPS), New Orleans, LA, USA, Nov. 28 - Dec. 9, 2022, 2022. URL: http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

[33] R. M. Alguliyev, Y. N. Imamverdiyev, L. V. Sukhostat, Cyber-physical systems and their security issues, Comput. Ind. 100 (2018) 212–223. doi:10.1016/J.COMPIND.2018.04.017.

[34] V. Cherepanova, V. Nanda, M. Goldblum, J. P. Dickerson, T. Goldstein, Technical challenges for training fair neural networks, CoRR abs/2102.06764 (2021). URL: https://arxiv.org/abs/2102.06764. arXiv:2102.06764.

[35] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, M. I. Jordan, Theoretically principled trade-off between robustness and accuracy, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 7472–7482. URL: http://proceedings.mlr.press/v97/zhang19p.html.

[36] Z. Nurlanov, F. R. Schmidt, F. Bernard, Adaptive certified training: Towards better accuracy-robustness tradeoffs, in: A. Bifet, P. Daniusis, J. Davis, T. Krilavicius, M. Kull, E. Ntoutsi, K. Puolamäki, I. Zliobaite (Eds.), Machine Learning and Knowledge Discovery in Databases. Research Track and Demo Track - European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part VIII, volume 14948 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 126–143. URL: https://doi.org/10.1007/978-3-031-70371-3_8. doi:10.1007/978-3-031-70371-3_8.

[37] N. Fijalkow, M. K. Gupta, Verification of neural networks: Specifying global robustness using generative models, CoRR abs/1910.05018 (2019). URL: http://arxiv.org/abs/1910.05018. arXiv:1910.05018.

[38] C. Xiao, B. Li, J. Zhu, W. He, M. Liu, D. Song, Generating adversarial examples with adversarial networks, in: J. Lang (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 3905–3911. doi:10.24963/IJCAI.2018/543.

[39] S. M. Katz, A. L. Corso, C. A. Strong, M. J. Kochenderfer, Verification of image-based neural network controllers using generative models, CoRR abs/2105.07091 (2021). URL: https://arxiv.org/abs/2105.07091. arXiv:2105.07091.

[40] K. Bjørner, S. Judson, F. C. Córdoba, D. Goldman, N. Shoemaker, R. Piskac, B. Könighofer, Formal XAI via syntax-guided synthesis, in: B. Steffen (Ed.), Bridging the Gap Between AI and Reality - First International Conference, AISoLA 2023, Crete, Greece, October 23-28, 2023, Proceedings, volume 14380 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 119–137. doi:10.1007/978-3-031-46002-9_7.

[41] S. Bassan, G. Katz, Towards formal XAI: formally approximate minimal explanations of neural networks, in: S. Sankaranarayanan, N. Sharygina (Eds.), Tools and Algorithms for the Construction and Analysis of Systems - 29th International Conference, TACAS 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Paris, France, April 22-27, 2023, Proceedings, Part I, volume 13993 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 187–207. doi:10.1007/978-3-031-30823-9_10.

[42] S. Sonnenburg, M. L. Braun, C. S. Ong, S. Bengio, L. Bottou, G. Holmes, Y. LeCun, K. Müller, F. Pereira, C. E. Rasmussen, G. Rätsch, B. Schölkopf, A. J. Smola, P. Vincent, J. Weston, R. C. Williamson, The need for open source software in machine learning, J. Mach. Learn. Res. 8 (2007) 2443–2466. doi:10.5555/1314498.1314577.