Bootstrapping Trust in ML4Nets Solutions with Hybrid Explainability

Abduarraheem Elfandi, Hannah Sagalyn, Ramakrishan Durairajan
University of Oregon

Walter Willinger
NIKSUN, Inc.

Abstract

The future success of ML4Nets—defined as the application of machine learning (ML) techniques to address real-world network security and performance problems—relies critically on convincing network operators to deploy ML-based solutions in their production networks. However, the black-box nature of many of these solutions has been a major impediment to both gaining the operators' trust in the underlying trained models and providing effective safety guarantees. Explainable AI (XAI) represents a recent approach to dealing with this problem and provides operators with global and local explainability techniques that enable them to reason about the decisions made by trained ML models. Unfortunately, in their current form, these solutions are lacking in simultaneously engendering the kind of trust and ensuring the type of safety guarantees that operators require for using ML-based solutions in practice.

This work proposes a novel hybrid explainability technique that combines global and local explainability methods to address network operators' dual requirements for proposed ML-based solutions. In particular, the proposed hybrid technique leverages global explainability methods to make a majority of a black model's decisions or predictions understandable and transparent and relies on local explainability methods to deal with the remaining "corner cases". While the practical application of this hybrid technique requires dealing with a delicate efficiency-accuracy tradeoff (i.e., weighing network operators' desire for trusting proposed ML-based solutions against their need to inspect the solutions' safety), its theoretical implication suggests examining an intriguing analog of the well-known CAP theorem for distributed systems. We present an illustrative use case to demonstrate the feasibility and potential of the proposed hyrid technique and sketch a proposed version of a CAP theorem analog for explainability of ML4Nets solutions.

1 Introduction and Overview

Ensuring the success of ML4Nets in practice hinges on convincing network operators to deploy ML4Nets solutions in their production networks. This, in turn, requires that network operators can trust these solutions and have means to assess their safety. Here, following [15], we say "a network operator has trust in a ML model" iff "the operator is comfortable with relinquishing control to the model." Moreover, referring to [6], by "assessing the safety of ML solutions", we mean "studying the problem of accidents, defined as unintended and harmful behavior that may emerge from poor design of real-world ML solutions." Unfortunately, due to the black-box nature of many of the currently considered ML models, today's network operators lack the means to reason about the decisions and predictions made by these models. These models' inability to provide explanations for their decision-making engenders distrust, prevents network operators from understanding the models' safety, and explains the operators' overall reluctance to using ML4Nets solutions in practice [13].

To address these issues, Explainable AI (XAI) has emerged as a field of study aimed at enhancing the comprehensibility of learning models and their decision-making processes (e.g., see the surveys [4, 7, 9]). At a high level, XAI encompasses two categories of techniques. The first category consists of *global explainability* techniques that leverage approximations in the form of explainable models to provide an overall explanation of a given black-box model and typically entail a tradeoff between the complexity (e.g., size) of the explainable model, its accuracy (e.g., number of input instances it explains), and the computational effort its generation requires. In theory, using such explainable approximation models, operators can reason about a given black-box model's decision-making (i.e., how and why the model arrives at a specific decision and not at some other decision) and gain confidence in the overall reliability of its decisions and predictions (i.e., when the model works or when and why the model does not work).

The second category is composed of *local explainability* techniques that are typically designed to provide explanations for individual instances that a trained model is given as input. Local techniques employ concepts such as feature importance scores, attention mechanisms, and rule-based explanations, and applying them at scale (number of instances) requires being aware of their per-instance computational complexity. These local techniques are useful vehicles for operators to reason about a given model's specific decisions or predictions and to assess the safety of a given ML4Nets solution in corner case scenarios (i.e., understanding the potential consequences of certain incorrect decisions for a given input instance).

While these techniques have been successfully applied in a number of different application domains (e.g., computer

vision [10] and autonomous vehicles [1, 3]), their suitability and effectiveness in the networking domain to address network performance and security problems of practical interest have attracted little to no attention to date, mainly because of data-related issues that are specific to networking [11, 13, 14]. Most critical among these issues are a general paucity of (labeled) data, the high volume and velocity of network data collected from real-world production networks, the one-off nature of existing data collection efforts, and important privacy and security concerns associated with collecting network data from operational networks.

Furthermore, faced with a growing number available explainability techniques, network researchers and operators alike are largely left in the dark about how to apply the latest techniques so as to simultaneously satisfy the dual requirements of network operators—gaining trust in ML4Nets solutions (by means of having a broad understanding of the solution's global behavior) and being able to assess the solutions' safety (by means of providing specific, case-by-case, local explanations that can be scrutinized with respect to the impact of the associated decisions and predictions). Finally, meeting both of these requirements concurrently also necessitates systems innovations that are aimed at striking a balance between the indiscriminate use of resource-intensive local explainability techniques and the selective application of efficient but inaccurate approximation models supplied by global explainability techniques.

In this work, we propose and present in Section 3 an initial evaluation of a novel hybrid explainability tool aimed at gaining network operators' trust in ML4Nets solutions. This tool entails three key steps. Step 1 involves improving the accuracy of explainable decision trees resulting from post-hoc applications of global explainability techniques. This step primarily involves integrating accurate local explainability methods opportunistically. We also use a simple majority voting mechanism to correct potential errors in the original decision tree, thus enhancing the trees accuracy and avoiding excessive computational overhead. Step 2 addresses situations where the global techniques fail to explain input instancess. For these "corner cases", we apply a majority voting mechanism with local explainability techniques to identify new branches of the decision tree generated in Step 1, thus enhancing that tree's utility. In Step 3, the new branches corresponding to these corner cases are integrated into the a final decision tree. This final step ensures compatibility with already existing nodes and branches, maintains traversal order along individual branches, and harmonizes interpretations to enhance trust in the model, assess its safety, and demonstrate its computational efficiency.

In addition to highlighting in Section 4 new research and systems-related challenges, this work is also a reminder that in application domains such as networking where ML-based solutions are touted for high-stakes decision making (i.e., dropping critical traffic if it is deemed malicious), the use of "responsible AI" to engender trust in the developed ML-models and assess their safety is still in its infancy. At the same time, there is, however, a great urgency for rapid advances in this area, especially in view of the current general excitement about large language models (LLMs) where, depending on the application domain, the need for trust and safety is magnified by the potential harm that erroneous or wrong decisions or predictions made by these latest generation of black-box models could cause [5, 20].

2 Hybrid Explainability

In this work, we present a novel hybrid explainability technique that is specifically designed to simultaneously satisfy network operators' dual requirements for deploying ML4Nets solutions—being able to trust these solutions and to assess their safety. At the core of this novel technique is the following three-step approach:

Step 1: Enhancing Explainable Model Accuracy. This step focuses on addressing accuracy issues that stem from using post-hoc global explainability techniques. These techniques include recently developed methods such as Trustee [12] or ARISE [13] and typically generate explainable models in the form of decision trees that approximate a given black-box models. Because of their approximate nature, the generation of such decision trees entails a complexity-accuracy tradeoff whereby high-accuracy decision trees necessitate large-sized (i.e., high-complexity) tree structures. To navigate this tradeoff, we incorporate computationally intensive yet highly accurate local explainability techniques in an opportunistic manner. Specifically, we enhance the explainability of certain branches of the approximate decision-tree model only if it results in improved accuracy. We determine this improvement through a straightforward majority voting mechanism that is aimed at resolving uncertainties or inconsistencies stemming from the fusion of global and local explainability techniques. This step essentially acts as a "model distillation" process, simplifying complex interpretations generated by various techniques and consolidating them into a more concise explanation. Moreover, the employed voting mechanism prevents unnecessary computational overhead when multiple techniques are in agreement.

Step 2: Handling Exceptional Cases. In this step, we address situations where global techniques fail to provide an explanations or don't produce a meaningful explanation. Such situations are common for training data collected from operational networks [13, 14], but network operators nevertheless want to be able to reason about a black-box models' decision-making process when faced with such "corner cases." For each data point in the training data that cannot be explained by any of the branches of the decision tree generated in Step 1, we apply the same majority voting mechanism used earlier, but this time exclusively with findings derived from applying local explainability techniques [16, 17]. This approach saves computational resources by applying local

techniques to a subset of the training data (i.e., corner cases). It also avoids unnecessary resource overhead when there is consensus among the employed local techniques. At the end of this step, we compile a comprehensive list of corner cases, complete with rule-based explanations, potentially adding new branches to the decision tree generated in Step 1.

Step 3: Expanding the Global Decision Tree. The final step involves integrating the new branches that emerge from the corner cases considered in Step 2 into the global decision tree constructed in Step 1. The goal is to expand this tree while ensuring that both the existing and new branches are integrated cohesively. In effect, this step serves as an "explanation summarization" process, where interpretations are harmonized to facilitate engendering trust, assessing safety, and guaranteeing computational efficiency. To determine the proper placement of each of the corner cases into the already existing decision tree, we start by examining the nodes of the decision tree where the conditions of the new branches (i.e., corner cases) align with the those in the existing decision tree. We then check if any existing nodes can accommodate the new branches with some modifications. If so, we update the nodes of the decision tree to incorporate the rules from the corner cases. Otherwise, we create new nodes in the decision tree. These new nodes are added as child nodes to existing parent nodes in the decision tree, ensuring that the conditions of the new nodes are compatible with the rules of their parent nodes and that they lead to the intended outcomes. After applying this process to each considered corner case, we review the entire new set of branches to maintain the order of traversal. In operational networks, these trees have to be updated on an ongoing basis by means of real-world feedback and new training data to enhance their accuracy.

3 Preliminary Results

To illustrate the practicality of our proposed hybrid explainability technique, we evaluate in the following the effectiveness of the above-described Step 1 in the context of ARISE, a previously-published weak-supervision-based framework for labeling different network datasets in an automated fashion and at scale [13]. In short, ARISE leverages network operators' domain knowledge in the form of *labeling functions* to programmatically label network datasets and uses multi-task learning to enable concurrent learning of network classification tasks (e.g., congestion vs. non-congestion). Its workflow (see Appendix C in [13] for details) requires to first create a noisy generative model and then train a predictive LSTM model. We choose ARISE because of its ability to produce a decision tree that enables operators to reason about the labeling decisions made by ARISE. Here we show how to embellish this decision tree by applying Step 1 of our proposed hybrid explainability technique.

Preliminary results. For our evaluation, we use CAIDA's Ark dataset, which comprises over 1.2 million round-trip time (RTT) measurements between 28 source-destination pairs collected over the course of a day [2]. Using this dataset, we trained a predictive LSTM model by utilizing the labeling function that classifies a data point as "experiencing congestion" if the RTT value falls within the range of [1.2 times β, 1.5 times α], where α and β represent the RTT values corresponding to the 75th and 25th percentiles, respectively. Our data partitioning scheme allocates 80% of the data for training, 10% for validation, and 10% for testing for each link. Additionally, we randomly selected 1,000 measurements from a single source-destination pair and manually labeled them with many false negatives to create a dataset for evaluating the decision tree generated by ARISE.

Figure 1 shows three key outcomes, along with the percentage of times the data points were labeled as congested ("VOTE") or not ("NORMAL"). On the left, we show the LSTM model created by ARISE. In the center, we shown the decision tree generated through ARISE's task-specific explainability capability. On the right, we show the explainable model with our integrated majority voting mechanism. Table 1 complements Figure 1 and lists the model evaluation metrics. In particular, the LSTM model achieves a good balance between precision (0.816) and recall (0.964), with an F1 score of 0.884 and an accuracy of 0.881. This indicates that the LSTM model performs well in labeling data points as congested or not while minimizing false positives. The task-specific explainable model achieves perfect precision (1.000) but has a lower recall (0.645), resulting in an F1 score of 0.784 and an accuracy of 0.833. However, combining this explainable model with our majority voting mechanisms yields predictions with improved recall (0.998) and adequate precision (0.826), resulting in both a high F1 score of 0.904 and high accuracy of 0.900. These preliminary findings indicate that including a simple majority voting mechanism can produce a more balanced and accurate classification, addressing the accuracy issues of post-hoc global explainability techniques. Furthermore, given that the task-specific decision tree model's accuracy of 83% and our sample size of 1,000 for model evaluation, the voting scheme effectively diminishes the number of "corner cases" from 17% (i.e., 170 corner cases) to 10% (i.e., 100 corner cases). Applying steps 2-3 results in a perfect precision (1.000) and a high recall (0.989), resulting in an improved F1 score of 0.994 and a high accuracy of 0.995. Considering the 10% corner cases after the application of majority voting, applying the rest of the steps further diminishes the number of corner cases from 100 to 4.

4 Future Work

In the short term, we plan to extend the described new tool as follows. In the fusion process (Step 1), we plan to optimize the resource allocation further by selecting the most relevant features from both global and local techniques, ensuring that only essential information is used. For Step 2, akin to the feedback loop in ARISE [13], we plan to use feedback from

	Precision	Recall	Accuracy	F1 score
LSTM model	0.816216	0.963830	0.881000	0.883902
Task-specific explainable model	1.000000	0.644681	0.833000	0.783959
After majority voting	0.825704	0.997872	0.900000	0.903661
Other steps	1.000000	0.989362	0.995000	0.994652

Table 1. Evaluation metrics for the three models depicted in Figure 1.

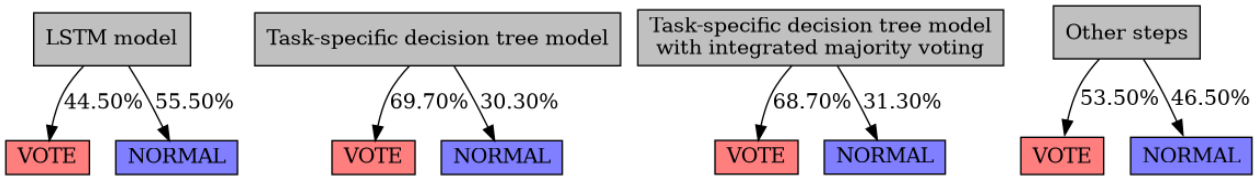

Figure 1. Initial LSTM model for congestion detection on a link (first), decision-tree-based explainable model derived using ARISE [13] (second), and the explainable model corrected using majority voting (third) and other steps (fourth).

operators to fine-tune the majority voting process, adjusting the weight of individual interpretations based on their accuracy and relevance. As part of the feedback loop, we intend to provide operators with the ability to choose the level of explainability they desire (also known as, selective explainability), allowing them to allocate resources according to the specific context and importance of the decision. In addition, we intend to explore the scalability of hybrid explainability tool in terms of handling large-scale network datasets and complex ML models.

A CAP Theorem Analog for Explainability of ML4Nets solutions. In the long term, similar to the CAP theorem in distributed systems [19], which notes that only two out of three characteristics can be achieved in distributed systems, our work suggests an intriguing analog in the context of explainability in ML4Nets. In particular, consider the following three dimensions: (1) **C**omplexity of explainable approximation model (e.g., size of generated decision tree given by the number of nodes); this metric reflects the operators' perspective and measures how big a tree operators are comfortable scrutinizing to satisfy their need for model explainability. (2) **A**ccuracy of explainable approximation model (e.g., fidelity of constructed decision tree); this metric measures how faithful the decision tree is in terms of explaining how the original black-box model makes its decisions and determines the pool of "corner cases" that requires further investigations to ensure the model's safety. (3) **C**omputation efficiency of explainable approximation model; this metric quantifies how many resources to expend on achieving the operators' required level of model explainability and accounts for the use of global techniques (i.e., generation of the initial approximation model in the form of a decision tree) and the use of local techniques (i.e., for improving the accuracy of the tree and augmenting via examinations of the identified corner cases).

For example, when an operator seeks to improve accuracy, it necessitates increased computational resources, which, in turn, inevitably leads to a higher level of complexity (i.e., less manageable for operators as humans are inherently limited in effectively processing large-sized trees). Conversely, reducing computational overhead results in an improved operator experience that is based on having to inspect only smaller-sized trees (i.e., low complexity), but this benefit comes at the cost of reduced accuracy. In fact, leveraging examples considered in [12], violations of the suggested CAP theorem analog for explainability of ML4Nets solutions can be readily associated with certain underspecification issues in currently used ML pipelines [8]. For example, shortcut learning is a case where a given model satisfies all three characteristics (i.e., low complexity, high accuracy, high computational efficiency). In contrast, models that are vulnerable to out-of-distribution input samples satisfy none of the three characteristics. In our future work, we plan to more formally and empirically investigate this analog by carefully considering several such illustrative examples.

Post-hoc vs. Ante-hoc Explainability Methods. While [18] argues against trying to create a second (post-hoc) model to explain an originally-trained black-box model and instead advocates using inherently interpretable models such as decision trees in the first place (i.e., ante-hoc or ex-ante), [21] takes a more nuanced view with respect to post-hoc explainability methods and sees potential benefits of their use in ML4Nets. However, in both cases, an open problem of significant importance is understanding the extent to which the almost exclusive current focus on decisison trees limits the explorations of ante-hoc or post-hoc explainability efforts.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback. This work is supported by the National Science Foundation through CNS-2145813, OAC-2126281, CICI-2319944, and SaTC-2132651. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF.

References

[1] ARGOVERSE. https://www.argoverse.org/.

[2] CAIDA Ark Datasets. www.caida.org/projects/ark/topo_datasets.xml.

[3] nuScenes. https://www.nuscenes.org/.

[4] A. Adadi and M. Berrada. Peeking Inside the Black-box: A Survey on eXplainable Artificial Intelligence (XAI). *IEEE access*, 6:52138–52160, 2018.

[5] C. Adam and R. Carter. Large Language Models and Intelligence Analysis. *CETaS Expert Analysis*, 2023.

[6] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete Problems in AI Safety. *arXiv preprint arXiv:1606.06565*, 2016.

[7] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information fusion*, 58:82–115, 2020.

[8] A. D'Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman, et al. Underspecification Presents Challenges for Credibility in Modern Machine Learning. *The Journal of Machine Learning Research*, 23(1):10237–10297, 2022.

[9] A. Das and P. Rad. Opportunities and Challenges in eXplainable Artificial Intelligence (XAI): A Survey. *arXiv preprint arXiv:2006.11371*, 2020.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] A. Gupta, C. Mac-Stoker, and W. Willinger. An Effort to Democratize Networking Research in the Era of AI/ML. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, pages 93–100, 2019.

[12] A. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville. AI/ML for Network Security: The Emperor has No Clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1537–1551, 2022.

[13] J. Knofczynski, R. Durairajan, and W. Willinger. ARISE: A Multitask Weak Supervision Framework for Network Measurements. *IEEE Journal on Selected Areas in Communications*, 40(8):2456–2473, 2022.

[14] Y. Lavinia, R. Durairajan, R. Rejaie, and W. Willinger. Challenges in Using ML for Networking Research: How to Label If You Must. In *Proceedings of ACM SIGCOMM Workshop on Network Meets AI ML*, 2020.

[15] Z. C. Lipton. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is both Important and Slippery. *Queue*, 16(3):31–57, 2018.

[16] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[17] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[18] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[19] S. Simon. Brewer's CAP theorem. *CS341 Distributed Information Systems, University of Basel (HS2012)*, 2000.

[20] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting. Large Language models in Medicine. *Nature medicine*, pages 1–11, 2023.

[21] W. Willinger, A. Gupta, A. S. Jacobs, R. Beltiukov, R. A. Ferreira, and L. Granville. A netai manifesto (part i): Less explorimentation, more science. *ACM SIGMETRICS Performance Evaluation Review*, 51(2):106–108, 2023.