Proceedings of the DataNinja sAIOnARA 2024 Conference 72-74 DOI: 10.11576/dataninja-1176

Interpretable Machine Learning via Linear Temporal Logic

Simon Lutz SIMON.LUTZ@TU-DORTMUND.DE
Research Center "Trustworthy Data Science and Security", University Alliance Ruhr, TU Dortmund University, Germany

Daniel Neider DANIEL.NEIDER@TU-DORTMUND.DE
Professor for Verification and Formal Guarantees of Machine Learning, Research Center "Trustworthy Data Science and Security", University Alliance Ruhr, TU Dortmund University, Germany

Abstract

In recent years, deep neural networks have shown excellent performance, outperforming even human experts in various tasks. However, their inherent complexity and black-box nature often make it hard, if not impossible, to understand the decisions made by these models, hindering their practical application in high-stakes scenarios. We propose a framework for learning LTL formulas as inherently interpretable machine learning models. These models can be trained both in a supervised and unsupervised setting. Furthermore, they can easily be extended to handle noisy data and to incorporate expert knowledge.

Keywords: Explainable AI, Learning of logic formulas, Linear Temporal Logic

In the last years, Artificial Intelligence (AI) has received tremendous attention and is nowadays used in a wide variety of application domains including medicine, law enforcement, autonomous systems, and natural language processing, to name but a few. In most cases, these AI systems are based on deep neural networks with hundreds of layers and billions of parameters. Trained on large amounts of training data, these models have shown excellent performance, outperforming even human experts in various tasks. However, their inherent complexity and black-box nature often make it hard, if not impossible, to understand the decisions made by a neural network. This is especially problematic in high-stakes application and often a severe obstacle to employing AI systems in practice. Consider, for instance, a medical system assisting doctors with diagnosing patients. If the system diagnoses a specific disease, it is imperative to understand the reason to ensure correct treatment is prescribed.

To overcome this drawback of intransparent decision-making, the field of explainable artificial intelligence (XAI) has evolved in recent years. Instead of just computing the decision of a neural network, XAI methods also provide a human-readable explanation of how the network concluded this decision (see [1] for a more detailed introduction). Broadly speaking, these methods can be separated into post hoc explanations and inherently interpretable models. Post hoc explanations do not interfere with the architecture or training of a neural network but aim at inferring an explanation by analyzing its decision-making post hoc. State-of-the-art methods include using game-theoretic Shapley values as a measure for feature importance (SHAP [2]), the use of surrogate models for local explanations (LIME [3]), and the visualization of feature importance using heat maps (Grad-CAM [4]), to name but a few. Instead of training a complex neural network, the second paradigm opts for training simpler models such as decision trees, decision rules, or linear regression. Even though these models are inherently interpretable, they may lack the ability to generalize well from the training data leading to worse overall performance. Nevertheless, multiple papers have recently introduced deterministic finite automata (DFAs) as capable (i.e., on par with state-of-the-art LSTM models) yet interpretable models for sequence classification [5, 6] and anomaly detection [7].

In this paper, we follow the second paradigm and introduce a framework for learning formulas in Linear Temporal Logic (LTL) [8] as interpretable machine learning models for time series data. This specific choice of model is motivated by the following observations: first, a description of the observed, temporal behavior can often be captured in a concise logical formula; second, there is a straightforward way to translate an LTL formula to natural language, making it easy for experts and lay people to comprehend; and third, many engineers are familiar with Linear

Temporal Logic, being the de facto standard for specifying temporal properties. We consider two different learning setups, one supervised and one unsupervised learning scenario. Furthermore, we will discuss possible extensions to handle noisy data and to insert domain and expert knowledge.

In the first, supervised setup, we are given a finite set $\mathcal{S}$ of sequences together with their corresponding class labels. Then the task is to learn an LTL-based classifier that predicts the class label of yet unseen data. Toward this goal, we construct a set of minimal LTL formulas, each characterizing one of the possible output classes (thus functioning as a one-vs-rest classifier). Here, minimality refers to a minimal number of subformulas which we use to ensure high interpretability in the sense of Occam's razor (i.e., smaller formulas are generally easier to understand than larger ones [6, 9]). For each class, we construct the corresponding formula by splitting the set $\mathcal{S}$ in a one-vs-rest manner. Then, we use the algorithm proposed by Neider and Gavran [10] to infer an LTL formula from this data. In order to classify a sequence, we query each LTL formula in our set, giving us a distribution over the decisions of the one-vs-rest classifiers. Then we adopt the approximate Bayesian method of Shvo et al. [6] to infer a posterior probability distribution over the true class label and conclude a prediction.

For the second setup, we consider the task of anomaly detection, i.e. identifying patterns in data that do not conform to expected behavior [11]. As anomaly detection often plays a major role in safety-critical applications such as medical diagnosis or autonomous control, collecting anomalous data can often be dangerous and labeled data is scarce. Therefore, anomaly detection methods are often trained in an unsupervised setting where the labels of the data are a priori not known but assumed to be normal. The objective of the anomaly detection method is then to learn the underlying concept of normality in the data. Whenever unseen data diverges from this concept of normality, it will be considered an anomaly. In addition to the data, these anomaly detection methods usually also require further auxiliary information or fine-tuned hyper-parameters to prevent them from producing a degenerate solution (i.e., one that classifies all or no data as anomalies). When adopting the above concept of anomaly detection in our second, unsupervised learning scenario we rely on the approach proposed by Roy et al. [12]. Given a set $\mathcal{S}$ of unlabeled sequences and a size bound n, their algorithm produces an LTL formula of size n which is language minimal with respect to $\mathcal{S}$, i.e., it accepts all sequences in $\mathcal{S}$ and no formula of the same size accepts fewer sequences (outside of $\mathcal{S}$). Here, the size n functions as an additional parameter regulating the trade-off between interpretability and capability to generalize. We can use an LTL formula learned by this algorithm to capture normality in the given data and thus to detect anomalies.

In both setups, the proposed algorithms incorporate the learning task as a constraint-solving problem. This allows them to utilize the advances and years of engineering work of modern SAT-solver. Furthermore, it provides a rich framework for further optimization and extension of the learning algorithm.

Similar to neural networks, both learning frameworks are susceptible to noisy data. While in the second setup, this may cause drastically worse performance, it can even render the learning task unsolvable in the first setup (if two inputs are the same but have different labels). One way of mitigating the effect of noise could be the incorporation of the ideas of Gaglione et al. [13] who propose a framework for learning LTL formulas from noisy data.

A second benefit of the learning framework is that it allows for easy incorporation of domain or expert knowledge. Lutz et al. [14] proposed a framework where expert knowledge in the form of a so-called sketch (i.e., a partial LTL formula) can be provided to the learning process. The learning algorithm then completes the sketch based on the given data. Combining this framework with the proposed learning frameworks allows utilizing expert knowledge to improve the quality of the learned LTL formulas and also speed up the learning process.

In conclusion, we introduced a framework for learning LTL formulas as inherently interpretable machine learning models both in a supervised and unsupervised setting. Furthermore, we presented two extensions of the learning framework, allowing the mitigation of noisy data and the incorporation of expert knowledge.

Acknowledgments

This work has been financially supported by Deutsche Forschungsgemeinschaft, DFG Project number 459419731, and the Research Center Trustworthy Data Science and Security (https://rc-trust.ai), one of the Research Alliance centers within the UA Ruhr (https://uaruhr.de).

References

[1] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[2] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[3] Marco Tulio Ribeiro, Sameer Singh, and Carlos 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.

[4] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[5] Christian Albert Hammerschmidt, Sicco Verwer, Qin Lin, and Radu State. Interpreting finite automata for sequential data. *arXiv preprint arXiv:1611.07100*, 2016.

[6] Maayan Shvo, Andrew C Li, Rodrigo Toro Icarte, and Sheila A McIlraith. Interpretable sequence classification via discrete optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9647–9656, 2021.

[7] Simon Lutz, Florian Wittbold, Simon Dierl, Benedikt Böing, Falk Howar, Barbara König, Emmanuel Müller, and Daniel Neider. Interpretable anomaly detection via discrete optimization. *arXiv preprint arXiv:2303.14111*, 2023.

[8] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 46–57. ieee, 1977.

[9] Rajarshi Roy, Dana Fisman, and Daniel Neider. Learning interpretable models in the property specification language. *arXiv preprint arXiv:2002.03668*, 2020.

[10] Daniel Neider and Ivan Gavran. Learning linear temporal properties. In *2018 Formal Methods in Computer Aided Design (FMCAD)*, pages 1–10. IEEE, 2018.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[12] Rajarshi Roy, Jean-Raphaël Gaglione, Nasim Baharisangari, Daniel Neider, Zhe Xu, and Ufuk Topcu. Learning interpretable temporal properties from positive examples only. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6507–6515, 2023.

[13] Jean-Raphaël Gaglione, Daniel Neider, Rajarshi Roy, Ufuk Topcu, and Zhe Xu. Learning linear temporal properties from noisy data: A maxsat-based approach. In *Automated Technology for Verification and Analysis: 19th International Symposium, ATVA 2021, Gold Coast, QLD, Australia, October 18–22, 2021, Proceedings 19*, pages 74–90. Springer, 2021.

[14] Simon Lutz, Daniel Neider, and Rajarshi Roy. Specification sketching for linear temporal logic. In *International Symposium on Automated Technology for Verification and Analysis*, pages 26–48. Springer, 2023.