Towards Tractable Formal Explanations for Neural Network Models

Anonymous submission

Abstract

Neural networks are fragile, it is widely reported that trivial disturbances on the inputs can make a neural network model flip its decision, and as a black box model, it cannot provide any explanations. Many of the current neural network explanation methodologies are heuristic, cannot guarantee their accuracy. Thus, formal explanation methodologies for neural network models are imperative, for machine learning models to be trustworthy and widely applied in real-life applications. Prime implicant (PI) explanations (also known as abductive explanations) are able to provide logical sufficient explanations for a neural network that are guaranteed to be correct. However, formally extracting PI explanations from a neural network model is NP-hard. In this paper, by converting the neural network model as a game, we propose an algorithm to extract formal explanations out of the neural network model with linear complexity. We apply the algorithms on real-life zero-day intrusion detection use case, and demonstrate the formal explanations extracted. Valuable insights and conclusions are discussed.

Introduction

With the pervasion of Artificial Intelligence, more and more research work rely on machine learning, especially neural network models for decision making. However, neural network are known to be fragile and untrustworthy. It has been widely reported that trivial disturbances on the input can make the neural network model flips its decision, deteriorating the performance significantly and abruptly, without any explanations (LeCun, Bengio, and Hinton. 2015; Gunning and Aha 2019; Shi et al. 2020; Stewart 2020; Mnih et al. 2015; Domingos 2015; Zhao and Hastie 2021; Mathur 2015; Banks, Plant, and Stanton 2018; Deeks 2019; Srinivasan and Chander 2020; Kuppa and Le-Khac 2021; Muna, Maliha, and Hasan 2021). Neural network models are opaque blackboxes, they cannot explain the decisions they made (Zhao and Hastie 2021; Mathur 2015; Banks, Plant, and Stanton 2018; Deeks 2019; Srinivasan and Chander 2020; Kuppa and Le-Khac 2021; Muna, Maliha, and Hasan 2021; Lipton 2018; Arrieta et al. 2020). In domains where security is of utmost importance, such as cybersecurity, trust is the fundamental basis and guarantee of the validity and prosperity of AI-based decision making. People's trust on the decisions made is based on the interpretability and transparency of the machine learning models make them (Gunning and Aha 2019). Consequences of the decision made by uninterpretable machine learning models are occasionally catastrophic, for example the fatal car crushes by Google's autonomous car (Mathur 2015) and Tesla's autopilot system (Banks, Plant, and Stanton 2018); an automatic bail risk assessment algorithm is believed to be biased and keep many people in jail longer than they should without explicit reasons, and another machine learning based DNA trace analysis software accuses people with crimes they did not commit [1]; Millions of African-American could not get due medical care by a biased machine learning assessment algorithm [2]; In Scotland, a football game is ruined because the AI camera mistakes the judge's bald head as the ball and keep focusing on it even during the goal scene[3].

That gives rise to the surge of research interest in Explainable Artificial Intelligence (XAI) or Interpretable Machine Learning (IML), numerous machine learning explanation methodologies have been proposed (Lundberg et al. 2020; Ignatiev 2020; Darwiche 2023; Shi et al. 2020; Ignatiev, Narodytska, and Marques-Silva 2019b; Johansson, König, and Niklasson 2003; Van den Broeck et al. 2021; Ribeiro, Singh, and Guestrin 2016; Guo et al. 2018; Lundberg and Lee 2017; Ribeiro, Singh, and Guestrin 2018; Johansson, König, and Niklasson 2004; Grover et al. 2019; Kim, Khanna, and Koyejo 2016), most of them are heuristic, such as SHapley Additive exPlanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME), and ANCHOR, providing approximated explanations by simulating the behavior of the model with a simpler and more explainable model, such as decision trees, but as discussed in previous work, although many believe tree-based machine learning models are explainable, the explanations they provide are "shallow", containing lots of redundancies of rules and features. Explanations extracted with formal methods, such as ***Prime Implicant (PI) explanations***, on the contrary, are logical sufficient and complete, and are guaranteed to be correct (Ignatiev, Narodytska, and Marques-Silva 2019a;

[1] See https://www.nytimes.com/2017/06/13/opinion/how-computers-are-harming-criminal-justice.html

[2] See https://www.wsj.com/articles/researchers-find-racial-bias-in-hospital-algorithm-11571941096

[3] see https://www.ndtv.com/offbeat/ai-camera-ruins-football-game-by-mistaking-referees-bald-head-for-ball-2319171

Ignatiev 2020; Darwiche 2023; Shi et al. 2020; Chan and Darwiche 2003). Heuristic explanation methodologies are not extracted with formal methods, and thus cannot guarantee its accuracy, the explanations may change at different run times, even for the same instance. However, extract formal explanations from neural network models are NP-hard, because converting each individual neuron into a boolean expression is an NP-hard problem; representing the boolean expressions into more tractable representations, such as OBDD, MDD or SDD is an NP-hard problem; extracting prime implicants from the boolean expression is also at least an NP-hard problem (Quine 1952). Decades of effort has been devoted to relax these problems into tractable ones (Ignatiev, Narodytska, and Marques-Silva 2019a; Ignatiev 2020; Darwiche 2023; Shi et al. 2020; Chan and Darwiche 2003), it has been recently reported that compiling a neuron into OBDD can be accomplished in polynomial time (Chan and Darwiche 2003; Shi et al. 2020) and prime implicants can now be extracted in linear time (Cooper and Marques-Silva 2021).

By representing a neuron as a threshold based linear classifier, and represent the decision making behavior of a neuron as a Game, this paper proposed a methodology to extract formal prime implicant explanations as game changing strategies in O(1) time complexity. In this paper, we formulate an Zero-Day intrusion detection neural network model as a linear classifier, and then represent the linear classifier into a multi-valued decision tree with multi-valued logic. By treating the multi-valued linear classifier as a game, we can decompose the prime implicates extracting problem into a game strategy making problem, which can be solved in O(1) time for each instance given. The prime implicants can be represented as the thresholds of game changing points, thus are also the explanations for the decision strategies. User cases studies based on real life Zero-Day intrusion detection demonstrate that for a specific instance, the prime implicant explanations can be calculated in linear time complexity.

From Neural Network to Game

To extract minimum and formal explanations from the model, we need to compile the model into a logical expression with formal method – knowledge compilation. The final model is a MLP neural network with fully connected three layers, include one input layer with 14 neurons, one hidden layer $y^{(1)}$ with 100 neurons and an activation function σ, and one output layer $y^{(2)}$ with one neuron and an activation function δ, as shown in Fig. 2a. The output calculation process of matrix multiplication of weight matrixes and bias matrixes are shown in Fig. 2a. For simplicity, the hidden layer activation function σ is the *identity function:* $y = x$, the output layer activation function δ is *logistic function*. Although the function expression is different from *identity function*, the decision logic of *logistic function* is the same with $y = x$, *"y is positively correlated with x, when* $x \geqslant 0$*, y is classified as 'Evil', otherwise, y is classified as 'Benign' "*. Thus, we calculate the expression of the MLP neural network as Fig. 2b. The "if-clause" in yellow background is the output of the entire neural network, also shown in Fig. 2c. This is a threshold based linear classifier (Borowski and Choi 2023; Shi et al. 2020), formally defined below:

Definition 1. *Let $f(x)$ be a function has input $\mathbb{X}$, where $\mathbb{X}$ is a set of n variables $X_1, X_2, \ldots, X_n$ (n is a positive integer), and a **theshold** T. Each variable X_i has a real valued weight ω_i. The function is a **Threshold-Based Linear Classifier** if it has the following format (Shi et al. 2020):*

$$f(x) = \begin{cases} 1, \ if \quad \sum_{i=0}^{n} \omega_i \cdot X_i \geqslant T \\ 0, \ otherwise \end{cases} \tag{1}$$

It can be considered as a mapping between input variables $\mathbb{X}$ and the boolean value of output $f(x)$. Depend on the input $\mathbb{X}$ values, it can be further classified as discrete linear classifier and positive discrete linear classifier, formally defined below.

Definition 2. *Let $f(x)$ be a threshold based linear classifier, if all of its inputs X_i are positive integer step values, starting from $1, 2, 3, \ldots$ and are finite, it is a **Threshold-Based Discrete Linear Classifier**.*

It is easy to see that threshold-based discrete linear classifier is a special case of threshold-based discrete linear classifier. Depending on the weights, input variables $\mathbb{X}$ can be further classified into two categories: positive inputs $\mathbb{X}_+$ and negative inputs $\mathbb{X}_-$, formally defined below.

Definition 3. *In a threshold-based positive discrete linear classifier $f(x)$, if a weight of an input $\omega_i \geqslant 0$, then it is a **Positive Weight** ω_+, otherwise, it is a **Negative Weight** ω_-.*

*In a threshold-based positive discrete linear classifier $f(x)$, if a input X_i has a **Positive Weight**, then it is a **Positive Input** x_+, if it has a **Negative Weight**, then it is a **Negative Input** x_-, the set of all the positive input of $f(x)$ is named **Positive Inputs** $\mathbb{X}_+$, the set of all negative input of $f(x)$ is defined as **Negative Inputs** $\mathbb{X}_-$.*

The *Positive Inputs* and *Negative Inputs* of the model in our use case are presented in Fig. 2d and 2e, and are supported by Fig. 2f and 2g. In Fig. 2d, the *Positive Inputs*, which are $\mathbb{B}$, $\mathbb{C}$, $\mathbb{E}$, $\mathbb{G}$, $\mathbb{H}$, $\mathbb{I}$, $\mathbb{J}$, $\mathbb{N}$ in our model, are labelled in color red, *Negative Inputs*, $\mathbb{A}$, $\mathbb{D}$, $\mathbb{F}$, $\mathbb{K}$, $\mathbb{L}$, $\mathbb{M}$ are labelled in color blue, the threshold T in our model is the bias 0.4485. To get a clearer view, we transformed the equation into Fig. 2e, and further to 2f, where the *Positive Inputs* are denoted as $\mathfrak{X}_+$, *Negative Inputs* are denoted as $\mathfrak{X}_-$.

Fig. 2b describes the decision behavior of the entire neural network, it is the multi-valued logical expression of the model. For any specific instance, if its input feature make this equation holds, then it is classified into "Evil" by this model, which means it is an intrusion, otherwise, the model will label it as "Benign", a normal traffic flow.

The task of extracting logical sufficient and complete explanations is to find the minimum rules that leads to *"always true"* or *"always fail"* results, formally defined below.

Definition 4. *An equation is **always pass** iff for any value within the given input range, the equation will always hold. Similarly, the equation is **always fail** iff for any value within the given input range, the equation never hold.*

*An **always pass zone** of an equation is a range of input values, that for any value within that range, the equation al-*

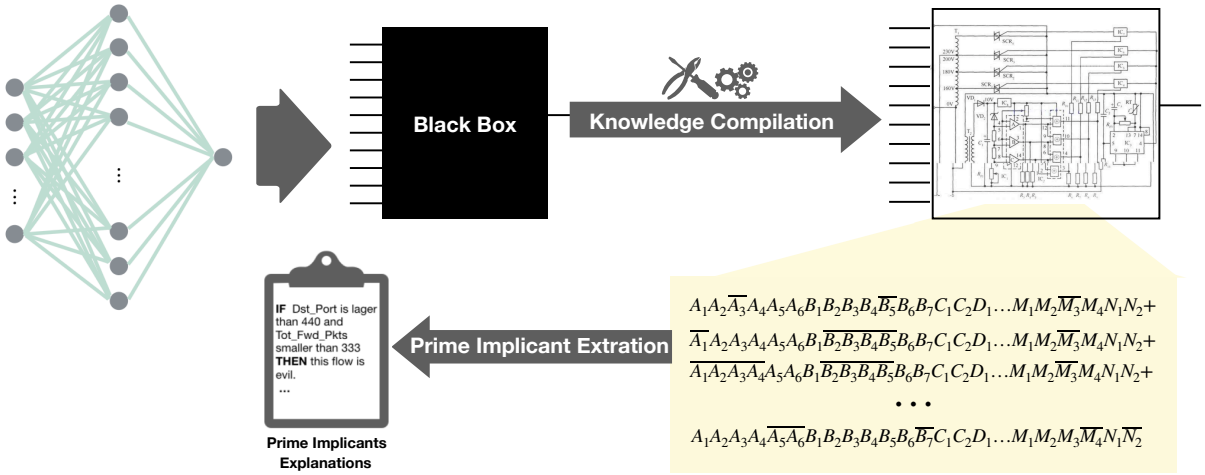

Figure 1: The overview of the study design.

*ways hold. On the contrary, an **always fail zone** is a range of input values, that for any value within that range, the equation always fail.*

Based on that definition, we have the following observation:

Observation 1. *To find the logical complete explanations of the neural network model, we need to find all the "always pass zone" and "always fail zone" for the corresponding threshold-based linear classifier equation.*

Proposition 1. (Boundary Input Values) *For an equation $f(x)$ with the following form,*

$$\sum_{i=0}^{n} \omega_i \cdot x_i \geqslant \mathfrak{X}$$

*where x_i is the discrete input variable, n is the total number of variables, $\omega_i > 0$ is the weight of each input variable, and $\mathfrak{X}$ is a constant. Given a specific value for $\mathfrak{X}$, the **boundary input values** of $f(x)$'s **always pass zone** is a set of combinations of inputs $\mathcal{B}$: $x_1, x_2, \ldots, x_n$ that $\sum_{i=0}^{n} \omega_i \cdot x_i \geqslant \mathfrak{X}$, and, let ω_{min} be the minimum of weights ω, and its corresponding input x_{min}, then $\sum_{x \neq x_{min}} \omega_i \cdot x_i + \omega_{min} \cdot (x_{min} - 1) < \mathfrak{X}$, and the combination of inputs $\mathcal{B}' = \mathcal{B}$ that each combination $\mathcal{B}'$, there is one and only one $x_i > 1$, that $x_i' = x_i - 1$,* is the **boundary input values** of $f(x)$'s **always fail zone**.

The proof of this proposition is given in Appendix . To explain the algorithm in a clearer way, we use card game as an analogy, as shown in Fig. 2g, the left hand side shows all the *positive input* as red bars, the *positive weights* as the length of the bars, analog to *the pips of the cards*, and the radius of the spheres at the end of each bar are the largest values it could take, analog to *the total number of cards of the same pip*. The corresponding *negative input* and the absolute value of the *negative weights* are shown in blue on the right hand side, as *the partner's suit in the card game*. Thus, the formal explanation extraction problem turns to a card game strategy problem, *"for each time the partner play, which cards should I play to win by the smallest margin?*[4]" or to put it in a more formal way, *"what is the least necessary combination of cards, that is larger than the partner's play?"*

Use Case: Formal Explanations for Zero-Day Intrusions

With the observations, we can now extract formal explanations for Zero-Day intrusions. As shown in Fig. 2, the *positive inputs* for our model are $\mathfrak{X}_+ = \mathbb{B}, \mathbb{C}, \mathbb{E}, \mathbb{G}, \mathbb{H}, \mathbb{I}, \mathbb{J}, \mathbb{N}$, corresponding *positive weights* are $0.104, 0.049, 0.157, 0.143, 0.296, 0.165, 0.024, 0.296$, the largest values they are allowed to take are $31, 28, 23, 17, 17, 19, 13, 7$, *positive sensitivity* is $\mathfrak{S}_+ = 0.022$, *positive inputs*' range is $[0.96, 16.53]$, and the *Negative Inputs* are $\mathfrak{X}_- = \mathbb{A}, \mathbb{D}, \mathbb{F}, \mathbb{K}, \mathbb{L}, \mathbb{M}$, the corresponding absolute *negative weights* are $0.074, 0.418, 0.348, 0.065, 0.23, 0.287$, the largest value allowed are $15, 10, 24, 29, 13, 10$, *negative sensitivity* is $\mathfrak{S}_- = 0.065$, together with the threshold $T = 0.4485$, the *negative inputs*' range is $[1.8705, 21.8355]$.

From Observation 4 and 5, we can directly get the following sample rules:

Rule 1: *"If a instance has any one and only one of the following situations: $\mathbb{N} \leqslant 3$, or $\mathbb{I} \leqslant 5$, or $\mathbb{E} \leqslant 5$, $\mathbb{G} \leqslant 6$, or $\mathbb{B} \leqslant 8$, or $\mathbb{C} \leqslant 18$, the other positive inputs $= 1$, whatever the negative inputs are, the model will classify it as benign."*

According to the discretization algorithm, the semantic behind the **Rule 1** is:

"If a traffic flow has `Idle_Min` $\leqslant$ 60072774, *or* `Fwd_Header_Len` $\leqslant$ 18, *or* `Flow_IAT_Min` $\leqslant$ 11.5, *or* `Fwd_IAT_Std` $\leqslant$ 49980.80, *or* `Flow_Duration` $\leqslant$ 13158.0, *or* `Fwd_Pkt_Len_Max` $\leqslant 338.5$, *and if not given, the other positive inputs have* `Idle_Min` $\leqslant 6104707$, *and* `Fwd_Header_Len` $\leqslant 4$, *or* `Flow_IAT_Min` $\leqslant -0.5$, *and*

[4]Here "by the smallest margin" does not mean the total value of the cards has to be the smallest among all possible winning strategies, it means "no unnecessary cards", for example, if $3A + 2B \geqslant \mathfrak{Target}$, $3A + B \geqslant \mathfrak{Target}$, $3A < \mathfrak{Target}$, $3C + 3D \geqslant \mathfrak{Target}$, $3C + 2D \geqslant \mathfrak{Target}$, and $3C + D < \mathfrak{Target}$, then the extra B in $3A + 2B$ and extra D in $3C + 3D$ are unnecessary, but both $3A + B$ and $3C + 2D$ are deemed as "winning strategies by the smallest margin" in our paper.

Fwd_IAT_Std $\leqslant$ 4.60, *and* Flow_Duration $\leqslant$ 156.5, *and* Fwd_Pkt_Len_Max $\leqslant$ 0.5, *and* Fwd_IAT_Max $\leqslant$ 0.5, *and* Fwd_Pkts_s $\leqslant$ 0.05, *no matter what the other features' values are, this flow is benign.*"

Rule 2: "*If a instance has* $\mathbb{M}$ *and* $\mathbb{A}$ *take whatever values they are allowed, the other negative inputs take their largest value, whatever the positive inputs are, the model will classify it as benign.*"

The semantics of **Rule 3** is:

"*If a traffic flow has* Fwd_IAT_Tot $>$ 66039593.5, *and* Fwd_Pkt_Len_Min $>$ 314.5, *and* Subflow_Fwd_Pkts $>$ 7983.0, *and* Fwd_Seg_Size_Avg $>$ 373.15, *no matter what the other features are, the model will classify it as benign traffic.*"

Rule 3: "*If a instance has* $\mathbb{M}$ *and* $\mathbb{K}$ *take whatever values they are allowed, the other negative inputs take their largest value, whatever the positive inputs are, the model will classify it as benign.*"

The semantics of **Rule 3** is:

"*If a traffic flow has* Fwd_IAT_Tot $>$ 66039593.5, *and* Fwd_Pkt_Len_Min $>$ 314.5, *and* Subflow_Fwd_Pkts $>$ 7983.0, *and* Dst_Port $>$ 49778.5, *no matter what the other features are, the model will classify it as benign traffic.*"

Rule 4: "*If a instance has* $\mathbb{K}$ *and* $\mathbb{A}$ *take whatever values they are allowed, the other negative inputs take their largest value, whatever the positive inputs are, the model will classify it as benign.*"

The semantics of **Rule 4** is:

"*If a traffic flow has* Fwd_IAT_Tot $>$ 66039593.5, *and* Fwd_Pkt_Len_Min $>$ 314.5, *and* Subflow_Fwd_Pkts $>$ 7983.0, *and* Idle_Mean $>$ 63254398, *no matter what the other features are, the model will classify it as benign traffic.*"

Rule 5: "*If a instance has all the positive inputs* $= 1$, *and* $\mathbb{M} \geqslant 2$ *or* $\mathbb{D} \geqslant 2$ *or* $\mathbb{F} \geqslant 2$, *the other negative inputs take whatever value they are allowed, the model will classify it as benign.* "

The semantics of **Rule 5** is:

"*If a traffic flow has* Idle_Min $\leqslant$ 6104707, *and* Fwd_Header_Len $\leqslant$ 4, *or* Flow_IAT_Min $\leqslant -0.5$, *and* Fwd_IAT_Std $\leqslant$ 4.60, *and* Flow_Duration $\leqslant$ 156.5, *and* Fwd_Pkt_Len_Max $\leqslant$ 0.5, *and* Fwd_IAT_Max $\leqslant$ 0.5, *and* Fwd_Pkt_Len_Max $\leqslant$ 0.5, *and* Fwd_Pkts_s $\leqslant$ 0.05, *and* Idle_Mean $\geqslant$ 6128868, *and* Fwd_Pkt_Len_Min $\geqslant$ 3.5, *and* Fwd_IAT_Tot $\geqslant$ 78811.0, *no matter what the other features are, the model will classify it as benign traffic.*"

Rule 6: "*If a instance has all the positive inputs* $= 1$, *and* $\mathbb{N} > 4$ *and satisfy one of the following situations* $\mathbb{D} \geqslant \mathbb{N} - 3$, *or* $\mathbb{M} \geqslant \mathbb{N} - 3$, *or* $\mathbb{F} \geqslant \mathbb{N} - 3$, *the other negative inputs take whatever value allowed, the model will classify it as benign.* "

The semantics of **Rule 6** is:

"*If a traffic flow has the positive inputs* Idle_Min $\geqslant$ 61783648, *and* Fwd_Header_Len $\leqslant$ 4, *or* Flow_IAT_Min $\leqslant -0.5$, *and* Fwd_IAT_Std $\leqslant$ 4.60, *and* Flow_Duration $\leqslant$ 156.5, *and* Fwd_Pkt_Len_Max $\leqslant$ 0.5, *and* Fwd_IAT_Max $\leqslant$ 0.5, *and* Fwd_Pkts_s $\leqslant$ 0.05, *and when* Idle_Min $\geqslant$ 61783648\65038801\63254398, *one of the following situations satisfy*, Fwd_Pkt_Len_Min $\geqslant$ 0.5\3.5\38.5 *respectively, or* Idle_Mean $\geqslant$ 5898747\6128868\25400460 *respectively, or* Idle_Mean $\geqslant$ 0.5\78811.0\1475745.0 *respectively, then no matter what the other features' values are, this flow is benign.*"

Discussions and Conclusions

From the explanations, we get a conclusion that, ***"for neural networks, only number matters, the semantic does not matter"***. For example in **Rule 1**, the equations $\mathbb{N} \leqslant 3$ (Idle_Min $<$ 60072774), $\mathbb{I} \leqslant 5$ (Fwd_Header_Len $<$ 18), $\mathbb{E} \leqslant 5$ (Flow_IAT_Min $<$ 11.5), $\mathbb{G} \leqslant 6$ (Fwd_IAT_Std $<$ 49980.80), $\mathbb{B} \leqslant 8$ (Flow_Duration $<$ 13158.0), and $\mathbb{C} \leqslant 18$ (Fwd_Pkt_Len_Max $<$ 4106868.0) are **EQUAL** to the model, although to human experts, these features and values have totally different semantics and should never be deemed as equal, or even related. Take **Rule 8** for another example, the $\mathbb{D}, \mathbb{M}, \mathbb{F}$ have the same effect when the model is making decisions and can be replaceable, however, Fwd_Pkt_Len_Min, Idle_Mean, and Fwd_IAT_Tot have no direct relations, and it is absurd to treat them as the same when making decisions as a human. From Fig. 2, it is obvious that the neural network model is in its essential an equation, it does not understand the semantics or can reason with logic, all it cares are the values of the numbers. Thus, it is essential that the formal explanations for the rules neural network has learned, be interrogated by human experts before any neural network models can be trusted or used in any security critical real-life domains.

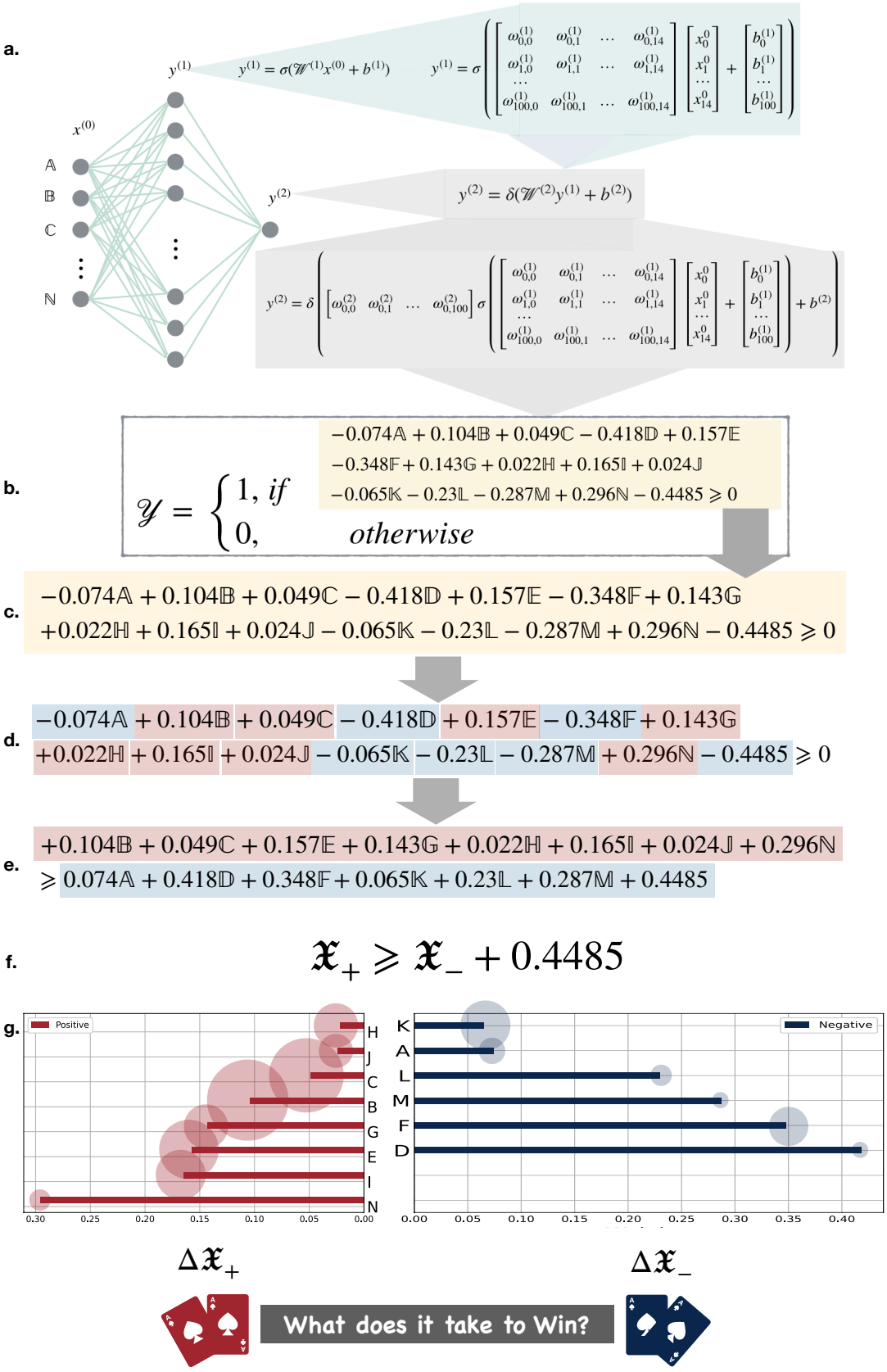

Figure 2: The knowledge compilation process.

References

Arrieta, A. B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58: 82–115.

Audemard, G.; Bellart, S.; Bounia, L.; Koriche, F.; Lagniez, J.; and Marquis, P. 2021. On the Explanatory Power of Decision Trees. *CoRR*, abs/2108.05266.

Banks, V. A.; Plant, K. L.; and Stanton, N. A. 2018. Driver error or designer error: Using the Perceptual Cycle Model to explore the circumstances surrounding the fatal Tesla crash on 7th May 2016. *Safety science*, 108: 278–285.

Borowski, R.; and Choi, A. 2023. On Bounding the Behavior of a Neuron. In *The International FLAIRS Conference Proceedings*, volume 36.

Chan, H.; and Darwiche, A. 2003. Reasoning About Bayesian Network Classifiers. In *19th Uncertainty in Artificial Intelligence (UAI)*, 107–115. San Francisco, California: Morgan Kaufmann Publishers.

Cooper, M. C.; and Marques-Silva, J. 2021. On the tractability of explaining decisions of classifiers. In *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Darwiche, A. 2023. Logic for Explainable AI. In *38th ACM/IEEE LICS*.

Deeks, A. 2019. The judicial demand for explainable artificial intelligence. *Columbia Law Review*, 119(7): 1829–1850.

Domingos, P. 2015. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.

Grover, S.; Pulice, C.; Simari, G. I.; and Subrahmanian, V. 2019. Beef: Balanced english explanations of forecasts. *IEEE Transactions on Computational Social Systems*, 6(2): 350–364.

Gunning, D.; and Aha, D. 2019. DARPA's explainable artificial intelligence (XAI) program. *AI Magazine*, 40(2): 44–58.

Guo, W.; Mu, D.; Xu, J.; Su, P.; Wang, G.; and Xing, X. 2018. Lemna: Explaining deep learning based security applications. In *ACM SIGSAC*, 364–379.

Ignatiev, A. 2020. Towards Trustable Explainable AI. In *IJCAI*, 5154–5158.

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019a. Abduction-based explanations for machine learning models. In *AAAI*, volume 33, 1511–1519.

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019b. On Validating, Repairing and Refining Heuristic ML Explanations. arXiv:1907.02509.

Johansson, U.; König, R.; and Niklasson, L. 2003. Rule extraction from trained neural networks using genetic programming. In *13th International Conference on Artificial Neural Networks*, 13–16.

Johansson, U.; König, R.; and Niklasson, L. 2004. The Truth is In There-Rule Extraction from Opaque Models Using Genetic Programming. In *FLAIRS*, 658–663. Miami Beach, FL.

Kim, B.; Khanna, R.; and Koyejo, O. O. 2016. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29.

Kuppa, A.; and Le-Khac, N.-A. 2021. Adversarial XAI Methods in Cybersecurity. *IEEE Transactions on Information Forensics and Security*, 16: 4924–4938.

LeCun, Y.; Bengio, Y.; and Hinton., G. 2015. Deep Learning. *Nature*, 521(7553): 436.

Lipton, Z. C. 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57.

Lundberg, S. M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J. M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; and Lee, S.-I. 2020. From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence*, 2(1): 56–67.

Lundberg, S. M.; and Lee, S. I. 2017. A unified approach to interpreting model predictions. In *The 31st NIPS*, 4768–4777.

Mathur, V. 2015. Google autonomous car experiences another crash. *Government Technology*, 17.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; and Georg Ostrovski, e. a. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529.

Muna, R. K.; Maliha, H. T.; and Hasan, M. 2021. *Demystifying machine learning models for IOT attack detection with explainable AI*. Ph.D. thesis, Brac University.

Quine, W. V. 1952. The problem of simplifying truth functions. *The American mathematical monthly*, 59(8): 521–531.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *22nd ACM SIGKDD*, 1135–1144.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI*, volume 32.

Shi, W.; Shih, A.; Darwiche, A.; and Choi, A. 2020. On Tractable Representations of Binary Neural Networks. In *17th KR*.

Srinivasan, R.; and Chander, A. 2020. Explanation Perspectives from the Cognitive Sciences-A Survey. In *IJCAI*, 4812–4818.

Stewart, M. 2020. Guide to Interpretable Machine Learning. https://www.topbots.com/interpretable-machine-learning/.

Van den Broeck, G.; Lykov, A.; Schleich, M.; and Suciu, D. 2021. On the tractability of SHAP explanations. In *AAAI*.

Zhao, Q.; and Hastie, T. 2021. Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, 39(1): 272–281.

Appendix A: Proof of Proposition 1.

Proof: We already know $\mathbb{B}$ is in the *always pass zone* of $f(x)$, if $\mathbb{B}^{'}$ is not the *boundary input value* of $f(x)$'s *always fail zone*, it means that there is a $x_p \in \mathbb{B}$ that satisfies:

$$\sum_{i=0}^{n} \omega_i \cdot X_i \geqslant \mathfrak{X} \tag{2}$$

there is also a $x_p^{'} = x_p - 1 \in \mathbb{B}^{'}$ that satisfies:

$$\sum_{i \neq p} \omega_i \cdot x_i + \omega_p \cdot (x_p - 1) \geqslant \mathfrak{X} \tag{3}$$

we already know that $x_p^{'}$'s weight

$$\omega_p \geqslant \omega_{min} > 0 \tag{4}$$

and

$$\sum_{x_i \neq x_{min}} \omega_i \cdot x_i + \omega_{min} \cdot (x_{min} - 1) < \mathfrak{X} \tag{5}$$

Eq. 3 can transform to

$$\sum_{i \neq p} \omega_i \cdot x_i + \omega_p \cdot x_p - \omega_p \geqslant \mathfrak{X} \tag{6}$$

which is equivalent with

$$\sum_{i} \omega_i \cdot x_i - \omega_p \geqslant \mathfrak{X} \tag{7}$$

and with

$$\sum_{i} \omega_i \cdot x_i - \mathfrak{X} \geqslant \omega_p \tag{8}$$

Similarly, Eq. 5 can transform to:

$$\sum_{x_i \neq x_{min}} \omega_i \cdot x_i + \omega_{min} \cdot x_{min} - \omega_{min} < \mathfrak{X} \tag{9}$$

which can be further transform to:

$$\sum_{i} \omega_i \cdot x_i - \omega_{min} < \mathfrak{X} \tag{10}$$

and to

$$\sum_{i} \omega_i \cdot x_i - \mathfrak{X} < \omega_{min} \tag{11}$$

put Eq. 8 and Eq. 11 together, we get:

$$\omega_p < \omega_{min} \tag{12}$$

which is contrast to Eq. 4, thus, the hypothesis does not hold.

Done.