Preprint: 9th International Workshop on Combinatorial Testing (IWCT 20), Porto, Portugal, October 24-28, 2020.

Combinatorial Methods for Explainable AI

D. Richard Kuhn[1], Raghu N. Kacker[1], Yu Lei[2], Dimitris E. Simos[3]

[1] *National Institute of Standards and Technology Gaithersburg, MD 20899, USA* {kuhn, raghu.kacker}@nist.gov

[2] *Computer Science & Engineering University of Texas at Arlington Arlington, TX, USA* ylei@uta.edu

[3] *SBA Research Vienna, Austria* dsimos@sba-research.org

***Abstract* –This short paper introduces an approach to producing explanations or justifications of decisions made by artificial intelligence and machine learning (AI/ML) systems, using methods derived from fault location in combinatorial testing. We use a conceptually simple scheme to make it easy to justify classification decisions: identifying combinations of features that are present in members of the identified class and absent or rare in non-members. The method has been implemented in a prototype tool, and examples of its application are given.**

Keywords – artificial intelligence; combinatorial testing; explainable AI; machine learning; t-way testing;

I. Introduction

Artificial intelligence and machine learning (AI/ML) systems have exceeded human performance in nearly every application where they have been tried and are increasingly incorporated into consumer products. As the current trend continues, AI will be increasingly used in safety-critical systems such as self-driving cars, medical devices, and weapons systems. Current AI systems are generally accurate, but sometimes make mistakes, and human users will not trust their decisions without explanation. Consequently, there is a significant need for improvements in explainability of AI/ML system functions and decisions [1][2][3][4][5][6][7].

The central problem for explainability, according to the Defense Advanced Research Progress Agency (DARPA), is to provide sufficient justification for an AI/ML conclusion such that users know why a conclusion was reached, or why not, and to allow the user to know when an algorithm will succeed or fail, and when it can be trusted [1]. Many conventional approaches leave users wondering what inputs caused a particular conclusion. More than curiosity is involved, as many AI/ML applications may be safety critical, and accuracy rates that are high enough for some applications are inadequate when safety and lives are at risk. Analysis within the aerospace industry concludes that the "artificial intelligence (AI) technology that has made spectacular progress in the consumer world is thus far unsuited to air transport safety standards", and explainability will be essential for certification by regulatory authorities [8]. Ideally, the ML algorithm should be able to explain its conclusion in a manner similar to a human expert, so that other human experts can have confidence in a conclusion, or spot a flaw in the reasoning. This is a significant challenge for methods such as neural networks.

Typically, there is a tradeoff between AI/ML accuracy and explainability: the most accurate methods, such as convolutional neural nets (CNNs), provide no explanations, while more understandable methods, such as rule-based systems, tend to be less accurate [1][2]. Black-box statistical predictions are inadequate, and explanations must be understandable to non-specialists, such as physicians, financial analysts, and in many cases everyday users.

The need for explainability in AI was recognized early, and was an inherent component of many of the first AI diagnostic systems. These were often expert systems using programming-style if/then rules to make decisions. For example: "if patient has symptoms A and B, or has B with C and D, then illness is X". Such systems provide natural explanations, but rules can be difficult to identify, and in many cases are less accurate than other approaches.

While neural networks and related methods often provide better accuracy, they are opaque to users. Decisions are produced using a vast number of internal connections, and some efforts have been aimed at adding explanations to neural nets, but this is an ongoing area of research and the approach has not been widely adopted.

A third way of adding explanations is model induction, inferring an explainable model from black-box inputs and outputs. Systems using this approach have been produced to attempt to identify the most relevant features used in decisions, typically using statistical methods. For example, LIME, one of the more widely used methods, determines features that are most strongly associated with an output [7]. Our method is most similar to these systems, except that we identify combinations of feature values. The result is a system that can infer explanations that incorporate predicates similar to rule based systems, using input/output combinations. Predicates that identify distinguishing features can also assist in validating the model generated by the ML algorithm, by providing more information for human experts in validation. Thus, this method adds value in AI/ML for both users, who need explanation, and model developers seeking to validate a black box model.

II. Human Factors Aspects

A key question for explainability is the degree to which an explanation will be acceptable and trusted by users, which necessarily deals with both technical and human factors. Full development of explainable AI will require extensive validation through human testing, which has not been included in most work in the field [9]. However, the applicability of human factors research to explainable AI has been studied, building on extensive research from psychology on models of human explanation. Here we summarize the major findings of this work, as documented in surveys of the field [9] [10][11][12] [13][14].

Miller et al. [10] suggest that "the most important result from this work is that explanations are *contrastive*: or more

accurately, *why-questions* are contrastive. That is, why-questions are of the form 'Why P rather than Q?'" They note that the psychological research indicates that generally all why-questions are contrastive, seeking an implicit contrast case even if one is not stated explicitly. This is suggested as a potential approach to explainable AI, because providing a contrast case may be easier than a full set of causes [10][15].

Another aspect of explainability identified in human factors research is causal attribution, i.e., the manner in which causes are attributed to events. Relevant findings in this area include research showing that users may consider counterfactuals, what would have happened if some event was not present [16][17], and that only a subset of a full event chain is typically used [17]. It is perhaps not surprising that users prefer simple explanations, with fewer causes or factors, but it was also found that simpler explanations were preferred over more likely explanations [18].

III. METHOD

The classification problem in machine learning is in some ways similar to the problem of fault location in combinatorial testing for software. The objective in both cases is to identify a small number of interactions, out of possibly billions or more, that trigger a failure (in testing) or produce a conclusion (in machine learning). We have methods and tools for fault location in combinatorial testing that can be adapted to ML problems, to identify the rare combinations of variable values that produce conclusions in AI systems. This approach has not been applied to explainable AI before.

A basic approach to fault location for testing is to subtract the set of combinations in passing tests from the set of combinations in failing tests, then using appropriate strategies to narrow down the remaining set to the most likely failure inducing combinations. Similar strategies involve identifying combinations that are more common in failing than in passing tests, to find the most likely cause of a failure.

We can apply this general strategy to the AI/ML explanation problem. Suppose a given object has been identified as a member of a particular class, one of the most common operations in machine learning. A vast number of algorithms and statistical methods can be used to make this identification, but it must be explained to users why the object belongs to the selected class and not some other class. That is, what inputs to the classification algorithm are a convincing justification for concluding that the object is in class *X* and not any other class? This is very similar to the problem of determining what inputs are the reason for a test to produce a failure rather than a passing result.

For explainability, we will also consider two sets of combinations – class and non-class member features, where 'class' refers to a particular group that an object is assigned to. For example, as illustrated in Fig. 1, we may want to explain why an animal is classified as a cat, noting that it shares features with other class members - brown & furry, whiskers, claws – and it does not have features of animals outside the cat class - not aquatic, not venomous. Some features are shared by both the class and non-class members.

While a variety of statistical methods are available for identifying one or a few features that contribute to a conclusion, more information can be provided by using methods from combinatorial testing fault location. We will consider *t*-way combinations, seeking to identify combinations that are unique to class members, i.e., not present in non-class members. It is likely that single features will not be unique, and many 2-way or higher strength feature combinations will also not be unique. But by considering *t*-way combinations with increasing values of *t*, we are likely to reach a point where some t-way combinations are uniquely associated with the class under consideration, or are never associated with the class and can be used to exclude it.

Individual features (orange) – brown & furry, whiskers, claws, not aquatic, not venomous, 4 legs, **...**

Class features (yellow) - brown & furry, black & furry, whiskers, claws, **...**not aquatic, not venomous, 4 legs,

Fig. 1. Feature identification

Looking at combinations of features makes sense intuitively for explanations, because individual features are normally too widely shared among objects of different types. Among animals, thousands of types have four legs, or claws, or pointed ears, but only a limited number have all of these features. For explanations we will look for combinations that are unique, or extremely rare. This is of course essentially the same process used to identify members of a taxonomy, by looking for features an object shares with members of a defined class and for other features that exclude it from a specific class. Thus we argue that the method introduced here is intuitive for users. By adapting combinatorial fault location processes, we can enhance and improve this intuitive method, by considering huge numbers of feature combinations, and quantifying their degree of association with members/non-members of classes. In the following section we illustrate the effectiveness of this approach with an example.

Example. For a more comprehensive example, and to illustrate the application of a prototype tool referred to as ComXAI, we will use the Animals with Attributes (AwA) database to explain the classification of an animal as a reptile. The AwA database describes a large collection of animals using 16 features, 15 boolean and one with six values. For example, Testudo the tortoise [22] (University of Maryland mascot), is shown in Fig. 2, with the following attributes (where 0=false, 1=true): *hair*=0, *feathers*=0, *egg-laying*=1, *milk-producing*=0, *airborne*=0, *aquatic*=0, *predator*=0, *toothed*=0, *backbone*=1, *breathes*=1 *venomous*=0, *fins*=0, *num-legs*=4, *tail*=1, *domestic*=0, *cat-size*=1.

Suppose that an AI/ML algorithm has assigned the class

reptile to Testudo. The prototype ComXAI tool analyzes the presence of combinations of these features in the AwA database animals that are not reptiles. The objective is to identify combinations of reptile features, present in Testudo, that are not present (or extremely rare) in non-reptiles. The presence of these feature combinations should be sufficiently convincing that a reptile has been identified correctly.

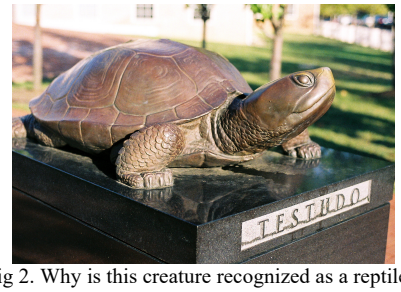

Fig 2. Why is this creature recognized as a reptile?

```
---------------
0053 occurrences = 0.552 of cases, hair = 0
0076 occurrences = 0.792 of cases, feathers = 0
0055 occurrences = 0.573 of cases, eggs = 1
0055 occurrences = 0.573 of cases, milk = 0
0072 occurrences = 0.750 of cases, airborne = 0
0061 occurrences = 0.635 of cases, aquatic = 0
0044 occurrences = 0.458 of cases, predator = 0
0039 occurrences = 0.406 of cases, toothed = 0
0078 occurrences = 0.813 of cases, backbone = 1
0076 occurrences = 0.792 of cases, breathes = 1
0090 occurrences = 0.938 of cases, venomous = 0
0079 occurrences = 0.823 of cases, fins = 0
0036 occurrences = 0.375 of cases, nlegs = 4
0070 occurrences = 0.729 of cases, tail = 1
0083 occurrences = 0.865 of cases, domestic = 0
0043 occurrences = 0.448 of cases, catsize = 1
```

Fig. 3. non-reptile single feature combinations.

As shown in Fig. 3, no single feature is sufficient explanation for classifying Testudo as a reptile, as he shares features with non-reptiles. For example, 55.2 % of other animals in the database have no hair, and 79.2 % have no feathers. Additionally, no pair of features is sufficient, as Testudo also shares 2-way combinations with non-reptiles. As shown in Fig. 4, 2.1 % of the animals in the AwA database have the feature pair *toothless & four-legged*, and 5.2 % have the feature pair *milk-producing & four-legged*.

```
0002 occurrences = 0.021 of cases, toothed,nlegs = 0,4
0005 occurrences = 0.052 of cases, hair,nlegs = 0,4
0005 occurrences = 0.052 of cases, milk,nlegs = 0,4
0006 occurrences = 0.063 of cases, eggs,nlegs = 1,4
0008 occurrences = 0.083 of cases, toothed,catsize = 0,1
0011 occurrences = 0.115 of cases, milk,catsize = 0,1
0012 occurrences = 0.125 of cases, eggs,catsize = 1,1
0013 occurrences = 0.135 of cases, hair,catsize = 0,1
0015 occurrences = 0.156 of cases, predator,catsize = 0,1
0015 occurrences = 0.156 of cases, predator,nlegs = 0,4
0017 occurrences = 0.177 of cases, airborne,toothed = 0,0
0019 occurrences = 0.198 of cases, feathers,toothed = 0,0
0020 occurrences = 0.208 of cases, predator,toothed = 0,0
0021 occurrences = 0.219 of cases, hair,predator = 0,0
0021 occurrences = 0.219 of cases, toothed,backbone = 0,1
0022 occurrences = 0.229 of cases, hair,aquatic = 0,0
```

Fig. 4. 2-way non-reptile feature combinations.

```
00000 occurrences = 0.000 of cases, aquatic,toothed,nlegs = 0,0,4
00000 occurrences = 0.000 of cases, eggs,aquatic,nlegs = 1,0,4
00000 occurrences = 0.000 of cases, hair,aquatic,nlegs = 0,0,4
00000 occurrences = 0.000 of cases, hair,nlegs,catsize = 0,4,1
00000 occurrences = 0.000 of cases, milk,aquatic,nlegs = 0,0,4
00000 occurrences = 0.000 of cases, milk,nlegs,catsize = 0,4,1
00000 occurrences = 0.000 of cases, predator,toothed,nlegs = 0,0,4
00001 occurrences = 0.010 of cases, eggs,nlegs,catsize = 1,4,1
00001 occurrences = 0.010 of cases, eggs,predator,nlegs = 1,0,4
00001 occurrences = 0.010 of cases, feathers,toothed,backbone = 0,0,1
```

Fig. 5. Non-reptiles in the database do not have these 3-way combinations

Looking at 3-way combinations produces much more useful results. As seen in Fig. 5, several 3-way feature combinations uniquely identify reptiles among the animals in the AwA database. No other genus is *non-aquatic* & *toothless* & *four-legged*; no other is *egg-laying* & *non-aquatic* & *four-legged*, and so on. Only reptiles, among the animals in the database, have the 3-way combinations of features shown in Fig. 5.

It is important to note that a different picture emerges from simply listing the individual features that are the strongest differentiators: *four legs*, *toothless*, *cat-size*. As seen in Fig. 3, none of these individual features is anywhere near adequate for identifying a reptile. Additionally, the 3-way combination of these individually-identified features does not appear in the list of 3-way combinations that uniquely identify a reptile among animals in the database. There are in fact many animals in the database with the features *four legs* & *toothless* & *cat-size*. This is a significant difference between the ComXAI approach and methods of statistically identifying the most significant features individually. This example shows why it is necessary to check the rate of occurrence of *t*-way combinations, rather than assume that the *t* strongest associations individually are sufficient to explain a classification.

IV. Discussion

Validation of this method using human subjects is outside the scope of this work, but we can consider the ComXAI approach with respect to human factors research on explanation, introduced in Sect. II. The method and tool described in this paper have been designed to provide intuitive explanations by identifying *t*-way combinations that are present in a given member of a class, and not present or extremely rare in non-members. We believe this is a natural form of explanation because it relies on observable features but quantifies the degree to which feature combinations occur in the class and non-class sets. In particular, this approach provides explanations that are *contrastive*, often considered the most important characteristic of explanations in the psychological literature [10][11]. We identify combinations of attributes that characterize the class to be identified, and that are not found in non-members of this class. This process naturally produces explanations that are contrastive – the combinations presented in the explanation are uniquely associated with the class identified. This provides a clear answer to the "Why *P* and not *Q*?" question implicit in explanations. The class is *P* because these combinations occur only in *P*, and do not occur with any other class *Q*. Using methods developed for fault location makes it possible

to apply the approach across many *t*-way combinations, providing strong justifications for AI/ML conclusions.

It should also be noted that identifying t-way combinations of features that distinguish a class member is essentially the same as specifying predicates in a rule-based expert system. Referring back to Example 1, the six 3-way combinations could be mapped directly to a rule such as "if (*not aquatic && not toothed && four legs*) || (*egg-laying && not aquatic && four legs*) …. then genus = testudo". It is often suggested that rule-based expert systems are the most interpretable, so this correspondence between *t*-way combinations and rule-based predicates also suggests that the ComXAI explanations can be understood well by users.

This method can also be compared with a decision tree approach, where leaf nodes are *t*-way combinations of features (Fig. 6). Note that the tree uses more attributes, leading to more complex predicates, while ComXAI identifies unique combinations of only three features. We plan to investigate the potential for such decision minimization in the future.

```
feathers = false
|   milk = false
|   |   backbone = false
|   |   |   airborne = false
|   |   |   |   predator = false
|   |   |   |   |   legs <= 2: invertebrate (2.0)
|   |   |   |   |   legs > 2: insect (2.0)
|   |   |   |   predator = true: invertebrate (8.0)
|   |   |   airborne = true: insect (6.0)
|   |   backbone = true
|   |   |   fins = false
|   |   |   |   tail = false: amphibian (3.0)
|   |   |   |   tail = true: reptile (6.0/1.0)
|   |   |   fins = true: fish (13.0)
|   milk = true: mammal (41.0)
feathers = true: bird (20.0)
```

Fig. 6. J48 decision tree for AwA produced by Weka [20].

It is also possible to use combinatorial methods to check for gaps in ML models [21]. This approach might be used in concert with ComXAI to validate ML models.

V. Conclusions

Explainability is a critical problem in the acceptance of artificial intelligence/machine learning, especially for critical applications. Human users may not trust AI if conclusions cannot be explained. Methods from combinatorial testing can be applied to the problem of explainable AI, by determining combinations of variable values that differentiate an example from other possible conclusions. That is, we identify *t*-way combinations that are present in members of a class and not present in objects outside the class. A prototype tool ComXAI that applies this approach has been developed.

Acknowledgement and disclaimer: This paper is an extended version of a NIST technical report [19]. Products may be identified in this document, but such identification does not imply recommendation by NIST, nor that those identified are necessarily the best available for the purpose.

References

[1] Gunning D. Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA). 2017. http://www.cc.gatech.edu/~alanwags/DLAI2016/(Gunning)%20IJCAI-16%20DLAI%20WS.pdf

[2] Biran O, Cotton C. Explanation and justification in machine learning: A survey. *IJCAI-17 Workshop on Explainable AI (XAI)* 2017 (p. 8).

[3] Brinton C. A Framework for explanation of machine learning decisions. *IJCAI-17 Workshop on Explainable AI (XAI)* 2017 .

[4] Lomas M, Chevalier R, Cross II EV, Garrett RC, Hoare J, Kopack M. Explaining robot actions. Proceedings of the seventh annual *ACM/IEEE international conference on Human-Robot Interaction* 2012 Mar 5 (pp. 187-188). ACM.

[5] Belle V. Logic meets probability: towards explainable AI systems for uncertain worlds. *26th Intl Joint Conference on Artificial Intelligence, IJCAI* 2017

[6] Holzinger A, Biemann C, Pattichis CS, Kell DB. What do we need to build explainable AI systems for the medical domain?. arXiv:1712.09923. 2017 Dec 28.

[7] Shakerin, F., & Gupta, G. (2018). Induction of Non-Monotonic Logic Programs to Explain Boosted Tree Models Using LIME. *arXiv:1808.00629*.

[8] T. Dubois, "No AI in Cockpit Anytime Soon, Onera, Thales Say", *Aviation Week and Space Technology*, Nov. 26, 2018.

[9] Tjoa, E., & Guan, C. (2019). A survey on explainable artificial intelligence (XAI) *arXiv:1907.07374*.

[10] Miller, T., Howe, P., & Sonenberg, L. (2017). Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioral sciences. *arXiv:1712.00547*.

[11] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intell.*, *267*, 1-38.

[12] Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, *6*, 52138-52160.

[13] Zhang, Y., & Chen, X. (2018). Explainable recommendation: A survey and new perspectives. *arXiv:1804.11192*.

[14] F. K. Došilović, M. Brčić and N. Hlupić, "Explainable artificial intelligence: A survey," *2018 41st Intl Convention on Information and Communication Tech., Electronics and Microelectronics (MIPRO)*, Opatija, 2018, pp. 0210-0215.

[15] Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplements*, *27*, 247-266.

[16] Kahneman, D., & Tversky, A. (1981). *The simulation heuristic* (No. TR-5). Standford Univ. DOI or website needed

[17] Hilton, D. J., & JOHN, L. M. (2007). The course of events: counterfactuals, causal sequences, and explanation. In *The psychology of counterfactual thinking* (pp. 56-72). Routledge.

[18] Lombrozo, T. (2007). Simplicity and probability in causal explanation. *Cognitive psychology*, *55*(3), 232-257.

[19] Kuhn, R., & Kacker, R. (2019). *An Application of Combinatorial Methods for Explainability in Artificial Intelligence and Machine Learning*. NIST, 5/22/19.

[20] Weka data mining. https://www.cs.waikato.ac.nz/ml/weka/

[21] Barash, G., Farchi, E., Jayaraman, I., Raz, O., Tzoref-Brill, R., & Zalmanovici, M. Bridging the gap between ML solutions and their business requirements using feature interactions. *2019 27th ACM Joint Meeting European Software Eng. Conf* (pp. 1048-1058

[22] https://en.wikipedia.org/wiki/Diamondback_terrapin#/media/File:Testudo_2.jpg