Mitigating the Rashomon Effect in Counterfactual Explanation: A Game-theoretic Approach

MGM Mehedi Hasan and Douglas A. Talbert
Department of Computer Science
Tennessee Tech University
Cookeville, USA
mmehediha42@tntech.edu, dtalbert@tntech.edu

Abstract

Counterfactual examples (CEs) are generally created to interpret the decision of a model. In this case, if a model makes a certain decision for an instance, the CEs of that instance reverse the decision of the model. There are many advantages of using counterfactuals as a way of explaining model decisions; however, there is one issue known as the *Rashomon Effect* that might dissuade target/intended users from using counterfactuals. If someone is presented with too many options, this might be overwhelming to them, and they might end up choosing an option that is not optimal or ideal to them. In this case, the Rashomon Effect is an impediment to realizing the full potential of counterfactual explanations. To utilize the full power of CEs and make them more helpful, we need to address the Rashomon Effect. In this work we focus on this issue from game-theoretic perspectives to help target users make informed and feasible decisions by finding highly suitable CEs. In this case, finding good counterfactuals will be a game between two players where each of them tries to find better CEs.

Introduction

There has been a desire for explanations of how complex computer systems make decisions for quite some time. The need for explanations can be dated back to some of the earliest work on expert systems (Buchanan and Shortliffe 1984). Explanations are critical for machine learning (ML), especially as machine learning-based systems are being used to inform decisions in societally critical domains such as finance, healthcare, education, and criminal justice. However, most explanation methods depend on an approximation of the ML model to create an interpretable explanation. For example, consider a person who applied for a loan and was rejected by the loan distribution algorithm of a financial company. Typically, the company may provide an explanation as to why the loan was rejected, for example, due to "poor credit history". However, such an explanation does not necessarily provide the person with sufficient information regarding what they need to do to improve their chances of being approved in the future. Critically, the most important feature may not be enough to flip the decision of the algorithm and, in practice, may not even be changeable such as gender or race.

Wachter et al. (Wachter, Mittelstadt, and Russell 2017) argue that there are three important aims for explanations: (1) to inform and help the person understand why a particular decision was reached, (2) to provide grounds to contest the decision in the case of an undesirable outcome, and (3) to understand what would need to change in order to get a desirable result in the future, based on the current decision making model. A counterfactual explanation (CE) of a prediction describes the smallest change to the feature values that changes the prediction to a predefined output, and it can be a good candidate to fulfill the three aims proposed by Wachter et. al. In interpretable machine learning, counterfactual explanations can be used to explain predictions of individual instances. In this paper, we use the terms counterfactuals and counterfactual examples interchangeably.

Counterfactual examples are increasingly seen as enhancing the autonomy of people subject to automated decisions by allowing people to navigate the rules that govern their lives (Barocas, Selbst, and Raghavan 2020). This helps people recognize whether to contest the decision making process and facilitates direct oversight and regulation of algorithms (Wachter, Mittelstadt, and Russell 2017; Selbst and Barocas 2018). Specifically, counterfactual examples provide this information by showing feature-perturbed versions of the same person who would have otherwise received the loan (Mothilal, Sharma, and Tan 2020).

Motivation and Contribution: There are many advantages of using counterfactuals as a way of explaining model decisions, however, there is one issue known as the *Rashomon Effect* (Anderson 2016) that might dissuade target/intended users from using counterfactuals. *Rashomon* is a Japanese movie in which the murder of a Samurai is told by different people. Each of the stories explains the outcome equally well, but the stories contradict each other. The same can also happen with counterfactuals, since there are usually multiple different counterfactual explanations. For each instance, you will usually find multiple counterfactual explanations. This is inconvenient – most people prefer simple explanations over the complexity of the real world. If someone is presented with too many options, this might be overwhelming to them, and they might end up choosing an option that is not optimal or ideal to them. In this case, the Rashomon Effect is an impediment to realizing the full potential of counterfactual explanations. To utilize the full

power of CEs and make CEs more helpful, we need to address the Rashomon Effect and help the target users make informed and feasible decisions by finding highly suitable CEs.

We address the Rashomon Effect question from two perspectives. The first one will deal with targeted counterfactuals i.e., CEs generated based on the inputs/criteria from the 'end user.' The second perspective is about generating 'Good/Plausible' counterfactual based on domain knowledge. In this case, we propose a solution that has three dimensions. One dimension deals with domain knowledge, another deals with user preferences, and the last is about the solution approach. For the solution approach we apply game theory. In this case, finding optimal counterfactuals will be a game between two players where each of them tries to find the better counterfactuals.

Background

In interpretable machine learning, counterfactual explanations can be used to explain predictions of individual instances. The counterfactual explanation method is model-agnostic, since it only works with the model inputs and output, and the interpretation can be expressed as a summary of the differences in feature values. Counterfactuals are human-friendly explanations, because they are contrastive to the current instance and because they are selective, meaning they usually focus on a small number of feature changes.

We can generate counterfactual explanations using a simple and naive approach, searching by trial and error (Molnar 2019). In this approach, we randomly change feature values of the instance of interest and stop when the desired output is predicted. There are, however, better, more practical approaches than trial and error. We can start by defining a loss function that takes as input the instance of interest and the output is a counterfactual or the desired outcome. This loss function measures how far the predicted outcome of the counterfactual is from the predefined outcome and how far the counterfactual is from the instance of interest (Wachter, Mittelstadt, and Russell 2017). There are two ways to optimize the loss function. One way is to optimize the loss directly with an optimization algorithm like Adam (Adaptive Moment Estimation) (Kingma and Ba 2014). Another way is to search around the instance. Wachter et. al (Wachter, Mittelstadt, and Russell 2017) proposed an approach by minimizing the following loss function, which was later refined by Molnar (Molnar 2019):

$$L(x_i, x_i', y', \lambda) = \lambda \cdot (\hat{f}(x_i') - y_i')^2 + d(x_i, x_i') \quad (1)$$

Here, the term $\lambda \cdot (\hat{f}(x_i') - y_i')^2$ represents the quadratic distance between the model prediction ($\hat{f}(x_i')$) for the counterfactual x_i' for an instance of interest x_i and the desired outcome y_i', which the user must define in advance. The second term $d(x_i, x_i')$ is the distance d between the instance of interest x_i to be explained and the desired counterfactual x_i'.

The parameter λ plays an important role here, which balances the distance in prediction *i.e.* $\lambda \cdot (\hat{f}(x_i') - y_i')^2$ against the distance in feature values *i.e.* $\hat{f}(x_i')$. The loss is solved by choosing an appropriate value of λ, and the solution returns a counterfactual x_i'. The value of λ dictates the kind of compromise we want to make in our preference for counterfactuals. For example, if we choose a higher value of λ that means we prefer counterfactuals that are closer to the desired outcome y_i'. On the other hand, if we go for a lower value λ, we prefer counterfactuals x_i' that are very similar to the instance of interest, x_i, in the feature values. A very large value of λ indicates that, the instance with the prediction that comes closest to y_i' will be selected, no matter how far it is away from x_i.

The choice of λ depends on the user, as he/she must decide how to balance the requirement that the prediction for the counterfactual matches the desired outcome with the requirement that the counterfactual is similar to x_i. Wachter et. al suggest instead of selecting a value for λ, we can select a tolerance ϵ. The tolerance indicates how far away the prediction of the counterfactual instance is allowed to be from y_i'. We can write this constraint in the following way:

$$|\hat{f}(x_i') - y_i'| \leq \epsilon \quad (2)$$

We can use any suitable optimization algorithm to minimize this loss function in Eq. (2). For example, if we have access to the gradients of the machine learning model, we can use gradient-based methods like RMSprop optimizer (Tieleman and Hinton 2012) or Adam. To the best of our knowledge, all the recent works on counterfactual examples generation use Eq. (1) with little or no modification (Mahajan, Tan, and Sharma 2019; Mothilal, Sharma, and Tan 2020; Russell 2019; Sokol and Flach 2019).

Game Theory

Game theory is the study of strategic decision-making, which provides a framework for understanding choice in situations among competing players (Gibbons 1992). In this case, game theory can facilitate competing players to reach optimal decision-making when confronted by independent and competing actors in a strategic setting. In our work, we apply game theory to find optimal CEs that help us reducing the Rashomon Effect. As a game model we choose the Stackelberg Leadership model, which is a sequential game model and is more in line with the kind of problem we are trying to solve. This game model combines theory with practice and provides a practical research model (Roughgarden 2004; Fiez, Chasnov, and Ratliff 2020; Chen et al. 2020; Sinha et al. 2018).

According to game theory, the interaction among participating players are considered as a game. A game can be multi-player, two-player, or mono-player game. Irrespective of the number of players in the game, each player tries to maximize his/her payoff. A gain in one player might result in either gain or loss in other players. This payoff motivation dictates each player's move. In game theory, every participating player is considered as rational, and they make their move intelligently. The heart of every game is the players who makes decision regarding the next course of action. While making a decision a player believes that other players will also try to to maximize their payoff which will result

in the best possible payoff for them. Game theory helps to determine which action will result in maximum payoff and provides a solution concept. A solution concept provides the best possible solution on the strategies or actions to be taken, and at the same time, it gives an idea of possible payoff.

The important part of a solution concept is to formulate a payoff function. Based on this payoff function, the action and strategies are chosen. The *action* is the move the player will take in the next step, whereas *strategy* is a complete set of actions in all possible situations that a player will take over the course of the game. Strategy can be of two types. *pure strategy* or *mixed strategy*. A pure strategy is one where a player takes unique set of actions given the action of other players. If the strategy requires a randomization given a situation, then it called mixed strategy.

A *Nash Equilibrium* is a solution concept that describes a steady state condition of the game where no player would have motivation to change action unilaterally as this would not increase his/her gain. This solution concept only specifies the equilibrium state but does not specify how that steady state is reached in the game. The Nash equilibrium is the most famous equilibrium, and in almost every security game, this concept is used. In truth, it does not ensure the best possible outcome but it does ensure that a player has made the best response given other players' responses. This is when the Nash Equilibrium is achieved.

The general terminologies of Game Theory are described below (Gibbons 1992; Roy et al. 2010; Osborne and Rubinstein 1994).

Game: A description of the strategic interaction between opposing, or co-operating interests, where the constraints and payoff for actions are taken into consideration.

Player: A player is a basic entity in a game that is tasked with making choices for actions. A player can represent a person, machine, or group of persons within a game.

Action: An action constitutes a move in the given game.

Payoff: The positive or negative reward to a player for a given action within the game.

Strategy: Plan of actions that a given player can take during game play.

Related Work

Mothilal et. al extended the work of Wachter et. al (Wachter, Mittelstadt, and Russell 2017) and provided a method to construct a set of counterfactuals with diversity (Mothilal, Sharma, and Tan 2020). Ribeiro et al. (Ribeiro, Singh, and Guestrin 2016) proposed a feature-based approach, LIME, that fits a sparse linear model to approximate non-linear models locally. Guidottiet al. (Guidotti et al. 2018) extended this approach by fitting a decision tree classifier to approximate the non-linear model and then tracing the decision-tree paths to generate explanations. Similarly, Lundberg and Lee (Lundberg and Lee 2017) provided human-comprehensible approximations for linear models and presented a unified framework that assigns each feature an importance value for a particular prediction. Russel worked on efficiently finding coherent counterfactuals avoiding the need for brute-force enumeration (Russell 2019). Ustun et. al worked on evaluating a linear classification model in terms of recourse, which behaves similarly to counterfactuals (Ustun, Spangher, and Liu 2019). In this case, the recourse provides a person the ability to change the decision of the model through actionable input variables. Mahajan et. al addressed the challenge of the feasibility of counterfactual examples by preserving causal relationships among input features (Mahajan, Tan, and Sharma 2019).

Even though there have been much work going on in the counterfactual explanation area, there is not much that addresses the Rashomon Effect. As we mentioned before it is important to address the Rashomon Effect to utilize the full potential of CEs. To the best of our knowledge, we are the first to use game theory to address this issue. In this work, we explore the problem of selecting highly suitable CEs using game theory, which is a strong tool used to analyze and find good solution (Gibbons 1992).

Game Theoretic approach to address the Rashomon Effect

Addressing the Rashomon Effect is very important to utilize full potential of CEs and make CEs more helpful for end users. Game theory is very popular for resolving strategically conflicting issues (Gibbons 1992). In this work we apply game theory based on the fact that choosing the best CE(s) is a strategically challenging decision, and game theory is appropriate for such situations.

Game Theory in Action A number of CEs can be generated for a single instance using different approaches. However, too many can do a disservice to its very own purpose. The target user might get confused when there are too many options to choose from. Choosing at random might not be the best choice for the user. Again, spending lots of time in deciding the correct one is also not the best idea. Out of all these choices of counterfactual examples to choose from, the user should be given only a handful of curated examples based on the needs of the use and feasibility of the CE. Finding these curated CEs can be considered as a competition between two agents (**Leader** and **Follower**), which is influenced by several factors. In this work, we analyze this competition by modeling the problem as a sequential game (Myerson 2013), in particular, using the Stackelberg Leadership Model (Osborne and others 2004). Solving the game provides optimal CEs. In this work, we explore the problem of selecting the optimal CEs using game theory, which is a strong tool used to analyze and settle down strategically conflicting issues (Gibbons 1992). In this game-theoretic analysis, we primarily map the competition between two agents using the sequential (multistage) Stackelberg game (Gibbons 1992; Osborne and Rubinstein 1994) and solve the game to find the Nash Equilibrium (NE) (Gibbons 1992). When a game reaches Nash Equilibrium i.e. we find the required number of optimal CEs, there is no benefit for any player to switch strategies. In this situation, all players in the game are satisfied with their game choices at the same time, so the game remains at equilibrium.

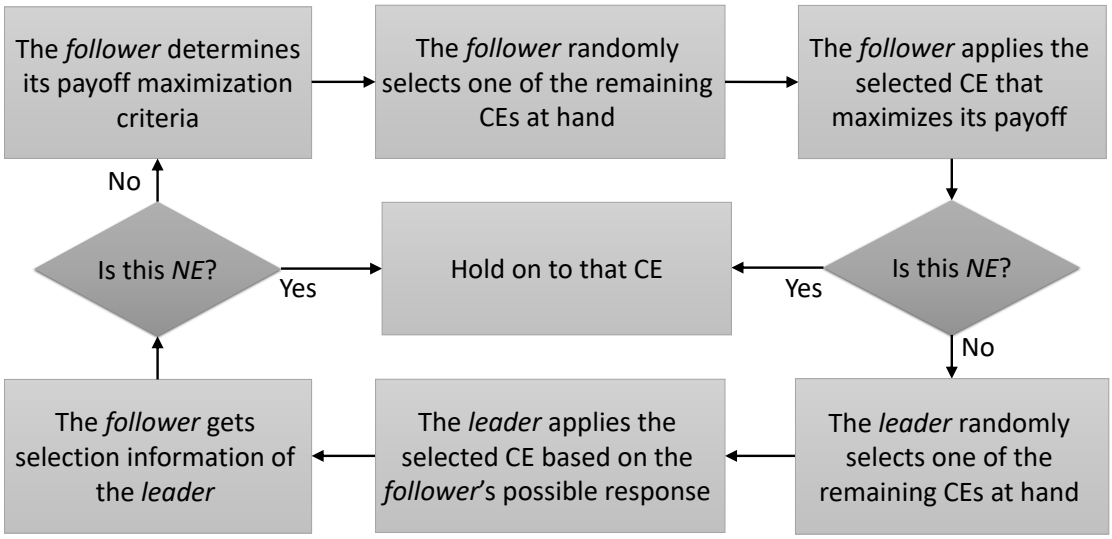

Fig. 1: Process flow of Stackelberg game for selecting the optimal CEs.

Algorithm 1: Rashomon-Game

Result: Finds optimal counterfactual(s)
Input: Generated Counterfactual example pool $\mathbb{C}_{\mathbb{P}}$, *NE* Threshold (N_ϵ), Desired no. of CEs , Maximum number of iterations (I_{max});
$\mathbb{S} \leftarrow \emptyset$
$I \leftarrow 0$
while $\mathbb{C}_{\mathbb{P}}$ *is not empty and* $|\mathbb{C}_{\mathbb{P}}| < R_N$ *and* $I < I_{max}$ **do**

1. $I \leftarrow I + 1$
2. *Nature* chooses the *Leader*.
3. The *Leader* randomly selects one CE (L_{CE}) from C_P. $C_P \leftarrow C_P - L_{CE}$
4. *Nature* chooses the *Follower*.
5. The *Follower* observes the selection of the *Leader*.
6. The *Follower* determines it payoff maximization criteria (P_{MC}).
7. The *Follower* randomly chooses a CE (F_{CE}) from C_P. $C_P \leftarrow C_P - F_{CE}$
8. The *Follower* applies the P_{MC} to the selected F_{CE}.
   if *(L_{CE} and F_{CE} are in* NE*) or* $P_{MC}(L_{CF}) - P_{MC}(F_{CF}) \leq N_\epsilon$ **then**
   $\mathbb{S} \leftarrow F_{CE}$
   $C_P \leftarrow C_P \cup L_{CE}$
   else
   $C_P \leftarrow C_P \cup \{L_{CE}, F_{CE}\}$
   end
9. Return $\mathbb{S}$.

end

Fig. 1 is a simplified diagram of the proposed process corresponding to a Stackelberg game. In this sequential game, the follower has the perfect and complete information about the leader's choice. At the first step, we need to find the best response of the follower based on the actions of the leader. The *Leader* and the *Follower* both can be chosen by **nature** (Osborne and Rubinstein 1994). The *Leader* will be assigned one of the generated CEs randomly. The *Follower* will have the rest of the CEs of the CE pool. In this case, we have designed Algorithm 1 to find the optimal CEs using game theory. The schematic diagram of the process is shown in Fig. 1.

Cost and Benefit In this section, we discuss the potential cost each of the players needs to consider while choosing a move to find the optimal counterfactual(s). The players need to consider the following costs:

1. Difficulty of changing the feature (hardness criteria).
2. Magnitude of the feature change.
3. Willingness to change the feature.
4. Time length to change the features.

Difficulty of changing the feature (hardness criteria): Some features might be difficult to change if not impossible. We call this difficulty of changing features the 'hardness' criteria. The hardness will be rated on a scale of 10 with 1 being the easiest to change and 10 being the hardest to change. Some features are on the extreme end of hardness criteria (i.e. 10 in that hardness criteria scale), we call these features as *immutable* features. Immutable features (like race) should not be suggested to change. While looking for feasible/optimal counterfactuals, considerations of immutable features should be accounted for. We use indicator function to indicate immutability of a feature. In general, we can determine the hardness from the domain knowledge. However, in other cases, we need to consult a domain expert.

Magnitude of the feature change: To what extent we want to or can change the value of a feature is an important aspect to consider when looking for a good counterfactual. For example, in the case of loan application, increasing the 'salary' value might be an option to get accepted. However, if the change amount is so big that the user is unable or unlikely to reach that, it's should not be considered as a good feature change to get the desired counterfactual.

Willingness to change the feature: Some users might be reluctant to change some features suggested in the CEs even though they sound feasible. In this case, we have the option to take the user's feedback. The willingness will be rated on a scale of 10 with 1 being the 'very willing' to change and 10 being the 'not willing' to change.

Time length to change the features: Some feature changes might be time sensitive and the user might not be willing to commit to that. Furthermore, by the time the user achieves the desired feature change, the time might have impact on other features as well. For example, to get loan approval if one needs to have a Graduate degree and by the time the user achieves the degree, it might affect other features (e.g. age) that might negatively impact the outcome.

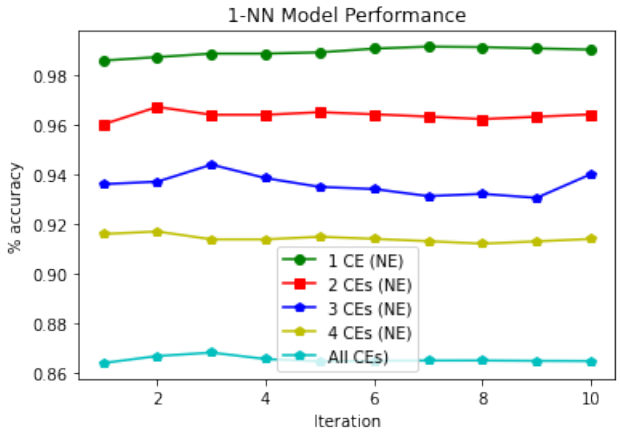

Fig. 2: Feasibility of game-theory curated CEs.

Implementation and Evaluation

We applied a technique introduced by Mothilal et. al (Mothilal, Sharma, and Tan 2020) to generate the CEs. We generated CEs using a shallow artificial neural network (ANN) and then used those CEs in other models. We used different models to experiment with the generated CEs to avoid potential bias that might arise when the same model that generated the CEs is again used to test those CEs. At the same time, we also wanted to make sure that CEs generated by one model are transferable to another model.

Dataset

In this experiment, we consider the LendingClub dataset (Davenport 2015). LendingClub is the first peer-to-peer lending company to register its offerings as securities with the Securities and Exchange Commission (SEC). Their operational statistics are public and available for download (Serrano-Cinca, Gutiérrez-Nieto, and López-Palacios 2015). We preprocess the data based on previous analyses (Davenport 2015; Tan et al. 2018) and obtain 8 features, namely, employment years, annual income, number of open credit accounts, credit history, loan grade as decided by LendingClub, home ownership, purpose, and the state of residence in the United States. The ML model's task is to decide loan decisions based on a prediction of whether an individual will pay back their loan.

Experiments and Discussion

We trained an artificial neural network (ANN) model using the LendingClub dataset. We randomly selected 400 instances and generated a maximum of 10 CEs for each of the instances. We would like to evaluate how feasible our game-theoretically-curated CEs i.e. CEs found in Nash Equilibrium. Our target was to reduce the Rashomon Effect. In this case, we cut down the number CEs from 10 to to 4, which are feasible and optimal. As a tool for explanation, CEs help a user intuitively explore specific points on the other side of the ML model's decision boundary, which then help the user to "guess" the workings of the model. To construct a metric for the accuracy of such guesses, we approximate a user's guess with another machine learning model that is trained on the generated CEs and the original input. Specifically, given a set of CEs and the input example, we train a 1-nearest neighbor (1-NN) classifier that predicts the output class of any new input. Thus, an instance closer to any of the CEs will be classified as belonging to the desired counterfactual outcome class, and instances closer to the original input will be classified as the original outcome class. We chose 1-NN for its simplicity and connections to people's decision-making in the presence of examples. We then evaluate the feasiblity of the CEs obtained from the game model in a mechanistic way. In this case, we hand pick a CE for each of the test instances based on the practical criteria stated above. Then we use our game model to find up to 4 optimal CEs. We can use our game model to pick any number of optimal CEs depending on the user's preference, but for simplicity, we cap this up at 4. It may be noted that originally we generated as many as 10 CEs for each instance. Here the 4 CEs obtained from the game model are supposed to reduce the Rashomon Effect as now the user will only have to choose from 4 feasible options rather than 10.

The accuracy of optimal CEs returned by the game model is shown in Fig. 2. We observe that when the game model provides only one CE in the equilibrium, the model performs better. As in this case, the 'neighbor' (our hand picked CE in this case) returned by the 1-NN model closely matches with the CE returned by the game model. When the game model returns more than 1 CE, we randomly choose 1 CE from the returned CEs (i.e. optimal CEs from the game model) and the 1-NN model return the 'neighbor'. In this case, the performance is slightly weaker. As we increase the number of returned CEs, we observe weaker performance. However, when we use all the generated CEs (i.e. we do not use

game model to return optimal CEs), the performance becomes worse. This indicates that our game-theoretic model is helping to reduce Rashomon Effect.

Conclusion and Future Work

In recent literary work, counterfactual explanations have garnered a lot of attention as an effective way to explain a machine learning model's decision. However, to utilize the full potential of counterfactual explanations, it's important to mitigate the Rashomon Effect. Our preliminary results shows that a game-theoretic model can effectively address this issue. Incorporating our game-theoretic model on top of a CE-generation model can mitigate the Rashomon Effect and make CEs more appealing to the intended users. In the future work, we want to further this idea by replacing the randomness in the CE choice by a more-informed process and run more extensive experiments with other datasets.

References

Anderson, R. 2016. The rashomon effect and communication. *Canadian Journal of Communication* 41(2).

Barocas, S.; Selbst, A. D.; and Raghavan, M. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 80–89.

Buchanan, B. G., and Shortliffe, E. H. 1984. Rule-based expert systems: the mycin experiments of the stanford heuristic programming project.

Chen, Y.; Li, Z.; Yang, B.; Nai, K.; and Li, K. 2020. A stackelberg game approach to multiple resources allocation and pricing in mobile edge computing. *Future Generation Computer Systems* 108:273–287.

Davenport, K. 2015. Lending Club Data Analysis Revisited with Python. https://kldavenport.com/lending-club-python/.

Fiez, T.; Chasnov, B.; and Ratliff, L. 2020. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study. In *International Conference on Machine Learning*, 3133–3144. PMLR.

Gibbons, R. 1992. Game theory for applied economics. Princeton University Press.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Pedreschi, D.; Turini, F.; and Giannotti, F. 2018. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lundberg, S. M., and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, 4765–4774.

Mahajan, D.; Tan, C.; and Sharma, A. 2019. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*.

Molnar, C. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

Mothilal, R. K.; Sharma, A.; and Tan, C. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 607–617.

Myerson, R. B. 2013. *Game theory*. Harvard university press.

Osborne, M. J., et al. 2004. *An introduction to game theory*, volume 3. Oxford university press New York.

Osborne, M. J., and Rubinstein, A. 1994. *A course in game theory*. MIT press.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " 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*, 1135–1144.

Roughgarden, T. 2004. Stackelberg scheduling strategies. *SIAM journal on computing* 33(2):332–350.

Roy, S.; Ellis, C.; Shiva, S.; Dasgupta, D.; Shandilya, V.; and Wu, Q. 2010. A survey of game theory as applied to network security. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, 1–10. IEEE.

Russell, C. 2019. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 20–28.

Selbst, A. D., and Barocas, S. 2018. The intuitive appeal of explainable machines. *Fordham L. Rev.* 87:1085.

Serrano-Cinca, C.; Gutiérrez-Nieto, B.; and López-Palacios, L. 2015. Determinants of default in p2p lending. *PloS one* 10(10):e0139427.

Sinha, A.; Fang, F.; An, B.; Kiekintveld, C.; and Tambe, M. 2018. Stackelberg security games: Looking beyond a decade of success. IJCAI.

Sokol, K., and Flach, P. A. 2019. Counterfactual explanations of machine learning predictions: opportunities and challenges for ai safety. In *SafeAI@ AAAI*.

Tan, S.; Caruana, R.; Hooker, G.; and Lou, Y. 2018. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 303–310.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* 4(2):26–31.

Ustun, B.; Spangher, A.; and Liu, Y. 2019. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 10–19.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.* 31:841.